La Computación en México

por Especialidades Académicas

ACADEMIA MEXICANA DE COMPUTACIÓN

2017

ACADEMIA MEXICANA DE COMPUTACIÓN A. C.

Presidente
Dr. Luis Alberto Pineda Cortés

Vicepresidente
Dr. Luis Enrique Sucar Succar

Tesorero
Dr. Carlos Artemio Coello Coello

Secretarios
Dr. Christian Lemaitre León
Dr. Francisco Javier Cantú Ortiz

Vocal
Dr. Jesús Favela Vara

Coordinadores de Secciones Académicas

Conocimiento y Razonamiento
Dr. Ramón Felipe Brena Pinero

Aprendizaje e Inteligencia Computacional
Dr. Carlos Alberto Reyes García

Tecnologías de Lenguaje
Dr. Luis Villaseñor Pineda

Robótica de Servicio
Dr. Alejandro Aceves López

Ingeniería de Software
Dra. Hanna Oktaba

Interacción Humano-Computadora
Dr. Luis Adrián Castro

Análisis de Señales y Procesamiento de Patrones
Dra. Pilar Gómez-Gil

Computación Evolutiva
Dr. Carlos Artemio Coello Coello

La Computación en México

por especialidades académicas

Luis Alberto Pineda Cortés
Coordinador y editor

ACADEMIA MEXICANA DE COMPUTACIÓN, A, C.

2017

La Computación en México por especialidades académicas

Coordinador general: Luis Alberto Pineda Cortés.

Primera edición: 2017
Academia Mexicana de Computación, A. C.

ISBN: 978-607-97357-1-5

Corrección de estilo: Luis Alberto Pineda Cortés
Formación: Homero Buenrostro Trujillo.
Diseño de portada: Mario Alberto Vélez Sánchez.
Cuidado de la edición: Nydia De Ávila Jiménez y Homero Buenrostro Trujillo.

Este libro se realizó con el apoyo del CONACyT: Proyecto 271979
"Programa Anual de Actividades de la Academia Mexicana de Computación 2016"

Impreso en México.
Printed in Mexico.

Capítulo 1: Conocimiento y Razonamiento

- Luis Alberto Pineda Cortés, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad de México
- Ramón Felipe Brena Pinero, Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey, Campus Monterrey
- Luis Enrique Sucar Succar, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla
- Juan Manuel Ahuactzin Larios, T-Systems México, Puebla
- Rocío Aldeco Pérez, Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey, Campus Querétaro
- José Matías Alvarado Mentado, Departamento de Computación, Centro de Investigaciones y Estudios Avanzados (CINVESTAV-IPN), México
- Ofelia Delfina Cervantes Villagómez, Departamento de Computación, Electrónica y Mecatrónica, Universidad de las Américas Puebla, Cholula, Puebla
- Juan Pablo García Vázquez, Universidad Autónoma de Baja California, Mexicali
- Genoveva Vargas Solar, CNRS, Francia
- Jose Luis Zechinelli, Universidad de Las Américas, Laboratorio Franco-Mexicano de Informática, Puebla
- Carlos Zozaya Gorostiza, Grupo BAL, México

Capítulo 2: Aprendizaje e Inteligencia Computacional

- Eduardo F. Morales Manzanares, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- Carlos A. Reyes García, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- Alicia Morales-Reyes, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- Hugo J. Escalante Balderas, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

Capítulo 3. Lingüística Computacional

- Luis Alberto Pineda Cortés, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad de México

- Hiram Calvo, Centro de Investigación en Computación, Instituto Politécnico Nacional, Ciudad de México

- Luis Villaseñor Pineda, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- Noé Alejandro Castro Sánchez, Centro Nacional de Investigación y Desarrollo Tecnológico, Tecnológico Nacional de México

- Alexander Gelbukh, Centro de Investigación en Computación, Instituto Politécnico Nacional, Ciudad de México

- Yasmín Hernández Pérez, Instituto Nacional de Electricidad y Energías Limpias, Morelos

- Héctor Jiménez Salazar, Universidad Autónoma Metropolitana, Unidad Cuajimalpa

- Manuel Montes y Gómez, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- David Pinto Avendaño, Benemérica Universidad Autónoma de Puebla, Puebla

- Fernando Sánchez Vega, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- Grigori Sidorov, Centro de Investigación en Computación, Centro de Investigación en Computación, Instituto Politécnico Nacional, Ciudad de México

- Gerardo Sierra Martínez, Instituto de Ingeniería, Universidad Nacional Autónoma de México, Ciudad de México
- Esaú Villatoro Tello, Universidad Autónoma Metropolitana, Unidad Cuajimalpa

Capítulo 4. Robótica

- Luis Alberto Pineda Cortés, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad de México

- Luis Enrique Sucar Succar, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- Jesús Savage Carmona, Facultad de Ingeniería, Universidad Nacional Autónoma de México, Ciudad de México

- Alejandro Aceves López, Instituto Tecnológico y de Estudios Superiores de Monterrey, Estado de México

- Héctor Manuel Becerra Fermín, Departamento de Ciencias de la Computación, Centro de Investigación en Matemáticas, Guanajuato

- Gibrán Fuentes Pineda, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad de México

- Marco Antonio Morales Aguirre, Departamento de Sistemas Digitales, Instituto Tecnológico Autónomo de México, Ciudad de México

- Hernando Ortega Carrillo, Departamento de Probabilidad y Estadística, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad de México

- Caleb Rascón, Consejo Nacional de Ciencia y Tecnología -Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad de México

- Arturo Rodríguez García, Posgrado de Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Ciudad de México

Capítulo 5. Ingeniería de Software

- Raúl Antonio Aguilar Vera, Facultad de Matemáticas, Universidad Autónoma de Yucatán

- Hanna Jadwiga Oktaba, Facultad de Ciencias, Universidad Nacional Autónoma de México, Ciudad de México

- Reyes Juárez Ramírez, Facultad de Ciencias Químicas e Ingeniería, Universidad Autónoma de Baja California

- Jorge Rafael Aguilar Cisneros, Decanato de Ingenierías, Universidad Popular del Estado de Puebla

- Carlos Alberto Fernández y Fernández, Instituto de Computación, Universidad Tecnológica de la Mixteca

- Oscar Mario Rodríguez Elias, División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Hermosillo

- Juan Pablo Ucán Pech, Facultad de Matemáticas, Universidad Autónoma de Yucatán

Capítulo 6. Interacción Humano Computadora

- Luis A. Castro, Departamento de Computación y Diseño, Instituto Tecnológico de Sonora (ITSON)

- Mónica Elizabeth Tentori Espinosa, Departamento de Ciencias de la Computación, Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), Baja California

- Jesús Favela Vara, Departamento de Ciencias de la Computación, Centro de Investigación Científica y Educación Superior de Ensenada, Baja California

- Marcela Deyanira Rodríguez Urrea, Facultad de Ingeniería, Universidad Autónoma de Baja California, Mexicali

- J. Alfredo Sánchez, Laboratorio de Tecnologías Interactivas y Cooperativas, Universidad de las Américas Puebla, Cholula, Puebla

Capítulo 7. Análisis de Señales y Reconocimiento de Patrones

- Pilar Gómez-Gil, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- Adolfo Guzmán Arenas, Centro de Investigación en Computación, Instituto Politécnico Nacional, Ciudad de México

- Felipe Orihuela-Espina, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- Ernesto Bribiesca, Departamento de Ciencias de la Computación, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad de México

- Caleb Rascón, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Consejo Nacional de Ciencia y Tecnología - Universidad Nacional Autónoma de México, Ciudad de México

Capítulo 8. Computación Evolutiva

- Carlos Artemio Coello Coello, Departamento de Computación, Centro de Investigaciones y Estudios Avanzados (CINVESTAV-IPN), México

- Carlos Alberto Brizuela Rodríguez, Departamento de Ciencias de la Computación, Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), Baja California

- Nareli Cruz Cortés, Centro de Investigación en Computación, Instituto Politécnico Nacional, Ciudad de México

- Arturo Hernández Aguirre, Departamento de Ciencias de la Computación, Centro de Investigación en Matemáticas, Guanajuato

- Efrén Mezura Montes, Centro de Investigación en Inteligencia Artificial, Universidad Veracruzana, Xalapa

- Alicia Morales-Reyes, Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla

- Diego Alberto Oliva Navarro, Departamento de Ciencias Computacionales, Instituto Tecnológico de Monterrey, Campus Guadalajara

- Katya Rodríguez Vázquez, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad de México

- Hugo Terashima Marín, Escuela Nacional de Ingeniería y Ciencias, Instituto Tecnológico de Monterrey, Campus Monterrey

- Edgar Emmanuel Vallejo Clemente, Escuela de Medicina, Instituto Tecnológico de Monterrey, Campus Monterrey

Índice general

Introducción

El presente texto surge de la preocupación de la Academia Mexicana de Computación (Amexcomp) por difundir la investigación científica y tecnológica en Computación que se realiza en nuestro país. La investigación es la actividad orientada a enriquecer el acervo de conocimiento de la disciplina y se materializa principalmente en artículos en revistas y congresos de prestigio internacional; esta actividad consiste también en la creación de tecnología de vanguardia y en la generación de patentes, así como en la creación de prototipos y sistemas de hardware y software con el potencial de incorporarse a productos de alta tecnología dirigidos al mercado internacional.

Las actividades de investigación científica y tecnológica en Computación se llevan a cabo en México principalmente en el entorno académico. Esta problemática se discute en relación al ecosistema de la Computación en el texto "Políticas y Estrategias para el Desarrollo de la Computación en México" publicado en 2016 por la propia Amexcomp.[1] El propósito del presente texto es profundizar sobre dicha discusión desde una perspectiva académica y difundir las contribuciones específicas. Más que ofrecer estadísticas, un directorio de investigadores e instituciones o un catálogo de productos, nuestro objetivo es dar una introducción de las especialidades que se practican en México y en este contexto presentar qué se hace, dónde y por quién.

De manera más general nuestra intención es contribuir a la formación y maduración de la cultura computacional en nuestro país. La Computación es ya

[1] Disponible en formato .pdf en http://amexcomp.mx

omnipresente e involucra prácticamente todos los ámbitos de la vida cotidiana, por lo que es necesario hacer consciente a la sociedad que la investigación en esta disciplina va mucho más allá de utilizar paquetes genéricos, como editores de textos o gráficos, hojas de cálculo e incluso sistemas especializados, como aquellos para el análisis estadístico o el diseño; va también más allá de programar en los lenguajes de programación más populares, incluso si se trata de resolver grandes sistemas de ecuaciones para los centros de supercómputo, y de ofrecer servicios de apoyo técnico en redes o seguridad, o de desarrollar aplicaciones, como los sistemas de información centrados en bases de datos en el entorno de internet, como las que se hacen en los departamentos de sistemas de las grandes empresas, o por las empresas de desarrollo de sistemas y en las fábricas de software, e incluso de desarrollar aplicaciones para los teléfonos portátiles tan populares hoy en día.

En este momento el lector podría preguntar de qué se trata entonces hacer investigación y qué se hace en nuestro país. Esta es la pregunta que pretendemos responder. El texto consta de ocho capítulos dedicados a las temáticas de las Secciones Académicas de la Académica Mexicana de Computación divididos en tres partes: i) Inteligencia Artificial, ii) Práctica y iii) Algoritmos. Esta clasificación se adoptó simplemente por motivos de la exposición y no presupone un orden o importancia de las partes o los capítulos que contienen. Asimismo, los capítulos son autocontenidos y se pueden leer independientemente, aunque el orden en cada parte va de lo general a lo particular y conviene abordar la lectura en el mismo.

La primera parte de Inteligencia Artificial (IA) reporta la actividad de una porción significativa de los investigadores que se dedican a la investigación en Computación en México y los conceptos de esta especialidad impactan en mayor o menor medida a las disciplinas incluidas en la segunda y tercera parte de práctica y algoritmos. Consta de cuatro capítulos: 1. Conocimiento y Razonamiento; 2. Aprendizaje e Inteligencia Computacional; 3. Lingüística Computacional y 4. Robótica de Servicio. Los tres primeros abordan los conceptos centrales de la especialidad: Pensamiento, Aprendizaje y Lenguaje, de manera consistente con la

propuesta de Alan Turing[2] quien planteó originalmente el programa de investigación de la Inteligencia Artificial.[3] En el capítulo 4 se muestra cómo se incorporan dichas facultades a mecanismos autónomos con capacidades perceptuales y motoras para asistir a los seres humanos en tareas de la vida cotidiana: los robots de servicio, muy populares en el imaginario colectivo contemporáneo.

La segunda parte aborda dos temáticas orientadas a la construcción y uso de sistemas de cómputo: la Ingeniería de Software (IS) y la Interacción Humano-Computadora (IHC). La primera es posiblemente la especialidad de cómputo más conocida en México, ya que se enseña en todos los programas académicos de licenciatura y posgrado con una orientación hacia la Informática, y cuyos egresados ingresan al mercado laboral de la Computación mayoritariamente. Este capítulo se enfoca a métodos, procedimientos y estándares para el desarrollo de sistemas de software y en este sentido difiere del resto de los capítulos que se enfocan a la investigación propiamente. El segundo capítulo de esta sección se enfoca a que los sistemas de cómputo sean usables, por lo que sus interfaces deben capitalizar las facultades humanas así como la forma y funcionalidad de los artefactos de cómputo. Por esta razón la IHC tiene un pie en la práctica y otro en la investigación. En conjunto los dos capítulos de la sección de práctica ofrecen un panorama de lo que se hace en México para la creación de sistemas que no sólo cumplan con su especificación funcional sino también sean usables.

La tercera parte aborda diversos tipos de algoritmos cuyo estudio ha sido objeto por parte de nuestra comunidad. El capítulo 7 de Análisis de Señales y Reconocimiento de Patrones tiene una larga tradición en México, especialmente por sus aplicaciones para el sensado, recolección y análisis de grandes volúmenes de información en una gran variedad de aplicaciones. Por otra parte, el capítulo 8 de Cómputo Evolutivo se centra en un tipo particular de algoritmos, muy útiles para la resolución de grandes sistemas de ecuaciones y para la optimización multi-objetivo y ha sido el foco de interés de nuestra comunidad desde hace dos o tres décadas.

[2] Turing, A. **Computing Machinery and Intelligence.** *Mind* 59:439-460.

[3] Aunque el término Inteligencia Artificial (Artificial Intelligence) propiamente lo introdujo John McCarthy en el workshop que se llevó a cabo en Dartmouth College en el verano de 1956 (ver https://en.wikipedia.org/wiki/Artificial_intelligence).

Hay varias especialidades de la Computación que no se incluyen en el presente texto, como la teoría de la computación, los lenguajes de programación, la computación gráfica, las grandes bases de datos, las redes de cómputo y la seguridad informática, entre otras. Aunque algunas de éstas se practican en México por individuos o pequeños grupos de investigación no están suficientemente representados todavía en nuestra Academia y posiblemente en México. Esperamos que en el futuro sea posible hacer un nuevo texto con una cobertura más amplia.

Por su parte los materiales incluidos en cada capítulo no son mutuamente excluyentes y hay temáticas que se abordan en varias especialidades, aunque en diferentes contextos, y con diferentes énfasis y enfoques. Por ejemplo, la representación del conocimiento y la inferencia automatizada se presentan originalmente en el capítulo 1 pero se retoman y elaboran en los capítulos 2, 3, 4, 6 y 7. Asimismo, el aprendizaje automático se presenta en el capítulo 2, pero se retoma y extiende en el 3, 4, 6 y sobre todo en el 7, aunque desde una perspectiva muy distinta e incluso con una terminología diferente. También los algoritmos evolutivos que se presentan en el capítulo 8 se aplican al aprendizaje de máquina en el capítulo 2 y al análisis de señales y reconocimiento de patrones en el capítulo 7. Hay también una relación muy importante entre los 4 capítulos de la primera parte de Inteligencia Artificial y el capítulo 6 de Interacción Humano-Computadora ya que mientras los primeros se centran en la construcción de sistemas autónomos que piensen, se comuniquen y aprendan por sí mismos, la IHC utiliza conceptos, métodos y herramientas muy similares desde el punto de vista técnico, pero puestas al servicio de los seres humanos para amplificar su competencia y destreza en dichas facultades. Asimismo, los capítulos 5 y 6 de la segunda parte están relacionados ya que la IS involucra al IHC y de manear recíproca el IHC toma y extiende algunos métodos de la IS. Estas relaciones aparecen a lo largo de todo el texto y esperamos que de la lectura en conjunto surja una idea coherente de la naturaleza de estas especialidades y sus puntos en común.

La estrategia adoptada en la exposición consiste en presentar a cada especialidad desde una perspectiva conceptual e histórica desde los orígenes hasta nuestros días. En cada capítulo se presentan las ideas centrales de la especialidad y su evolución a través del tiempo. En varios capítulos se refieren los artículos

seminales que dieron lugar a un esfuerzo de largo aliento[4] y los libros de texto que informaron el estudio de las disciplinas por varios años e incluso décadas y que han sido de cabecera para una o más generaciones de científicos y tecnólogos. En algunos capítulos se citan también las revistas paradigmáticas en las que se ha reportado el desarrollo de la disciplina, así como los principales congresos y conferencias en los que se reúne la comunidad internacional.

Cada capítulo muestra una perspectiva de largo plazo que enfatiza la continuidad de la disciplina y el progreso hasta nuestros días, pero también se describen ciclos locales iniciados por ideas o propuestas novedosas que se desarrollaron con gran entusiasmo, a veces durante varios años, pero que llegaron a sus límites, para dar lugar a un nuevos ciclos, ya sea de inmediato o con períodos de letargo que en algunos de los casos fueron de varios años.

A lo largo de esta narrativa se introducen las aportaciones realizadas en México. Para este efecto se usan notas al pie de página con referencias formales a artículos, libros u otras fuentes de diversa índole, o con comentarios que presentan o detallan esfuerzos particulares. La intención es presentar la contribución en contexto al tiempo que se dan las referencias concretas a los autores con sus instituciones y se hace accesible la documentación técnica para quienes deseen estudiar o aplicar dichos conceptos de manera más profunda e incluso ponerse en contacto directo con los autores.

Algunos de los capítulos incluyen una lista explícita de las instituciones nacionales en que se lleva a cabo investigación en la especialidad, así como las conferencias y congresos nacionales o regionales promovidas o con una alta participación nacional. En otros capítulos esta información se vierte implícitamente a través de las citas a los productos de los investigadores.

Por otra parte, la información no es exhaustiva ya que no todos los investigadores activos en México pertenecen a nuestra Academia y aunque se contó con la participación de investigadores que no son aún miembros y se intentó dar

[4] En particular cuatro de los capítulos (1, 2, 3, 8) hacen referencia al artículo de Allan Turing de 1950 por su influencia en el programa de largo plazo de la Inteligencia Artificial y la fuente de inspiración que representó para muchos de los investigadores en el área.

la mayor cobertura posible, sin duda hay omisiones. No obstante, esperamos que este ejercicio promueva la participación de una mayor parte de la comunidad y en el futuro se cuente con un panorama más preciso de la investigación en Computación que se realiza en nuestro país.

Con el fin de contextualizar muchos de los contenidos, en diversas ocasiones se hace referencia a páginas de la Web, en particular a Wikipedia.[5] Aunque esta práctica no se considera apropiada en entornos muy formales debido a la variabilidad potencial de los contenidos e incluso a la disponibilidad de la fuente, los artículos de este medio son muy accesibles y estables, y dan una idea muy directa y clara de los conceptos generales. Por lo mismo, dentro del espíritu práctico de este libro nos permitimos usar este tipo de referencias de manera flexible.

El presente texto se pensó para una audiencia amplia en varios niveles y ciclos de interés. La más inmediata es la propia membresía de Amexcomp y en general los investigadores en Computación en México. El objetivo en este nivel es hacer a la comunidad más consciente de quiénes somos, qué hacemos y dónde estamos. Ante la extensión de la disciplina es frecuente encontrar métodos, teorías y algoritmos que se desarrollan de manera paralela en dos o más áreas, posiblemente con diferentes perspectivas e incluso nomenclaturas, y esperamos que esta obra permita identificar tanto dichos traslapes como las áreas de oportunidad comunes.

Este libro está también dirigido a la comunidad estudiantil tanto a nivel licenciatura como de posgrado. En particular la gran mayoría de los recursos humanos que se forman en Computación en México tienen la perspectiva de insertarse al mercado laboral de la Computación y la Informática, que requiere mayoritariamente perfiles de Ingeniería de Software. Este es el caso porque en México no existen empresas que desarrollen alta tecnología computacional por lo que el mercado laboral para la investigación en Computación es prácticamente inexistente con excepción del ámbito académico, aunque en éste las plazas para investigadores son muy escasas. Como consecuencia un porcentaje muy elevado de los egresados tienen un conocimiento muy limitado e incluso sesgado de la disciplina, de sus horizontes y de las oportunidades que ofrece.

[5] https://en.wikipedia.org/wiki/Main_Page

Con el fin de contribuir a un mejor entendimiento de la investigación en Computación en oposición a su uso, lo cual es fuente de frecuentes confusiones, el texto está también dirigido a la comunidad científica y tecnológica. Se dirige asimismo a todos los interesados en las aplicaciones científicas y tecnológicas de la Computación y a los actores de la definición de la política científica y tecnológica de nuestro país. Finalmente, el texto se dirige también a todos los actores del ecosistema de la Computación en México[6] y esperamos sea también accesible a sectores informados e interesados del público en general.

Por lo anterior, en el presente texto se evita la presentación de teorías formales y notaciones matemáticas que podrían reducir significativamente el número de lectores potenciales. Sin embargo, los conceptos se presentan con profundidad y rigor y la lectura requiere concentración y análisis. Esperamos sinceramente que el lector enriquezca su comprensión de la investigación en Computación y quede mejor informado acerca del esfuerzo que se realiza actualmente en México.

Para la elaboración del presente texto el Consejo Directivo de la Academia Mexicana de Computación convocó a sus ocho Secciones Académicas a desarrollarlo de manera colegida. A lo largo de 2016 se realizó una reunión presencial por cada sección en la que se planteó el objetivo del libro y se discutió su formato y la estrategia de desarrollo. En dichas reuniones los participantes tuvieron la oportunidad de presentar sus perspectivas de la especialidad y sus propias contribuciones. Posteriormente los participantes enviaron diversos materiales y referencias a los coordinadores de su Sección Académica y se inició el proceso de redacción. Esta tarea se asumió por un grupo de colegas y los textos se distribuyeron y retroalimentaron con todos los miembros de la Sección. Una vez que se contó con un primer borrador se realizaron dos reuniones presenciales en las que participaron el Consejo Directivo y los coordinadores de las Secciones Académicas y los autores principales para hacer una lectura en voz alta de los textos y recibir comentarios y sugerencias de los participantes. Posteriormente los textos pasaron por un proceso de revisión de estilo y lectura de prueba y fueron sujetos a una segunda vuelta de revisión por

[6] Como se describe en el libro referido en la nota 1.

los autores de cada uno de los capítulos. El texto final surgió de la incorporación de los comentarios recibidos.

Todas estas actividades así como el proceso de producción y la publicación del libro se realizaron con el apoyo del CONACyT a través del proyecto 271979 "Programa de Actividades de la Academia Mexicana de Computación 2016" por lo que extendemos nuestro profundo agradecimiento.

2a. Reunión Anual de la Academia Mexicana de Computación

Palacio de la Autonomía, Ciudad de México, 17 de noviembre, 2016

1. Conocimiento y Razonamiento

1.1. Introducción

La tecnología computacional se pensó inicialmente para automatizar cálculos aritméticos que rebasaban por mucho las capacidades humanas. En particular, dentro de las ciencias exactas y la ingeniería, las computadoras se requerían para resolver sistemas de ecuaciones de grandes dimensiones por medio de métodos numéricos. Sin embargo, muy pronto se hizo claro que esta tecnología era también útil para representar el conocimiento y razonar de forma automatizada. Ya en 1950 Alan Turing publicó el artículo *Computing Machinery and Intelligence*[1] (Maquinaria Computacional e Inteligencia) en el que se plantea por primera vez —de manera explícita en un entorno académico y científico— la pregunta de si las máquinas pueden pensar. En este artículo Turing presentó el "Juego de imitación", mejor conocido como la Prueba de Turing, que pretende mostrar que una máquina capaz de comunicarse con un ser humano de manera natural se tiene que considerar como inteligente; en este artículo se propuso también el programa de investigación de la Inteligencia Artificial (IA) junto con dos tareas para desarrollarlo: crear máquinas capaces de jugar ajedrez, es decir de razonar, y máquinas capaces de comunicarse con los seres humanos en el lenguaje natural, para lo cual es necesario que entiendan, es decir que tengan la capacidad de representar y expresar conocimiento. En

[1] Turing, A.M. (1950). **Computing Machinery and Intelligence.** *Mind,* 59:433-460.

ambas tareas ha habido un gran avance: desde 1997 la computadora *Deep Blue* venció al campeón mundial de ajedrez, G. Kasparov, y hoy en día muchos teléfonos celulares han llevado la capacidad de conversación en lenguaje natural a la sociedad en general; sin embargo, esto se debe tomar con cautela ya que la comprensión profunda del lenguaje es una tarea que está todavía lejos de resolverse plenamente.

Desde luego, los términos "conocimiento" y "razonamiento" evocan en principio una actividad mental humana y el término "inteligencia artificial" ha tenido sus detractores; sin embargo, es un hecho que las máquinas tienen la capacidad de representar conocimiento y razonar, por ejemplo, para hacer diagnósticos, tomar decisiones y planear en una gran variedad de entornos y aplicaciones.

1.2. Representaciones proposicionales

Durante los años cincuenta y sesenta del siglo pasado se hizo un gran esfuerzo para representar conocimiento y razonar de manera automática, con énfasis en los procesos deductivos. El sello de esta época fue la creación de programas de cómputo capaces de hacer pruebas lógicas y matemáticas, y de resolver problemas de carácter general mediante la definición de un espacio del problema (*problem space*) y búsqueda heurística. En particular se adoptó a la lógica como el lenguaje para expresar conocimiento, y el razonamiento se conceptualizó como el uso de reglas deductivas para derivar teoremas a partir de axiomas. Es decir, mediante inferencias que van de las causas a los efectos y que se pueden demostrar como correctas o "válidas" desde una perspectiva formal.

Sin embargo, muy pronto se descubrió que expresar problemas de representación y razonamiento en términos lógicos es muy difícil ya que, entre varios factores, las inferencias de la vida cotidiana van frecuentemente de los efectos observables a sus causas, por lo que pueden tener excepciones y no

son válidas desde una perspectiva formal. Por ejemplo, los médicos observan síntomas, que son efectos, a partir de los cuales tienen que inferir las enfermedades, es decir, sus causas, y a veces se equivocan. Esta forma de inferencia en reversa se conoce como "abductiva" o "de diagnóstico".

Más aún, la inferencia abductiva presupone la toma de decisiones, la planeación y la acción. Por ejemplo si hay un charco enfrente de mi casa me puedo preguntar si llovió o si se rompió una tubería, e indagar ambas posibilidades. Si el diagnóstico es que llovió no hay que hacer nada, pero si se rompió una tubería es necesario controlar la fuga. Es en este contexto donde surge la necesidad de tomar decisiones, las cuales deben ser coherentes con el diagnóstico más plausible, tomando en cuenta ciertos valores, por ejemplo, que hay que cuidar el agua. Una vez que se toma la decisión de qué hacer (la decisión y el objetivo son lo mismo: controlar la fuga) es posible establecer un plan y llevarlo a cabo. Una reflexión intuitiva muestra que es éste el ciclo de razonamiento en el que estamos inmersos la mayor parte del tiempo los seres humanos, quienes somos *expertos* en la vida cotidiana.

Estas ideas dieron lugar a los llamados *Sistemas Expertos* (SE), muy populares en la década de los ochenta del siglo pasado, como MYCIN[2] —para el diagnóstico de ciertas enfermedades— y en dicha época se les auguraba un futuro muy prometedor. Alrededor de éstos se crearon plataformas de desarrollo o *shells* (como el muy popular pero limitado CLIPS, o los sofisticados LEVEL5, NEXPERT o FLEX), así como metodologías para transferir el conocimiento de los expertos humanos a las bases de conocimiento; surgió incluso la disciplina de Ingeniería de Conocimiento (*Knowledge Engineering*) para estos propósitos, así como estándares de desarrollo tales como *Common-KADS*.

Independientemente de que el conocimiento se exprese mediante un lenguaje lógico o un sistema de reglas, estas representaciones tienen un ca-

[2] https://en.wikipedia.org/wiki/Mycin

rácter lingüístico, por lo que se les conocen como "proposicionales" y en algunos ámbitos como "simbólicas".

1.3. Representaciones estructuradas

Por otra parte, razonar con representaciones proposicionales es muy costoso en recursos computacionales, tanto por la complejidad de los algoritmos como por los requerimientos de memoria, por lo que en los años setenta y ochenta hubo también una gran cantidad de investigación para representar conocimiento y razonar de forma más efectiva mediante el uso de estructuras como árboles, gráficas dirigidas acíclicas y grafos de diversos tipos.

Posiblemente la representación estructurada más intuitiva es la estructura de árbol o jerárquica para representar taxonomías. Éstas consisten, idealmente, en partir el universo o dominio de conocimiento en un número de clases mutuamente exclusivas, las cuales se parten a su vez de manera recurrente, hasta llegar a las clases más básicas o concretas. Por ejemplo, en la Figura 1.1.a se muestra un árbol que representa a la taxonomía con la información de que los pingüinos y las águilas son aves, que las aves, los peces y los mamíferos son animales, que las aves vuelan y que las águilas comen peces.

En esta estructura los nodos representan clases y las líneas representan a la relación de contención, de tal forma que las clases representadas por los nodos superiores dominan o contienen a las clases representadas por los nodos inferiores; es decir, todo individuo de una clase inferior lo es también de la clase superior. En esta notación cada nodo puede tener una o varias etiquetas textuales asociadas que denotan las propiedades y relaciones de todos los individuos u objetos de la clase, las cuales se indican con cursiva. Por ejemplo, *vuelan* en el nodo **aves** representa que todas las aves vuelan, y *comen peces* en el nodo **águilas** representa que todas las águilas, sin importar cuál en particular, comen individuos de la clase **peces**, también sin importar qué peces.

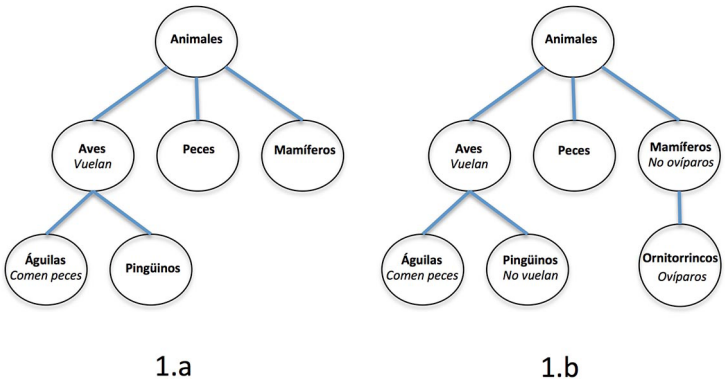


Figura 1.1. Representaciones Estructuradas: (a) ilustra una jerarquía estricta o árbol; (b) una jerarquía con excepciones.

En esta representación las instancias o individuos concretos de cada clase heredan las propiedades de la clase o clases que las dominan, además de que pueden tener propiedades y relaciones específicas entre ellos mismos. Las inferencias se hacen "navegando" a través de la estructura de manera muy eficiente, en vez de aplicar un esquema de inferencia, como en el caso de la lógica, que puede ser muy costoso. En el ejemplo, se sigue, entre varias proposiciones, que las águilas vuelan.

Las representaciones estructuradas se usan comúnmente para expresar lo que se sabe acerca del mundo, especialmente si tiene carácter positivo, como en nuestro ejemplo, pero es frecuente omitir lo que no se considera relevante, es falso o no se sabe, que tiende a ser de carácter negativo, a pesar de que hay muchas más proposiciones negativas que positivas. Es decir, normalmente se expresan las propiedades y relaciones que tienen los individuos u objetos, pero no las que no tienen; por ejemplo, que los peces no comen aves. En este sentido, una representación estructurada es como la "figura" con los rasgos salientes de un cuadro o dominio de conocimiento, que se expresa, pero presupone muchísimo conocimiento de "fondo", que no se expresa. Aunque esto es muy claro en las representaciones estructuradas, éste es un principio general de todo sistema representacional, incluyendo a las

representaciones proposicionales, ya que lo que se expresa es la figura, y toda figura se interpreta siempre en relación a un fondo.

Por lo mismo, toda representación deja varias informaciones indeterminadas, como si los peces comen animales o si los peces comen aves, a pesar de que éstas son preguntas genuinas que un usuario podría hacer al sistema de conocimiento. La respuesta dependerá de la forma de interpretar a la representación: si se asume que el conocimiento es "completo", y que si algo no se expresa es simplemente porque es falso, la respuesta en nuestro ejemplo será que los peces no comen animales y que los peces no comen aves, lo que es falso y cierto respectivamente. Esta forma de interpretación se conoce como la hipótesis del mundo cerrado (*closed-world assumption*) y se adopta por la mayoría de las representaciones estructuradas (y algunas proposicionales, como el lenguaje de programación Prolog), ya sea explícita o implícitamente.

La estructura de grafo pura no tiene una forma natural de expresar la negación, por lo que no es posible distinguir entre lo que no es el caso y lo que no se sabe. En este sentido las representaciones estructuradas contrastan radicalmente con el lenguaje natural y los lenguajes lógicos, como la lógica proposicional o la lógica de predicados, donde es trivial expresar que los peces comen animales y los peces no comen aves, y responder correctamente a las preguntas correspondientes. Más aún, si la base de conocimiento no contiene una proposición (los peces comen aves) ni su negación (los peces no comen aves) y el usuario pregunta *¿los peces comen aves?* el sistema podrá contestar simplemente *no s*é. Por esta razón se dice que las representaciones proposicionales que incluyen la negación permiten la expresión de "conocimiento incompleto", sin que esto conlleve a inferencias incorrectas.

Un compromiso entre las representaciones proposicionales y estructuradas es adoptar el grafo para llevar a cabo la representación e inferencia, pero incluir entre las etiquetas al prefijo de negación, de manera adicional a las etiquetas para identificar propiedades y relaciones, como se muestra en las etiquetas de los mamíferos y los pingüinos en la Figura 1.1.b. En este sis-

tema si no se expresa una proposición ni su negación, o si ninguna de éstas se puede inferir a través de la relación de herencia, la respuesta a la pregunta correspondiente no se sabe.

Una posibilidad adicional es que el sistema incluya la negación pero en vez de responder *no sé*, se extienda la hipótesis del mundo cerrado y responda que no es el caso. Esta podría ser la estrategia adoptada para responder preguntas como si las ballenas caminan o los elefantes vuelan, ya que a nadie se le ocurriría especificar de manera explícita que estos animales carecen de dichas propiedades. El riesgo es por supuesto que existe la posibilidad de equivocarnos.

La discusión anterior ilustra además otro concepto central a la representación del conocimiento: un sistema que asume que el conocimiento es completo y la hipótesis del mundo cerrado es menos expresivo que uno que incluye la negación y asume que el conocimiento es incompleto. Sin embargo, se sabe también que mientras más expresivo es un lenguaje o esquema representacional, mayor es también la complejidad de hacer inferencias en el mismo, y este compromiso de la representación del conocimiento (*Knowledge Representation Trade-Off*),[3] vincula y restringe de manera fundamental a los esquemas de inferencia y a los formatos de representación, al menos en los modelos computacionales.

1.4. Consistencia, defaults[4] y excepciones

Otro aspecto fundamental a la representación del conocimiento y al razonamiento automatizado es que las premisas y conclusiones de los argumentos

[3] Brachmann, R., Levesque, H. (1985). **A Fundamentel Tradeoff in Knowledge Representation and Reasoning.** En R. Beachman y H. Levesque, Readings in Knowledge Representation, Morgan Kaufmann, pp. 41-70.

[4] El término del inglés *default* se refiere a una propiedad o relación que se especifica de manera genérica para todos los individuos de una clase. Se traduce en México como "valores por omisión" y en ocasiones "valores asumidos", pero además de que estas traducciones no expresan claramente el concepto, el término default es el que se utiliza normalmente en México.

deben ser consistentes. Sin embargo, las taxonomías naturales ofrecen excepciones que ponen un reto a los esquemas de representación. Continuando con nuestro ejemplo, sabemos que todas las aves vuelan y que los pingüinos son aves, pero no vuelan; asimismo, que los mamíferos no ponen huevos, pero que los ornitorrincos son mamíferos que sí los ponen, como se ilustra en la Figura 1.1.b. La paradoja es que a pesar de que el sistema es contradictorio, los seres humanos somos capaces de responder sin ningún problema a las preguntas de si los pingüinos vuelan o los ornitorrincos ponen huevos. En representación del conocimiento las propiedades o relaciones de la clase se refieren como *defaults* y los individuos o entidades que tienen la propiedad contraria son las "excepciones"; en nuestro ejemplo los pingüinos son la excepción del default positivo de que todas las aves vuelan, y los ornitorrincos son la excepción del default negativo de que los mamíferos no ponen huevos.

El estudio de estas propiedades desde el punto de vista lógico cae dentro de las llamadas lógicas no-monotónicas, en las cuales el valor de verdad de una proposición puede cambiar con nuevos descubrimientos, en oposición a la lógica clásica y a las matemáticas, donde una vez que un teorema se prueba su valor de verdad permanece para siempre.

Una forma intuitiva de enfrentar el problema es utilizando el llamado "principio de especificidad" que establece que en caso de que haya conflicto entre los valores de verdad de las proposiciones representadas en el sistema, se prefiere a la información más específica. De acuerdo con este principio, en caso de que haya defaults, ya sea de carácter positivo o negativo, se prefieren siempre las excepciones. Este principio, además de ser muy intuitivo, es independiente de si la taxonomía se explora de arriba hacia abajo (*top-down*) o de abajo hacia arriba (*bottom-up*), ya que la oposición entre lo abstracto o más general versus lo concreto o particular es independiente a la estrategia de búsqueda en la estructura representacional.

Sin embargo, una vez que aparecen defaults con excepciones, aparecen otros fenómenos que van más allá de lo que se puede resolver apelando al

principio de especificidad. Por ejemplo, supongamos que se establece que todos los depredadores y sus presas viven en el mismo lugar, pero al mismo tiempo que todos los animales nacieron donde viven. Supongamos adicionalmente que se establece que las águilas son los depredadores de los pingüinos, que Pepe es una águila, que Paco es un pingüino y que los pingüinos viven en la Patagonia. Se sigue entonces que Paco nació en la Patagonia y que Pepe vive en la Patagonia. Sin embargo, supongamos que Paco recibe una oferta de trabajo del Zoológico de Chapultepec y ahora vive en la Ciudad de México. Se sigue entonces que Pepe vive también en la Ciudad de México. Si adicionalmente establecemos que el lugar de nacimiento y de residencia de todos los animales es único, el sistema se hace ahora inconsistente, pero por una nueva razón, ya que hay diferentes líneas argumentales que aunque son consistentes en sí mismas no lo son con otras líneas posibles. Este problema se conoce como el de las "extensiones múltiples" y aqueja también de manera fundamental a los sistemas de razonamiento con defaults y excepciones.

Sin embargo, no todas estas proposiciones tienen la misma calidad y decir que todas las aves vuelan tiene mucho mayor fundamento, o es más necesario, que decir que el lugar donde uno vive es donde uno nació, o que los depredadores y sus presas viven en el mismo lugar, que son verdades mucho más contingentes. Por lo anterior, las propiedades o relaciones de carácter general se pueden dividir en defaults y preferencias, donde los defaults son más necesarios y las preferencias más contingentes, y se puede establecer el principio adicional de que en caso de conflicto se prefieren los defaults. Por supuesto, un sistema que haga esta última distinción será más expresivo que uno que no la haga, por lo que la inclusión de preferencias incrementará el costo de la inferencia.

Una complicación adicional surge cuando la noción de clase, que tiene un carácter ontológico o de existencia, se relaja, y se dice que una "clase" se forma por individuos que comparten una propiedad o una relación.[5] Conti-

[5] Hay una tradición muy antigua que sostiene que hay ciertas propiedades "esenciales" que todos los objetos de la clase tienen necesariamente; si éste fuera el caso sería posible dar

nuando con nuestro ejemplo, si en la taxonomía de la Figura 1.1.b se incluye la "clase" adicional de "animales con pulmones" que domine a mamíferos y aves, la estructura pasa de ser una jerarquía a una *látice,* como se ilustra en la Figura 1.2. A diferencia de los árboles o jerarquías estrictas, en las que un nodo sólo tiene un nodo superior que lo domina, en las látices todo nodo puede ser dominado por varios nodos, por lo que las clases representadas no son mutuamente exclusivas necesariamente.

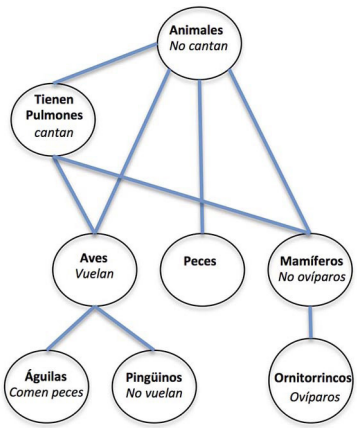


Figura 1.2. Jerarquía múltiple o *látice.*[6]

Si en esta última estructura se aumenta la propiedad de que los animales no pueden cantar pero los animales con pulmones si pueden hacerlo, tanto las aves como los mamíferos tendrán tanto la propiedad de cantar como su negación y el sistema se volverá inconsistente por esta razón adicional. Una vez más, la forma de lidiar con esta situación es apelando a la noción de pre-

definiciones analíticas para las clases utilizando precisamente dichas propiedades en conjunto con el nombre de la clase que la domina de manera más inmediata; por ejemplo, en las aves son animales que vuelan, aves es la clase que se define, animales la que la domina y vuelan la propiedad esencial. Sin embargo, esta postura esencialista se puede refutar con argumentos contundentes. Por ejemplo, ver la discusión Saúl Kripke en **Naming and Necessity,** Harvard University Press, 1980.

[6] El término *látice* viene del inglés *lattice* cuya traducción al español es *retícula* o *celosía*, pero estos términos no se usan en este contexto y la forma que se emplea por la comunidad de computólogos en México es *látice*.

ferencias, para ponderar el peso de una propiedad de acuerdo a la ruta por la que se hereda, aunque también hay que pagar el incremento del costo computacional, que puede ser muy significativo.

Este último ejemplo sugiere adicionalmente que la definición de estructuras conceptuales tiene un carácter subjetivo: ¿Por qué hay mamíferos pero no "pulmoníferos"? La razón es que las taxonomías se hacen siempre bajo un foco de interés y una perspectiva de análisis particular, y diferentes perspectivas pueden dar lugar a diferentes particiones del mismo universo o dominio de conocimiento, cuya representación integral puede dar lugar a látices, que a su vez pueden ser responsables de inconsistencias de carácter más profundo.

Todos estos aspectos se tienen que atender en las bases de datos de conocimiento así como sus esquemas inferenciales asociados, y la utilidad de la representación e inferencia automatizada dependerá en buena medida de la forma como se toman en cuenta, especialmente cuando se requiere la construcción de recursos de conocimiento muy significativos, como ocurre en los dominios de conocimiento complejos, o como los que están disponibles en recursos de grandes dimensiones como Internet.

1.5. Incertidumbre, probabilidad y redes bayesianas

Otro aspecto que es necesario considerar es que muchos problemas de la vida real involucran razonar y actuar no sólo con conocimiento incompleto, defaults y preferencias de diversos tipos, sino adicionalmente con incertidumbre. Este último concepto es ortogonal a las nociones de incompletez y consistencia mencionadas arriba. La incertidumbre se refiere más bien a la carencia y/o confiabilidad de la información, por lo que la teoría de probabilidad provee un marco adecuado para representar y razonar acerca de dicho conocimiento. Hay alternativas a la teoría de la probabilidad para razonar con incertidumbre, como la lógica difusa, pero la teoría de la probabilidad cuenta con fundamentos matemáticos sólidos y métodos bien establecidos.

Por otra parte, la aplicación directa de la probabilidad implica una alta complejidad computacional, lo cual frenó su desarrollo en los inicios de la computación, en particular en el área de inteligencia artificial y sistemas expertos. Sin embargo, el surgimiento de las redes bayesianas,[7] así como de otros paradigmas relacionados, en los ochenta del siglo pasado, hicieron posible el desarrollo de métodos computacionales eficientes para la representación de conocimiento e inferencia basadas en probabilidad. La idea esencial es representar las relaciones de dependencia e independencia entre las variables de cierto problema mediante grafos, con ahorros importantes en memoria y en las operaciones de cómputo requeridas para modelar problemas complejos.

Por ejemplo, en la Figura 1.3 se ilustra un modelo simplificado de un problema de diagnóstico médico en el que se expresa que la fiebre y el dolor dependen de tener gripa o tifoidea, que las reacciones dependen también de esta última, y que la tifoidea depende de haber comido alimentos de procedencia dudosa. Estas relaciones se representan como los nodos del grafo y se cuantifican con probabilidades condicionales de cada variable dados los nodos que los domina (*padres* en el grafo); por ejemplo, para la variable Fiebre: *P(Fiebre/Tifoidea, Gripe)*. El cálculo de la probabilidad de cada variable dada cierta información, por ejemplo de la enfermedad dados los síntomas, se puede hacer en forma muy eficiente mediante la regla de Bayes con la ayuda de las relaciones de independencia implícitas en el grafo. En general este proceso de inferencia probabilista es eficiente incluso para modelos con cientos o miles de variables (mientras la topología del grafo no sea muy densa).

Las redes bayesianas son una instancia de los Modelos Gráficos Probabilistas —grafos que representan las dependencias entre variables y parámetros locales asociados, así como mecanismos eficientes de inferencia— que incluyen a los modelos ocultos de Markov, los campos de Markov, los clasificadores bayesianos, así como representaciones que incorporan decisiones

[7] Pearl, J. **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,** Morgan Kaufmann, San Francisco, 1988.

y utilidades, como los diagramas de influencia y los procesos de decisión de Markov. Se han desarrollado lenguajes de programación que facilitan la implementación de estos modelos como el API *ProBTR* que implementa la Programación Bayesiana, la cual consiste en una metodología y formalismo para especificar y resolver modelos probabilistas.

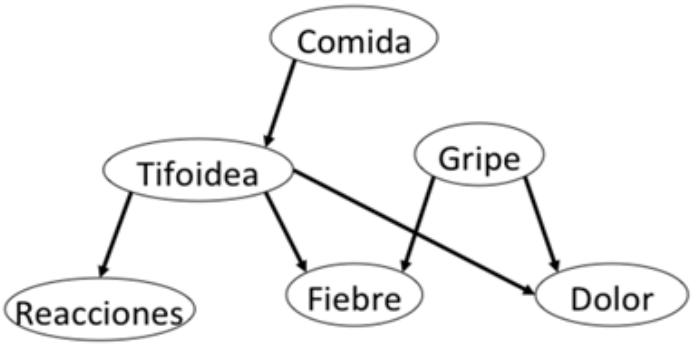


Figura 1.3: Ejemplo de una red bayesiana.

1.6. Inferencia inductiva y aprendizaje

Además de las inferencias deductiva y abductiva, existe la inferencia inductiva o de aprendizaje, que consiste en la generación de reglas generales a partir de un número de instancias o ejemplos de un fenómeno. En la modalidad simbólica esta inferencia consiste en la generación de una descripción general a partir de descripciones de situaciones particulares, o mediante el razonamiento por analogía; alternativamente, en el nivel de computación sub-simbólico, esta inferencia subyace a los procesos de clasificación y predicción por métodos probabilísticos, estocásticos o mediante redes neuronales o, de manera más general, el llamado "conexionismo". La inferencia inductiva se aborda de manera detallada en el capítulo de Aprendizaje e Inteligencia Computacional.

En resumen, toda inferencia se puede asimilar a los tres tipos principales: deductiva, abductiva e inductiva o de aprendizaje. Por su parte, desde una

perspectiva teórica y metodológica, ha habido varios formatos representacionales para implementar los tres tipos de inferencia, donde destacan i) las representaciones simbólicas, a veces llamadas proposicionales, lógicas o lingüísticas, ii) las representaciones estructuradas, iii) las representaciones estocásticas o probabilísticas y iv) las llamadas representaciones sub-simbólicas, donde juegan un papel protagónico las redes neuronales. Aunque cada uno de estos cuatro tipos de representaciones se asocia a un modo de inferencia preferencial, desde una perspectiva amplia los tres tipos de inferencias se han investigado y pretendido asimilar desde la perspectiva de los cuatro tipos de esquemas o formatos representacionales.

1.7. Interacción Humano-Computadora

En otra dimensión, hasta cierto punto independiente de lo que se ha dicho hasta ahora, es necesario considerar si las computadoras se utilizan para representar el conocimiento y razonar de manera automática, o si quienes realizan estos procesos son los seres humanos, y la tecnología computacional se utiliza para apoyarnos en los procesos de representación e inferencia. Esta bifurcación se dio de manera explícita en la historia de la computación. Un ejemplo muy conocido es el caso del investigador del MIT Terry Winograd,[8] quien después de crear el programa de cómputo SHLDRU[9] a finales de los sesenta, que era capaz de entender inglés y razonar de una manera no trivial a juicio de la comunidad de investigación, consideró que las máquinas no pueden realmente entender y razonar en un sentido humano, y que dotarlas de estas capacidades, con un nivel de competencia que escalara a problemas reales, era muy distante. Por estas razones reorientó su investigación para utilizar las metodologías y herramientas de la IA para apoyar la representación del conocimiento y el razonamiento de los seres humanos, con grandes aplicaciones potenciales, contribuyendo de esta forma a la creación de la especialidad del cómputo conocida como Interacción Humano-Computadora (IHC), que

[8] https://es.wikipedia.org/wiki/Terry_Winograd
[9] https://es.wikipedia.org/wiki/SHRDLU

se puede considerar la otra cara de la moneda de representación e inferencia, la cual se aborda extensamente en el capítulo 6 de este texto. Una cantidad considerable de la investigación en conocimiento y razonamiento realizada por la comunidad mexicana ha tenido esta última orientación y hay un traslape importante entre las comunidades de representación y razonamiento y la comunidad de IHC.

1.8. Sistemas Multi-Agentes

En otra dirección independiente a lo ya dicho, durante los años ochenta y noventa del siglo pasado, empezó a tomar fuerza la idea de conjuntar varias entidades inteligentes a las que denominaremos "agentes" capaces de colaborar y competir y, de esta forma, dar lugar a una conducta colectiva. A estas colecciones se les conoce como Sistemas Multi-Agentes (SMA) y su estudio retomó influencias de muchas áreas aparentemente dispares. En primer lugar, desde luego, hereda las técnicas de búsqueda y de razonamiento de la inteligencia artificial clásica y del aprendizaje automático, pero en vez de suponer una inteligencia central "omnisciente" con un conocimiento completo del mundo, toma el punto de vista de una inteligencia limitada, y sobre todo situada, en la que el agente inteligente percibe la realidad desde su ubicación y circunstancia particular a través de sus sensores y ejerce su influencia en su entorno mediante actuadores o efectores, con la mediación de un proceso de razonamiento deliberativo.

Los SMA también reciben la influencia de áreas como la Teoría de Juegos, que inicialmente se estudiaba en Economía, pero que es muy útil como paradigma de interacción de múltiples entidades racionales. Conceptos como el equilibrio de Nash, las estrategias dominantes, los juegos de suma cero, las estrategias óptimas de Pareto y otros, encontraron naturalmente aplicación en los SMA. Asimismo, áreas tan aparentemente distintas como la lingüística, la biología y la psicología también han contribuido al desarrollo de los SMA. Otra área relacionada con los SMA es el estudio de los autómatas, principal-

mente los de estados finitos, con su contraparte probabilista en los procesos de Markov.

El período entre el año 2000 y la actualidad se ha desarrollado entre una multitud de paradigmas de IA compitiendo y complementándose, hasta que en los últimos años el aprendizaje automático (*Machine Learning* o ML) ha destacado por la enorme rentabilidad que aporta a las empresas, especialmente en áreas como mercadotecnia y ventas, y es un ejemplo muy directo de la aplicación de la inferencia inductiva. Herramientas basadas en IA, y en particular en aprendizaje, se han vuelto comunes en los últimos años, incluyendo la detección de caras y reconocimiento de personas en cámaras y redes sociales, los sistema que filtran el *spam* en nuestros correos, las recomendaciones de libros y películas de acuerdo a nuestras preferencias, etcétera.

1.9. Ontologías

Por otra parte, y ligado a la explosión del Internet, desde el inicio de este milenio han surgido tecnologías asociadas a la representación del conocimiento, como las llamadas "Ontologías", que son parte de la propuesta del "Web Semántico",[10] el cual tiene por objetivo que las páginas Web se auto-describan usando un código estándar *Resource Description Framework* (RDF)[11] de forma tal que puedan ser interpretadas por la computadora, a diferencia de las páginas Web usuales, que están destinadas al consumo humano. Las tecnologías asociadas a la representación del conocimiento en ontologías han tenido mucho desarrollo, y han dado lugar a estándares de lenguajes tales como XML,[12] RDF,[13] OWL[14] y otros, así como múltiples herramientas para manejarlos, tanto para su almacenamiento como para su explotación usando formas limitadas de razonamiento automático.

[10] https://en.wikipedia.org/wiki/Semantic_Web
[11] https://en.wikipedia.org/wiki/Resource_Description_Framework
[12] https://en.wikipedia.org/wiki/XML
[13] Ibid
[14] https://en.wikipedia.org/wiki/Web_Ontology_Language

Algunos aspectos de las ontologías son: i) se utilizan para describir un dominio específico; ii) los términos y las relaciones están claramente definidos en ese dominio; iii) existe un mecanismo para organizar los términos (comúnmente se utiliza una estructura jerárquica o una látice, como se ilustra en las figuras 1.1 y 1.2) y iv) existe un acuerdo entre los usuarios de una ontología de tal manera que el significado de los términos se utiliza de manera coherente. Otras funciones y usos más relevantes y generales de las ontologías incluyen la descripción de la semántica de los datos, para compartir el conocimiento y reutilizar los recursos de información al comunicar agentes humanos y/o de software, facilitando la interoperabilidad del conocimiento.

Este desarrollo simplemente corresponde a la extensión de los tipos de inferencia y esquemas de representación, como grafos y látices, que anteriormente estaban circunscritos a procesos locales, a su expresión en redes masivas de cómputo como es el caso del Internet. Este movimiento ha tenido un gran impacto en la capacidad de expresar información y utilizarla de manera distribuida, pero con una gran limitación en la capacidad de razonamiento, dado su costo computacional, además de que no es siempre posible garantizar la consistencia de la información, por las razones abordadas anteriormente en las secciones 1.3 y 1.4.

El papel que las tecnologías de conocimiento y razonamiento tendrán en los próximos años seguramente será altamente relevante, siempre y cuando logre una alta integración con otras tecnologías de IA, tales como las que se mencionan en este texto.

En las secciones restantes de este capítulo presentaremos las aportaciones de la comunidad mexicana a la investigación en los esquemas de representación y los tipos de inferencia, así como las aplicaciones que se han hecho en México.

1.10. Investigación en México

La investigación realizada en México en esta área sigue las tendencias de la investigación a nivel mundial, y desde luego puede ser clasificada siguiendo los mismos lineamientos que hemos presentado en la sección anterior. Así, los formatos de representación pueden ser proposicionales, estructurados, probabilísticos o conexionistas, mientras que las formas de inferencia pueden ser de naturaleza deductiva, abductiva o inductiva. Asimismo, los sistemas pueden ser centralizados o multiagentes, y los sistemas de conocimiento se pueden concebir como autónomos y causales de la conducta de agentes computacionales, como los robots, o pueden concebirse como servicios para apoyar la representación y el razonamiento humano. En otra dimensión, los sistemas pueden adquirir conocimiento a través de la interacción con seres humanos, aprender nuevo conocimiento mediante la inducción de aprendizaje, o utilizar grandes repositorios como las ontologías disponibles en Internet.

En los párrafos siguientes se presentan algunas investigaciones de nuestra comunidad siguiendo estas categorías conceptuales.

1.10.1. Razonamiento proposicional

Una aportación a la literatura mundial con este enfoque fue la teoría para la representación e interpretación de información diagramática y textual,[15] con aplicaciones al razonamiento multimodal, como el que se requiere para interpretar mapas. Esta investigación se inició en el Instituto de Investigaciones Eléctricas (IIE) y se concluyó en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), de la Universidad Nacional Autónoma de México (UNAM). Este tipo de representaciones se aplicaron posteriormente en el IIMAS al razonamiento que involucra el uso de diagramas, y en particular para la automatización del descubrimiento y pruebas geométricas; la teoría resultante permitió crear por primera vez un programa de Inteli-

[15] Pineda, L. A., Garza, G. (2000). **A Model for Multimodal Reference Resolution.** *Computational Linguistics* 26(2):136-192.

gencia Artificial que descubre y prueba (aunque de manera semi-automática) el Teorema de Pitágoras así como algunos teoremas aritméticos de carácter inductivo que se pueden representar con diagramas.[16]

1.10.2. Representaciones estructuradas

En el contexto del proyecto Golem en el IIMAS, UNAM, se desarrolló un sistema de representación del conocimiento para razonar con taxonomías con conocimiento incompleto, default y excepciones, utilizando el principio de especificidad, como se presenta en las secciones 1.3 y 1.4. Aunque el sistema está codificado en Prolog, por lo que también tiene un aspecto proposicional, se puede orientar a diversos dominios de conocimiento, especialmente donde se requiera actualizar a la base de conocimiento dinámicamente, incluyendo defaults y excepciones, tanto de carácter positivo como negativo.[17] Estas representaciones son también útiles para modelar la cadena inferencial de diagnóstico, toma de decisiones y planeación, que lleva a cabo el robot Golem-III cuando sus expectativas no se cumplen en el mundo.[18]

Otros trabajos han utilizado representaciones del conocimiento basadas en grafos. El Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) ha explorado el uso de los grafos conceptuales para minería de texto;[19] estos trabajos han permitido confirmar que estas representaciones, aunque sencillas y fáciles de obtener y analizar, restringen los patrones descubiertos a nivel temático. Para contar con una mayor semántica se propone usar a los grafos conceptuales para representar el contenido de los textos y

[16] Pineda, L. A. (2007). **Conservation principles and action schemes in the synthesis of geometric concepts.** *Artificial Intelligence*, 171(4):197-238.

[17] Pineda, L. A., Rodríguez, A., Fuentes, G., Rascón, C., Meza, I. (2017). **A Light non-monotonic knowledge-base for service robots, Intel Serv Robotics** 10(3):159-171. doi:10.1007/s11370-017-0216-y.

[18] http://golem.iimas.unam.mx/~golem/opportunistic_inference/

[19] Montes-y-Gómez, M. Gelbukh, A., López-López, A. (2002). **Text mining at detail level using conceptual graphs.** *International Conference on Conceptual Structures*. Springer Berlin Heidelberg.

se han obtenido algunos patrones descriptivos de los documentos al aplicar varios tipos de operaciones sobre estos grafos.

La Universidad de las Américas en Puebla (UDLA) ha trabajado en el modelado de textos basados en grafos con propósito de realizar identificación,[20] y verificación de autoría,[21] así como para el perfilado del usuario y el análisis de sentimientos.[22] Los grafos propuestos son estructuras con mayor riqueza semántica y han permitido obtener mejor precisión en las tareas de minería de textos. El grupo ha usado también las ontologías para el desarrollo de sistemas inteligentes de recomendación, particularmente turística, combinadas con el modelado de redes sociales usando grafos sobre los cuales las medidas de centralidad permiten elaborar recomendaciones que consideran los perfiles de las preferencias de los usuarios, así como las aportaciones de sus redes sociales y las características que distinguen a los sitios de interés.[23]

Otro enfoque de razonamiento no monotónico es la argumentación rebatible, que es una forma argumentativa con defaults, argumentos y contra argumentos; esta metodología se aplicó a problemas de distribución inteligente de información en el proyecto JITIK del Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) con la U. N. del Sur en Argentina.[24]

---

[20] Castillo, E., Cervantes, O., Vilariño, D., Pinto, D., León S., (2014). **Unsupervised Method for the Authorship Identification Task.** Artículo presentado en CLEF PAN (Working Notes).

[21] Castillo, E., Vilariño, D., Cervantes, O., Pinto D., (2015). **Author attribution using a graph based representation.** CONIELECOMP.

[22] Castillo, E., Cervantes, O., Vilariño D., Báez, D., Sánchez, A., (2015). **UDLAP: Sentiment Analysis Using a Graph Based Representation.** Artículo presentado en SEMEVAL.

[23] Cervantes, O., Gutiérrez, F., Gutiérrez, E., Sánchez, J. A., Muhammad, R., Wan, W. (2015). **A Recommendation engine based on social metrics.** 6th Workshop on Semantics for Smarter Cities, 14th International Semantic Web Conference (ISWC2015).

[24] Brena, R., Chesnevar, C., Aguirre, J. L. (2006). **Argumentation-supported Information Distribution in a Multiagent System for Knowledge Management.** Lecture Notes in Computer Science 4049, pp. 279-296, Springer Verlag.

1.10.3. Manejo probabilista de incertidumbre y modelos gráficos

Desde los años noventa del siglo pasado diversas instituciones mexicanas, como el IIE, el ITESM y luego el INAOE, el Tecnológico de Acapulco y la Universidad Veracruzana, entre otros, han realizado investigaciones en modelos gráficos probabilistas donde se destaca su aplicación a diversas áreas:

- La aplicación de redes bayesianas en análisis de confiabilidad de sistemas complejos.[25] Fue el primer trabajo en esta área en el mundo y dio origen a una línea de investigación con talleres y congresos propios.

- El uso del esquema de validación de información basado en redes bayesianas en diversos dominios, incluyendo plantas eléctricas, transformadores, fabricación de tubos de acero y campos petroleros.[26]

- La aplicación de las redes de eventos temporales en diagnóstico de plantas eléctricas y la predicción de mutaciones del virus del VIH.[27]

- El reconocimiento de ademanes con modelos ocultos de Markov y redes bayesianas dinámicas; el grupo fue uno de los pioneros en esta aplicación, la cual es muy utilizada actualmente.[28]

---

[25] Torres-Toledano, J. G., Sucar, L. E. (1998). **Bayesian Networks for Reliability Analysis of Complex Systems.** En H. Coelho (Ed.), IBERAMIA'98, Lecture Notes in Computer Science, Vol.1484, Springer-Verlag, Berlin, pp. 195-206.

[26] Ibargüengoytia, P. H., Vadera, S., Sucar, L. E. (2006). **A Probabilistic Model for Information Validation.** *British Computer Journal*, 49(1):113-126.

[27] Hernández-Leal, P., Rios-Flores, A., Ávila-Rios, S., Reyes-Terán, G., González, J. A., Fiedler-Cameras, L., Orihuela-Espina, F., Morales, E. F., Sucar, L. E. (2013). **Discovering HIV Mutational Pathways using Temporal Bayesian Networks.** *Artificial Intelligence in Medicine,* 57(3):185-195.

[28] Avilés-Arriaga, H. H., Sucar, L. E., Mendoza-Durán, C. E., Pineda, L. A. (2011). **Comparison of Dynamic Naive Bayesian Classifiers and Hidden Markov Models for Gesture Recognition.** *Journal of Applied Research and Technology*, 9(1):81-102.

- La adaptación de sistemas de rehabilitación virtual basada en procesos de decisión de Markov.[29]

- La aplicación de Campos de Markov para mejorar los procesos de anotación y recuperación de imágenes.[30]

- El uso de redes bayesianas y diagramas de decisión para la selección de pozos para inyección en campos petroleros maduros.

- Modelado del estudiante para tutores inteligentes, incluyendo modelos relacionales probabilistas para laboratorios virtuales, representación del estado afectivo del estudiante basado en redes de decisión y un modelo basado en redes de decisión dinámicas para la secuencia y navegación de objetos de aprendizaje en ambientes de educación en línea.[31]

- Modelos para planeación basados en procesos de decisión de Markov que permiten coordinar robots de servicio al realizar tareas complejas, incluyendo un esquema para realizar acciones concurrentes y resolver conflictos.[32]

---

[29] Avila-Sansores, Sh., Orihuela-Espina, F., Sucar, L. E., and Álvarez-Cárdenas, P. (2013). **Adaptive Virtual Rehabilitation Environments.** ICML Workshop: Role of Machine Learning in Transforming Health, Atlanta, USA.

[30] Hernández-Gracidas, C., Sucar, L. E., Montes, M. (2013). **Improving Image Retrieval by Using Spatial Relations.** *Journal of Multimedia Tools and Applications,* Vol. 62:479-505.

[31] Sucar, L. E., Noguez, J. (2008). **Student Modeling.** En: O. Pourret, P. Naim, B. Marcot (eds.) Bayesian Belief Networks: **A Practical Guide to Applications,** Wiley and Sons, pp. 173-186.

[32] Corona, E., Sucar, L. E. (2011). **Task Coordination for Service Robots Based on Multiple Markov Decision Processes.** En: Sucar, L. E., Hoey, J., Morales, E. (eds.) Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions, IGI Global, Hershey.

Asimismo se han desarrollado en México herramientas de software genéricas, tales como VALIDATOR[33] (herramienta para validar información en bases de datos, que puede detectar y corregir diferentes tipos de errores), clasificadores bayesianos, semi-bayesianos, multi-dimensionales[34] y jerárquicos,[35] que se han incorporado a las herramientas abiertas WEKA/MEKA,[36] ASISTO[37] (sistema para ayuda de operadores de plantas eléctricas), PROMODEL[38] (ambiente Web orientado a servicios y dirigido por modelos para el desarrollo de sistemas bajo incertidumbre que permite generar aplicaciones Web de forma automática). También se publicó un libro sobre programación bayesiana[39] y otro sobre modelos gráficos probabilistas y sus aplicaciones.[40]

Además se han fundado en México empresas que comercializan productos basados en modelos probabilistas, con aplicaciones para el sector financiero, económico, médico, gobierno y de seguridad, entre otros; incluyendo Promagnus[41] (antes Probayes Américas), Cytron Medical y Sistemas Box.

[33] Herrera Vega, J., Orihuela-Espina, F., Morales, E. F., Sucar, L. E. (2012). **A framework for oil well production data validation.** Workshop on Operations Research and Data Mining (ORADM'2012) en 10th International Conference on Operations Research, Cancún, México.

[34] Sucar, L. E., Bielza, C., Morales, E., Hernández, P., Zaragoza, J., Larrañaga, P. (2014). **Multi-label Classification with Bayesian Network-based Chain Classifiers.** *Pattern Recognition Letters,* 41:14-22.

[35] Ramírez, M., Sucar, L. E., Morales, E. (2014). **Path Evaluation for Hierarchical Multi-label Classification.** *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference* (FLAIRS), pp. 502-507.

[36] http://www.cs.waikato.ac.nz/ml/weka/

[37] Reyes, A., Sucar, L. E., Morales, E. F. (2009) **AsistO: A Qualitative MDP-Based Recommender System for Power Plant Operation.** *Computación y Sistemas,* 13(1):5-20.

[38] López-Landa, R., Noguez, J. (2012). **PRoModel: a model-driven software environment that facilitates and expedites the development of systems that handle uncertainty.** En *Proceedings of the 2012 Symposium on Theory of Modeling and Simulation- DEVS* Integrative M&S Symposium, Society for Computer Simulation International, pp. 41.

[39] Bessière, P., Mazer, E., Ahuactzin, J. M., Mekhnacha, K., **Bayesian Programming,** Chapman and Hall/CRC, 2013.

[40] Sucar, L. E. **Probabilistic Graphical Models: Principles and Applications,** Springer, 2015.

[41] http://www.promagnuscompany.com

Entre las aplicaciones desarrolladas destacan:

- Redes bayesianas para el cálculo de riesgo operacional en los bancos.
- Suavizado de series de tiempo económicas basado en cadenas de Markov para la toma de decisiones en los ciclos de negocios.
- Sistemas para el conteo de vehículos o personas basados en filtros bayesianos, utilizados para auditorías de peajes, conteo en estacionamientos y medición de grado de actividad en centros comerciales.
- Modelos predictivos basados en redes bayesianas para integrar la información de las unidades de salud remotas en epidemiología, como fue el caso de la epidemia H1N1 y la predicción de apariciones de brotes de dengue.
- Medición del denominado efecto *bullwhip* (fluctuación de los pedidos a lo largo de una cadena de suministro) y la generación de rutas y órdenes de visita para el recorrido óptimo de vehículos de reparto en última milla.

1.10.4. Sistemas Multi-Agentes

En el área de los Sistemas Multi-Agentes ha habido desarrollos en México desde mediados de los años noventa. En el aspecto teórico, se ha trabajado en los formalismos de comunicación de agentes[42,43] por el grupo del Laboratorio de Informática Avanzada (LANIA) en Xalapa. Asimismo, investigadores mexicanos tuvieron una destacada participación en las propuestas para pro-

[42] Fallah-Seghrouchni, A. E., Lemaître, Ch. (2002). **A Framework for Social Agents' Interaction Based on Communicative Action Theory and Dynamic Deontic Logic.** MICAI'02: *Proceedings of the Second Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence.*

[43] Lemaître, Ch. (2000). **A Comprehensive Theory of Meaning for Communication Acts in Multi-Agent Systems.** ICMAS'00: *Proceedings of the Fourth International Conference on MultiAgent Systems (*ICMAS-2000).

tocolos de mercados basados en agentes[44] así como de "Instituciones Electrónicas".[45]

Otras propuestas mexicanas de investigación relacionada con SMA incluyen una investigación conjunta del Tecnológico de Monterrey con la Universidad de Carnegie Mellon[46] para modelar el problema del agente adversario. El grupo internacional "Agentes, Interacción y Complejidad" en el que participa el ITESM, Campus Querétaro,[47] investiga sistemas socio-técnicos, socio-económicos y socio-ecológicos, aplicados al diseño de infraestructura sostenible y resistente. Este grupo investiga y diseña sistemas en que entidades autónomas (ya sea humanos, organismos biológicos, agentes de software o robots) interactúan y ha desarrollado técnicas para obtener información de redes sociales y agentes humanos para retroalimentar un sistema de mejora de respuesta ante desastres naturales (proyecto ORCHID[48]). Este sistema tiene además la particularidad de presentar la información de manera que cualquier usuario pueda entenderla y reaccionar apropiadamente cuando ocurre un desastre.

Varios grupos de investigación mexicanos han trabajado en computación ubicua e inteligencia ambiental; en particular en el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), en el ITESM y en el INAOE. Estos trabajos involucran inteligencia ambiental y conciencia de la situación y el contexto, redes sociales, sensado participativo

[44] Rodríguez-Aguilar, J. A., Martín, F. J., Noriega, P., Garcia, P., Sierra, C. (1998). **Towards a test-bed for trading agents in electronic auction markets.** *AI Communications:* 11(1).

[45] Aldewereld, H., Dignum, F., García-Camino, A., Noriega, P., Rodríguez-Aguilar, J. A., Sierra, C. **Operationalisation of Norms for Electronic Institutions.** December 2006 Coordination, Organizations, Institutions, and Norms in Agent Systems II: AAMAS 2006 y ECAI 2006 International Workshops, COIN 2006 Hakodate, Japan, May 9, 2006 Riva del Garda, Italy, August 28, 2006. Revised Selected Papers.

[46] Garrido-Luna, L., Brena, R., Sycara, K. P. (1998). **Towards Modeling Other Agents: A Simulation-Based Study.** *Proceedings of the First International Workshop on Multi-Agent Systems and Agent-Based Simulation.*

[47] http://www.aic.ecs.soton.ac.uk/

[48] http://www.orchid.ac.uk/

y oportunista y cuidado pervasivo de la salud, entre otros. En el CICESE se desarrolló la plataforma SALSA,[49] específicamente diseñada para dar soporte a las aplicaciones de inteligencia ambiental. Varias instituciones mexicanas colaboraron en el proyecto europeo Ubi-Health,[50] orientado a la aplicación de cómputo ubicuo a la salud, incluyendo el desarrollo de técnicas de representación, inferencia y aprendizaje, particularmente cuando se tiene información escasa e incompleta.

Los agentes, ya sean humanos o computacionales, frecuentemente cambian sus políticas de acción. Recientemente en el INAOE se ha trabajado en agentes que se pueden adaptar a los cambios de otros agentes, basados en conceptos de teoría de juegos y aprendizaje por refuerzo; esto le permite al agente tomar la mejor decisión considerando un modelo de sus oponentes o colaboradores, según sea el caso. Para ello aprende un modelo del otro(s) agente(s) que se representa como un MDP, el cual se revisa y actualiza continuamente.[51] Esto se ha aplicado a agentes para los mercados de energía, área en la que el INAOE colabora con varias instituciones mexicanas en el proyecto del Centro Mexicano de Innovación en Energía Eólica.[52]

1.10.5. Ontologías

La investigación en ontologías se ha desarrollado desde finales de los años ochenta en varias instituciones como la Universidad de las Américas de Puebla (UDLAP), el Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV), la Benemérita Universidad Autónoma de Puebla (BUAP), el Colegio de Postgraduados de Chapingo, el Centro de Investigación en Compu-

[49] Rodríguez, M. D., Favela, J. (2012). **Assessing the SALSA architecture for developing agent-based ambient computing applications.** *Science of Computer Programming,* 77(1):46-65.

[50] http://www.ubihealth-project.eu/

[51] Hernandez-Leal, P., Rosman, B., Taylor, M. E., Sucar, L. E., Muñoz de Cote, E. (2016). **A Bayesian Approach for Learning and Tracking Switching, Non-Stationary Opponents.** *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS),* pp. 1315-1316.

[52] http://www.cemieeolico.org.mx/

tación del Instituto Politécnico Nacional (CIC-IPN) y la Universidad Autónoma Metropolitana (UAM), entre otras.

El proceso de adquisición de conocimiento de un dominio específico realizado por los humanos es una tarea lenta, costosa y con alta probabilidad de inconsistencia. Para abordar este problema, en el CINVESTAV se han hecho propuestas para la construcción automática de ontologías a partir del análisis de grandes cantidades de texto.[53] En dicha investigación se propusieron dos modelos de aprendizaje a partir de texto proveniente de documentos no estructurados en inglés. Estos modelos se basan en *Latent Dirichlet Allocation* (LDA)[54] y la Hipótesis Distribucional,[55] los cuales permiten descubrir de manera efectiva los temas cubiertos por los documentos del corpus de texto.

El enriquecimiento automático de las ontologías es un tema de creciente interés debido al enorme volumen de datos disponibles para ser incorporados como conocimiento activo en los sistemas inteligentes. En el CIC-IPN se han propuesto técnicas novedosas para adquirir nuevo conocimiento de manera incremental y automática, manteniendo la consistencia de la base de conocimientos.[56] En particular, se propuso el método *Ontology Merging*[57] (OM) que incluye un algoritmo para fusionar/unir dos ontologías (obtenidas de documentos de la Web) de manera automática (sin intervención humana) para producir una tercera ontología que considere el manejo de inconsistencias y redundancias entre las ontologías originales. El uso

[53] Ocampo-Guzman, I., Lopez-Arevalo, I., Sosa-Sosa, V. (2009). **Data-driven approach for ontology learning.** En *6th International Conference on Electrical Engineering, Computing Science and Automatic Control,* Toluca, México, pp. 1-6.

[54] Blei, D. M., Ng, A. Y., Jordan, M. I. (2003). **Latent dirichlet allocation.** *Journal of machine Learning research,* 3:993-1022.

[55] Sahlgren, M. (2008). **The distributional hypothesis.** *Italian Journal of Linguistics,* 20(1):33-54.

[56] Cuevas A.D., Guzmán-Arenas A. (2010). **Automatic Fusion of knowledge stored in Ontologies.** *Intelligent Decision Technologies* 4(1):5-19.

[57] Idem.

repetido de OM permite la adquisición de mucha información del mismo tópico. Otro procedimiento para construir ontologías de manera supervisada encuentra conceptos relevantes en forma de frases temáticas y relaciones no jerárquicas en el corpus de documentos.[58]

Otras aplicaciones mexicanas importantes de las ontologías incluyen:

- Proyecto GeoBase, del Colegio de Postgraduados de Chapingo, cuyo objetivo es construir una base ontológica geoespacial para apoyar la investigación geomática en la gestión de recursos agrícolas y naturales.[59] Los elementos de la base de datos geoespaciales se originan del análisis de imágenes y de la manipulación de datos geográficos. Este proyecto posibilita las consultas semánticas basadas en ontologías.

- Ontología para la creación del Sistema Sinóptico de Calidad Ambiental (EQSS),[60] que integra los datos requeridos a partir de sitios de Internet y datos concentrados por diferentes organismos como INEGI, CONABIO, SEMARNAT, CNA, entre otros. La ontología propuesta se basa en el conocimiento del sistema EQSS el cual posee una arquitectura similar a la de los sistemas expertos para la toma de decisiones con conocimiento sobre la calidad ambiental y la interacción con el Sistema de Información Geográfica (SIG).

- Ontología genérica para video-vigilancia, que incluye los elementos visuales, objetos y acciones que son relevantes para los sistemas de

[58] Toledo I., Martínez-Luna G., Guzmán-Arenas A. (2012). **Automatic building of an ontology from a corpus of text documents using data mining tools.** *Journal of Applied Research and Technology* 10(3):398-404.

[59] Fernández Y., Medina-Ramírez C., Soria-Ruíz J. (2014). **Geographic metadata and ontology based satellite image management.** *IEEE Geoscience and Remote Sensing Symposium,* pp. 117-120.

[60] Cabrera-Cruz, R. B. E., Alarcón-Ruiz, E., Rolón-Aguilar, J. C., Nava-Díaz, S. W., Otazo-Sánchez, E. M. Aviléz, R. P. (2015) **Developing Ontology Systems as a Base of an Environmental Quality Management Model in México.** *Journal of Environmental Protection,* 6:1084-1093.

video vigilancia automática.[61] Mediante la ontología se pueden realizar procesos de razonamiento para inferir situaciones de interés a partir de detecciones elementales; por ejemplo, si se detecta a una persona que lleva un objeto, lo deja y sigue caminando, se podría inferir una posible situación peligrosa al haber un objeto abandonado.

1.10.6. Combinaciones de métodos

Otras investigaciones combinan varios de los métodos descritos en las secciones 1.2, 1.3 y 1.4, con teoría de juegos, razonamiento probabilista, métodos sub-simbólicos, etc. Por ejemplo, en el CINVESTAV se desarrolló un modelo matemático y de simulación computacional de juegos complejos, colectivos y de tablero,[62] así como la modelación y simulación de la selección de estrategias en dichos juegos. Algunas situaciones específicas incluyen: i) Deportes colectivos como béisbol, fútbol americano y fútbol soccer,[63] ii) De bienes sociales como la "Tragedia de los Comunes"; iii) De tablero como el juego de Go.[64] Para la selección de estrategias en los juegos deportivos y sociales se utiliza el concepto de equilibrio, en particular el de Nash. Para el aprendizaje de estrategias se utilizan las redes neuronales, tanto clásicas como profundas. La probabilidad y estadística se utilizan para determinar la frecuencia de ocurrencia de las jugadas y el nivel de los jugadores o equipos particulares. La estrategia se complementa con el modelo de Ising,[65] sobre interacción de elementos en fenómenos de alta incertidumbre y con transiciones de fase.

---

[61] Hernández-Leal, P., Escalante, H. J., Sucar, L. E. (2017). **Towards a Generic Ontology for Video Surveillance.** En E. Sucar et al. (Eds.): AFI 2016, LNICST 179, Springer, pp. 3-7.

[62] Alvarado, M., Yee, A. (2012). **Nash equilibrium for collective strategy reasoning.** *Journal of Expert Systems with Applications,* 39(15):12014-12025.

[63] Yee, A., Campirán, E., Alvarado, M. (2016). **Gaming and strategic choices to American football game.** *Int. journal of mathematical and computational methods,* 1:355-371..

[64] Yee, A., Alvarado, M. (2012). **Pattern Recognition and Monte-CarloTree Search for Go Gaming Better Automation.** En *Ibero-American Conference on Artificial Intelligence, IBERAMIA-2012,* Springer Berlin Heidelberg, pp. 11-20.

[65] Lee, T. D., Yang, C. N. (1952). **Statistical theory of equations of state and phase transitions. II. Lattice gas and Ising model.** *Physical Review,* 87(3):410.

1.11. Retos y perspectivas

Las tecnologías de conocimiento y razonamiento darán lugar en los próximos años a nuevos desarrollos y aplicaciones de los diferentes enfoques, metodologías y técnicas.

Aunque en la actualidad los sistemas proposicionales, lógicos y lingüísticos no son muy populares, debido principalmente al auge del aprendizaje automático, este tipo de representaciones seguirán siendo de gran utilidad especialmente cuando se requiera transparentar lingüísticamente la cadena inferencial y las relaciones entre los efectos y sus causas.

Los modelos probabilistas, por su parte, tendrán un campo de aplicación muy amplio en problemas con incertidumbre, incluyendo aplicaciones en la salud, el sector energía, juegos (serios y de entretenimiento), robótica de servicio, etc. Otra área con gran dinamismo es la de los modelos gráficos causales, que no sólo representan relaciones de dependencia estadística sino también relaciones causa-efecto. En México se empieza a incursionar en esta área en el modelado de las relaciones de conectividad efectiva en el cerebro mediante redes bayesianas causales,[66] así como en modelos predictivos para la detección temprana de fragilidad en los adultos mayores.

Un área más en desarrollo es la combinación de representaciones basadas en la lógica de predicados y representaciones probabilistas, con el fin de capitalizar las bondades de ambos sistemas: alta expresividad y manejo de incertidumbre. Estas representaciones tienen diversas variantes y se conocen como modelos relacionales probabilistas. En México se han aplicado al reconocimiento de objetos en imágenes.[67]

---

[66] Montero-Hernández, S. A., Orihuela-Espina, F., Herrera-Vega, J., Sucar, L. E. (2016). **Causal Probabilistic Graphical Models for Decoding Effective Connectivity in Functional Near InfraRed Spectroscopy,** FLAIRS, AAAI Press, 2016.

[67] Ruiz, E., Sucar, L. E. (2014). **Recognizing Visual Categories with Symbol-Relational Grammars and Bayesian Networks,** CIARP, LNCS 8827, pp. 540-547.

Una de las tendencias principales en las propuestas de sistemas basados en agentes es la integración con ambientes de cómputo ubicuo e Internet de las Cosas (IoT, por sus siglas en inglés).[68] Los escenarios del cómputo ubicuo, en que múltiples procesadores y sensores están integrados al ambiente, proveen servicios de manera transparente, pero se requiere coordinarlos de forma distribuida, siguiendo los métodos de los sistemas multi-agentes, tales como la negociación. Por ejemplo, la generación actual de domótica, basada en dispositivos conectados que se controlan por medio del celular y de reglas simples pronto llegará a su límite —no podemos controlar 50 dispositivos de forma centralizada si hay interacciones complejas entre ellos— y será necesario emplear un enfoque de delegación, típico de los agentes inteligentes. Al pasar del Internet de los usuarios humanos al IoT se abren grandes oportunidades. Para abordar esta problemática se integró la línea de investigación "Sistemas Ubicuos Multi-agentes" de la Red Temática RedTIC desde el 2011 en la que participaron varios miembros de AMEXCOMP.

Otra tarea pendiente es el diseño e implementación de procesos autónomos denominados agentes ontológicos para modelar y recuperar información disponible en repositorios digitales y resolver los problemas de interoperabilidad semántica.

Aunque la investigación sobre ontologías y modelos basados en grafos ha sido intensa durante varias décadas todavía queda mucho por hacer. Particularmente, es necesario realizar mayor experimentación y explorar más profundamente las propuestas de modelado del conocimiento aplicadas a grandes volúmenes de datos como los disponibles en Internet.

[68] Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. (2013). **Internet of Things (IoT): A vision, architectural elements, and future directions.** Future generation computer systems, 29(7):1645-1660.

2. Aprendizaje e Inteligencia Computacional

2.1. Breve historia

Desde los inicios de la computación a mediados de los años cuarenta del siglo XX se pensó en la posibilidad de crear máquinas capaces de aprender. Posiblemente el primer artículo sobre aprendizaje computacional es el de McCulloch y Pitts[1] que muestra cómo emular compuertas lógicas proposicionales con modelos artificiales de redes neuronales. Por su parte, en el artículo seminal de 1950 Turing[2] propuso crear una máquina capaz de aprender como parte del programa para la construcción de computadoras que exhibieran una inteligencia general, a la cual le llamó la Máquina Niño (*Child Machine*).[3] Desde entonces, en buena medida inspirados por estas ideas, los científicos computacionales empezaron a crear sistemas básicos de aprendizaje desde diferentes perspectivas. Algunos intentaron emular los procesos dentro del cerebro mediante redes neuronales artificiales. Un poco más tarde se empezaron a demostrar pruebas de convergencia en redes neuronales sencillas o perceptrones (e.j., Rosenblatt[4]). Esta línea de investigación ha evolucionado

[1] McCulloch, W. S., Pitts, W. (1943). **A logical calculus of the ideas immanent in nervous activity**. *Bulletin of Mathematical Biophysics* 5: 115-133.

[2] Turing, A. (1950). **Computing Machinery and Intelligence**. *Mind* 59: 433-460.

[3] Morales, E. (2013). **Educando al "niño computacional" de Turing**. *Ciencias*, pp. 32-39.

[4] Rosenblatt, F. (1958). **The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain**. *Cornell Aeronautical Laboratory, Psychological Review* 5(6):386-408.

de manera continua hasta nuestros días y actualmente se cuenta con redes neuronales de muchas capas para implementar el llamado aprendizaje profundo (*Deep Learning*). Otros investigadores buscaron cómo aprender a jugar juegos sencillos mediante estrategias de búsqueda, aprendizaje de posiciones y ajuste de una función de recompensa.[5] De esta última se derivó el aprendizaje por refuerzo, en el que un agente computacional aprende cuál es la mejor acción a realizar en cada instante de tiempo para maximizar una función de recompensa. Recientemente se desarrolló un sistema que le ganó al campeón mundial de Go[6] el cual combina aprendizaje por refuerzo con aprendizaje profundo. En los años sesenta y setenta se desarrollaron algoritmos para aprender representaciones simbólicas, como Hunt[7] y Feigenbaum.[8] Algunos de los algoritmos más utilizados en aprendizaje computacional generan árboles de decisión y sus variantes, las cuales se han aplicado a una gran cantidad de situaciones reales. En la misma época otros investigadores se enfocaron al desarrollo de sistemas inspirados en genética y evolución, por ejemplo Rechenberg,[9] Schwefel,[10] Fogel[11] y Holland.[12] De aquí surgieron varios algoritmos que se engloban en la computación evolutiva y se utilizan principalmente para resolver problemas de optimización complejos en los cuales no es posible utilizar las técnicas matemáticas tradicionales.[13]

---

[5] Samuel, A. (1959). **Some Studies in Machine Learning Using the Game of Checkers**. *IBM Journal*. 3 (3): 210-229.

[6] Silver, D., et. al. (2016). **Mastering the game of Go with deep neural networks and tree search**. *Nature*, 529: 484-489.

[7] Hunt, E. B., Hovland, C. I. (1963). **Programming a Model of Human Concept Formation**. en E. A. Feigenbaum y J. Feldman (Eds.) *Computers and Thought*. MacGraw-Hill, 310-325.

[8] Feigenbaum, E. A. (1963). **The Simulation of Verbal Learning Behaviour**. En E.A. Feigenbaum and J. Feldman (Eds.) *Computers and Thought*. MacGraw-Hill, 310-325.

[9] Rechenberg, I. (1971). **Evolutionsstrategie: Optimierung technischer Systeme nach prinzipien der biologischen Evolution**, TU Berlin.

[10] Schwefel, H. P. (1975). **Evolutionsstrategie und numerische Optimierung**, TU Berlin.

[11] Fogel, L. J. Owens, A. J., Walsh, M. J. **Artificial Intelligence through Simulated Evolution**, John Wiley & Sons, (1966).

[12] Holland, J. (1962). **Outline for a Logical Theory of Adaptive Systems**. *JACM* 9: 297-314.

[13] Ver capítulo de Computación Evolutiva en este texto.

Además de aprender modelos u optimizar funciones, algunos investigadores trataron de formalizar el razonamiento humano de manera alternativa a los enfoques lógicos tradicionales, como la lógica de predicados. Uno de éstos fue Lofti Zadeh que creó la lógica difusa[14] en la cual las expresiones no sólo toman valores de verdadero y falso (1 y 0) sino también valores reales entre estos dos extremos. Esta línea de investigación, al igual que las anteriores, se ha desarrollado y refinado de forma continua hasta nuestros días. Todas éstas forman parte del Aprendizaje e Inteligencia Computacional, que son el tema de este capítulo.

2.2. Aprendizaje Computacional

El Aprendizaje Computacional (*Machine Learning*) se dirige al estudio y desarrollo de algoritmos con la capacidad de mejorar su desempeño con la experiencia. Otros nombres comúnmente utilizados para denotar el área son Aprendizaje Automático, Aprendizaje Maquinal y Aprendizaje de Máquina. El énfasis de esta especialidad es crear algoritmos que puedan inducir modelos automáticamente sin la intervención o asistencia humana. Dado que la capacidad de aprender es una característica distintiva de la inteligencia no es de extrañar que el Aprendizaje Computacional sea un área central de la Inteligencia Artificial.

Para entender la utilidad práctica y relevancia del área podemos pensar en cómo se resuelven comúnmente muchos problemas. Generalmente las personas crean modelos o programas para resolverlos; sin embargo, algunos son difíciles de formalizar, no siempre se cuenta con expertos en el dominio o se tiene demasiada información. Por ejemplo, si se requiere escribir un programa para reconocer personas, no es claro cómo programarlo, a qué expertos consultar o cómo limitar el número de fotos y videos de personas existentes. Los algoritmos de aprendizaje permiten generar automáticamente dichos programas o modelos a partir de datos y con ellos resolver problemas.

[14] Zadeh, L. (1965). **Fuzzy Sets**. *Information Sciences* 8: 338-353.

Esto claramente abre una gran cantidad de posibles aplicaciones que son difíciles de formalizar o programar por una persona y que requieren contar con grandes cantidades de datos para su desarrollo.

El reciente éxito y publicidad que han recibido estos algoritmos se deben principalmente a tres factores: i) disponibilidad de una gran cantidad de datos de todo tipo, ii) máquinas más poderosas con altas capacidades de almacenamiento y mayores velocidades de procesamiento, y iii) la facilidad de utilizar estas herramientas por las empresas para resolver problemas reales.

Algunos autores clasifican al área de aprendizaje computacional en términos de tres tipos o categorías de modelos: geométricos, probabilístas y lógicos. Otra clasificación posible se da en términos de las tareas que abordan. Una caracterización burda, la cual se adopta en este capítulo, es: 1) Sistemas que crean descripciones, resúmenes, identifican prototipos o casos anómalos a partir de datos; 2) sistemas que segmentan/agrupan datos automáticamente; 3) sistemas que crean automáticamente modelos para predecir salidas, ya sea para tareas de clasificación (salidas discretas) o de regresión (salidas continuas); 4) sistemas que encuentran automáticamente dependencias entre variables, ya sean éstas probabilistas, funcionales, asociativas o causales; y 5) sistemas que aprenden a crear planes o estrategias de control para resolver problemas de decisión secuenciales. A continuación se presenta una breve descripción de cada una de éstas.

2.2.1. Resúmenes y detección de prototipos

Con el incremento acelerado en la generación de datos es importante tener la capacidad de crear automáticamente resúmenes que resalten la información más relevante. Los estadísticos entendieron rápidamente la utilidad de esta funcionalidad y generaron medidas, como la media o la moda, y medidas de dispersión, como la desviación estándar, para tener una idea rápida de la naturaleza de los datos. En aprendizaje computacional algunos investigadores han

tratado de encontrar los datos prototípicos dentro de una gran cantidad de datos para hacer inferencias, facilitando el análisis y reduciendo la velocidad de procesamiento. Las personas que trabajan con análisis de texto también han creado sistemas que generan resúmenes de textos automáticamente para su fácil lectura y comprensión. También se pueden usar técnicas para reducción de datos identificando únicamente los datos atípicos, los cuales son muchas veces los de mayor interés.

2.2.2. Segmentación

Los sistemas que segmentan automáticamente datos se conocen como algoritmos de agrupamiento o *clustering*.[15,16] También se conocen como algoritmos de aprendizaje no supervisado. La idea de estos algoritmos es agrupar automáticamente datos de tal forma que los elementos en cada grupo se parezcan entre sí, bajo algún criterio, y no se parezcan –o se parezcan poco– con los elementos de otros grupos, bajo el mismo criterio. Por ejemplo, los datos de clientes de empresas comerciales se pueden agrupar bajo ciertos criterios para diseñar campañas publicitarias diferenciadas. Los grupos pueden ser disjuntos o tener traslapes, organizarse en jerarquías, ser exhaustivos, la pertenencia a los grupos puede ser probabilísta, etc. La medida de similaridad se define generalmente por la proximidad que existe entre los elementos en un espacio multidimensional y hay una gran cantidad de medidas posibles para este efecto. Si los datos son exclusivamente numéricos se pueden usar medidas de distancia, como la Euclideana. Si los datos no sólo se describen por números, se usan otras medidas de similaridad que dejan de cumplir con las propiedades de las métricas. También se les puede dar más peso a algún tipo variable o atributo que a otro(s) para que sus valores tengan más influencia en la creación de los grupos. Por otro lado, también hay diferentes estrategias para formar los grupos, algunas agrupan a los elementos gradualmente, otras

[15] Estivill-Castro, V., Yang, J. (2001). **Non-crisp Clustering by Fast, Convergent, and Robust Algorithms**. En *Proc. of PKDD*, pp. 103-114.

[16] Estivill-Castro, V., Houle, M. E. (2001). **Data Structures for Minimization of Total Within-Group Distance for Spatio-temporal Clustering**. En *Proc. PKDD*, pp. 91-102.

parten de un grupo global el cual se separa en subgrupos de forma recurrente, otros parten de particiones formadas originalmente de forma aleatoria las cuales se modifican hasta cumplir con un criterio dado de antemano, etcétera.

Uno de los algoritmos más conocidos de *clustering* es el de *k-means*. El algoritmo es relativamente simple y parte del número de particiones *k* que se quieren formar. El algoritmo selecciona aleatoriamente *k* elementos y calcula la similaridad de todos los datos con dichos *k* elementos. Cada dato se asocia al elemento *k* más cercano con lo que se forman *k* grupos y se obtiene su centroide o elemento "promedio". El proceso se realiza de manera recurrente con respecto a los centroides, con lo que se forman, posiblemente, nuevos grupos. El proceso se repite hasta que los centroides no cambian. Existen muchas extensiones y variantes a este algoritmo.

Los datos se pueden agrupar desde diferentes perspectivas y de diferentes formas. Se han desarrollado medidas de calidad para los grupos que se forman tomando en cuenta la densidad de elementos, qué tan compactos son los grupos, etc. Algunos retos del área son cómo lidear con diferentes representaciones en los datos, cómo crear grupos con formas arbitrarias, cómo incluir restricciones y cómo hacerlos interpretables, entre muchos otros.

2.2.3. Predicción

Existen algoritmos que crean automáticamente modelos para predecir salidas, ya sea para tareas de clasificación (salidas discretas) o de regresión (salidas continuas). La tarea de predicción es posiblemente la que más atención ha recibido en la literatura por su gran cantidad de posibles aplicaciones. A estos algoritmos también se les conoce como aprendizaje supervisado. La información de entrada dada a estos algoritmos son ejemplos, descritos generalmente con atributos o variables, asociados con una etiqueta. Cuando la etiqueta es nominal se conocen como algoritmos de clasificación, cuando es numérica se conoce como regresión o estimación.

2.2.3.1. Clasificación

La meta de los algoritmos de clasificación es inducir automáticamente un modelo a partir de los datos para que dada la descripción de un nuevo ejemplo produzca una etiqueta de salida. Por ejemplo, a partir de una descripción de atributos de un paciente prediga una enfermedad o dados los valores de variables de una línea de producción detecte una falla. En la tarea de clasificación existe una gran cantidad de algoritmos como árboles de decisión, máquinas de soporte vectorial y aprendizaje basado en instancias, entre otros. En este capítulo se describen brevemente estos tres enfoques.

En un árbol de decisión cada nodo representa un atributo y sus ramas los posibles valores que dicho atributo puede tomar. En caso de atributos continuos, éstos se dividen automáticamente en dos rangos que son mayores o menores que un cierto valor. Las hojas de los árboles son los valores de las clases que queremos predecir. El algoritmo selecciona de manera heurística el atributo que mejor particiona los datos en términos de las clases. Una vez que selecciona un atributo, el algoritmo se vuelve a aplicar a los datos de cada una de sus ramas hasta que: i) se queda con ejemplos de una sola clase y etiqueta la hoja con esa clase, ii) se queda sin ejemplos y etiqueta la hoja con la clase mayoritaria de su padre, o iii) se queda sin atributos y etiqueta la hoja con la clase mayoritaria de sus datos. Para clasificar un nuevo ejemplo, se recorre el árbol de acuerdo a los valores de los atributos del ejemplo y al llegar a una hoja se regresa la clase asociada a esa hoja. Existen varias medidas de selección de atributos, pero la más conocida se basa en ganancia de información la cual selecciona el atributo que reduce más la entropía o dispersión de las clases en los datos. Esto es, el atributo en donde las datos contenidos en sus ramas tienen más claramente definidas alguna de las clases.

Otro algoritmo muy usado en clasificación son las máquinas de soporte vectorial o de vectores de soporte. Una forma de construir un clasificador entre dos clases es tratar de encontrar un plano, o hiperplano en caso de

más dimensiones, que mejor divida a las clases. En las máquinas de soporte vectorial el encontrar ese hiperplano se plantea como un problema de optimización en donde se busca el plano que divida a las clases y que esté más alejado de los ejemplos de las clases contrarias. Como en muchos problemas las clases no son linealmente separables (no existe un hiperplano que las divida perfectamente) se recurre a dos estrategias; la primera es usar variables de holgura, lo cual permite tener "excepciones" de algunos ejemplos, y la segunda, mucho más conocida e importante, es la que se conoce como el "*kernel trick*", que consiste en crear una representación de más dimensiones que la original con la esperanza de que en ese nuevo espacio se pueda encontrar un mejor hiperplano.

Finalmente, están los algoritmos de aprendizaje basados en instancias, los cuales en vez de inducir automáticamente un modelo que sirva para predecir una clase almacenan todos los datos y buscan cuál o cuáles son los datos más parecidos al ejemplo a clasificar y los utilizan para predecir un valor. El algoritmo más conocido es el de *k* vecinos más cercanos. La idea es obtener del conjunto de datos los *k* elementos que sean más parecidos –bajo cierta medida– al ejemplo que se quiere clasificar, analizar las clases de esos *k* vecinos y regresar la clase mayoritaria. Al igual que en los algoritmos de *clustering*, existen diferentes medidas de similaridad que se pueden utilizar para encontrar los vecinos más cercanos. Muy relacionado a estos algoritmos está el razonamiento basado en casos donde cada ejemplo, llamado caso, se representa por la descripción de un problema, la solución empleada y los resultados obtenidos. Cuando se quiere resolver un nuevo problema, se consultan las descripciones de los problemas en la base de casos, se obtienen los más parecidos y se utilizan sus soluciones, con posibles adecuaciones, para resolver el problema en cuestión.

La gran mayoría de los algoritmos de clasificación utiliza una representación de los ejemplos basada en pares artibuto-valor produciendo en muchos casos modelos proposicionales. Existen otros algoritmos que trabajan con

representaciones más expresivas utilizando grafos[17] o lógica de primer orden. Una de ellas es la programación lógica inductiva[18,19] la cual recibe de entrada datos e información relacional y produce modelos expresados en términos de relaciones. Esto permite inducir modelos que contengan funciones, variables y definiciones recursivas los cuales son más expresivos que los modelos generados por los algoritmos anteriores. Algunos de los retos de los sistemas de clasificación son el ruido (valores erróneos y faltantes) que puede existir en los datos, una gran cantidad de datos, pocos datos pero muchos atributos, muchas clases posibles, entre otros.

2.2.3.2. Regresión

La meta de los algoritmos de estimación o regresión es inducir automáticamente, a partir de los datos, un modelo que dada la descripción de un ejemplo produzca un valor continuo de salida. Por ejemplo, predecir el valor de una temperatura de salida en base a los valores de las variables de un proceso industrial. Algunos de los algoritmos más usados son las redes neuronales, regresión lineal, regresión localmente pesada, procesos gaussianos, etcétera.

Además del método clásico de mínimos cuadrados, posiblemente el algoritmo más conocido de regresión sea el de las redes neuronales. A continuación se describe la red más conocida como *feed-forward* con retro-propagación. Esta red neuronal es un grafo dirigido organizado por capas, en donde todos los nodos de una capa están conectados hacia todos los nodos de la siguiente capa y reciben información de todos los nodos de la capa anterior. Cada conexión está asociada con un peso. A cada nodo le llega la suma de todas las salidas de los nodos de la capa anterior multiplicadas por sus respectivos pesos y a ese valor generalmente se le aplica una función continua no

[17] Gonzalez, J.A. Holder, L.B. (2000). **Graph Based Concept Learning**. En *Proc. Seventeenth National Conference on Artificial Intelligence* (AAAI), 2000.

[18] Morales, E. F. (1997). **PAL: A Pattern-Based First-Order Inductive System**. *Machine Learning* 26: 227-252.

[19] MacKenney, R. (1999). **Learning using higher Order Functions**. En *Proc. Inductive Logic Programming (ILP)*, pp 42-46.

lineal diferenciable que produce la salida de ese nodo. Esa salida, junto con las salidas de todos los nodos, de esa capa, se propaga de la misma forma hacia la siguiente capa, hasta producir una o más salidas finales. Los pesos de las redes se inicializan aleatoriamente y el aprendizaje consiste en ajustar los pesos de la red para que con base en los valores de ejemplos conocidos se produzcan las salidas deseadas. Este ajuste se hace gradualmente alterando los pesos en la dirección que produce el máximo descenso en la superficie del error. Para esto se obtiene el negativo de la derivada o gradiente del error con respecto a los pesos y se suma en forma ponderada al peso actual. El error de salida se distribuye entre todos los pesos de las capas anteriores. Este proceso se realiza con todos los datos de entrenamiento varias veces hasta que los pesos producen resultados aceptables o cambian poco los pesos de la red.

Las redes nueronales fueron populares en los inicios de la Inteligencia Artificial y aunque dejaron de usarse por varios años recuperaron su popularidad cuando se difundió el algoritmo de entrenamiento que acabamos de describir; recientemente ha habido mucho interés a partir de que se logró entrenar redes de "muchas" capas (más de 5). Aunque algunas de las ideas originales de entrenamiento de muchas capas, conocido ahora como aprendizaje profundo, se plantearon hace tiempo, la disponibilidad de grandes cantidades de datos y las velocidades de procesamiento de las máquinas actuales fueron las que permitieron realizarlo. Estas redes se han utilizado con éxito en análisis de imágenes, procesamiento de lenguaje natural y en juegos.

Algunos de los retos del área son, como en todas las demás, la cantidad de ruido que exista en los datos, si se tienen datos numéricos y nominales, la posibilidad de realizar un sobre-ajuste en los modelos y el tiempo de entrenamiento de las redes.

2.2.4. Análisis de dependencias

Algunos algoritmos encuentran automáticamente dependencias entre variables, ya sean éstas probabilístas, funcionales, asociativas o causales. Estos mo-

delos permiten determinar si el valor de una variable depende de los valores de las otras. Por ejemplo, cuál es la probabilidad de que llueva dado que la presión atmosférica y la temperatura tienen ciertos valores, o qué tan frecuente ocurre y con qué confianza, que las personas que compran leche y pan, también compren mantequilla y mermelada.

Una de las técnicas más usadas para capturar dependencias probabilístas son los modelos gráficos probabilístas, como las redes bayesinas.[20] Estos modelos consisten de grafos acíclicos dirigidos en los que los nodos representan variables aleatorias y los arcos sus dependencias probabilistas, y cada grafo representa una distribución de probabilidad conjunta de varias variables. Este formalismo permite evaluar los valores más probables de un conjunto de variables dados los valores de otras variables, y modela el razonamiento probabilista basado en la regla de Bayes. Los algoritmos de aprendizaje de redes bayesianas inducen tanto la estructura de la red como las dependencias entre las variables y se expresan como tablas de probabilidad condicional de cada variable dados sus padres.[21] El aprendizaje paramétrico o de los valores de las tablas de probabilidad condicional se basa generalmente en obtener las frecuencias de los valores de las variables relacionadas. Las técnicas de aprendizaje estructural se basan en métodos que utiizan medidas de ajuste y estrategias de búsqueda, o en métodos basados en pruebas de independencia.

Uno de los métodos más usados, que forma la base de otros algoritmos de aprendizaje de redes bayesianas, es el de Chow y Liu.[22] En este algoritmo se obtiene la información mutua entre todos los pares de variables, se ordenan estas medidas de mayor a menor, donde la de mayor información mutua representa un grafo inicial (con dos nodos) y se agregan las siguientes ramas mientras no se formen ciclos hasta que se hayan incluido todas la variables.

[20] Ver el capítulo de Conocimiento y Razonamiento en este texto.

[21] Luis-Velázquez, R., Sucar, L. E., Morales, E. F. (2010). **Inductive Transfer for Learning Bayesian Networks**. *Machine Learning* 79 (1-2): 227-255.

[22] *Chow, C. K., Liu, C.N. (1968),* **Approximating discrete probability distributions with dependence trees.** *IEEE Transactions on Information Theory, IT-14 (3): 462–467.*

Algunas extensiones a este algoritmo permiten determinar la dirección de las ramas y construir grafos más sofisticados (poliárboles o redes multiconectadas). Dentro de los retos del área podemos mencionar el razonamiento con variables continuas y mixtas así como dominios en donde exista una gran cantidad de dependencias, lo cual complica el algoritmo de inferencia.

Otra forma de obtener relaciones entre variables es por medio de las reglas de asociación. Estos algoritmos buscan grupos de variables y valores que se repitan frecuentemente en los datos. Con estos grupos de variables y valores repetidos, construyen reglas que tienen subgrupos de pares atributo-valor tanto en el antecedente como en el consecuente. El algoritmo más conocido del área es el de *Apriori.*[23] En éste se buscan primero todos los pares atributo-valor que cumplan con un mínimo de repeticiones o soporte en los datos. Con estos se buscan todas las parejas de pares atributo-valor con el soporte deseado. Esto continúa con tercias y así sucesivamente, hasta que no exista ningún subconjunto de pares atributo-valor más grande con el soporte requerido. Con cada subconjunto encontrado se crean todas las posibles reglas que cumplan con un nivel de confianza (soporte del subconjunto entre el soporte del antecedente de la regla) formadas por las combinaciones de antecedentes y consecuentes de los elementos de los subconjuntos. Esta área está muy relacionada con los algoritmos de patrones frecuentes y los de descubrimiento de subgrupos, los cuales se basan en buscar subconjuntos de pares atributo-valor frecuentes. Uno de los retos de las reglas de asociación es cómo encontrar todos estos conjuntos frecuentes de manera eficiente.

Tanto las redes bayesianas como las reglas de asociación se pueden modificar para resolver problemas de clasificación, donde el interés es predecir el valor de la variable que determina a la clase. Aunque se han desarrollado también algoritmos para encontrar dependencias funcionales y causales, no han sido tan populares como las redes bayesianas y las reglas de asociación.

[23] Agrawal, R., Srikant, R. (1994). **Fast algorithms for mining assocation rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB,** pp. 487-499, Santiago, Chile.

2.2.5. Planeación

Finalmente, existen algoritmos que aprenden a crear planes o estrategias de control para resolver problemas de decisión secuencial. Por ejemplo, decidir una serie de acciones para cumplir una meta, como ganar un juego o controlar algún dispositivo, como un robot.

Posiblemente el más conocido en años recientes sea el aprendizaje por refuerzo.[24,25] En este tipo de aprendizaje un agente aprende por prueba y error cuál es la mejor acción a realizar en cada momento para maximizar el valor esperado de su recompensa total. Para aprender qué acción se debe realizar en cada estado (aprender una política) el agente selecciona una de sus posibles acciones, con la cual se mueve –posiblemente– a otro estado y recibe una recompensa. La selección de acciones es aleatoria inicialmente y se ajusta gradualmente tomando en cuenta la recompensa recibida y la acumulada en otros estados. Uno de los algoritmos más conocidos es el de Q-learning,[26] el cual ajusta gradualmente el valor de la recompensa total esperada de cada par estado-acción. El valor de cada par estado-acción se ajusta incrementalmente calculando la diferencia entre dicho valor y la suma de la recompensa recibida al tomar la acción y el máximo valor esperado de todos los siguientes pares estado-acción. Algunos de los retos de esta área son cómo lidiar con espacios de estados y acciones continuos, cómo aprender con muchas variables, qué hacer en ambientes no estacionarios y cómo reducir la velocidad de aprendizaje.

---

[24] Morales, E. F., Sammut, C. (2004). **Learning to Fly by Combining Reinforcement Learning with Behavioural Cloning.** En *Proc. of the Twenty-first International Conference on Machine Learning* (ICML), pp. 598-605.

[25] García, E. O., Muñoz de Cote, E. , Morales, E. F. (2014). **Transfer Learning for Continuous State and Action Spaces**. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)* 28 (7).

[26] https://en.wikipedia.org/wiki/Q-learning

2.2.6. Otros enfoques

Existen muchos otros algoritmos y estrategias de aprendizaje. Algunos que han perdido popularidad se basan en abducción como el aprendizaje basado en explicaciones, el aprendizaje de macro-operaciones, usados en planeación y los algoritmos de descubrimiento y aprendizaje por medio de analogías, entre otros.

Algunas de las variantes más utilizadas actualmente son los ensambles, en los cuales se inducen varios modelos a la vez y se combinan sus resultados para resolver alguna tarea. También se han desarrollado algoritmos de aprendizaje por transferencia, en donde se utiliza información de una tarea resuelta para resolver otra de manera más eficiente y/o más efectiva.

Dentro de los algoritmos de clasificación se han desarrollado algoritmos de aprendizaje semi-supervisado, en donde se combinan datos etiquetados y no etiquetados (esto es, con y sin una clase asociada) para tratar de construir un mejor clasificador. También se han desarrollado algoritmos para tratar de predecir un conjunto de clases al mismo tiempo o algoritmos multi-etiqueta.[27,28]

Finalmente, existen algoritmos que buscan que un agente nunca deje de aprender, lo que se conoce como *life-long learning*. Existe mucha expectativa y algunos investigadores creen que dada la velocidad con la cual se desarrolla el área, pronto se va a lograr crear una inteligencia artificial superior a la de los humanos.

---

[27] Zaragoza, J. H., Sucar, L. E., Morales, E. F., Bielza, C., Larrañaga, P. (2011). **Bayesian Chain Classifiers for Multidimensional Classification**. En *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2192-2197.

[28] Hernández, J., Sucar, L. E., Morales, E. F. (2014). **Multidimensional hierarchical classification.** *Expert Systems with Applications* 41 (17): 7671-7677.

2.3. Lógica Difusa

La Lógica Difusa es un enfoque de computación basado en grados de verdad en lugar del tradicional verdadero/falso (1 ó 0) de la lógica Booleana.[29] De acuerdo a Zadeh, la membresía de un elemento a un conjunto difuso es gradual con valores en el intervalo [0, 1]. Un ejemplo clásico es clasificar individuos en el conjunto de personas altas considerando su estatura. En los conjuntos típicos o duros, un elemento pertenece o no al conjunto, por lo que es necesario establecer un límite inferior (ej. 1.75 mts.) arriba del cual todos son altos y debajo ninguno lo es. En los conjuntos difusos la membresía es gradual y alguien que mide 1.75 mts. es alto en un grado, por ejemplo, de 0.7, e incluso uno que mide 1.70 mts. se puede considerar alto en un grado, por ejemplo, de 0.3; por su parte quien mida 1.82 mts. es indudablemente alto y su pertenencia al conjunto es 1. Todos los que midan más de 1.82 seguirán siendo altos en un grado de 1 y quienes midan, por ejemplo, 1.60 o menos no pertenecerán a dicho conjunto y su grado de pertenencia será 0.

Existen diferentes operaciones para combinar valores de membresía en conjuntos difusos: una intersección difusa es el valor menor o mínimo de la membresía de cada elemento en ambos conjuntos; la unión de conjuntos difusos es el valor más alto o máximo de los dos valores difusos y el complemento es la diferencia entre su valor y 1.

También existen definiciones para operaciones entre relaciones difusas, las cuales se establecen entre dos elementos en algún grado de pertenencia entre 0 y 1. Sean $R(X, Y)$ y $S(Y, Z)$ dos relaciones difusas donde X, Y y Z son conjuntos difusos; hay varias operaciones binarias aplicables sobre ellas que resultan en una tercera relación también difusa, como el Producto Círculo o *Circlet*, el Producto Subtriángulo, el Producto Supertriángulo y el Producto Cuadrado.

[29] Zadeh, L. (1963). **Fuzzy Sets**. *Information Sciences* 8: 338-353.

Los productos relacionales difusos se han utilizado como base de las Redes Neurales Relacionales Difusas con las que se han desarrollado diversas aplicaciones de clasificación por el grupo del INAOE,[30] principalmente para el reconocimiento del llanto del bebé con fines de diagnóstico.

Las operaciones entre relaciones difusas se aplican también a los sistemas expertos. Por ejemplo, sean X, Y y Z un conjunto de pacientes, síntomas y enfermedades respectivamente. Mediante relaciones difusas y sus operaciones se pueden representar proposiciones como: el paciente x muestra el síntoma y en grado G; el síntoma y es uno de los que caracterizan la enfermedad z en grado G; el paciente x tiene al menos uno de los síntomas de la enfermedad z en grado G; x es un síntoma de la enfermedad z en grado G; x tiene todos los síntomas de la enfermedad z y quizás más en grado G; o x tiene exactamente los síntomas de enfermedad z y no otros en grado G; donde G es un valor real entre 0 y 1.

Un Sistema Experto Difuso (SED)[31] utiliza este tipo de proposiciones difusas y lógica difusa en vez de booleana para razonar acerca del problema; específicamente, consiste de una colección de funciones de membresía y de un conjunto de reglas de forma:

SI x es A ENTONCES y es B

donde x y y son variables lingüísticas y A y B son valores lingüísticos determinados por conjuntos difusos en el universo de discurso X y Y, respectivamente; por ejemplo:

Regla 1: SI *la velocidad* es *rápida* ENTONCES *distancia_de_frenado* es *larga*

[30] Suaste-Rivas, I., Reyes-Galaviz, O. F., Diaz-Mendez, A., Reyes-Garcia, C. A. (2004). **Implementation of a Linguistic Fuzzy Relational Neural Network for Detecting Pathologies by Infant Cry Recognition**. En LNCS 3315: Advances in Artificial Intelligence, Edited by C. Lemaitre, C. A. Reyes and J. Gonzalez, Springer, Berlin, pp. 953-962.

[31] Baghel, A., Tilotma, S. (2013). **Survey on Fuzzy Expert System**. *International Journal of Emerging Technology and Advanced Engineering Website*, 3(12):230-233.

Regla 2: SI *la velocidad* es *lenta* ENTONCES *distancia_de_frenado* es *corta*

El rango de la variable *velocidad* (el universo del discurso) es de 0 a 220 km/h, pero esta gama incluye conjuntos difusos, tales como *lento*, *medio* y *rápido*. El universo del discurso de la variable *distancia_de_frenado* puede estar entre 0 y 300 mts. e incluye los conjuntos difusos *corta*, *media* y *larga*. Así, las reglas difusas se refieren a conjuntos difusos y los SED pueden reducir hasta en un 90 por ciento el número de reglas requeridas para modelar un dominio en relación a un sistema experto convencional.

Otra marcada diferencia con respecto a los sistemas expertos convencionales es el proceso de inferencia, el cual consiste básicamente de cuatro etapas:

1. Fusificación: las funciones de membresía definidas sobre las variables de entrada son aplicadas a sus valores actuales, para determinar el grado de verdad de cada premisa de la regla.

2. Inferencia: se puede definir como un proceso de mapeo de una entrada a una salida, utilizando la teoría de conjuntos difusos.

3. Composición: todos los subconjuntos difusos asociados a cada variable de salida se combinan para formar un subconjunto difuso único de salida.

4. Defusificación (opcional): se emplea cuando es útil convertir el conjunto de salidas difusas a un número preciso (número duro). Las dos técnicas más comunes de defusificación son *Centroide Ponderado* y *Máximo*.

Algunos de los beneficios de los SED son reducir el costo de desarrollo de aplicaciones, los costos de ejecución y los costos de mantenimiento, ya que al emplear un número limitado de reglas para codificar el conocimiento de alto

nivel son menos propensos a errores, manejan información vaga, incierta e imprecisa y son mucho más fáciles de mantener.

2.3.1. Sistemas de control difuso

Una de las áreas de aplicación más exitosa de la lógica difusa es el control difuso. Los sistemas de éste se forman con reglas de inferencia difusa, donde cada regla tiene como antecedentes y consecuentes las condiciones de pertenencia de variables de estado y las acciones de control respectivamente.[32] El primer sistema fue diseñado para mantener la presión de una caldera y la velocidad de su pistón constante en una máquina de vapor.

2.3.2. Mapas cognitivos difusos (Fuzzy Cognitive Maps, FCM)

Ron Axelord introdujo los mapas cognitivos como una manera formal de representar el conocimiento científico social y modelar la toma de decisiones en sistemas sociales y políticos.[33] El objetivo principal de construir un mapa cognitivo en torno a un problema es predecir las decisiones que se tomarán, dejando que los temas relevantes interactúen entre sí. Estas predicciones se pueden usar para determinar si una decisión es consistente con toda la colección de afirmaciones causales declaradas en el sistema.

El término mapa cognitivo difuso (FCM) fue acuñado por Bart Kosko[34] para describir un modelo con dos características principales:

1. Las relaciones causales entre los nodos son difusas. En vez de usar solamente los signos para indicar causalidad positiva o negativa, se asocia un número a la relación para expresar el grado de proximidad entre dos conceptos.

[32] Mc Neill, D., Freiberger P. **Fuzzy Logic**, Simon & Shuster, New York, 1993.
[33] Axelrod, R. **Structure of Decision**, Princeton University Press, US, 1976.
[34] Bart, K. (1986). **Fuzzy Cognitive Maps**, *Int. J. Man-Machine Studies* 24:65-75.

2. El sistema es dinámico con retroalimentación, por lo que cambiar un nodo afecta a otros, lo que a su vez puede afectar al nodo que inicia el cambio. La presencia de retroalimentación añade un aspecto temporal al funcionamiento de los FCM ya que su estructura se puede ver como una red neuronal artificial recurrente, donde los conceptos se representan por neuronas y las relaciones causales por enlaces ponderados o bordes que las conectan.

2.4. Algoritmos Evolutivos[35]

Los Algritmos Evolutivos (AE) son técnicas de solución de problemas de optimización y búsqueda que utilizan operadores de distintos tipos y que imitan procesos evolutivos como elementos clave para explorar y explotar los espacios de búsqueda. Ésta es una técnica de búsqueda basada en la teoría de la evolución de Darwin que ha cobrado tremenda popularidad alrededor del mundo durante los últimos años. Entre los algoritmos evolutivos más conocidos se encuentran los Algoritmos Genéticos, Programación Evolutiva, Estrategias Evolutivas y la Programación Genética. Los AE mantienen una población de individuos que evolucionan de acuerdo a reglas de selección y otros operadores, que se conocen como *operadores genéticos* como la *cruza, recombinación y mutación.*

El AE más conocido es el algoritmo genético. Básicamente, este algoritmo mantiene una "población" de soluciones del problema que se aborda. La idea es generar nuevas soluciones en un proceso iterativo que pretende converger a una solución que satisfaga cierto criterio de paro establecido por el usuario. Nuevas soluciones se generan mediante la combinación y alteración de las soluciones más "aptas", donde la aptitud de las soluciones se evalúa mediante la función objetivo del problema bajo análisis.

[35] Para mayor información ver capítulo de Cómputo Evolutivo en este texto.

2.5. Aplicaciones

En México se han desarrollado una gran cantidad de aplicaciones que utilizan técnicas de Aprendizaje e Inteligencia Computacional, como sigue:

2.5.1. Medicina

Un área en donde se han realizado varios desarrollos es la de análisis de llanto de bebés. En particular, una de estas aplicaciones selecciona automáticamente un modelo para clasificar patrones de llanto infantil con fines de diagnóstico.[36] La selección se hace por medio de un algoritmo genético y permite generar modelos personalizados para el tipo de muestras de llantos infantiles. Como parte de un proyecto integral para el análisis y clasificación de llanto infantil se han desarrollado otros sistemas con computación suave[37] donde se comparan sistemas evolutivos con redes neurales para clasificar patologías a partir del llanto infantil. En otro trabajo[38] se presenta un sistema hibrido que consiste en una Máquina de Soporte Vectorial Difusa dedicado al reconocimiento automático del llanto de bebé; en otro más del mismo grupo se implementa una Red Neuronal Relacional Difusa para la detección de patologías a partir del llanto infantil.[39]

---

[36] Rosales-Pérez, A., Reyes Garcia, C. A., Gonzalez, J. A., Reyes-Galaviz, O., Escalante, H. J., Orlandi, S. (2105). **Classifying Infant Cry Patterns by the Genetic Selection of a Fuzzy Model**, *Biomedical Signal Processing and Control*, 17, pp. 38-46.

[37] Reyes Galaviz, O. F., Reyes Garcia, C. A. (2005). **Infant Cry Classification to Identify Hypo Acoustics and Asphyxia comparing an Evolutionary-Neural System with a Neural Network System**. En LNAI 3789*: Advances in Artificial Intelligence* Gelbukh, A. Alvaro de de Albornoz, A., Terashima-Marin, H. (Eds.). Springer, Berlin. pp. 949-958.

[38] Barajas-Montiel, S. E., Reyes-García, C. A. (2006). **Fuzzy Support Vector Machines for Automatic Infant Cry Recognition**. En LNCIS 345, D. S. Huang, K. Li, G.W. Irwin (Eds.). Springer, Berlin, pp. 876-881.

[39] Suaste-Rivas, I., Reyes-Galaviz, O. R., Diaz-Mendez, A., Reyes-Garcia, C. A. (2004). **Implementation of a Linguistic Fuzzy Relational Neural Network for Detecting Pathologies by Infant Cry Recognition**. En LNCS 3315: **Advances in Artificial Intelligence,** Edited by Christian Lemaitre, Carlos A. Reyes and Jesus Gonzalez, Springer, Berlin, pp. 953-962.

Con el desarrollo y utilización masiva de teléfonos celulares cada vez más inteligentes, recientemente se han desarrollado una gran diversidad de aplicaciones. Dentro del ámbito médico se han desarrollado sistemas capaces de predecir el estado de ánimo de un paciente bipolar[40] o el nivel de estrés de un trabajador exclusivamente con la información que generan los sensores del teléfono.[41] La aplicación a pacientes bipolares es importante porque en estados de fuerte depresión los pacientes pueden cometer suicidio. La segunda aplicación ha demostrado su utilidad ya que el estrés percibido de las personas está asociado con problemas de salud. En ambas aplicaciones se utilizaron técnicas de aprendizaje computacional, como clasificadores, clasificación semi-supervisada y aprendizaje por transferencia.

Otras aplicaciones en salud tienen que ver con el análisis de imágenes médicas y las series de tiempo. Entre éstas, se puede mencionar la segmentación de imágenes de Cervix mediante AE;[42] la generación automática de modelos para clasificar tipos y subtipos de leucemia a partir de imágenes de médula;[43] la identificación de episodios epilépticos a partir de señales de electroencefalogramas,[44] entre otras.

---

[40] Maxhuni, A. Muñoz-Meléndez, A., Osmani, V., Perez, H., Mayora, O., Morales, E. F. (2016). **Classification of bipolar disorder based on analysis of voice and motor activity of patients**. *Pervasive and Mobile Computing* 31: 50-66.

[41] Maxhuni, A., Hernandez-Leal, P., Sucar, L. E. Osmani, V., Morales, E. F. Mayora, O. (2016). **Stress Modelling and Prediction in Presence of Scarce Data**. *Journal of Biomedical Informatics* 63: 344-356.

[42] Marquez-Grajales, A., Acosta-Mesa, H. G., Mezura-Montes, E., Hernández-Jiménez, R. (2016). **Cervical image segmentation using active contours and evolutionary programming over temporary acetowhite patterns**. *Proceedings of CEC 2016*: 3863-3870.

[43] Escalante, H. J., Montes-y-Gómez, M., González, J. A., Gómez-Gil, P., Leopoldo Altamirano Robles, L., Reyes García, C. A., Reta, C., Rosales-Pérez, A. (2012). **Acute leukemia classification by ensemble particle swarm model selection**. *Artificial Intelligence in Medicine* 55(3): 163-175.

[44] Gómez-Gil, P., Juárez-Guerra, E., Alarcón Aquino, V., Ramírez-Cortés, Rangel-Magdaleno, J. J. (2014). **Identification of Epilepsy Seizures Using Multi-resolution Analysis and Artificial Neural Networks**. *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, 337-351.

2.5.2. Industria

En México se han desarrollado aplicaciones para tratar de mejorar los procesos de producción de tubos. Por ejemplo, para optimizar el proceso de fabricación para que los tubos cumplan con los requerimientos mecánicos, como niveles de dureza. Igualmente, se han aplicado exitosamente técnicas de inteligencia y aprendizaje computacional para detectar fallas en baleros incrustados en electrodomésticos.[45] Otro aspecto importante ha sido la creación de herramientas de limpieza de datos[46] que pueden usarse en distintas aplicaciones.

El Sector Energético es otra área en donde es posible aplicar una gran cantidad de técnicas de Aprendizaje e Inteligencia Computacional. En particular, se desarrolló una metodología híbrida basada en regresión vectorial y algoritmos genéticos para la predicción de la velocidad del viento.[47] De igual manera se desarrolló un sistema inteligente de soporte a la operación de plantas de generación eléctrica.[48] Éste es un sistema de soporte para la toma de decisiones (DSS) basado en un formalismo conocido como redes bayesianas con nodos temporales (Temporal Nodes Bayesian Network: TNBN).

---

[45] Saucedo-Espinosa, M. A., Escalante, H. J., Berrones, A. (2017). **Detection of defective embedded bearings by sound analysis: a machine learning approach**. *J. Intelligent Manufacturing* 28(2): 489-500.

[46] Ibargüengoytia, P. H., García, U.A., Herrera-Vega, J., Hernández-Leal, P., Morales, E. F. Sucar, L. E., Orihuela-Espina, F. (2013). **On the Estimation of Missing Data in Incomplete Databases: Autoregressive Bayesian Networks.** En *Proc. of the Eigth International Conference on Systems ICONS-2013,* pp. 111-116.

[47] G. Santamaría-Bonfil, A. Reyes-Ballesteros, C. Gershenson. (2016). **Wind speed forecasting for wind farms: A method based on support vector regression.** *Renewable Energy* 85:790-809.

[48] Arroyo, G., Sucar, L. E. **Sistema Inteligente de Soporte a la operación de Plantas de Generación Eléctrica**. Por aparecer en *Komputer Sapiens*.

2.5.3. Sociedad

Los sistemas de Aprendizaje e Inteligencia Computacional pueden aplicarse a problemas que beneficien en general a la sociedad. Por ejemplo, se desarrolló un sistema de inferencia difuso entrenado como red neuronal dedicado al reconocimiento de emociones a partir de la voz. Este sistema es la base de una aplicación comercial de apoyo a *call centers*.[49] Otra aplicación consiste en predecir el ozono en la Ciudad de México a partir de información de las estaciones de monitoreo ambiental.[50] En una línea diferente de aplicación se desarrollaron algoritmos dedicados a mejorar la calidad de transporte de los individuos, capaces de aprender patrones de baches y otras anomalías de la superficie de la carretera, cuya representación utiliza bolsa de palabras (*bag of words*).[51,52]

2.5.4. Interfaces

Se han desarrollado una serie de sistemas basados en modelos de inteligencia computacional dedicados al procesamiento de diferentes tipos de bioseñales y al posterior reconocimiento de patrones o a su clasificación. Entre éstos se encuentran trabajos para clasificar automáticamente el habla imaginada en un ambiente multi-objetivo a partir de electroencefalogramas (EEG) utilizan-

---

[49] Pérez-Espinosa, H., Reyes-García, C. A., Villaseñor-Pineda, L. **Acoustic feature selection and classification of emotions in speech using a 3D continuous emotion model**. *Biomedical Signal Processing and Control*, Elsevier Ltd, Electronic Publication, 7(1):79-87, January 2012, doi: 10.1016/j.bspc.2011.02.008.

[50] Sucar, E., Perez-Brito, J., Ruiz-Suarez, C., Morales, E. (1997). **Learning Structure from Data and its Application to Ozone Prediction**. *Applied Intelligence* 7(4): 327-338.

[51] González L. C., Moreno, R., Escalante, H. J., Martínez, F., Carlos, M. R. (2017). **Learning Roadway Surface Disruption Patterns Using the Bag of Words Representation**. Por aparecer en *IEEE Transactions on Intelligent Transportation Systems*.

[52] Carlos, M. R., Aragón, M. E., González, L. C., Escalante, H. J., Martínez, F. **Evaluation of detection approaches for Road Anomalies based on accelerometer readings —Addressing who's who**. En revisión en *IEEE Transactions on Intelligent Transportation Systems*.

do sistemas de inferencia difusa.[53] La disciplina en que se desarrollan estos modelos se conoce como Interfaces Cerebro-Computadora (*Brain Computer Interfaces*).

2.6. Investigación en México

Existen diferentes grupos de investigación en México dedicados al Aprendizaje e Inteligencia Computacional. Entre éstos destacan el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, con sus laboratorios de Aprendizaje Computacional y Reconocimiento de Patrones y de Procesamiento de Bioseñales y Computación Médica, en los que se realiza investigación básica y aplicada; el Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM) con su grupo de Investigación en Sistemas Inteligentes, enfocado a la optimización y logística, inteligencia ambiental, semántica Web, salud, previsión e inteligencia de negocios, entre otros, y su grupo de Investigación en Modelos de Aprendizaje Computacional, enfocado a la innovación de técnicas estadísticas para el reconocimiento de patrones y de técnicas de aprendizaje de modelos computacionales (*machine learning*), así como de modelos de visualización, en aplicaciones como seguridad informática, integridad personal y resiliencia personal; el Instituto Tecnológico de Culiacán enfocado al desarrollo de Sistemas Tutoriales Inteligentes, que ha impulsado el *Workshop on Intelligent Learning Environments WILE* (Taller de Ambientes Inteligentes de Aprendizaje); la Universidad Veracruzana con su cuerpo académico de Investigación y Aplicaciones de la Inteligencia Artificial, enfocado al descubrimiento de conocimiento en datos, sistemas multi-agentes, procesamiento digital de imágenes y cómputo evolutivo; la Universidad Panamericana con su grupo de Visión, Máquinas y Señales, que realiza investigación aplicada en procesamiento de imágenes, señales biológicas y áreas médicas, diseño de algoritmos de aprendizaje y sistemas de control robóticos; el Insti-

[53] Torres-García, A. A., Reyes-García, C. A., Villaseñor-Pineda, Gregorio García-Aguilar, G. (2016). **Implementing a fuzzy inference system in a multi-objective EEG channel selection model for imagined speech classification**. *Expert Systems With Applications* 59:1–12.

tuto Tecnológico de Tijuana con sus laboratorios Sistemas Híbridos Difusos Inteligentes y Computación Evolutiva, y el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas de la UNAM, donde se hace investigación en Computación Evolutiva, Aprendizaje aplicado a la Interacción Humano-Computadora y aprendizaje de máquina aplicado al procesamiento de imágenes y lenguaje natural.[54,55] Otros grupos destacados son el grupo de Optimización Evolutiva Multi-Objetivo del CINVESTAV; el laboratorio Evovisión del CICESE, que realiza investigación en visión computacional mediante algoritmos evolutivos, el grupo de Análisis de datos multidimensionales y reconocimiento de patrones del Centro de Investigación en Matemáticas (CIMAT) en Guanajuato y el laboratorio de Big DATA del INFOTEC.

2.7. Conclusiones

Actualmente se vive un creciente interés en el área de Inteligencia Artificial y, en particular, en los sistemas de aprendizaje e inteligencia computacional. Esto se debe en parte a los avances recientes del área, impulsados por la gran cantidad de datos generados y disponibles para su análisis, los avances en las computadoras para procesar cada vez más rápidamente información y en las aplicaciones comerciales que se están desarrollando utilizando algoritmos maduros de aprendizaje e inteligencia artificial. La capacidad "inteligente de las cosas y sistemas" será determinada por su nivel de Coeficiente Intelectual Máquina (MIQ) que se convertirá en una de las especificaciones más importantes para seleccionar productos. En general se espera un futuro con inteligencia incremental en muchas áreas, como ciudades inteligentes, carreteras inteligentes, edificios inteligentes, vehículos autónomos extremadamente inteligentes, sistemas administrativos, educativos, legales y gubernamentales

[54] Gibran Fuentes-Pineda, G., Koga, H., Watanabe, T. (2011). **Scalable Object Discovery: A Hash-based Approach to Clustering Co-occurring Visual Words.** *Transactions on Information and Systems*, E94-D(10):2024–2035.

[55] Gibran Fuentes-Pineda, G., Meza-Ruiz, I. V. (2015). **Sampled Weighted Min-Hashing for Large-Scale Topic Mining.** *Proceedings of the Mexican Conference on Pattern Recognition,* Vol. 9116, pp. 203-213.

inteligentes, así como variantes de teléfonos, televisores y computadores inteligentes. Algunos autores creen que se va a lograr crear una super-inteligencia y se están cuestionando sus posibles consecuencias para la humanidad, desde los enfoques pesimistas, que vaticinan el fin de la raza humana, hasta los optimistas, que predicen un futuro en donde los problemas de salud, alimentación, energía y contaminación, por mencionar algunos, se van a resolver definitivamente con la ayuda de esta super-inteligencia.

3. Lingüística Computacional

3.1. Introducción

La Lingüística Computacional se inició desde la introducción de la Computación y la Inteligencia Artificial (IA) a principios de la década de los cincuenta del siglo pasado. La primera propuesta en el entorno científico y académico para construir y programar una máquina capaz de conversar en lenguaje natural con seres humanos, como el español o el inglés, aparece en el artículo *Computing Machinery and Intelligence*,[1] en el que Alan Turing propuso el programa de investigación para la IA y, en particular, planteó la meta de construir y programar una máquina capaz de entender el lenguaje natural. Es también en este artículo donde se presenta el "Juego de imitación", mejor conocido como la "Prueba de Turing", que propone que una máquina capaz de conversar con un ser humano en lenguaje natural con un alto nivel de desempeño, se tiene que considerar "inteligente". La subdisciplina de la IA que engloba los esfuerzos para lograr dicha meta se conoce como Lingüística Computacional.[2]

Al igual que la lingüística descriptiva tradicional, la Lingüística Computacional aborda una gama muy amplia de fenómenos del lenguaje, tanto hablado como escrito, pero con la restricción de que los modelos se deben ca-

[1] Turing, A. (1950). **Computing Machinery and Intelligence**. *Mind,* 59: 433-460.
[2] https://en.wikipedia.org/wiki/Computational_linguistics

racterizar mediante sistemas de reglas formales y/o procedimientos computacionales, lo que impone una restricción muy severa a los tipos de modelos admisibles. En este enfoque, el objeto de investigación es la representación computacional del conocimiento lingüístico así como los procesos que lo utilizan. Por lo mismo, esta disciplina caracteriza los diferentes niveles de representación lingüística así como sus relaciones. En general se reconocen seis niveles, como sigue: i) el fonético y fonológico, ii) prosódico y entonativo, iii) léxico y morfológico, iv) sintáctico, v) semántico y vi) pragmático.

Por la magnitud de la empresa y con el fin de abordar sus objetivos específicos, la Lingüística Computacional se ha desarrollado históricamente en varias especialidades, con metáforas, teorías y metodologías diferentes, y cada una de éstas ha representado un esfuerzo de investigación y desarrollo tecnológico de dimensiones colosales. Entre las más prominentes podemos mencionar: i) Reconocimiento del habla y síntesis de voz (1952),[3] ii) Traducción automática entre lenguajes naturales (1954),[4] iii) Procesamiento del lenguaje natural (1964)[5] y iv) Lingüística de corpus (1967).[6]

Los programas y dispositivos que se presentaron inicialmente –como prueba de concepto– generaron grandes expectativas y se pensó que la construcción de la máquina del lenguaje iba a ser sólo cuestión de tiempo. Sin embargo, las propuestas no se materializaron y estas disciplinas pasaron por varios ciclos con un fuerte impulso inicial hasta su agotamiento a lo largo de la segunda mitad del siglo XX. A pesar de ello, durante este período se creó un acervo de conocimiento que se reflejó en la aparición de varias revistas emblemáticas donde se detalla la historia de la disciplina, principalmente *Computational Linguistics, Artificial Intelligence* y otras más,[7] en libros de texto de

[3] https://en.wikipedia.org/wiki/Speech_recognition
[4] https://en.wikipedia.org/wiki/History_of_machine_translation
[5] https://en.wikipedia.org/wiki/Natural_language_processing
[6] https://en.wikipedia.org/wiki/Corpus_linguistics
[7] Por ejemplo, *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *Computer Speech and Language*, y *Speech Communication*s.

procesamiento de lenguaje natural[8] y reconocimiento del habla[9] y en varias conferencias internacionales de gran prestigio.[10] Asimismo, a finales de la década de los noventa y principios de este siglo aparecieron dos textos que son ahora referencia central para los estudios de posgrado en la especialidad.[11]

Con el fin de contextualizar las contribuciones de la comunidad mexicana a esta disciplina, en la Sección 3.2 se describen los seis niveles de representación lingüística principales, aunque sin comprometerse, en la medida de lo posible, con ninguna corriente de pensamiento o teoría del lenguaje en particular. Con este antecedente en la Sección 3.3 se describen los principales estudios teóricos y aplicaciones realizados por la comunidad mexicana, y en cada caso se hace referencia a los niveles de representación relevantes.

3.2. Modelos computacionales de la estructura del lenguaje

Los estudios computacionales de la estructura del lenguaje se iniciaron con la publicación de *Syntactic Structures*,[12] por Noam Chomsky, quien propuso por primera vez que la estructura sintáctica del lenguaje se puede modelar a través de un sistema de reglas completamente mecanizadas o formales. Esta teoría abrió la puerta para modelar dentro del paradigma computacional no sólo la sintaxis sino los diversos niveles de representación lingüística y tuvo

[8] Winograd, T. **Understanding Natural Language**, Academic Press, Nueva York, 1972; Allen, J. **Natural Language Understanding**, Benjamin-Cumming, 1987 y 1994; Gazdar, G. Mellish, C. **Natural language processing in Prolog**, Addison Wesley, 1989.

[9] Jelineck, F. **Statistical Methods for Speech Recognition,** Cambridge, Mass.: MIT Press, 1997; Huang, X. D., Yasuo Ariki, Y., Mervyn, Y., Jack, A. **Hidden Markov Models for Speech Recognition,** Edinburgh University Press, 1990.

[10] *SpeechTEK*, *SpeechTEK Europe, ICASSP*, *Interspeech/Eurospeech*, y la *IEEE ASRU* en reconocimiento del habla; asimismo *Meeting of the ACL*, *COLING*, *NAACL*, *EMNLP* y *HLT* en procesamiento del lenguaje natural.

[11] Manning, C., Schütze, H. **Foundations of Statistical Language Processing**, MIT Press, 1999; Jurafsky, D., Martin, J. **Speech and Language Processing: An introduction to Natural Language Processing, Speech Recognition and Computational Linguistics**, Prentice Hall, 2009.

[12] Chomsky, N. **Syntactic Structures**, The Hague/Paris: Mouton, 1957.

repercusiones en una gama muy amplia de disciplinas científicas, tecnológicas y en las humanidades. Los niveles principales de estructura lingüística son como sigue:

3.2.1. Nivel fonético y fonológico

Las unidades básicas de la estructura del lenguaje hablado se conocen como fonemas. Los fonemas se reconocen por sus características combinatorias o sintagmáticas, y de contraste o paradigmáticas,[13] y corresponden, hasta cierto punto, con los símbolos del alfabeto. Por ejemplo, la palabra *casa* está constituida por cuatro fonemas en secuencia o en relación sintagmática, es decir *c*, *a*, *s* y *a*, y contrasta paradigmáticamente con las palabras *tasa* y *cara* en la primera y tercera posición respectivamente, por lo que dichas palabras se distinguen entre sí y tienen significados diferentes. El sistema fonológico se caracteriza por las reglas que permiten o restringen las secuencias y contrastes entre estas unidades, y que generan el conjunto de palabras actuales y potenciales de una lengua, como el español o el inglés.

Por su parte, cada fonema da lugar a una realización acústica específica, aunque hay fonemas que tienen diferentes realizaciones. Por ejemplo, en el español de México la *ch* en la palabra *Chihuahua* se pronuncia de manera diferente por hablantes del centro y del norte del país (africada sorda y fricativa respectivamente) pero dicha alteración no cambia su significado, por lo que el fonema es el mismo pero tiene dos realizaciones acústicas o alófonos diferentes. Mientras la fonología estudia la estructura de estas unidades abstractas, la fonética estudia las propiedades físicas de sus realizaciones acústicas.

Las unidades acústicas de los lenguajes humanos se codifican en el alfabeto fonético internacional (AFI),[14] el cual se concibió originalmente para representar la pronunciación de las palabras en los diccionarios bilingües y

[13] Saussure, F. **Curso de Lingüística General**, Editorial Losada. S. A. Moreno 3362 Buenos Aires. 1945.

[14] https://es.wikipedia.org/wiki/Alfabeto_Fonético_Internacional

permitir la comunicación entre hablantes de lenguas maternas diferentes. Por otra parte, cada región lingüística utiliza un conjunto de unidades acústicas específicas y el estudio de la fonética de su dialecto se centra en la definición de su alfabeto fonético. Los alófonos se pueden representar por un símbolo en un alfabeto fonético computacional, el cual se puede asociar a la representación de las características físicas de la señal de audio.[15] Por razones prácticas, se han desarrollado alfabetos fonéticos para cada dialecto para habilitar el reconocimiento de voz y la traducción computacional entre lenguajes hablados.[16]

3.2.2. Nivel de prosodia y entonación

En este nivel se representa la acentuación de las palabras y la estructura tonal de los enunciados. Las unidades sobre las que recaen los tonos son las sílabas y la entonación se caracteriza por la variación de la frecuencia fundamental o F0 en la elocución, la cual permite distinguir las oraciones declarativas, interrogativas y admirativas. Al igual que en el nivel fonético-fonológico, la entonación se puede estudiar desde la perspectiva de la estructura tonal abstracta o en términos de su realización física. Este nivel de representación es necesario para reconocer los tipos de intenciones expresadas en el habla y es fundamental para la creación de voces sintéticas de calidad, que pongan el acento de las palabras en la sílaba correcta y que trasmitan la intención de la elocución de manera clara.

[15] Ver el capítulo 7.

[16] Se debe considerar también que no todo símbolo ortográfico corresponde a un fonema o tiene una realización acústica, ya que el texto es una representación convencional del lenguaje hablado. Por ejemplo, la letra *h* en el español de México es sorda por lo que no se asocia a ninguna unidad fonética; asimismo, los símbolos de puntuación no corresponden a ningún fonema y representan más bien a la prosodia y la entonación. Por lo mismo, es necesario distinguir tres conjuntos de símbolos: los ortográficos, los fonemas y los símbolos acústicos, así como las reglas que traducen a cada unidad o palabra entre estos tres conjuntos.

3.2.3. Nivel léxico y morfológico

El siguiente nivel de representación se enfoca al análisis de la estructura abstracta de las palabras. Esta estructura se puede pensar en términos de una lista de palabras o "lexicón" con una entrada por cada palabra; cada entrada a su vez contiene una secuencia de fonemas, la cual se asocia a uno o varios alófonos así como a la representación de su significado convencional.[17] Intuitivamente, el primer paso para comprender el lenguaje hablado es el reconocimiento de la voz, que consiste en "alinear" a las secuencias de unidades acústicas en la elocución con secuencias de realizaciones de palabras en el lexicón y, a través del mismo, con sus significados convencionales.

Es necesario considerar que la gran mayoría de las palabras tienen variaciones debido a partículas como prefijos, infijos y sufijos que las transforman para especificarlas en algún sentido predecible y reconocible por los hablantes de la lengua; por ejemplo las palabras *inmaterial*, *corredor* y *materialista* se forman respectivamente a partir del prefijo de negación *in*, el infijo *do* y el sufijo *ista*. El primero niega la propiedad adscrita por el adjetivo que se modifica (*material*), la segunda crea el nombre de quien realiza una actividad a partir del verbo que la nombra, y la tercera crea el nombre de quien cree o hace algo a partir de la propiedad que adscribe el adjetivo modificado (*Juan es un materialista*). Si pensamos que el lexicón contiene una forma nuclear de cada palabra, asociada a un significado básico, la morfología computacional estudia las reglas formales que producen las transformaciones posibles del núcleo, que a su vez producen una alteración correspondiente en la representación de su significado. Esta ampliación dinámica del lexicón permite que se puedan reconocer no sólo las palabras actuales sino también las potenciales en conjunto con sus significados convencionales.

[17] Los conceptos o "contenidos" de las palabras se representan en una base de conocimiento; por ejemplo, como predicados lógicos o en una taxonomía conceptual en una representación estructurada. Ver capítulo 1.

3.2.4. Nivel sintáctico

La sintaxis caracteriza la estructura de las frases y las oraciones. En este nivel se considera que dentro de la oración hay palabras cuya función es central en la oración y que las otras tienen un papel subordinado. Estas relaciones se representan como una jerarquía cuyo nodo superior o cabeza es el núcleo gramatical y los nodos inferiores son los constituyentes. Dicha estructura puede ser simple o muy compleja dependiendo de los fenómenos involucrados y del formato de representación y se conoce como "estructura sintáctica". Su relevancia se debe a que es "la portadora" del significado convencional de la oración. Mientras que el significado de las palabras se codifica directamente en el lexicón, el significado convencional de la oración depende de los significados de las palabras y de cómo se combinan en la estructura sintáctica. Por esta razón, desde la propuesta original de Chomsky, inducir la estructura sintáctica a partir de las reglas de la gramática y el estímulo lingüístico se considera como una de las tareas centrales del procesamiento del lenguaje. Este proceso se conoce como "parseo" y su estudio ha sido también sujeto de una investigación de grandes dimensiones en la Lingüística Computacional.

La estructura sintáctica debe tomar en cuenta que tanto las palabras básicas como las derivadas morfológicamente pueden sufrir transformaciones adicionales cuando se ponen en el contexto de una frase o una oración, como las inflexiones debidas al género y al número de los sustantivos, que en el español deben concordar con las inflexiones de los verbos. Por lo mismo, aunque las inflexiones son también partículas que modifican a las palabras su carácter es sintáctico y no morfológico.

Un fenómeno de características morfo-sintáctico muy singular y frecuente en nuestra lengua es el de los llamados pronombres clíticos, como *se* y *lo* en *se lo comió* o *dáselo*. Lo interesante de estas partículas, independientemente de la determinación de sus referencias, es que pueden aparecer como morfemas o como palabras independientes, tanto adelante como atrás del verbo,

de manera muy flexible, además de que llevan implícito su caso (acusativo o dativo) que indica si su referencia recibe la acción verbal directamente o es beneficiario de la misma: en *se lo comió*, *lo* es un pronombre acusativo que denota el objeto de comer, lo comido, y *se* es un pronombre dativo, cuya referencia es el beneficiario de dicha acción, es decir, quien realizó la acción de comer.

De forma muy general el análisis sintáctico se ha abordado desde dos enfoques principalmente: constituyentes y dependencias. Ambos se han estudiado durante más de 40 años y representan alternativas teóricas y metodológicas para el estudio y las aplicaciones de la Lingüística Computacional, como se verá a continuación.

3.2.4.1. Enfoque de constituyentes

En este enfoque, presentado originalmente por Chomsky,[18] un constituyente es una palabra o grupo de palabras que cumplen una función específica en la oración. Estos grupos se conocen como "categorías gramaticales". El proceso sintáctico consiste en encontrar a las categorías de una oración segmentándola en sus partes de manera recurrente hasta que las partes sean las palabras básicas o derivadas en el lexicón. Aunque el número de palabras en el lexicón y el número de reglas sintácticas sea finito, la aplicación sistemática de las reglas sintácticas y morfológicas genera un número infinito de oraciones. En la Figura 3.1 se ilustra una estructura sintáctica por constituyentes, donde *O* representa a la oración, *GN* al grupo nominal y *GV* al grupo verbal. Las estructuras jerárquicas se pueden representar linealmente poniendo la expresión o subexpresión entre paréntesis seguida de un subíndice que indica la categoría del constituyente, como se indica en la parte superior de la Figura 3.1.

[18] Ver nota 12.

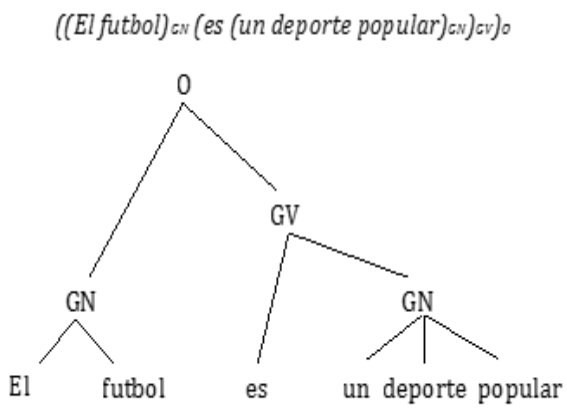


Figura 3.1. Análisis por constituyentes de *El futbol es un deporte popular*.

3.2.4.2. Enfoque de dependencias

Por su parte el enfoque de dependencias iniciado por Lucien Tesnière en 1959,[19] establece que la estructura sintáctica consiste en relaciones de dependencia entre pares de palabras donde una es la principal, rectora o cabeza, y la otra está subordinada. Si cada palabra de la oración tiene una palabra propia rectora (cabeza), la oración entera se ve como una estructura jerárquica de diferentes niveles, o como un "árbol de dependencias". La única palabra que no está subordinada es la raíz del árbol. En la Figura 3.2 se ilustra la estructura de la oración en la Figura 3.1 pero en términos de dependencias.

A diferencia del enfoque de constituyentes, en el enfoque de dependencias no se postulan categorías sintácticas abstractas por lo que las estructuras sólo contienen unidades léxicas concretas. Por lo mismo, la oposición entre el enfoque por constituyentes y el enfoque por dependencias proviene a su vez de una oposición más profunda entre la hipótesis de que la estructura sintáctica es una representación abstracta versus la hipótesis de que ésta tiene un carácter concreto.

[19] Tesnière, L., **Éléments de syntaxe structurale**, Klincksieck, Paris, 1959.

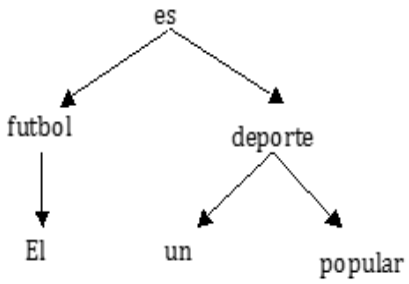


Figura 3.2. Representación en dependencias de *El futbol es un deporte popular*.

3.2.5. Nivel semántico

El propósito de la semántica es caracterizar el significado convencional o independiente del contexto de emisión o interpretación de una unidad lingüística, ya sea básica, como las palabras, o compuesta, como las frases y las oraciones. La semántica computacional se basa en el llamado "Principio de Composicionalidad",[20] que establece que el significado de una estructura compuesta es función del significado de sus partes y de su modo de combinación gramatical. En su formulación más general la semántica de una oración se caracteriza como una función proposicional, donde la palabra principal en la estructura es la función y las otras sus argumentos. Por ejemplo, la representación semántica o estructura de argumentos de *Juan se comió el pan* es la predicación *comer*(e_i, *juan*, *pan*) & *pasado*(e_i) & *beneficiario*(e_i) = *Juan*, cuya interpretación es que e_i es un evento de comer que ocurrió en el pasado, que el agente de dicho evento fue Juan, quien comió, y el paciente (el ente que es pasivo en la acción) fue el pan, lo comido, mientras que *se* es un pronombre dativo que duplica al sujeto e indica que éste se benefició con la acción de comer. Estas representaciones pueden ser de carácter concreto, como en el presente ejemplo, pero también se pueden referir a abstracciones muy profundas. Posiblemente el formalismo más general para caracterizar la semántica de los lenguajes naturales de forma computacional es la Semántica de

[20] http://en.wikipedia.org/wiki/Principle_of_compositionality

Montague[21] aunque existe una gran variedad de formatos representacionales, como redes semánticas, primitivas conceptuales, etc., que son materia de la representación del conocimiento.

3.2.6. Nivel pragmático

Toda unidad lingüística, ya sea hablada o escrita, se realiza e interpreta en relación a un "contexto". La noción de contexto se puede ilustrar con una analogía pictórica: todo cuadro tiene una imagen saliente, "la figura", que ocurre siempre en relación a una escena o "fondo". La unidad lingüística es la figura y el contexto el fondo. En general se pueden distinguir dos tipos de contextos: i) el lingüístico, que consiste en información aportada por el lenguaje mismo a lo largo de la conversación o el discurso y ii) el situacional, que consiste en información extra-lingüística y que tiene como parámetros al hablante, al oyente y posiblemente a uno o más terceros, con sus expectativas e intenciones, así como la situación espacial y temporal en la que se realiza o interpreta la elocución.

El contexto se puede apreciar, como una primera aproximación, en relación a las palabras que funcionan como índices o variables cuya referencia o denotación cambia en cada situación de uso. Los índices más directos y evidentes son los pronombres. Si éstos se interpretan en relación al contexto lingüístico, es decir a lo que se ha dicho en la conversación o el discurso, se llaman anafóricos; por su parte, si se interpretan en relación al contexto no lingüístico, como la situación espacial o temporal, se llaman deícticos o indexicales. Asimismo, las inferencias para encontrar los referentes de los pronombres se conocen como anafóricas e indexicales respectivamente. Esta distinción es esencial no sólo para interpretar el lenguaje en uso, sino también para interpretar representaciones multimodales, como los mapas o diagramas con sus anotaciones textuales.[22]

---

[21] Dowty, D. R., Wall, R. E., Peters, S. **Introduction to Montague Semantics**, Kluwer Academic Publishers, 1981.

[22] Pineda, L. A., Garza, G. (2000). **A Model for Multimodal Reference Resolution**. *Computational Linguistics,* 26 (2): 136-192.

Para ilustrar la complejidad de la inferencia anafórica considere el discurso compuesto por la oraciones *Pepe vio a Juan comprar el pan, se lo comió*. El problema es determinar quiénes son los referentes de los pronombres *se* y *lo*, para lo cual es necesario identificarlos y correferenciarlos con sus antecedentes. Una posibilidad es que *se* tenga como antecedente a *Pepe,* quien comió, y *lo* al pan, lo comido. Sin embargo, también puede ser el caso que quien se comió el pan haya sido Juan e incluso que quien comió haya sido Pepe y lo comido Juan. La inferencia anafórica es sumamente compleja y ha sido también objeto de un esfuerzo de investigación de grandes dimensiones en lingüística computacional.

Por su parte, los pronombres indexicales tienen que tomar su referente directamente del mundo en relación al hablante y al oyente, como en la interpretación de *yo* en *yo vi a Juan comerse el pan*, cuyo referente será quien profiera la elocución en el contexto particular. Hay también otras palabras como los adverbios *aquí*, *allá*, *ahora* o *ahorita* que se interpretan en relación a una locación espacial o un momento o un intervalo en el tiempo y tienen una connotación indexical pura.

Sin embargo, los pronombres que se usan normalmente como anafóricos también se pueden utilizar como indexicales, en cuyo caso el proceso de interpretación se vuelve más complejo. Por ejemplo, en una situación en que el pan esta sobre la mesa y Pepe le dice a Juan ¿*me lo pasas*? al tiempo que señala al pan con su dedo índice, la interpretación de *me* es quien profirió la elocución y es beneficiario de la acción, es decir Pepe, y de *lo* es el pan, donde el contexto relevante es el mundo y no el discurso. Aunque en principio se puede distinguir al contexto lingüístico y al que se establece por la situación en el mundo, en la práctica los contextos interactúan y el problema de interpretación es en general sumamente complejo. Un caso particular de índices son los nombres propios como Pepe, Juan y Pedro, que se refieren en cada caso al individuo en el contexto, aunque haya miles de individuos que han tenido, tienen y tendrán estos nombres.

La interpretación del lenguaje en cada situación de uso, en oposición a la interpretación convencional estudiada por la semántica, se complica aún más cuando se toman en cuenta las intenciones y las creencias de los interlocutores, así como la entonación de las elocuciones. En español las oraciones declarativas que expresan una proposición tienen una entonación relativamente plana, las interrogativas tienen una curva ascendente al final, y las imperativas como las órdenes enfatizan los tonos iniciales. Por lo mismo, es posible identificar si una elocución es un enunciado, una pregunta o una orden, sin siquiera interpretar el sentido convencional de la oración. Se dice que todo enunciado en cada situación de uso es un "acto del habla",[23] y que si la entonación corresponde al tipo del acto del habla, éste es directo, pero si esta relación se cambia, el acto del habla es indirecto. Por ejemplo, enunciar *¿Me puedes pasar el pan?* en un contexto normal de interpretación no es una solicitud de información –si el interlocutor tiene la capacidad de llevar a cabo dicha acción– sino una directiva de acción, y en ciertos contextos puede ser incluso una orden. Los actos del habla indirectos ocurren muy frecuentemente en la conversación cotidiana y su estudio es también objeto de la pragmática.

Los actos del habla indirectos pueden también alterar tanto el material léxico como la forma sintáctica del enunciado; por ejemplo, si Pepe y Juan están en una habitación donde hace mucho frio y el primero le dice al segundo, quien está junto a la ventana, que está abierta, ¿Qué tal tu veranito*?*, la intención es realmente la directiva de acción o incluso la orden de que Juan cierre la ventana. Este ejemplo ilustra cómo el contexto o situación de interpretación espacial y temporal, que involucra también a las expectativas e intenciones de los interlocutores, es indispensable para interpretar al lenguaje en uso.

En resumen, el propósito del análisis pragmático computacional es inferir la intensión o significado de una elocución de manera automática a partir de su significado convencional y del contexto. Asimismo, la salida del módulo

[23] Por ejemplo, ver Austin, J. L. **How to Do Things With Words**, Oxford University Press, 1962. Para una discusión más didáctica ver Levinson, S. C. **Pragmatics**, Cambridge University Press, 1983.

pragmático se puede conceptualizar como la especificación de las acciones que el intérprete tiene que realizar, entre un conjunto posiblemente muy amplio de tipos de acciones, como respuesta a la intención expresada por su interlocutor. Estas acciones pueden ser mentales, motoras e incluso emotivas. Mental, como consultar y reportar información, o hacer una inferencia conceptual, como determinar si dos expresiones son sinónimas, o deliberativa, como hacer un diagnóstico, tomar una decisión o inducir un plan; motora, como tomar un objeto y entregarlo a un destinatario; y emotiva, como ponerse muy contento al recibir una buena noticia.

3.2.7. Ambigüedad

Un problema que aqueja a los modelos computacionales de la interpretación del lenguaje natural es la ambigüedad. Este fenómeno se puede apreciar directamente en el lexicón, ya que hay muchas palabras que tienen más de un significado convencional, como *banco* o *gato*, y su resolución consiste en inferir qué es lo que quiso decir quien las expresó en la situación de uso, tanto en el lenguaje hablado como en el escrito. La ambigüedad aparece también en el nivel sintáctico; por ejemplo, en *Pepe vio a Juan en el banco comiéndose el pan*, es ambiguo quién estaba en el banco, quién estaba comiéndose el pan, y qué quiere decir *banco*. Se invita al lector a visualizar las diversas escenas descritas por esta oración, incluyendo la escena en que los dos están comiendo pan sentados en un banco en el banco. El problema es más agudo si se toma en cuenta que la ambigüedad puede también surgir en el nivel fonético y la entonación, y frecuentemente se sostiene que puede ocurrir en todos los niveles de representación lingüística.

Los modelos generativos o basados en reglas seleccionan una entrada léxica para cada palabra y asignan una estructura sintáctica particular a cada interpretación posible, que se refleja como una predicación particular en la semántica, por lo que la aplicación sistemática de estos procesos produce un número significativo de estructuras sintácticas y, consecuentemente, de

interpretaciones para cada elocución, aunque normalmente sólo una es la apropiada en el contexto. La resolución de la ambigüedad se ha abordado tradicionalmente tanto con la inclusión de restricciones que prevengan la generación de interpretaciones incorrectas como con la generación de todas las interpretaciones para después eliminar las incorrectas, o mediante combinaciones de estas dos estrategias. Sin embargo, si sólo se toma en cuenta el significado convencional de las palabras o las oraciones, el problema es realmente muy complejo, y frecuentemente se sostiene que éste es el mayor problema de la lingüística computacional y que todo lenguaje bien regimentado debería excluir completamente a la ambigüedad, como sucede en los lenguajes formales.

Sin embargo, la ambigüedad es también un recurso expresivo que se puede capitalizar para expresar varias figuras lingüísticas, como la generalización y la ironía, entre muchas otras, y normalmente los seres humanos no generamos interpretaciones incorrectas, cuando menos conscientemente; más aún, somos capaces de apreciar los diferentes sentidos posibles y entender el chiste. Esto se debe a que a diferencia de los modelos formales que utilizan tan sólo significados convencionales, los seres humanos interpretamos el lenguaje en relación al contexto. Considere que si las palabras u oraciones se expresan en un contexto espacial o temporal especifico, digamos por Pedro, quien profiere *Pepe vio a Juan en el banco comiéndose el pan* al tiempo que ve la escena cuando ocurre, o viendo una fotografía del evento, la información extralingüística previene parcial o totalmente la generación de hipótesis incorrectas.

3.2.8. Arquitectura de la máquina del lenguaje

A primera vista, la funcionalidad y relaciones entre los niveles de representación lingüística sugieren que el lenguaje se procesa, es decir, se interpreta y se genera, a través de una estructura "de tubería" (*pipeline*) o "de abajo hacia arriba" (*bottom-up*), en el que los módulos de procesamiento correspondientes a cada nivel de estructura se alinean en serie. Este modelo se adopta explícita

o implícitamente en la gran mayoría de los modelos y aplicaciones de la Lingüística Computacional.

En esta arquitectura el problema de la ambigüedad se resuelve ya sea imponiendo restricciones mutuas entre las representaciones que se tienen que combinar en cada nivel, o restricciones entre los objetos de diferentes niveles, filtrando las posibles interpretaciones a lo largo del proceso hasta el final de la tubería, donde se produce sólo la interpretación correcta. De manera análoga, la generación procede a partir de una especificación abstracta de la intención la cual se especifica de manera semántica, sintáctica y léxica, hasta su realización fonética o textual.

Sin embargo, es también posible pensar que el contexto no sólo restringe de manera activa las posibles interpretaciones durante el proceso de interpretación, sino que participa activamente en la síntesis de la interpretación correcta de manera simultánea con los procesos que actúan en cada uno de los niveles de representación lingüística. Desde este punto de vista, el modelo de tubería se puede sostener, pero en lugar de pensar en el nivel pragmático al final de la tubería, éste se tendría que pensar como un módulo paralelo, que impacta a todos los módulos de manera directa, tanto en el proceso de interpretación como en el de generación.

Sin embargo, en muchas de las aplicaciones tradicionales de la Lingüística Computacional no se tiene información del contexto o no es posible representarlo, por lo que se tiene que adoptar el modelo de tubería y enfrentar el problema de la ambigüedad. Pero, como se sugiere a partir de esta discusión, el problema de fondo no es el de la ambigüedad en sí, sino el de la representación del contexto junto con los problemas asociados a los procesos del módulo pragmático, y cómo estos procesos inciden en el resto de los niveles de representación lingüística.

3.3. Especialidades cultivadas en México

En esta sección se presentan las áreas de investigación desarrolladas por la comunidad mexicana. Iniciamos con el trabajo en lingüística de corpus en la Sección 3.3.1 ya que estos recursos se utilizan en todos los niveles de representación. En la sección 3.3.2 se abordan las contribuciones en tecnologías del habla, donde se hace referencia tanto a los niveles de estructura fonético-fonológica y léxica, como a la lingüística de corpus. La sección 3.3.3 se dedica a los sistemas conversacionales en lenguaje natural que involucran también a todos los niveles. En la sección 3.3.4 se abordan las especialidades relacionadas con el procesamiento de textos que involucran principalmente a los niveles léxico, sintáctico y semántico.

3.3.1. Lingüística de corpus

La lingüística de corpus se enfoca a la creación de recursos lingüísticos, tanto en la modalidad hablada como en la textual. Los corpus son recursos empíricos codificados en formatos computacionales para el estudio de todos los niveles de representación lingüística, así como para el desarrollo de aplicaciones diversas. El desarrollo de corpus exige procedimientos y criterios rigurosos por su magnitud y para su diseño, recolección y organización, de manera que sean confiables y apropiados para las tareas de interés.

La práctica de recabar corpus se inició en el entorno computacional de manera muy temprana con la recopilación del conjunto de textos de diferentes géneros del *Brown Corpus*[24] y más tarde con la construcción de textos anotados sintácticamente, llamados *Treebanks*,[25] particularmente con el *Penn Treebank Project*.[26] Un corpus muy conocido para el lenguaje hablado es el

[24] https://en.wikipedia.org/wiki/Brown_Corpus

[25] https://en.wikipedia.org/wiki/Treebank

[26] Taylor, A., Marcus, M., Santorini, B. (2003). **The Penn treebank: an overview**. En A. Abeille (Ed.) Treebanks: The state of the art in syntactically annotated corpora. Kluwer, pp. 41-70.

Switchboard.[27] Para el español, un corpus textual anotado o transcrito en varios niveles es el Cast3LB Corpus.[28]

Los corpus textuales de primera generación contenían alrededor de un millón de palabras, mientras que los mega corpus actuales contienen más de cien millones de palabras; esto es en parte posible gracias a la Web. Los corpus deben prepararse para su tratamiento computacional, por ejemplo, para la búsqueda de ocurrencias de determinado tipo y número de palabras, la extensión oracional, etc. A continuación se muestran diversas herramientas de utilidad para este tratamiento y en especial para corpora de gran magnitud.

3.3.1.1. Modelos del lenguaje

Una tarea básica en el análisis del lenguaje es la predicción de las unidades que se siguen en una elocución o un texto (por ejemplo, alófonos o palabras) dadas las unidades que se han observado. Esta predicción se puede hacer utilizando modelos probabilísticos, llamados modelos de *n-gramas*. Un *n*-grama es una secuencia de *n tokens* o instancias de palabras; un 2-grama (o bigrama) es una secuencia de dos palabras como *favor de*, *de guardar* o *guardar silencio*; un 3-grama (trigrama) es una secuencia de tres palabras como *favor de guardar* o *de guardar silencio*. Un modelo de lenguaje consiste en la asignación de una probabilidad para todos los *n*-gramas diferentes en relación al corpus de referencia, utilizando para este efecto un algoritmo, normalmente de carácter probabilístico. Por ejemplo, la probabilidad de un unigrama es simplemente la probabilidad de ocurrencia de cada palabra en relación al corpus, y la probabilidad de un bigrama es la probabilidad de que la palabra al final ocurra dada la probabilidad de que la palabra inicial ocurre. Este procedimiento se aplica de manera recurrente para obtener la probabilidad de *n*-gramas de orden mayor.

Es muy importante que el corpus sea representativo o informativo del dominio de interés ya que su calidad impacta directamente en la calidad de

[27] https://catalog.ldc.upenn.edu/ldc97s62
[28] http://www.aclweb.org/anthology/W04-0209.pdf

los modelos del lenguaje. Por lo mismo es común utilizar medidas entrópicas para evaluarlo. Una medida común es la perplejidad, que cuantifica el promedio de las secuencias del lenguaje que pueden seguir a una secuencia dada. Mientras menos sean éstas más predecible e informativo es el corpus, y su entropía tiene un valor más bajo, por lo que el corpus será de mayor utilidad.

Por su parte, en todos los corpus hay contextos que no aparecen de manera contingente aunque puedan ocurrir en el lenguaje, por lo que sus *n*-gramas tendrán un valor de cero. Para enfrentar este problema es común reasignar la masa de probabilidad de los *n*-gramas, quitándole a los que más tienen y distribuyéndola entre los que menos tienen, de forma que la masa total se conserve. Este proceso se conoce como "suavizado" y hay diversas estrategias y algoritmos para llevarlo a cabo. Por ejemplo, si un trigrama tiene probabilidad cero, se le puede asignar la misma probabilidad del trigrama que menos probabilidad tiene en el corpus, al tiempo que se les quita esta probabilidad de manera proporcional a los que más tienen. Este tipo de estrategias son muy útiles en la práctica ya que en un corpus suavizado todos los *n*-gramas tienen un valor, a pesar de que sus partes constitutivas no lo tengan.

Los modelos de lenguaje tienen una gran variedad de aplicaciones, como la corrección de errores gramaticales, reconocimiento de voz, análisis de opiniones, generación de lenguaje natural, medida de semejanza entre palabras, identificación de autoría, entre muchas otras, como se explica más adelante en este texto.

3.3.1.2. Corrección ortográfica

Supongamos por ejemplo que la frase *en quince monitos* ocurre en un texto. El error se puede detectar si la probabilidad del 3-grama *en quince monitos* es cero, pero, al mismo tiempo, la probabilidad del 2-grama *en quince* y del 1-grama *monitos* no lo son; por lo mismo, *monitos* en dicho contexto debe ser un error. Por otra parte *monitos* se parece a *minutos*, lo cual se puede evaluar por algún tipo

de medida de similitud entre palabras; supongamos adicionalmente que el 3-grama *en quince minutos* tiene una probabilidad diferente de cero. Consecuentemente es posible corregir el error substituyendo *monitos* por *minutos*. Por supuesto, esta explicación es sumamente simplificada pero se incluye para dar una idea de la intuición subyacente a este tipo de algoritmos. Algunos trabajos[29] proponen la corrección de errores gramaticales mediante un modelo de lenguaje de trigramas sintácticos[30] continuos y no continuos[31] obtenidos de un corpus de texto de dimensiones muy significativas.

3.3.2. Reconocimiento de voz

El proceso de traducir el habla a su representación textual por medio de un proceso computacional se conoce como "reconocimiento de voz". Este proceso se conceptualiza en relación a los niveles fonético-fonológico, léxico y morfológico. El reconocimiento se inicia con el análisis de la señal del habla para identificar la secuencia de unidades acústicas de la elocución. El proceso propiamente consiste en alinear dicha secuencia con una secuencia de pronunciaciones de un conjunto de palabras en el lexicón. Como el lexicón contiene a las palabras representadas como secuencias de fonemas, posiblemente agrupados en morfemas, se requiere adicionalmente traducir dicha forma abstracta a su realización ortográfica, lo cual se puede hacer conceptualmente mediante reglas que traduzcan fonemas y morfemas a grafemas (o símbolos ortográficos). Los algoritmos de reconocimiento de voz son normalmente probabilísticos, por lo que cada elocución puede dar lugar a un número muy significativo de hipótesis, cada una asociada a un valor de preferencia o *score*.

[29] Hernandez, S. D., Calvo, H. (2014). **Shared Task: Grammatical Error Correction with a Syntactic N-gram Language Model from a Big Corpora**. En *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task* (*CoNLL 2014)*, pp. 53–59.

[30] Sidorov, G., Gupta, A., Tozer, M., Catala, D., Catena, A., Fuentes, S. (2013). **Rule-based System for Automatic Grammar Correction Using Syntactic N-grams for English Language Learning (L2)**. En *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task* (*CoNLL 2013),* pp. 96–101.

[31] Sidorov, G. (2013). **N-gramas sintácticos no-continuos**. *Polibits*, 1(48):69–78.

Por otra parte, los modelos de lenguaje asignan una probabilidad a la elocución, la cual se puede tomar como su probabilidad a priori. Mediante este recurso adicional la hipótesis preferida por el reconocedor de voz será aquella cuyo producto de su *score* acústico y la probabilidad que le asigne el modelo del lenguaje sea mayor.

La creación de sistemas de reconocimiento de voz requiere de dos grandes esfuerzos de investigación: i) desarrollar los algoritmos de reconocimiento propiamente, incluyendo los algoritmos para crear los modelos del lenguaje y ii) crear los lexicones o diccionarios fonéticos de la lengua en cuestión para lo cual se requiere crear los modelos acústicos de todas las realizaciones de los fonemas del dialecto para el que se construye el reconocedor. Para esto último es necesario contar con un alfabeto fonético específico para el dialecto en el que se incluya un símbolo para cada alófono de cada fonema. Asimismo, se requiere contar con una gran cantidad de instancias de la realización acústica de cada alófono, recolectadas en los contextos acústicos en los que pueda ocurrir, para crear su modelo a nivel de la señal. También es necesario crear los diccionarios fonéticos propiamente, los cuales deben incluir todas las palabras del dialecto, normalmente en las decenas de miles, así como las diferentes formas en que cada palabra se puede pronunciar. Además, es necesario contar con corpus de grandes dimensiones en los dominios lingüísticos en los que se utilizará el reconocedor para crear los modelos del lenguaje correspondientes. La creación de estos recursos lingüísticos es también parte de la lingüística de corpus.

Con el fin de contar con un recurso lingüístico de esta naturaleza para el español de la Ciudad de México se desarrolló el alfabeto *Mexbet*,[32] el cual hizo posible diseñar, colectar y transcribir el corpus DIMEx100.[33] Asimismo, este corpus se utilizó para construir un diccionario fonético que incluye las

[32] Cuétara, J. **Fonética de la ciudad de México. Aportaciones desde las tecnologías del habla**, Tesis de Maestría, Universidad Nacional Autónoma de México, México, 2004.

[33] Pineda, et al. (2010). **The Corpus DIMEx100: Transcription and Evaluation**. *Language Resources and Evaluation*, 44:347–370. DOI: 10.1007/s10579-009-9109-9

realizaciones de cada palabra, transcritas adicionalmente en tres niveles de granularidad, y se validó con la construcción de un número muy significativo de reconocedores de voz utilizando los algoritmos del sistema Sphinx producido por la Universidad de Carnegie Mellon.[34]

3.3.3. Sistemas conversacionales en lenguaje natural

Desde la propuesta inicial de Turing el gran reto de la lingüística computacional ha sido la construcción de sistemas que puedan conversar con los seres humanos a través del lenguaje, especialmente hablado. Los esfuerzos en esta línea de investigación se iniciaron a mediados de la década de los sesenta con el programa Eliza[35] que simulaba ser un psicoterapeuta Rogeriano, y un poco más tarde por el programa SHRDLU,[36] ambos desarrollados en el MIT. Esta tradición se continuó en la década de los noventa con sistemas interactivos capaces ya de interactuar en inglés hablado, apoyados por máquinas inferenciales, principalmente para hacer planes, como los sistemas de la serie TRIPS y TRAINS desarrollados en la Universidad de Rochester.[37]

La construcción de sistemas conversacionales o de diálogo en lenguaje natural se ha abordado en México en el contexto del Proyecto Diálogos Inteligentes Multimodales en Español (DIME).[38] Este proyecto se enfocó originalmente en la creación del Corpus DIME[39] consistente en un conjunto de diálogos colaborativos para la solución de tareas, los cuales se etiquetaron ortográficamente, así como en los niveles fonético, prosódico y entonativo, léxico, sintáctico y pragmático. Para el nivel sintáctico se desarrolló una gramática del español de México con énfasis en el sistema de clíticos dentro de

[34] https://es.wikipedia.org/wiki/CMU_Sphinx
[35] https://en.wikipedia.org/wiki/ELIZA
[36] https://en.wikipedia.org/wiki/SHRDLU
[37] Allen, J., Ferguson, G. **TRIPS: An Integrated Intelligent Problem-Solving Assistant**. *Proceedings of AAI*, 1998.
[38] http://turing.iimas.unam.mx/~luis/DIME/
[39] Idem

la perífrasis.[40] Para el nivel pragmático se desarrolló el esquema de análisis DIME-DAMLS[41] mediante el cual se obtuvo la estructura de los actos del habla, directos e indirectos, relativos a la tarea propiamente, a la administración de la tarea y a la administración de la comunicación.

Estas ideas se desarrollaron en paralelo con un modelo para la administración de diálogos computacionales enfocado a la interpretación de los actos del habla en relación a un contexto, el cual se caracteriza como una gráfica recursiva de situaciones llamada "modelo de diálogo", donde cada situación se define en términos de las expectativas y acciones intencionales del agente computacional. En este modelo una situación puede también embeber a un modelo de diálogo subordinado, de tal forma que la estructura del grafo corresponde a la estructura de la conversación. Dichas ideas dieron lugar a la creación del lenguaje de programación SitLog,[42] para la especificación e interpretación de modelos de diálogo en sistemas conversacionales y se incorporaron al Proyecto Golem,[43] para el desarrollo de robots de servicio capaces de comunicarse con los seres humanos en lenguaje hablado, tanto en español como en inglés.

3.3.4. Procesamiento de textos

El procesamiento de textos en México, especialmente para el análisis de los niveles léxico y sintáctico, ha tenido una influencia muy significativa del formalismo de dependencias; esto se debe a la correspondencia directa con los componentes de roles semánticos que conforman la oración, para un rango de

[40] Ver, por ejemplo: Pineda, L. A., Meza, I. (2005). **The Spanish Pronominal Clitic System**. *Procesamiento del Lenguaje Natural*, 34:67–103.

[41] Pineda et al. (2006). **Balancing Transactions in Practical Dialogues.** *Proceedings of CICLing 2006, LNCS 3878*, pp. 331–342. Ver también: Pineda, L. A., Estrada, V., Coria, S., Allen, J. (2007). **The obligations and common ground structure of practical dialogues**. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 11(36): 9–17.

[42] Pineda, L. A. et al. (2013). **SitLog: A Programming Language for Service Robot Tasks**. *Int J Adv Robot Syst*, 10:358. doi: 10.5772/56906

[43] http://golem.iimas.unam.mx/

aplicaciones desde recuperación de información,[44] búsqueda de respuestas[45,46] y minería de texto[47,48] hasta especificaciones de software.[49] Una gran variedad de enfoques semánticos, como grafos conceptuales,[50] Recursión de Semántica Mínima (MRS)[51] o redes semánticas, tienen rasgos similares a un conjunto de predicados.

El trabajo que se ha realizado con el enfoque de dependencias (especialmente para el idioma español) ha sido relativamente reciente, principalmente a partir del año 2000. En particular se ha utilizado *Connexor*,[52] un analizador sintáctico de dependencias que está disponible comercialmente en varios idiomas (inglés, español, francés, alemán, sueco, finlandés, y posiblemente otros). Este analizador se basa en el formalismo de *gramáticas de dependencia funcional* (FDG).[53] En México se desarrolló el analizador sintáctico para el es-

[44] Villatoro-Tello, E., Chavéz-García, O., Montes-y-Gómez, M., Villaseñor-Pineda, L., Sucar, L.. (2010). **A Probabilistic Method for Ranking Refinement in Geographic Information Retrieval**. *Procesamiento de Lenguaje Natural*, 44:123–130.

[45] Montes-y-Gómez, M., Villaseñor-Pineda, L., López-López, A. (2008). **Mexican Experience in Spanish Question Answering**. *Computación y Sistemas*, 12(1):40–60.

[46] Téllez-Valero, A., Montes-y-Gómez, M., Villaseñor-Pineda, L., Peñas-Padilla, A. (2011). **Learning to Select the Correct Answer in Multi-Stream Question Answering**. *Information Processing and Management,* 47(6):859–869.

[47] Ramírez-de-la-Rosa, G., Montes-y-Gómez, M., Solorio, T., Villaseñor-Pineda, L. (2013). **A document is known by the company it keeps: Neighborhood consensus for short text categorization**. *Journal of Language Resources and Evaluation*, 47(1): 127–149.

[48] Montes-y-Gómez, M., Gelbukh, A., López-López, A. (2002). **Text Mining at Detail Level Using Conceptual Graphs**. *Uta Priss et al. (Eds.): Conceptual Structures: Integration and Interfaces, 10th Intern. Conf. on Conceptual Structures, ICCS-2002, Bulgaria, Lecture Notes in Computer Science, N 2393*, Springer-Verlag, pp. 122–136.

[49] Isabel, D., Moreno, L., Fuentes, I., Pastor, O. (2005). **Integrating Natural Language Techniques in OO-Method**. *Gelbukh, A. (ed.), Computational Linguistics and Intelligent Text Processing (CICLing-2005), Lecture Notes in Computer Science, 3406*, Springer-Verlag, pp. 560–571.

[50] Sowa, J. F. **Conceptual Structures: Information Processing in Mind and Machine**, Addison-Wesley, 1984.

[51] Copestake, A., Flickinger, D., Sag, I. **Minimal Recursion Semantics. An introduction**, CSLI, Stanford University, 1997.

[52] http://www.connexor.eu/technology/machinese/demo/syntax

[53] Tapanainen, P. y Järvinen, T. (1997). **A non-projective dependency parser**. *Proceedings of the 5th Conference on Applied Natural Language Processing, Washington, D.C.*, pp. 64–71.

pañol con el enfoque de dependencias DILUCT.[54] Su algoritmo utiliza reglas heurísticas para descubrir relaciones entre palabras, además de estadísticas de coocurrencias de palabras, las cuales aprende de una manera no supervisada para resolver ambigüedades, como la de adjunción de frase preposicional.[55] En las siguientes secciones se presentan varias aplicaciones desarrolladas con este enfoque.

3.3.4.1. Análisis de opinión

El análisis de opinión, también conocido como extracción de opiniones, minería de opiniones, análisis subjetivo o análisis de sentimiento (*sentiment analysis*), ayuda a conocer la percepción de la comunidad acerca de productos o servicios. Este análisis se basa frecuentemente en la información textual disponible en las redes sociales, como Facebook y *twitter*, incluyendo los llamados "emoticones".

Para este análisis se consideran la fuente o emisor, el objetivo o receptor y el tipo de actitud o polaridad, que puede ser positiva o negativa. El análisis de opinión puede ser simple, complejo o avanzado, dependiendo respectivamente de si sólo se reporta la polaridad, de si además se reporta el grado de la actitud, o si también se reporta la fuente y el objetivo, posiblemente con otras características, como teléfonos y fechas, así como los llamados *hashtags* en *twitter*.

La metodología consiste en crear corpus anotados a diferentes niveles de granularidad con los parámetros de opinión y utilizar estos recursos para crear clasificadores mediante toda la gama de algoritmos de aprendizaje au-

[54] Calvo H., Gelbukh A. (2006). **DILUCT: An Open-Source Spanish Dependency Parser Based on Rules, Heuristics, and Selectional Preferences**. *En Kop C., Fliedl G., Mayr H.C., Métais E. (eds) Natural Language Processing and Information Systems. NLDB 2006. Lecture Notes in Computer Science, vol 3999*. Springer, Berlin, Heidelberg, pp. 164–175.

[55] Calvo, H., Gelbukh, A. (2003). **Improving prepositional phrase attachment disambiguation using the web as corpus**. En *Proceedings of the 8th Iberoamerican Congress on Pattern Recognition (CIARP 2003),* Springer Berlin, Heidelberg, pp. 604–610.

tomático, para clasificar o anotar los textos analizados. Esta tarea se puede realizar también a través de lexicones de opinión[56] o mediante la medición de distancias semánticas a diversos conceptos.[57] Éstos se pueden crear de manera automática a partir de corpus anotados.[58]

3.3.4.2. Detección de ironía

Una tarea complementaria al análisis de opinión es el análisis de ironía, ya que esta figura retórica consiste precisamente en expresar una proposición mediante su negación, pero esperando que el interlocutor haga la interpretación correcta. El problema es que si una proposición se expresa irónicamente el análisis de opinión anotará las opiniones positivas como negativas y viceversa. La detección de ironía se ha abordado dentro del procesamiento de textos mediante modelos basado en *n*-gramas simples de categorías gramaticales, de palabras frecuentemente utilizadas en textos con tono humorístico, tales como aquellas relacionadas con sexualidad, relaciones humanas, parentesco, así como de medidas de palabras usualmente clasificadas como positivas o negativas.[59] También se han considerado características de afectividad usando el recurso de WordNet-affect,[60] y una medida de complacencia basada en diccionarios de términos afectivos.[61]

---

[56] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[57] Calvo, H. (2015). **Opinion analysis in social networks using antonym concepts on graphs**. En Proc. 2nd. Int. Conf. on Future Data and Security Engineering (FDSE 2015), L NCS, Springer 9446:109–120.

[58] Minqing, H., Liu, B. (2004). **Mining and summarizing customer reviews**. En *Proceedings of the tenth ACM SIGKDD Int. Conf. on Knowledge discovery and data mining, ACM*, *(KDD 2004),* pp. 168-177.

[59] Reyes, A., Rosso, P., Veale, T. (2013). **A multidimensional approach for detecting irony in Twitter**. *Language Resources and Evaluation*, 47(1):239–268.

[60] Strapparava, C., Valitutti, A. (2004). **WordNet-Affect: an Affective Extension of WordNet**. *Proceedings of the 4th* Int. Conf. on Language Resources and Evaluation *(LREC 2004)*, Lisbon, pp. 1083–1086.

[61] Antonio, R., Rosso, P. (2012). **Making objective decisions from subjective data: Detecting irony in customer reviews**. *Journal on Decision Support Systems*, 53(4):754–760.

3.3.4.3. Semejanza entre palabras y diccionarios de ideas afines

Los diccionarios de ideas afines o tesauros (*thesaurus* en inglés) son recursos lingüísticos que organizan las palabras de acuerdo a la relación que guardan entre sí, tales como sinonimia, antonimia, hiperonimia, hiponimia, meronimia, holonimia, etc. Mediante estas relaciones es posible establecer "distancias" entre palabras;[62] por ejemplo, las palabras más cercanas a *cebra* son *jirafa*, *búfalo*, *hipopótamo*, *rinoceronte*, *gacela* y *antílope*; las próximas a *excepción* son *exención*, *limitación*, *exclusión*, *instancia*, *modificación*; y a *jarrón* son *tazón*, *sartén*, *jarra*, *contenedor*, *platillo* y *taza*.

Los tesauros tienen muchas aplicaciones dentro del procesamiento de lenguaje natural, tales como el análisis sintáctico,[63] la interpretación de conjunciones, la resolución de anáforas, la medición de cohesión en textos, la desambiguación de sentidos de palabras (WSD), la corrección ortográfica y el reconocimiento de voz, entre muchas otras. Algunos de los tesauros más utilizados son: WordNet,[64] Roget,[65] WASPS,[66] Word Sketches[67] y Medical Subject Headings (*MeSH*).[68]

---

[62] Ortega, R. M., Aguilar, C., Villaseñor-Pineda, L., Montes, M., Sierra, G. (2011). **Hacia la identificación de relaciones de hiponimia/hiperonimia en Internet**. *Revista Signos*, 44(75):68–84.

[63] Calvo, H., Gelbukh, A. (2004). **Acquiring selectional preferences from untagged text for prepositional phrase attachment disambiguation**. En *Proc Int. Conf. on Application of Natural Language to Information Systems,* Springer, pp. 207–216.

[64] WordNet puede consultarse en línea en wordnet.princeton.edu

[65] Roget, P. M., Dutch, R. A., ed., **The Original Roget's Thesaurus of English Words and Phrases** (Americanized ed.), New York: Longmans, Green & Co./Dell Publishing Co., Inc., 1962.

[66] Kilgarriff, A, (2003). **WASPS: Thesauruses for natural language processing**. *Proceedings of Natural Language Processing and Knowledge Engineering*. DOI: 10.1109/NLP-KE.2003.1275859

[67] Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D. (2004). **Itri-04-08 the sketch engine**. *Information Technology*, pp. 105–116.

[68] Rogers, F. B. (1963). **Medical subject headings**. *Bull Med Libr Assoc.* 51:114–6.

WordNet es una base de datos organizada de manera jerárquica que contiene un tesauro en línea junto con un diccionario para el idioma inglés. A través del proyecto EuroWordNet también está disponible para otros idiomas, como el español, italiano, alemán, francés, sueco, checo y estonio. WordNet define los sentidos utilizando un concepto llamado *synset* (conjunto de sinónimos). El *synset* contiene un conjunto de palabras que están relacionadas de manera cercana con la definición de la palabra, como se ilustra en los ejemplos mencionados.

3.3.4.4. Desambiguación del sentido de las palabras

La desambiguación del sentido de las palabras se aborda desde tres enfoques diferentes: i) la desambiguación basada en conocimiento que usa fuentes léxicas externas tales como diccionarios y tesauros,[69, 70] aunado al uso de propiedades del discurso; ii) la desambiguación supervisada, la cual utiliza algoritmos de aprendizaje de máquina y se requiere contar con un corpus en el que se etiquete el sentido apropiado de las palabras[71] y iii) la desambiguación no supervisada, que requiere un conjunto de entradas codificadas, pero no necesariamente el etiquetado del sentido apropiado.[72] También hay enfoques mínimamente supervisados y el sentido más frecuentemente utilizado.[73] En

[69] Calvo, H., Gelbukh, A., Kilgarriff, A. (2005). **Distributional thesaurus versus WordNet: A comparison of backoff techniques for unsupervised PP attachment**. En *International Conference on Intelligent Text Processing and Computational Linguistics,* Springer, Heidelberg, pp. 177–188.

[70] Tejada-Cárcamo, J., Calvo, H., Gelbukh, A. (2008). **Improving unsupervised WSD with a dynamic thesaurus**. En *International Conference on Text, Speech and Dialogue*, Springer Berlin, Heidelberg, pp. 201–210.

[71] Rosso, P., Montes-y-Gómez, M., Buscaldi, D., Pancardo-Rodríguez, A., Villaseñor-Pineda, L. (2005). **Two Web-based approaches for Noun Sense Disambiguation**. *International Conference on Intelligent Text Processing and Computational Linguistics, CICLing-2005. Mexico City,. Lecture Notes in Computer Science 3406*, Springer, pp. 267–279.

[72] Calvo, H. (2008). **Augmenting word space models for Word Sense Discrimination using an automatic thesaurus**. En *Advances in Natural Language Processing.* Springer Berlin Heidelberg, pp. 100–107.

[73] Cárcamo, J. T., Gelbukh, A., Calvo, H. (2008). **An innovative two-stage WSD unsupervised method**. *Procesamiento del lenguaje natural*, 40:99–105.

los enfoques basados en aprendizaje de máquina se construye un vector de características que representan al contexto.[74] En particular para el español, se evalúan los trabajos a través de concursos internacionales como SENSEVAL-2, SENSEVAL-3, que después dieron lugar a SEMEVAL.[75]

3.3.4.5. Detección de engaño

El problema de detección de engaño se ha estudiado ampliamente, para determinar si una opinión es verdadera o falsa, es decir que tiene por fin engañar al lector. En esta tarea, el texto se representa también como vectores de características que se utilizan para entrenar clasificadores mediante técnicas de aprendizaje de máquina.[76] En la mayoría de los trabajos se usan herramientas que requieren información específica del dominio, tales como *n*-gramas sintácticos y diccionarios léxicos de emociones. Sin embargo, otros trabajos consideran únicamente aspectos distribucionales del texto,[77] por ejemplo mediante el uso de algoritmos de modelado de tópicos dentro de un entorno probabilístico. Estos trabajos combinan sus características con otras obtenidas a partir de diversas fuentes, como representación de espacios de palabras, categorías gramaticales o diccionarios de aspectos psicológicos de las palabras. Para efectos de evaluación existen diversos conjuntos de datos enfocados a dominios específicos. Aunque se ha tratado de encontrar un detector universal de engaño con alto desempeño, esto todavía no se ha logrado.

---

[74] Tejada-Cárcamo, J., Calvo, H., Gelbukh, A., Hara, K. (2010). **Unsupervised WSD by finding the predominant sense using context as a dynamic thesaurus**. *Journal of Computer Science and Technology*, 25(5):1030–1039.

[75] http://www.senseval.org

[76] Hernández Fusilier, D., Guzmán Cabrera, R., Montes-y-Gómez, M, Rosso, P. (2015). **Detection of Positive and Negative Deceptive Opinions with PU-learning**. *Information Processing and Management*, 51:433–443.

[77] Hernández-Castañeda, A., Calvo, H., Gelbukh, A., Flores, J. J. G. (2017). **Cross-domain deception detection using support vector networks**. *Soft Computing*, 21:585.

3.3.4.6. Detección de autoría

La identificación de autor es otro problema que se aborda dentro del procesamiento de textos. Por ejemplo, es necesario saber qué autor escribió un libro anónimo, identificar la autoría de las notas de un posible criminal, etc. Esta tarea puede ser muy compleja cuando se realiza en un entorno abierto, es decir, cuando no se tiene un conjunto predefinido de posibles autores. Es por ello que se han creado tareas más específicas donde se incluyen varios documentos de un autor conocido,[78,79] y un documento y un autor desconocido. Entre más cortos son los textos, la tarea se dificulta más, pues existen menos pistas que permitan distinguir el estilo de un autor. En casos reales como el campo forense, difícilmente se cuenta con textos largos. Otra tarea asociada con esta problemática es la identificación del perfil del autor, cuyo objetivo es inferir el género, lugar de origen, rango de edad e incluso rasgos de personalidad del autor a partir de los temas sobre los que versa el texto y su estilo.[80]

Se han considerado diversos enfoques para obtener características más informativas basadas en el estilo; también es posible generar características al extraer información léxica, sintáctica o semántica. La información léxica usualmente se limita a conteos de palabras y ocurrencias de palabras comunes. Por otra parte, mediante la información sintáctica es posible obtener, hasta cierto punto, el contexto de las palabras. Algunos trabajos usan información semántica léxica para encontrar características que permitan discriminar los textos mediante modelos probabilísticos de distribuciones latentes.

---

[78] Coyotl-Morales, R. M., Villaseñor-Pineda, L., Montes-y-Gómez, M., Rosso, P. (2006). **Authorship Attribution using Word Sequences**. *11th Iberoamerican Congress on Pattern Recognition, CIARP 2006. Lecture Notes in Artificial Intelligence 4225*, Springer, pp. 844–853.

[79] Escalante, H. J., Solorio, T., Montes-y-Gómez, M. (2011). **Local Histograms of Character n-grams for Authorship Attribution**. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*. Portland, Oregon, USA, pp. 19–24.

[80] López-Monroy, A. P., Montes-y-Gómez, M., Escalante, H. J., Villaseñor-Pineda, L., Stamatatos, E. (2015). **Discriminative Subprofile-Specific Representations for Author Profiling in Social Media**. *Knowledge Based Systems*, 89:134–147.

3.3.4.7. Reconocimiento de paráfrasis e implicación textual

Otro reto del procesamiento de textos es el parafraseo, el cual consiste en reconocer si dos expresiones significan lo mismo, o reformular una expresión con otra que tenga el mismo significado. El reconocimiento de paráfrasis puede ser utilizado en aplicaciones como extracción de información, sistemas de pregunta y respuesta, generación de resúmenes de múltiples documentos, detección de plagio, etcétera.

Considere las oraciones E1: *Carlos aprecia la comida francesa*; E2: *A Carlos le gusta la cocina francesa* y E3: *Carlos aprecia la comida francesa picante.* A pesar de que E1 y E2 son oraciones compuestas por distintas palabras la idea que expresan es la misma y por lo tanto se deben considerar como paráfrasis. Por su parte, aunque E2 y E3 transmiten la idea de que a Carlos le gusta la comida francesa, no se pueden considerar como paráfrasis ya que E3 es más específica que E2.

De forma general el procesamiento de paráfrasis puede ser dividido en tres grandes tareas: i) extracción, ii) generación y iii) reconocimiento. La extracción tiene como objetivo obtener el conjunto más grande posible de pares de expresiones que conforman paráfrasis; esto se realiza a partir de un corpus de referencia. La generación, tiene como objetivo generar el mayor conjunto de expresiones en lenguaje natural que sean paráfrasis de la expresión objeto de análisis. Las expresiones generadas deben tener los menos errores posibles. Finalmente, el reconocimiento tiene por objetivo determinar si dos expresiones de entrada son o no paráfrasis.[81] Esta tarea se requiere a su vez para abordar otras, como la detección de plagio.[82]

---

[81] Por ejemplo, Calvo, H., Segura-Olivares, A., García, A. (2014). **Dependency vs. constituent based syntactic n-grams in text similarity measures for paraphrase recognition**. *Computación y Sistemas*, 18(3):517–554.

[82] Sánchez, F., Villatoro, E., Montes, M., Villaseñor, L., Rosso, P. (2013). **Determining and Characterizing the Reused Text for Plagiarism Detection**. *Expert Systems with Applications*. 40(5):1804–1813.

Una tarea relacionada es el reconocimiento de implicación textual. Ésta consiste en identificar si un determinado texto, denominado hipótesis, se implica o se puede inferir a partir de otro texto. Existen trabajos que se basan en enfoques de reconocimiento léxicos, sintácticos y semánticos, los cuales se complementan con diversas técnicas como lematización, eliminación de palabras sin contenido, manejo de negación, relaciones semánticas y semejanza entre palabras.[83] Estos métodos se basan en medir la razón o porcentaje de cobertura de la hipótesis con respecto al texto dado. Para efectos de evaluación se utiliza el marco de referencia PASCAL de reconocimiento de implicación textual.[84]

3.4. Perspectivas

La Lingüística Computacional se seguirá desarrollando a lo largo de todo el siglo XXI. El incremento de la capacidad de cómputo, memoria y conectividad a costos muy bajos, aunado a la proliferación de repositorios de información y dispositivos móviles, aumentarán la infraestructura para explotar la tecnología así como la demanda de servicios de información de carácter lingüístico. En particular hay dos tipos de demanda de gran escala, que además son ubicuos en todo el espectro de aplicaciones. El primero es la necesidad de acceder a textos en formatos digitales en una amplia gama de formatos: libros, revistas, periódicos, páginas personales, etc., además de la información disponible en las redes sociales, cuya diversidad, amplitud y alcance continuarán extendiéndose en los próximos años. En este rubro también se incluyen los recursos textuales relacionados con dominios específicos, como la salud, las leyes, la educación y la capacitación, etc. Es necesario considerar adicionalmente que estos recursos estarán disponibles en prácticamente todos los lenguajes y dialectos. Por lo mismo será cada vez más necesario

[83] Segura-Olivares, A., García, A., Calvo, H. (2013). **Feature Analysis for Paraphrase Recognition and Textual Entailment**. *Research in Computing Science*, 70:119–144.

[84] Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B. (2007). **The Third PASCAL Recognizing Textual Entailment**. RTE '07 *Proceedings of the ACL-PASCAL Workshop on Textual Entailmente and Paraphrasing,* pp. 1–9.

localizar información, extraerla, traducirla, resumirla y expresarla de forma relevante y accesible. El segundo tipo de demanda provendrá de la necesidad de comunicarse con dispositivos computacionales a través del lenguaje natural, especialmente hablado.

Desde el punto de vista de los enfoques teóricos y de la práctica de la especialidad, los métodos estadísticos y el aprendizaje de máquina, como el aprendizaje profundo o *Deep Learning* actualmente de moda, continuarán siendo populares en los próximos años, hasta que estas líneas se agoten y se abran o se retomen otras metáforas. En particular, los métodos actuales producen descripciones globales con información cualitativa que orientan acerca de las tendencias de los fenómenos estudiados y permiten tomar decisiones de carácter genérico. Por ejemplo, la extracción de información, el análisis de sentimiento e incluso la implicación textual producen respuestas cualitativas con un indicador de su grado de certeza o confiabilidad, lo cual permite saber, respectivamente, que hay un número de documentos en Google relacionados con cierta temática, que un producto comercial tiene cierto grado de aceptación en una comunidad o que posiblemente una proposición se siga de un texto.

Sin embargo, también es importante analizar al lenguaje de manera determinada y poder utilizar la información lingüística de manera causal, como cuando Juan cierra la ventana porque Pepe le dice "¿qué tal tu veranito?". Este nivel de comprensión, que es el que realizamos los seres humanos durante la conversación o en la lectura se puede abordar con las técnicas estadísticas sólo de manera limitada; por lo mismo, es posible que su estudio se retome con un grado mayor de madurez cuando se tenga una comprensión más profunda de la facultad lingüística y del cómputo natural, que es el que realizamos los humanos y también otros seres vivos, en oposición al cómputo artificial o de ingeniería enfocado a algoritmos, que es producto de la invención humana.

La meta propuesta por Turing de crear una máquina capaz de entender el lenguaje natural no se ha logrado todavía y no se alcanzará en el corto plazo. Programas como Eliza, que sostenía una conversación aparentemente inteligente simplemente reflejando el lenguaje del interlocutor humano iniciaron una tradición que busca "ganar la prueba de Turing" con trucos superficiales. Esta propuesta ha resultado tenaz y su manifestación actual son los llamados "Chatbots" integrados a sistemas de diálogo, que se utilizan en los *call centers* para comprar boletos de avión o para hacer reservaciones en restaurantes. Una aplicación más ubicua es conversar con un teléfono celular, como Siri,[85] que se adapta al usuario con el apoyo de algoritmos de aprendizaje. Sin embargo, por más inteligentes que estos sistemas parezcan su nivel de comprensión es superficial y son muy sensibles al tipo de información que reciben ya que más que comprender, su meta es engancharse con el usuario humano, a quien tienen que agradar. Un ejemplo reciente de las limitaciones de este tipo de "inteligencia por asociación" es Tay, un Chatbot producido por Microsoft que a las pocas horas de su lanzamiento empezó a enviar twitts racistas y misóginos, además de que su conducta se hizo repetitiva y predecible.[86]

Un escenario posible en el futuro es que los sistemas conversacionales con modelos superficiales se sigan desarrollando y que prácticamente todos los dispositivos con los que interactuamos los humanos, desde las licuadoras hasta los coches y aviones, tengan su sistema de diálogo que les permita conversar acerca de sus objetivos y funcionalidades. En este entorno habrá una percepción de que las máquinas hablan pero siempre de forma esquemática y superficial.

Por otra parte, para decir que las máquinas realmente entienden, éstas tendrían que interpretar y generar expresiones en relación al contexto, desplegando una amplia gama de actos del habla directos e indirectos que sean causales a su conducta. Esto a su vez dependerá de contar con una noción

[85] https://en.wikipedia.org/wiki/Siri
[86] https://en.wikipedia.org/wiki/Tay_(bot)

coherente y comprensiva del contexto, es decir, del nivel de representación pragmático y de su interacción con los demás niveles de representación lingüística. Estas máquinas podrían llegar a existir pero a la luz del estado actual del conocimiento del lenguaje y de la tecnología computacional no es posible afirmarlo todavía.

4. Robótica de Servicio

4.1. El espacio de la Robótica

La robótica estudia los mecanismos físicos controlados por un programa de cómputo que interactúan con los seres humanos y/o el entorno y realizan tareas que involucran un proceso informacional y una acción motora o perceptual de manera coordinada. Aunque la palabra "robot" proviene de la ciencia ficción y tiene como antecedentes los autómatas mecánicos que se han construido a lo largo de la historia,[1] la robótica como disciplina científica y tecnológica se inicia con la invención de los brazos autónomos para asistir la manufactura en líneas de producción, especialmente en la industria automotriz, con el robot *Unimate* en 1961.[2] Esta invención dio lugar al desarrollo de una industria de grandes dimensiones con gran impacto económico y estratégico en el mundo.

A partir de estos robots iniciales, en conjunto con el desarrollo de las disciplinas relacionadas, como el Procesamiento de Señales, la Teoría de Control y la Inteligencia Artificial, durante las últimas décadas se han diseñado y construido una gran cantidad de robots o dispositivos robóticos que se pueden conceptualizar o clasificar en relación a dos dimensiones principales: i) autonomía y ii) complejidad del entorno. La primera depende de la riqueza

[1] https://en.wikipedia.org/wiki/Robotics
[2] https://en.wikipedia.org/wiki/Unimate

de sus sensores y de la variedad y estructura de sus mecanismos motores e informacionales o "mentales" y, consecuentemente, de la riqueza de estímulos o informaciones que el robot es capaz de percibir y/o interpretar, y de la cantidad y estructura de los tipos de acciones que puede realizar para conseguir sus objetivos de manera autónoma.

La segunda corresponde a la variedad y variabilidad del entorno que el robot es capaz de enfrentar e incluso transformar. Esta dimensión no se trata de la extensión espacial donde los robots pueden funcionar e interactuar, sino de la cantidad y complejidad de entornos cualitativamente diferentes; por ejemplo, aunque las hormigas, las cucarachas o los osos de agua son ubicuos en el planeta, sus micro-entornos locales, en relación a su escala, son muy similares en todos lados, en oposición a los entornos globales complejos, cualitativamente diferentes, como los que podemos enfrentar los seres humanos. Aunque la diversidad y la variabilidad del entorno son dimensiones diferentes están relacionadas ya que entornos simples tienden a variar poco y entornos complejos mucho, por lo que se pueden colapsar en una sola dimensión a la que nos referimos como "complejidad". Mientras mayor sea la autonomía del robot al actuar e interactuar en entornos complejos, mayor será su inteligencia.

De acuerdo con estas consideraciones es posible describir el mapa de la robótica en el espacio definido por estas dos dimensiones, como se ilustra en la Figura 4.1. En el origen están los robots con un solo tipo de acción en un entorno fijo, como los brazos industriales y en la esquina superior derecha los robots humanoides de la ciencia ficción. Asimismo la gran mayoría de los robots actuales y potenciales estarían sobre una nube a lo largo de la diagonal principal ya que la complejidad del robot debe ser coherente con la complejidad de su entorno, aunque habría muchas excepciones, como los robots completamente autónomos dirigidos a actuar en entornos de complejidad baja o moderada.

Por otra parte, desde una perspectiva funcional y de su utilidad, el espacio de los robots se puede dividir en tres grandes regiones o semiplanos paralelos verticales de izquierda a derecha: i) industriales, ii) controlados de manera remota y iii) completamente autónomos. La primera región corresponde a los robots de las líneas de producción cuyo entorno no varia o varía muy poco y su protocolo de acción está definido de antemano, por lo que sus procesos de percepción, toma de decisiones y acción intencional son muy limitados. La segunda incluye mecanismos con la capacidad de explorar y manipular el entorno pero la información que obtienen a través de sus sensores se envía a un operador humano quien los controla de manera remota y decide sus acciones. Estos robots tienen grandes aplicaciones actuales y potenciales, ya que pueden operar en ambientes peligrosos o inaccesibles para los seres humanos, como inspeccionar el núcleo de un reactor nuclear o buscar sobrevivientes después de un terremoto. La tercera categoría corresponde a robots capaces de percibir e interpretar el mundo y tomar decisiones de manera autónoma, principalmente en entornos complejos, aunque, salvo algunas excepciones, como los vehículos autónomos, los robots de esta última categoría están todavía en los laboratorios de investigación. En la Figura 4.1 se ilustran algunos ejemplos de robots en este espacio.

Figura 4.1. Ejemplos de robots en el Espacio de la Robótica.

4.2. Robots de Servicio

Con base en el trabajo de investigación y desarrollo de la comunidad mexicana de computación, en este capítulo nos enfocamos a los robots de servicio, un tipo de robot de media o alta autonomía capaz de asistir a los seres humanos en entornos de complejidad baja o moderada. Es importante enfatizar que estos robots son diferentes a los industriales ya que su objetivo es simplificar el trabajo humano en casas, oficinas, hospitales, etc., para lo cual tienen que moverse de manera intencional, evitando obstáculos,[3] reconocer e identificar a los humanos, comunicarse a través del lenguaje hablado, ver, tomar, llevar y entregar objetos entre otros tipos de acciones o conductas intencionales. Los robots de servicio deben también ser capaces de hacer diagnósticos, tomar decisiones y hacer planes para llevar a cabo sus funciones de manera exitosa. Se espera que en un futuro cercano dentro de la casa habrá robots pequeños que limpien, aspiren y trapeen el piso; robots fijos encargados del lavado y planchado de la ropa; y robots de tipo humanoide que fungirán como asistentes generales. En el exterior habrá robots que cortarán el pasto y otros que harán rondines de vigilancia. Así como se incorporaron a la vida cotidiana los televisores, las computadoras y los teléfonos celulares, los robots de servicio también lo harán y llegarán a ser muy familiares, si se cumplen las expectativas que tenemos actualmente. Se espera que una de las aplicaciones principales de esta tecnología en el mediano plazo sea asistir a adultos mayores como acompañantes, para explicar acciones rutinarias, llevar la lista y toma de medicinas, apoyar en el ejercicio físico, proveer información de calendario y hora, proveer información espacial, detectar accidentes, como caídas, o mejor aún, mantener la casa segura para evitar accidentes o situaciones de riesgo (ej. apagar la estufa, recoger objetos del suelo, etc.). En el ámbito hospitalario para transportar medicinas, análisis y alimentos, guiar a los médicos en las visitas y monitorear los signos vitales de los pacientes.

[3] En la terminología de la especialidad esta acción se conoce como "navegar" y a la sub-disciplina que aborda la problemática relacionada como "navegación".

4.3. Conceptualización y arquitectura de Robots de Servicio

En el desarrollo de los robots de servicio se han observado de forma genérica tres paradigmas o enfoques de robots móviles, cada uno de los cuales determina hasta cierto punto la arquitectura del robot. El primero, llamado "deliberativo", se desarrolló desde los años sesenta del siglo pasado, basado en el paradigma simbólico y representacional del programa de la Inteligencia Artificial. Este enfoque parte de una representación del mundo y a través de un proceso de planeación se determina la secuencia de acciones que el robot debe realizar para lograr sus objetivos. Un ejemplo de este tipo de arquitecturas es la del robot Shakey[4] desarrollado en la Universidad de Stanford. El segundo enfoque, llamado "reactivo",[5] se desarrolló en los años ochenta en el Instituto Tecnológico de Massachusetts (MIT) por el investigador Rodney Brooks; en éste no se cuenta con una representación del medio ambiente, ni tampoco con módulos de planeación de acciones ni movimiento, sino que se utilizan varios módulos que responden a estímulos específicos provenientes del entorno y se generan salidas inmediatas las cuales son combinadas o seleccionadas por un árbitro para generar lo que el robot debe hacer bajo ciertas circunstancias. Los módulos pueden utilizar máquinas de estados, redes neuronales o campos potenciales. El tercer enfoque, llamado "probabilista" se basa en el concepto de que tanto el sensado como las acciones del robot en ambientes complejos son inciertos y se representan mediante variables aleatorias, las cuales pueden ser manipuladas para obtener un plan de acción robusto. Un ejemplo de este enfoque es el automóvil autónomo que obtuvo el primer lugar en el *DARPA Grand Challenge* del 2005 desarrollado por Sebastian Thrun de la Universidad de Stanford.[6]

[4] http://www.ai.sri.com/shakey/

[5] https://en.wikipedia.org/wiki/Behavior-based_robotics

[6] Thrun, S., et al. (2006). **Stanley: The Robot that Won the DARPA Grand Challenge**. *Journal of Field Robotics,* 23(9): 661–692.

4.4. Modelo conceptual y niveles de sistema en Robótica

Desde el punto de vista de su uso, diseño y construcción, los robots se pueden conceptualizar en tres niveles de sistema: i) funcional, ii) de dispositivos y algoritmos y iii) de implementación.[7] El primer nivel corresponde a la perspectiva de los seres humanos o usuarios que interactúan con el robot: qué servicios ofrece en términos prácticos, cómo se estructuran las tareas que puede realizar y cuál es la estructura de la comunicación o la interacción con los agentes en su entorno. Si las expectativas que tenemos para los robots autónomos se llegaran a materializar éstos se comprarán en agencias, como las de coches, y vendrán con un manual de usuario con la especificación de sus conductas básicas y las formas de combinarlas para realizar tareas compuestas. Dicho manual correspondería con la especificación del nivel funcional. Este nivel funcional tiene también un impacto directo en la arquitectura del robot ya que determina en buena medida cómo se utilizan los dispositivos y algoritmos robóticos así como la forma en que el robot interactúa con otros agentes y cómo responde a los eventos que ocurren en su entorno, tanto naturales como los producidos por la acción de otros agentes de manera intencional.

El nivel de dispositivos y algoritmos corresponde al diseño y construcción de artefactos robóticos junto con los algoritmos o programas que los habilitan: brazos con manos capaces de tomar objetos frágiles con la fuerza suficiente para sostenerlos pero sin romperlos; cabezas con cámaras y algoritmos de visión para reconocer objetos y determinar su posición y orientación, que en conjunto con los brazos y las manos habiliten al robot para tomar objetos; plataformas móviles o piernas que le permitan al robot trasladarse de una ubicación a otra, asociados a algoritmos que le permitan construir un mapa de su entorno y moverse evadiendo obstáculos; dispositivos y algoritmos que lo habiliten para interactuar o comunicarse a través del lenguaje hablado, o reconocer ruidos y voltear hacia la fuente sonora y ponerla en su

[7] Pineda, L. A., Rodríguez, A., Fuentes, G., Rascon, C., Meza, I.V. (2015). **Concept and Functional Structure of a Service Robot**. *Int J Adv Robot Syst*, 12:6. doi: 10.5772/60026

foco de atención. En este nivel también se incluyen sistemas de representación del conocimiento que le permiten hacer inferencias conceptuales y sistemas de razonamiento deliberativo para hacer diagnósticos, tomar decisiones y planear sus acciones futuras. Este nivel de sistema es el que se asocia más directamente con la práctica de la robótica, o de manera más coloquial, con lo que hacen los científicos y tecnólogos "roboticistas" o los "roboteros". Asimismo, muchos dispositivos y algoritmos en este nivel se apoyan en técnicas de análisis de señales, reconocimiento de patrones, aprendizaje de máquina y teoría de control, entre otras, por lo que desde la perspectiva de la robótica estas disciplinas se subordinan al nivel algorítmico.

Finalmente el nivel de implementación corresponde a los programas de apoyo que se requieren para habilitar los diversos dispositivos y funciones, pero que no son tecnología robótica propiamente. Se pueden mencionar aquí los sistemas operativos y sistemas de comunicaciones que se utilizan para integrar las partes físicas o computacionales del robot como un todo. Esta tarea la realizan programadores de sistemas como apoyo a los roboticistas.

Todos los robot se pueden conceptualizar en términos de estos tres niveles de sistema pero es frecuente que sólo el nivel de dispositivos y algoritmos se considere explícitamente, en particular en el diseño y construcción de robots simples. Sin embargo, cuando el robot está constituido por varios dispositivos y requiere de varios algoritmos complejos, que tienen que operar de manera coordinada, es indispensable distinguir los tres niveles de manera explícita y abordarlos con metodologías específicas y grupos de investigación y desarrollo enfocados a los niveles funcional y de implementación, de manera adicional a los grupos enfocados a cada una de las especialidades en el nivel de dispositivos y algoritmos.

4.5. Aproximaciones al nivel funcional en México

En esta sección se presentan tres métodos para organizar y estructurar robots de servicio en el nivel funcional; estos enfoques son los adoptados y desarro-

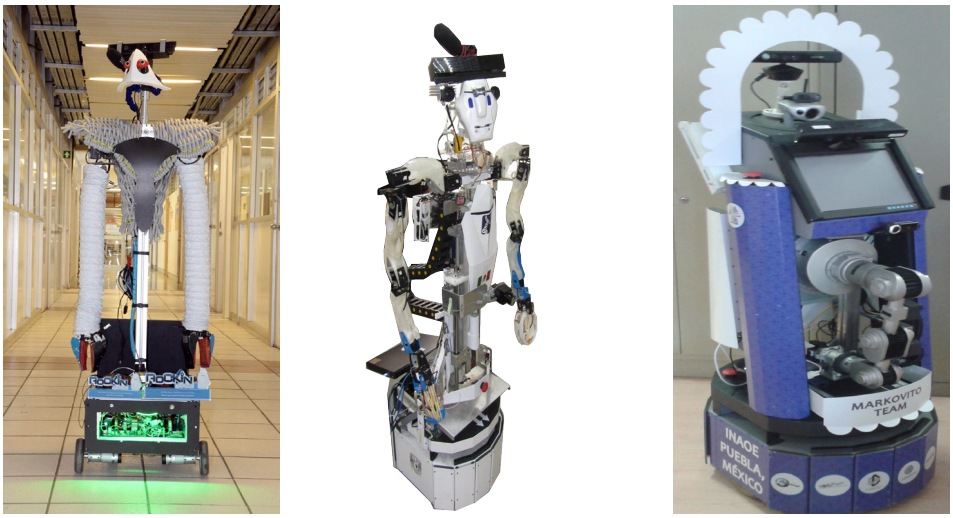
llados por los grupos mexicanos que trabajan con robots de servicio, como sigue: i) máquinas de estado, ii) lenguajes de especificación de tareas y iii) procesos de decisión de Markov. El primero se ha adoptado por el grupo Pumas de la Facultad de Ingeniería de la UNAM y se implementa en el robot Justina; el segundo se ha desarrollado por el Grupo Golem[8] del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) de la UNAM y se ha implementado en los robots Golem, Golem-II+ y Golem-III, y el tercero se ha adoptado y desarrollado en el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) y se ha implementado en los robots Markovito y Sabina.

4.5.1. Máquinas de Estados Finitos

La forma más directa para programar la conducta de un robot de servicio es mediante el uso de máquinas de estados. En éstas se tiene una representación del estado en el cual se encuentra el sistema y dadas ciertas condiciones de entradas y el estado presente se calcula el estado siguiente. La parte fundamental de las máquinas de estado es el Autómata de Estados Finitos (AEF) que especifica y guía la conducta. Por ejemplo, la conducta de un robot omnidireccional con dos motores que le permiten ir hacia adelante, hacia atrás y hacer giros de 45 grados hacia la derecha e izquierda, que cuente con dos sensores que le permiten detectar obstáculos (izquierdo y derecho) y capaz de realizar un conjunto finito de acciones, se puede especificar como sigue: si el robot no detecta ningún obstáculo avanza en la dirección en que está orientado; si detecta un obstáculo con su sensor izquierdo se mueve hacia atrás y gira hacía la derecha 45 grados; si detecta un obstáculo con su sensor derecho se mueve hacia atrás y gira hacía la izquierda 45 grados; si detecta un obstáculo con los dos sensores entonces se mueve hacia atrás, gira dos veces hacia la izquierda, avanza hacia adelante y finalmente gira hacia la derecha dos veces. Las entradas y salidas del autómata se pueden asociar a diferentes dispositivos

[8] http://golem.iimas.unam.mx/

y algoritmos, como redes neuronales implementadas en hardware (FPGAs)[9] para interpretar una señal de entrada o a campos de potenciales, para decidir una acción de navegación que se ejecuta cuando el control de estados se mueve de un estado al siguiente. Estas conductas pueden ser externas, como moverse hacia atrás o girar, o internas, como consultar a una base de datos de conocimiento. En el caso de que la interpretación y la conducta sean concretas el robot será "reactivo" y en la medida en que sus conductas internas sean más ricas tendrá una orientación más representacional y deliberativa. El robot Justina, el cual se ilustra en la Figura 4.2 (a), es una realización de este modelo conceptual en el nivel funcional.



(a) (b) (c)

Figura 4.2. Robots de servicio desarrollados en México: (a) Justina (FI-UNAM), (b) Golem-III (IIMAS-UNAM), (c) Sabina (INAOE).

[9] Savage, J., Cruz, J., Matamoros, M., Rosenblueth, D.A., Muñoz, S., Negrete, M (2016). **Configurable Mobile Robot Behaviors Implemented on FPGA Based Architectures**. En *16th IEEE Int. Conf. on Autonomous Robot Systems and Competitions*, pp. 317-322.

4.5.2. Lenguajes de especificación e interpretación de tareas

El modelo basado en autómatas de estado finito es limitado cuando la complejidad de la tarea y la comunicación aumenta, por lo que es útil incrementar el nivel de abstracción para especificar y ejecutar tareas complejas. Esto se puede lograr substituyendo la noción de "estado" por la noción más estructurada de "situación". La diferencia es que mientras un estado tiene un contenido nulo de información, excepto por las etiquetas asociadas a sus transiciones y sus siguientes estados, una situación es rica informacionalmente, como en los cuadros de historietas en las caricaturas, donde hay uno o más personajes en el entorno o contexto. Una situación se puede caracterizar computacionalmente mediante la especificación de las expectativas de quien juega el papel del *yo*, en este caso el robot, las cuales corresponden con sus creencias acerca de los eventos naturales que pueden ocurrir en el mundo y a las creencias e intenciones que adscribe a sus interlocutores. La situación contiene también las acciones intencionales que el robot debe realizar cuando una de las expectativas se cumple. En este modelo la acción del robot se concibe como un tránsito continuo entre situaciones, como cuando se lee un libro de historietas, que corresponde con las tareas que el robot realiza en el mundo.

Por ejemplo, la conducta de un mesero se puede caracterizar de manera simplificada por un conjunto de situaciones, como estar en la puerta del restaurante esperando al cliente; llevar al cliente a su mesa; darle la carta y tomarle la orden; servirle; vigilar que todo esté en orden; llevarle la cuenta; cobrarle, llevarle el cambio y despedirle. En cada situación el mesero tiene un conjunto, de expectativas normalmente pequeño, así como la capacidad de realizar de manera inmediata la acción que se requiere cuando una de éstas se cumple. La noción de situación es una abstracción con una extensión temporal y espacial relativamente amplia y de ahí su utilidad: el mesero puede estar toda una hora moviéndose nerviosamente en el vestíbulo del restaurant en la situación "esperando al cliente" y sólo pasará a la situación de "llevando cliente a su mesa" cuando llegue un cliente. De hecho el agente cambia de situación justamente

cuando cambian sus expectativas. Por supuesto, hay situaciones complejas como "ver si todo está en orden" pero éstas se pueden partir a su vez en situaciones más simples, hasta llegar a las situaciones en que las expectativas correspondan con actos de interpretación concretos y las acciones se realicen directamente, como moverse o proferir una orden. Por estas razones, tareas relativamente complejas, como la conducta de un mesero, de un asistente en un supermercado o de un recepcionista en un hotel, se pueden modelar como una gráfica prototípica de situaciones, que a su vez se extiende o "desdobla" como una gráfica concreta cada vez que el robot lleva a cabo una tarea.

En este modelo las expectativas y las acciones guían a los dispositivos percepción y acción; sin embargo, las acciones pueden ser también internas o "mentales" y pueden consistir en hacer inferencias conceptuales o invocar a mecanismos deliberativos, como los sistemas de diagnóstico, toma de decisiones y planeación, que se utilizan de manera oportunista según los requerimientos del ciclo de la comunicación.

La pregunta obligada en este punto es qué pasa si no se cumple ninguna expectativa y/o sucede un evento inesperado, que además sea relevante. En este punto el robot pasa de estar situado a no estarlo y no sabe qué hacer. No estar situado es también perder momentáneamente la capacidad de interpretar y actuar en relación a la tarea o perder el contexto. Por lo mismo, cuando no se cumple ninguna expectativa u ocurre un evento inesperado se activa un modelo de diálogo cuyo propósito es simplemente volver a situarse, o alternativamente, invocar un ciclo de razonamiento para saber qué pasó y actuar de manera coherente para volverse a situar y realizar sus objetivos. Un robot que adopta este modelo conceptual es un robot situado.

Para implementar este modelo conceptual se pueden emplear lenguajes de programación orientados a la definición de tareas. Por ejemplo, el Grupo Golem desarrolló el lenguaje de programación de tareas robóticas SitLog.[10]

[10] Pineda, L. A., Salinas, L., Meza, I.V., Rascon, C., Fuentes, G. (2013). **SitLog: A Programming Language for Service Robot Tasks**. *Int J Adv Robot Syst*, 10:358. doi: 10.5772/56906

Este lenguaje ofrece dos tipos de datos abstractos para especificar e interpretar tareas robóticas: el tipo *situación* y el tipo *modelo de diálogo*. El primero permite declarar directamente las expectativas y acciones de una situación y el segundo especificar una gráfica de situaciones que corresponde con la estructura prototípica de la tarea y, de manera paralela, a la estructura de la comunicación entre el robot y otros agentes en el entorno. El intérprete de SitLog consiste del intérprete de una red de transición recursiva (RTR),[11] que extiende a los AEF con una estructura de pila o *stack*, y del intérprete de un lenguaje funcional para la especificación e interpretación de expectativas y acciones, donde ambos intérpretes funcionan de manera coordinada. El intérprete de SitLog es el componente central de la arquitectura IOCA[12] utilizada en la serie de robots Golem y, en particular, en el robot Golem-III, el cual se ilustra en la Figura 4.2 (b). El intérprete de SitLog utiliza a su vez servicios de una base de conocimiento para llevar a cabo inferencias conceptuales[13] para explorar la base de datos de conocimiento, así como inferencias deliberativas de manera oportunista.[14]

4.5.3. Procesos de Decisión de Markov

A medida que los robots de servicio interactúan en ambientes más variables y complejos se tienen que enfrentar a la incertidumbre tanto en su percepción del ambiente como en el resultado de sus acciones. Una forma de considerar y lidiar con dicha incertidumbre es a través de los procesos de decisión de Markov (MDPs, por sus siglas en inglés). Un MDP se puede ver como una extensión de las máquinas de estado que considera que las transiciones de un estado a otro, dada cierta acción, son probabilistas. Por ejemplo, si el robot realiza cierto movimiento como desplazarse 10 metros en línea recta, lo más

[11] https://en.wikipedia.org/wiki/Recursive_transition_network

[12] Pineda, L. A., Meza, I., Avilés, H., Gershenson, C., Rascón, C., Alvarado, M. and Salinas, L. (2011). **IOCA: Interaction-Oriented Cognitive Architecture**. *Research in Computing Science,* 54: 273–284.

[13] Ver video en: http://golem.iimas.unam.mx/lightkb/

[14] Ver video en: http://golem.iimas.unam.mx/opportunistic_inference

probable es que no avance exactamente 10 metros dado que se puede patinar, haber imperfecciones en el piso o imprecisión en sus motores, etc. Los MDPs consideran la incertidumbre en el resultado de las acciones del robot; si también existe incertidumbre sobre el estado actual del robot (que tiene que ver con su percepción del ambiente) existe otro modelo llamado MDP parcialmente observable (POMDP). Mediante MDPs y POMDPs se pueden modelar los estados y acciones de un robot para realizar cierta tarea de forma que su comportamiento sea más robusto.

Los MDPs se pueden utilizar para representar y resolver problemas robóticos a diferentes niveles, tanto muy básicos como para desplazarse en su ambiente, como de alto nivel para coordinar sus diferentes capacidades y resolver tareas complejas.[15] En esta sección nos enfocamos a este último aspecto. Para realizar una tarea compleja un robot requiere utilizar diferentes capacidades en el nivel de dispositivos y algoritmos. Se requiere además que dichas capacidades se ejecuten de una forma coordinada para realizar una tarea, de preferencia en forma óptima (menor tiempo, mayor efectividad, etc.).

Mediante MDPs (o POMDPs) se pueden coordinar las diferentes capacidades básicas de un robot para realizar tareas complejas bajo incertidumbre. Para ello se define un *modelo* que especifica:

1. Un conjunto de estados *S* que define la situación actual del robot a alto nivel; por ejemplo su ubicación, la meta, lo que conoce del ambiente, los comandos de sus usuarios, etcétera.

2. Un conjunto de acciones *A*, que establece las posibles decisiones que puede realizar el robot, como localizarse, desplazarse de un lugar a otro, percibir el ambiente con sus cámaras u otros sensores, preguntar algo, etcétera.

[15] Elinas, P., Sucar, L.E., Reyes, A., Hoey, J. (2004). **A decision theoretic approach for task coordination in social robots**. En *IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, pp. 679–684.

3. Una función de transición $P(S_i, A, S_j)$ que define las probabilidades de que el robot pase a un nuevo estado S_j dado que realizó cierta acción A y se encontraba en cierto estado S_i. Estas probabilidades se pueden establecer en forma aproximada por una persona (subjetivas) o se pueden aprender a través de interactuar con el ambiente mediante algoritmos de aprendizaje por refuerzo.

4. Una función de recompensa que establece que tan deseable es cada estado (o estado-acción) de acuerdo a los objetivos de la tarea. Normalmente se establece una recompensa alta para el estado meta de la tarea; por ejemplo, el que el robot entregue a la persona cierto objeto, si la tarea es que el robot vaya y busque dicho objeto, lo tome y se lo lleve a una persona.

Una vez establecido el modelo de una tarea mediante un MDP existen algoritmos que permiten establecer una *política* o plan general para que el robot realice la tarea de forma óptima, es decir que maximice la utilidad esperada, que normalmente es la suma de las recompensas que recibe de acuerdo a la función establecida previamente. La política establece para cada estado cuál es la mejor acción que debe hacer el robot de forma que se lleve a cabo la tarea de la mejor manera, incluso habiendo incertidumbre en los resultados de sus acciones.

Este esquema permite que se pueda hacer un desarrollo modular y eficiente de robots de servicio para realizar diferentes tareas. Si se cuenta con un conjunto de capacidades básicas, desarrollar nuevas tareas implica simplemente cambiar el modelo del MDP y resolverlo para tener el plan correspondiente. Una analogía sería el del director de una orquesta que con los mismos músicos e instrumentos puede interpretar diferentes obras cambiando la partitura.

Existen diversas extensiones al esquema básico descrito anteriormente. Una es considerar incertidumbre en el estado actual del robot de forma que

se modela como un POMDP. Esto hace más compleja la solución ya que hay que considerar la probabilidad de cada estado (estado de creencia), lo que en principio implica un número infinito de estados; la solución exacta es sólo posible para problemas muy pequeños pero se han desarrollado soluciones aproximadas que permiten resolver casos más complejos como los de robótica. Otra extensión es el considerar que el robot puede realizar varias acciones a la vez, como desplazarse, comunicarse y reconocer personas. Para esto se han desarrollado MDPs concurrentes que toman en cuenta las restricciones para evitar conflictos entre los diversos tipos de acciones.[16]

El robot Markovito,[17] el cual se ilustra en la Figura 4.2 (c), utiliza MDPs para representar y realizar diferentes tareas, entre las cuales están: i) llevar mensajes y objetos entre personas; ii) buscar un objeto en un ambiente doméstico; iii) desplazarse a diferentes cuartos en una casa; iv) recibir y reconocer a diferentes personas, entre otras.

4.6. Nivel de dispositivos y algoritmos

Este nivel se organiza en relación a las modalidades principales de la percepción, como la visión y el lenguaje así como de la acción motora incluyendo la navegación y manipulación. Cada una de estas facultades se habilita mediante sensores y actuadores físicos en conjunto con algoritmos especializados que interpretan la señal proveniente del entorno para construir una representación del mundo. Es importante recalcar que el robot no mantiene representaciones del mundo directamente sino de las interpretaciones del mundo que es capaz de crear con su equipamiento sensorial y sus procesos perceptuales. A continuación se describen las facultades robóticas principales que se han estudiado por la comunidad mexicana en i) percepción y acción motora, ii) audición y lenguaje y iii) representación del conocimiento y razonamiento.

[16] Corona, E., Morales, E., Sucar, L.E. (2009). **Executing Concurrent Actions with Multiple Markov Decision Processes**. En *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. IEEE Press, pp. 82–89.

[17] http://ccc.inaoep.mx/~markovito/

4.6.1. Percepción y acción motora

La percepción robótica involucra dos tareas principales: i) el sensado de una señal proveniente del mundo mediante algún tipo de dispositivo, y ii) asignar una interpretación a dicha señal de manera coherente con el contexto o situación en que se hace la observación. Las señales pueden ser de cualquier modalidad de la percepción. Normalmente se presentan al sistema de cómputo como arreglos o matrices de datos, donde cada celda contiene un número que codifica a una propiedad de la señal sensada; por ejemplo, en el caso de las cámaras, la escena fotografiada se registra en una matriz de pixeles, donde cada celda contiene un número que representa el color del punto correspondiente en el espacio; a partir de estas informaciones se inicia el proceso de interpretación.[18] Entre los tipos de sensores más comúnmente utilizados en robótica se encuentran los siguientes:[19]

- Sonares: emiten una señal sonora y a partir del tiempo de ida y vuelta se puede estimar la distancia a los obstáculos (análogo a cómo los murciélagos detectan obstáculos).
- Láser: emiten rayos láser y también a partir del tiempo de ida y vuelta (o la fase) se estima la distancia a los objetos; son más precisos que los sonares.
- Cámaras:[20] obtienen imágenes/videos del entorno que pueden utilizarse para múltiples propósitos.
- Cámaras de profundidad: como el Kinect, no sólo obtienen la información de color de cada pixel de la imagen sino también un estimado de su distancia.

[18] Ver capítulo 7.

[19] Estos tipos de dispositivos de alta tecnología se desarrollan en México sólo de manera muy limitada; por lo mismo, la investigación en robótica en nuestro país se hace comúnmente con componentes básicos disponibles en el mercado internacional.

[20] Dado su bajo costo, tamaño y versatilidad, las cámaras y las cámaras de profundidad se han vuelto muy populares recientemente. En muchos casos se combinan diferentes sensores, ya sea del mismo o diferente tipo, mediante lo que se conoce como *fusión sensorial*.

- Micrófonos: permiten percibir los sonidos en el ambiente, en particular la voz humana para la interacción con las personas.
- Narices electrónicas: recientemente se han desarrollado sensores olfativos que permiten detectar ciertos compuestos como el alcohol.[21]

Por su parte, la acción motora consiste en moverse en el entorno, caminar o tomar objetos con las manos, entre muchas otras formas. Estos procesos son inversos con respecto a la percepción ya que su entrada es una señal producida por un proceso computacional y la salida es la acción mecánica producida por un servo-motor.[22]

No todas las tareas robóticas requieren que la escena se interprete completamente y es común que la información proporcionada por el sensor sólo permita una interpretación muy limitada; sin embargo, ésta puede ser suficiente para producir la conducta motora. Por ejemplo, para evadir objetos durante la navegación es suficiente detectar que hay un objeto enfrente para que el robot cambie de trayectoria sin tener que saber cuál es el objeto particular que se evade o a qué clase pertenece. Conductas de este tipo son reactivas en oposición a aquellas que involucran un proceso de interpretación profundo, que son deliberativas. Dentro de la robótica clásica, la mayoría de las conductas son reactivas e involucran al menos un sensor y un dispositivo de acción, por lo que la percepción y la acción motora se tienen que abordar juntas. Algunas de las tareas paradigmáticas de carácter reactivo son la navegación, el seguimiento o *tracking* y el movimiento supervisado por la visión o control visual. Por su parte, si el proceso incluye interpretaciones e inferencias deliberativas es posible diferenciar más claramente a las tareas de la percepción de las acciones motoras y estudiar ambos problemas por separado. Algunos procesos con un carácter más perceptual son la detección y el reconocimiento de personas, el reconocimiento de objetos y el análisis de escenas.

[21] http://tecreview.itesm.mx/esta-nariz-robotica-podria-salvarte-la-vida/

[22] Los servo-motores de calidad se producen también por corporativos internacionales y en México se usan como componentes básicos.

También hay tareas que involucran la construcción de representaciones pero que a su vez están estrechamente vinculadas a conductas motoras, por lo que la percepción y la conducta motora se abordan también en conjunto. Un ejemplo es la construcción de mapas al tiempo que se explora el espacio, aunada a la localización del robot en el mismo mapa; otro es el reconocimiento visual y la manipulación de objetos. Para enfrentar estos retos se han desarrollado una diversidad de algoritmos, algunos más genéricos, por ejemplo en el campo de la visión computacional y otros más específicos a la robótica. A continuación se mencionan algunos desarrollos de la comunidad mexicana en i) mapeo, auto-localización y navegación, ii) seguimiento, iii) reconocimiento y manipulación de objetos, iv) detección y reconocimiento de personas y v) monitoreo y vigilancia.

4.6.1.1. Mapeo, auto-localización y navegación

Uno de los problemas fundamentales para los robots de servicio es la movilidad, lo que incluye la construcción de mapas, la localización y la navegación:

1. Construcción de mapas: los robots móviles requieren una representación de su entorno o mapa para moverse en el entorno espacial; en general es conveniente que ellos mismos lo construyan por medio de sus capacidades sensoriales y algorítmicas. Existen diferentes tipos de mapas, incluyendo los métricos (rejillas de ocupación espacial, representaciones geométricas, etc., en 2 y 3 dimensiones), los topológicos, los semánticos[23] y los de configuraciones (que tienen tantas dimensiones como los grados de libertad del robot).

2. Localización: este problema consiste en que el robot determine su ubicación en el mapa de manera dinámica. La localización tiene dos variantes principales: i) global, en la cual el robot se tiene que localizar de manera absoluta en relación al mapa (por ejemplo cuando no sabe

[23] Ver sección 4.6.3.

donde está al iniciar una tarea); ii) local o relativo a una posición de referencia durante su desplazamiento. Para localizarse, el robot puede utilizar información distintiva del ambiente, como marcas (naturales o artificiales), objetos específicos o puntos característicos (esquinas, SIFT,[24] etc.). Es importante destacar que el robot tiene que actualizar su posición continuamente ya que si sólo utiliza el registro de su propio movimiento, lo que se conoce como odometría, el error de localización se incrementa continuamente hasta que eventualmente el robot se pierde.

3. SLAM: frecuentemente la construcción del mapa y el cálculo de la localización o ubicación se tienen que realizar de manera simultánea; este proceso coordinado se conoce como *mapeo y localización simultáneos* (SLAM, por sus siglas en inglés).

4. Navegación: Los robots requieren construir un plan o definir una trayectoria para desplazarse de un lugar a otro; esto se conoce como planeación de trayectorias o de movimiento (*motion planning*). Para esta tarea se han desarrollado diversos algoritmos, incluyendo determinísticos y probabilísticos. Una vez establecido, el plan debe ejecutarse, lo que involucra el control de los actuadores del robot para poder seguir la trayectoria deseada, como las llantas en robots móviles, las articulaciones en brazos robóticos y las piernas en robots humanoides. En ambientes variables deben detectarse posibles cambios en el mundo u obstáculos dinámicos y evitarlos o ajustar el plan.

Para resolver estas tareas en el INAOE y el CIMAT se han desarrollado algoritmos probabilísticos que utilizan sensores láser para obtener una representación del espacio libre (en 2-D) basados en rejillas de ocupación espacial, donde cada celda contiene un número real que representa la probabilidad

[24] SIFT (*Shift Invariant Feature Transform*) es uno de múltiples esquemas para identificar puntos característicos salientes y estables en imágenes: https://en.wikipedia.org/wiki/Scale-invariant_feature_transform.

de que esté libre u ocupada; con base al mapa resultante se han desarrollado técnicas de navegación robustas ante las limitaciones perceptuales del robot.[25] Asimismo, dicho mapa se utiliza para que el robot se localice mediante la detección de marcas naturales en el ambiente, por ejemplo esquinas, puertas y paredes.[26] También se ha trabajado en técnicas de navegación mediante el aprendizaje de reglas tele-reactivas a partir de ejemplos provistos por una persona.[27]

Se ha trabajado además en la navegación de robots con ruedas basada en un conjunto de imágenes objetivo[28] las cuales se capturan previamente y constituyen una representación del ambiente que se denomina "memoria visual", la cual le permite al robot navegar posteriormente de forma autónoma. Se tienen ya resultados para robots humanoides. Otro reto es hacer mapas utilizando una sola cámara, lo que se conoce como SLAM monocular, ya que no se tiene información de profundidad; sin embargo, ésta se puede estimar mediante conocimiento previo del mundo.[29] Un reto más, abordado en el CINVESTAV, IPN y el ITAM es la navegación para coches autónomos que requiere la integración a alta velocidad de componentes de localización basada en entrada visual y de lidiar con planificación de tareas y de movimientos.

---

[25] Romero, L., Morales, E., Sucar, L.E. (2001). **A robust exploration and navigation approach for indoor mobile robots merging local and global strategies**. En *International Conference on Robotics and Automation (ICRA)*, IEEE Press, pp. 3092–3097.

[26] Romero, L., Morales, E., Sucar, L.E. (2001). **A Hybrid Approach to Solve the Global Localization Problem for Indoor Mobile Robots Considering Sensor's Perceptual Limitation's**. En *Int. Joint Conf. on Artificial Intelligence (IJCAI),* AAAI Press, pp. 1411–1416.

[27] Vargas, B., Morales, E. (2009). **Learning Navigation Teleo-Reactive Programs using Behavioural Cloning**. En *Proc. IEEE International Conference on Mechatronics (ICM).*

[28] Becerra, H. M. , Sagues, C., Mezouar, Y., Hayet, J. B. (2014). **Visual navigation of wheeled mobile robots using direct feedback of a geometric constraint**. *Autonomous Robots*, 37(2): 137–156.

[29] Mota-Gutierrez, S. A., Hayet, J. B., Ruiz-Correa, S., Hasimoto-Beltran, R., Zubieta-Rico, C. E. (2015). **Learning depth from appearance for fast one-shot 3-D map initialization in VSLAM systems**. En *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA),* pp. 2291–2296.

Un problema relacionado es la caminata estable de robots humanoides. Se han hecho desarrollos para simultáneamente realizar el control visual y controlar la caminata dinámica de un robot humanoide.[30] Para este efecto se utiliza un generador de patrón de movimiento basado en control predictivo (MPC) que adapta los pasos de la caminata y las trayectorias del centro de masa del robot para seguir perfiles de velocidad provistos por el usuario. La contribución de este trabajo es de reformular el problema de MPC al considerar información de retroalimentación visual, en lugar de usar la velocidad de referencia. Un problema relacionado es la regulación de postura por medio de visión para robots humanoides.[31] Para este efecto se utiliza un esquema de control capaz de lograr la convergencia de la tarea visual en un tiempo predefinido. El esquema es un control jerárquico basado en tareas que puede manejar al mismo tiempo la tarea visual y una tarea de evasión de obstáculos, los cuales se detectan con una cámara de profundidad. Genera velocidades continuas para el robot y considera las capacidades de caminata omnidireccional de los robots humanoides.

4.6.1.2. Seguimiento

El seguimiento de personas en ambientes dinámicos donde pueden transitar libremente vehículos y personas, como en las pruebas de navegación de la competencia RoboCup, es una tarea sumamente compleja. Este problema se ha abordado por los robots Markovito, Justina y Golem con diferentes estrategias y algoritmos.

El método de seguimiento de personas del equipo Markovito del INAOE combina información del esqueleto obtenido con el Kinect con información

[30] Garcia, M., Stasse, O., Hayet, J. B., Dune, C., Esteves, C., Laumond, J. P. (2015). **Vision-guided motion primitives for humanoid reactive walking: Decoupled versus coupled approaches**. *International Journal of Robotics Research,* 34(4-5): 402–419.

[31] Delfin, J., Becerra, H. M., Arechavaleta, G. (2016). **Visual servo walking control for humanoids with finite-time convergence and smooth robot velocities**. *International Journal of Control,* 89(7): 1342–1358.

de color de la cámara del robot. Inicialmente se detecta a la persona mediante el algoritmo de detección de esqueletos del Kinect y luego se aprende un modelo de color de la ropa de dicha persona. Para el seguimiento se integra la información de las observaciones (basadas en color) con la estimación de movimiento mediante un filtro de Kalman.

La estrategia que se desarrolla actualmente por el Grupo Golem se basa en el uso combinado de un láser que localiza el torso de la persona a seguir y el esqueleto que se obtiene del Kinect. El método utiliza adicionalmente el parche de colores del objeto a seguir, como la camisa o el pantalón del sujeto, para distinguirlo de otras personas u objetos, fijos o móviles, que se encuentren en la trayectoria de seguimiento.

Un problema relacionado que se considera clásico en robótica móvil desde el punto de vista de los sistemas de control es el de persecución–evasión, donde un robot debe seguir y alcanzar a un blanco móvil. Este problema se ha analizado en el CIMAT desde el punto de vista teórico para la captura de un evasor omnidireccional usando un robot de manejo diferencial en un ambiente sin obstáculos.[32] El método propuesto utiliza técnicas de control óptimo para obtener representaciones de las primitivas de movimiento y estrategias (en equilibrio de Nash) de tiempo mínimo para seguidor y evasor. También aborda el problema de decisión del juego y las condiciones que definen al ganador. Esta metodología se ha aplicado además al problema de persecución–evasión en un ambiente con obstáculos,[33] con especial atención al problema combinatorio que surge para mantener visibilidad de un evasor que visita varias ubicaciones.

---

[32] Ruiz, U., Murrieta-Cid, R. Marroquin, J. L. (2013). **Time-optimal motion strategies for capturing an omnidirectional evader using a differential drive robot**. *IEEE Transactions on Robotics*, 29(5): 1180–1196.

[33] Becerra, I., Murrieta-Cid, R., Monroy, R., Hutchinson, S., Laumond, J. P. (2016). **Maintaining strong mutual visibility of an evader moving over the reduced visibility graph**. *Autonomous Robots*, 40(2): 395–423.

4.6.1.3. Reconocimiento y manipulación de objetos

Los robots de servicio deben ser capaces de tomar y llevar objetos de un sitio a otro, lo que se conoce como el problema de manipulación. Para ello los robots móviles y humanoides incorporan brazos robóticos con pinzas o manos para sujetar diferentes tipos de objetos. La manipulación presupone a su vez que el robot debe ver los objetos para lo cual es necesario identificarlos y determinar su posición y orientación espacial. Por esta razón los problemas de reconocimiento y manipulación se relacionan de manera muy estrecha. Desde el punto de vista computacional el problema de manipulación involucra varios subproblemas:

- Reconocimiento y ubicación del objeto: el primer paso para manipular un objeto es reconocerlo y ubicar su posición en el ambiente (coordenadas 3-D).

- Modelado: en el caso de objetos complejos se requiere conocer su forma; es decir, obtener un modelo tridimensional que permita identificar posibles puntos de agarre. Para ello se obtienen diferentes vistas del objeto con algún sensor de profundidad, como el Kinect, que se integran en una representación 3-D. El determinar las mejores posiciones del sensor para obtener dichas vistas es un problema de planificación que se conoce como "siguiente mejor vista" (*next best view*).

- Determinar puntos de sujeción (*grasping*): en base al modelo 3-D del objeto se deben establecer puntos de sujeción estables (normalmente dos o tres, dependiendo del tipo de pinza o mano); es decir, que permitan sujetar y levantar al objeto en forma segura.

- Planeación/ejecución de movimientos: una vez que se establece la posición y punto de agarre del objeto es necesario llevar al efector final del brazo a las coordenadas deseadas, para lo cual se establece un plan de los movimientos del brazo/base móvil que luego se ejecuta,

normalmente con cierta retroalimentación sensorial que asegure tomar al objeto.

Enseguida resumimos algunas aportaciones de grupos en nuestro país.

El Grupo Golem del IIMAS, UNAM, aborda la tarea de reconocimiento de objetos mediante el algoritmo MOPED[34] el cual requiere crear un modelo de antemano para cada objeto conocido. Los modelos se forman a partir de un conjunto de fotografías tomadas alrededor del objeto desde diferentes distancias y perspectivas para configurar una envolvente. Las fotografías se procesan mediante el algoritmo SIFT.[35] Con estos descriptores se forma una nube de puntos alrededor del objeto, la cual se calibra con respecto a un punto externo de observación. Esta calibración permite determinar cuál será el tamaño del objeto con relación a un nuevo punto de observación. Para efectos de reconocimiento, la vista del objeto desde el punto de observación se procesa también mediante el algoritmo SIFT y sus descriptores se alinean parcialmente con respecto a alguna vista arbitraria del modelo. Si este proceso es exitoso el objeto se reconoce en conjunto con su posición y orientación o *pose* en el espacio 3-D en relación al punto de observación, que corresponde con la cámara del robot.

Para efectos de la manipulación el robot alinea su brazo a la misma altura que el objeto y, dado que conoce el tamaño de su brazo y la distancia al objeto, le es posible dirigir su brazo hacia la posición del objeto por un método de triangulación directo. Con esta metodología el equipo Golem obtuvo el 2º lugar con el robot Golem-II+ en el reto técnico de la competencia RoboCup México 2012 y el 3er. lugar en la prueba de visión y manipulación en el RoboCup Leipzig 2016 con el robot Golem-III. En esta misma competencia Golem-III se desempeñó como un asistente de supermercado en la prueba

[34] Collet, A., Martinez, M., Srinivasa, S.S. (2011). **The MOPED framework: Object Recognition and Pose Estimationfor Manipulation**. *The International Journal of Robotics Research,* 30: 1284–1306.

[35] Ver nota 23.

del *Open Challenge* en la que tomó varios objetos exitosamente y obtuvo el 2º lugar.

El algoritmo MOPED es muy preciso (tiene muy pocos falsos positivos) pero sólo funciona con objetos ricos en textura, además de que no puede reconocer objetos para los que no se cuente con modelos, por lo que su cobertura es pobre y puede tener muchos falsos negativos (es decir objetos que no ve). Para superar esta limitación el grupo Golem utiliza una segunda estrategia basada en la nube de puntos, que corresponden a objetos en la escena, que se obtiene a través del sensor de distancia del Kinect. Las nubes se segmentan en la imagen y se asocian a una máscara de color, la cual se utiliza como un identificador único para objetos de baja textura. Con esta nueva funcionalidad, el robot Golem-III pondera los objetos que reconoce y otorga un mayor peso a los objetos reconocidos con MOPED, un peso menor a los objetos modelados por nube de puntos y máscara de color, y un peso aún más bajo a los objetos que reconoce por la nube de puntos, pero cuya máscara de color no corresponde con ninguno de los modelos conocidos. Estos últimos objetos se catalogan como "no conocidos". Con esta estrategia en cascada se espera incrementar significativamente la cobertura visual y dar al robot Golem-III la capacidad de manipular una gama amplia de objetos tanto conocidos como desconocidos.

Sin embargo, es difícil que un robot de servicio cuente con modelos previos de todos los posibles objetos en el mundo, por lo que en el INAOE se ha desarrollado un método de reconocimiento de objetos que aprende nuevos modelos a partir de ejemplos de imágenes que obtiene de Internet.[36] Para tener más ejemplos se generan nuevas imágenes mediante transformaciones de las imágenes originales y luego se aprende un clasificador del nuevo objeto a partir de los ejemplos, que se puede usar para reconocerlos en el ambiente.

[36] Navarrete, D., Morales, E., Sucar, L. E. (2012). **Unsupervised learning of visual object recognition models**. En *IBERAMIA - LNAI 7637*, Springer-Verlag, pp. 511–520.

En el INAOE se desarrolló también un algoritmo para resolver el problema de planificación de vistas para modelado tridimensional de objetos; éste toma en cuenta la incertidumbre en la posición final del sensor, el cual está montado en el eslabón final de un brazo sobre un robot móvil.[37] El desplazamiento de la base móvil introduce errores en la posición deseada del sensor, lo que puede ocasionar que no se obtenga una vista adecuada. Para compensar esto y tener posiciones robustas ante el error de posicionamiento, el planificador ejecuta varias veces el algoritmo de planeación basado en RRT[38] (*Rapid exploring Random Tree*) simulando errores y selecciona aquel plan que da un valor esperado óptimo. Para ello se define una función de utilidad que establece un compromiso entre varios aspectos deseables: i) observar superficies no vistas del objeto, ii) garantizar un traslape mínimo con las vistas anteriores (para poder juntar las vistas en un modelo 3-D), y iii) minimizar la distancia que el robot debe moverse.

En el campo del control visual de robots móviles se han propuesto esquemas genéricos de control robusto que usan restricciones geométricas y relacionan múltiples vistas de una escena. Por ejemplo, el problema de regulación de postura se ha abordado mediante un esquema de control visual basado en imágenes (*image-based visual servoing*) que usa restricciones geométricas para conducir a un robot de manejo diferencial a una posición y orientación deseadas.[39] Un problema relevante en este contexto es el seguimiento de objetos, por ejemplo, para tomar un objeto con la mano. En el CIMAT se ha investigado también el problema de confirmar la identidad de un objeto candidato; para este efecto se propuso un método que mezcla la localización del robot relativa al objeto candidato y la confirmación de que dicho objeto es el

[37] Vasquez-Gomez, J.I., Sucar, L.E., Murrieta-Cid, R. (2016). **View/state planning for three-dimensional object reconstruction under uncertainty**. *Autonomous Robots*, 41 (1):89–109.

[38] https://en.wikipedia.org/wiki/Rapidly-exploring_random_tree

[39] Becerra, H. M., Hayet, J. B., Sagués, C. (2014). **A single visual-servo controller of mobile robots with super-twisting control**. *Robotics and Autonomous Systems*, 62 (11): 1623–1635.

blanco buscado.[40] El proceso de confirmación con esta meta dual se modela como un POMDP que se resuelve con Programación Dinámica.

Otro problema relevante es la búsqueda de objetos; por ejemplo si el usuario hace una petición (robot: "tráeme una manzana") en una casa es necesario combinar aspectos de percepción, planeación y navegación. Se desea que el robot encuentre el objeto en el menor tiempo posible, para lo cual debe estimar los lugares más probables (por ejemplo, una manzana en la cocina o el comedor) y establecer una estrategia de búsqueda que minimice el tiempo esperado. En el INAOE se desarrolló una estrategia que inicialmente estima la probabilidad de encontrar cierto objeto en los diferentes tipos de habitaciones; ésta se basa en estimar la correlación entre objeto-habitación mediante el uso de diversas bases de datos en Internet.[41] Combinando la probabilidad de las habitaciones, la distancia del robot a las habitaciones y el área de las habitaciones se establece una heurística que permite planear una ruta para explorar las habitaciones en el menor tiempo posible, la cual da resultados cercanos al óptimo.[42, 43] Finalmente, el robot busca el objeto dentro de cada habitación enfocándose en superficies planas donde se espera encontrarlo.

4.6.1.4. Reconocimiento de personas

El reconocimiento de personas forma parte de las tareas que un robot de servicio debe realizar. Este problema se puede descomponer en tres subpro-

[40] Becerra, I., Valentín-Coronado, L. M., Murrieta-Cid, R., Latombe, J. C. (2016). **Reliable confirmation of an object identity by a mobile robot: A mixed appearance/localization-driven motion approach**. *International Journal of Robotics Research*, 35(10): 1207–1233.

[41] Izquierdo-Córdova, R., Morales, E.F., Sucar, L.E. (2016). **Object Location Estimation in Domestic Environments through Internet Queries**. *RoboCup Symposium*, 2016

[42] Izquierdo-Cordova, R., Morales, E.F., Sucar, L.E., Murrieta-Cid, R. (2016). **Searching Objects in Known Environments: Empowering Simple Heuristic Strategies**. *IJCAI Workshop on Service Robots*.

[43] Garantizar el tiempo óptimo es un problema difícil de resolver eficientemente, lo que se conoce en computación como problemas "NP" (no existe algoritmo conocido que pueda resolverlo en un tiempo polinomial respecto al tamaño del problema).

blemas: detección, localización e identificación, cuyo objetivo es determinar cuántas personas hay, dónde están y quiénes son respectivamente.[44]

La detección está relacionada con la capacidad de clasificar cuáles objetos en el ambiente son personas y cuáles no. Una estrategia común es primero detectar caras en las imágenes;[45] para este efecto hay un algoritmo muy popular que utiliza técnicas de aprendizaje computacional para construir un ensamble de clasificadores[46] que permite determinar si un pixel forma parte de una cara o no. La localización, por su parte, consiste en determinar la ubicación exacta de la persona detectada. Por ejemplo, si la detección se realiza con Kinect es posible obtener las coordenadas 3-D de los puntos más relevantes de su cuerpo, incluyendo las manos, los codos, los hombros y la cabeza. Finalmente, la identificación consiste en determinar si la persona detectada está incluida en la base de datos del robot. Este último proceso se realiza principalmente mediante el uso de rasgos biométricos;[47] por ejemplo, mediante la identificación de puntos característicos en la zonas de los ojos-nariz-boca. El reconocimiento se hace más robusto al combinar varias imágenes mediante un esquema bayesiano,[48] que es la estrategia que usa el robot Markovito para reconocer personas.

En particular, el robot Golem-III realiza la detección de personas mediante el algoritmo de Viola-Jones para detectar rostros. También puede detectar la región completa de la cabeza y los hombros utilizando un algoritmo desarrollado por el grupo de investigación Golem, el cual se basa en la técni-

---

[44] Cielniak, G., Duckett, T. (2004). **People Recognition by Mobile Robots**. *Journal of Intelligent and Fuzzy Systems*, 15:21–27.

[45] Viola, P., Jones, M (2001). **Robust Real-time Face Detection**. *International Journal of Computer Vision*, 57(2):137–154.

[46] Ver capítulo 2.

[47] Jain, A.K., Ross, A., Prabhakar, S. (2004). **An Introduction to Biometric Recognition**. *IEEE Transactions on Circuits and Systems for Video Technology,* 14(1):4–20.

[48] Cruz, C., Sucar, L. E., Morales, E. (2008). **Real-Time Face Recognition for Human-Robot Interaction**. En *IEEE International Conference on Automatic Face and Gesture Recognition*.

ca de Histogramas de Gradientes Orientados.[49] El robot utiliza un Kinect 2 para poder detectar cuerpos y sus respectivos esqueletos, los cuales permiten clasificar la posición en la que se encuentra una persona –de pie, sentada o recostada en el piso– o determinar si está realizando un gesto –ya sea apuntar o alzar la mano para llamar la atención. Para la identificación de personas, el robot utiliza la técnica llamada Eigenrostros,[50] la cual es un enfoque biométrico que utiliza algoritmos de aprendizaje.

En años recientes, un tipo especial de rasgos llamados biométricos suaves[51] (como el color de la ropa) han atraído la atención de distintas investigaciones debido a su utilidad en escenarios no controlados. En particular, el robot Golem-III puede identificar a una persona a partir de su ropa con la información que provee el detector de esqueletos de Kinect 2 y sin necesidad de ver el rostro. Para ello, durante la fase de aprendizaje, el robot guarda múltiples vistas de la misma persona en distintas orientaciones y extrae parches de color que se etiquetan semánticamente de acuerdo a la parte del cuerpo en la que se encuentran.

4.6.1.5. Monitoreo y vigilancia

Los robots no sólo deben ser capaces de analizar imágenes estáticas sino también secuencias de imágenes o videos. En el CIMAT se desarrolló una metodología para aprender y utilizar modelos de movimiento probabilísticos de múltiples objetivos, incorporada a un esquema de rastreo visual por filtro de partículas en secuencias de video.[52] Dada una secuencias de video para

[49] Dalal, N., Triggs, B. (2005). **Histograms of oriented gradients for human detection**. *Computer Vision and Pattern Recognition*, 1:886-893.

[50] Turk, M., Pentland, A. (1991). **Eigenfaces for recognition**. *Journal of Cognitive Neuroscience*, 3(1):71–86.

[51] Jain, A. K., Dass, S. C., Nandakumar K. (2004). **Soft Biometric Traits for Personal Recognition Systems**. En *Proc. of Inter. Conference on Biometric Authentication*, pp. 731-738.

[52] Madrigal, F., Hayet J. B. (2015). **Goal-oriented visual tracking of pedestrians with motion priors in semi-crowded scenes**. En *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA),* pp. 720-725.

entrenamiento se extraen primero las posiciones (u "objetivos") donde los peatones entran/salen de la escena observada, o simplemente donde cambian de dirección con frecuencia. Posteriormente, se aprende un modelo de movimiento a partir de estadísticas para cada uno de estos objetivos. Finalmente, se utilizan los modelos de movimiento en un esquema de tipo IMM (Modelos Múltiples en Interacción) para el seguimiento y la estimación del objetivo de cada persona rastreada.

4.6.2. Audición y Lenguaje

Los robots de servicio requieren comunicarse a través del lenguaje, especialmente hablado. Para este efecto se deben incorporar sistemas de reconocimiento e interpretación del lenguaje así como de generación de lenguaje y síntesis de voz. Las contribuciones de la comunidad mexicana al reconocimiento de voz[53] se han utilizado directamente para habilitar el reconocimiento del español hablado a los robots producidos en el Proyecto Golem, especialmente a Golem-I y al sistema Golem en Universum. Asimismo, el lenguaje de programación SitLog se ha utilizado para crear los modelos de diálogo de los robots Golem-II+ y Golem-III. En estos robots, así como en Justina y Markovito, se ha utilizado también el sistema de reconocimiento de voz de Microsoft, especialmente para habilitar la comunicación en inglés en el contexto de las competencias RoboCup en la liga @Home,[54] en la que estos robots participan regularmente.

La producción del lenguaje, por su parte, tiene dos componentes principales: la generación de los contenidos y la síntesis de voz propiamente. Para el primero se han usado diferentes estrategias que van desde el uso de plantillas o "templates" hasta textos cuya generación está asociada a un proceso inferencial; para el segundo se utilizan sintetizadores con voces sintéticas disponibles en Internet, ya que no hay productos locales lo suficientemente robustos para estas tareas.

[53] Ver capítulo 3.

[54] http://www.robocupathome.org/

4.6.2.1. Audición robótica y análisis de imágenes acústicas

El reconocimiento de voz en robots de servicio presenta tres retos adicionales a lo descrito anteriormente: i) detección y selección de varias fuentes sonoras, por ejemplo varias personas que interactúan al mismo tiempo con el robot, ii) separación de estas fuentes sonoras, y iii) su clasificación.

El problema de la localización de fuentes sonoras se ha abordado con una configuración de hardware, con un arreglo de tres micrófonos dispuestos en un triángulo equilátero que presenta tres frentes a los sonidos generados en el entorno, de tal forma que la diferencia de fases entre cada par de micrófonos en cada frente permite establecer una hipótesis acerca de la posición de la fuente sonora. Asimismo, la posición de la fuente en relación a los tres frentes del arreglo debe ser lo más coherente posible, por lo que las hipótesis incoherentes se descartan directamente. Este arreglo, en conjunto con su algoritmo asociado, permite localizar más fuentes que el número de micrófonos utilizados, superando las limitantes de los sistemas tradicionales.[55] El único requisito es que las fuentes sonoras sean de voz humana ya que el algoritmo aprovecha que si dos o más personas hablan a la vez, el empalme de la voz en micro segmentos de tiempo (por ejemplo del orden de 100 milisegundos) es muy bajo. En los momentos de poco empalme se detecta a una persona y, a lo largo del tiempo, se agrupan las detecciones por cada fuente. Este sistema es muy liviano en requisitos de hardware y software, por lo que es apropiado para aplicaciones de robótica de servicio. Dicho desarrollo fue integrado en el robot de servicio Golem y se utiliza en varias aplicaciones como[56] recuperarse al perder visualmente a la persona que se sigue, tomar asistencia en una clase, jugar el juego Marco Polo (versión americana del juego "Gallinita Ciega")

[55] Rascon, C., Fuentes, G., Meza, I. (2015). **Lightweight multi-DOA tracking of mobile speech sources**. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(11). doi: 10.1186/s13636-015-0055-8.

[56] Meza, I., Rascón, C., Fuentes, G., Pineda, L. A. (2016). **On Indexicality, Direction of Arrival of Sound Sources, and Human-Robot Interaction.** *Journal of Robotics*, Article ID 3081048, 2016. doi:10.1155/2016/3081048.

y ser mesero en un restaurante. La aplicación de mesero[57] fue evaluada y se encontró que los usuarios sintieron que el robot Golem se comportaba como un mesero al ser llamado desde lejos. También se encontró una solución más natural al tomar la orden de varios comensales que hablan al mismo tiempo: cuando el robot detecta a más de una persona hablando, descarta la orden, pide que sólo una persona hable a la vez y, utilizando la información de dirección, encara a la persona que le toca hablar. Básicamente, el robot es el que impone el orden de la interacción.[58]

Una vez ubicadas las fuentes sonoras es necesario separar las voces de personas que estén hablando al mismo tiempo. Para este efecto es necesario separar las señales y alimentar la información de un sólo hablante a la vez al reconocedor de voz. Adicionalmente, si una de las fuentes es ruido (por ejemplo, tráfico en la calle, un abanico de aire, música, etc.) se filtra de manera natural. Más aún, el robot puede ser capaz de reconocer a dos personas que estén hablando al mismo tiempo. Actualmente, dicho reto se aborda en la UNAM por medio de técnicas de *Aprendizaje Profundo*.

El último paso de este proceso es clasificar las fuentes sonoras; si una fuente es una persona y la otra ruido se puede omitir alimentar la información del ruido al reconocedor y hacer más eficiente el reconocimiento de voz. Adicionalmente, si varias personas están hablando a la vez se puede identificar quién está hablando y, posiblemente, darle más prioridad al dueño del robot que a un tercero.

---

[57] Rascon, C., Meza, I., Fuentes, G., Salinas, L., Pineda, L. A. **Integration of the Multi-DOA Estimation Functionality to Human-Robot Interaction**. *Int J Adv Robot Syst*, 2015, **12**:8. doi: 10.5772/59993

[58] Esta aplicación se demostró como la prueba *Open Challenge* del equipo Golem en el *RoboCup Eindhoven 2013*, en Holanda, en la cual se obtuvo la puntuación más alta y se premió con el *Innovation Award* de dicha competencia.

El corpus DIMEx100[59] (descrito en el capítulo 3) se ha utilizado también para la evaluación de desarrollos de Audición Robótica en México y es la base del corpus de evaluación *Acoustic Interactions for Robot Audition* (AIRA)[60] que simula fuentes sonoras por medio de las mismas grabaciones de DIMEx100 reproducidas en localizaciones conocidas, con múltiples micrófonos con posiciones conocidas, otorgando transcripciones y señales originales limpias.

4.6.3. Representación del conocimiento y razonamiento

En robótica de servicio es muy importante contar con un sistema de representación de conocimiento y razonamiento. Entre los diversos aspectos que debe conocer el robot destaca la necesidad de contar con un mapa del entorno físico. Los principales tipos de mapas utilizados en robótica son:

1. *Mapas métricos*. Consisten en una representación geométrica del espacio en dos o tres dimensiones en la que se distinguen al menos las zonas libres (por donde el robot puede desplazarse) de las zonas ocupadas (obstáculos). Esto le permite al robot conocer su ubicación en el mundo y navegar. Existen diversas formas de representación incluyendo: i) mapas de celdas (voxels en 3-D) en el cual el espacio se divide en una rejilla de 2 ó 3 dimensiones a cuyas celdas se les asigna un valor de ocupado o libre (cuantificado por una probabilidad); ii) mapas que se representan con elementos geométricos básicos, como líneas, superficies, cilindros, etc., que al integrarse proveen un modelo aproximado del mundo; iii) mapas en base a puntos característicos que se extraen de las imágenes y cuyo objetivo principal es permitir al robot mantener su ubicación, sin incluir un modelado preciso de los objetos en el mundo.

[59] Pineda, L. A., Castellanos, H., Cuétara J., Galescu, L., Júarez, J., Llisterri, J., Pérez P., Villaseñor, L. (2010). **The Corpus DIMEx100: Transcription and Evaluation**. *Language Resources and Evaluation* 44:347–370. doi: 10.1007/s10579-009-9109-9

[60] Ver nota 15.

2. *Mapas topológicos*. El ambiente se representa en forma más abstracta en base a un grafo de conectividad, donde los nodos representan áreas o espacios (como cuartos en ambientes interiores) y los arcos denotan relaciones de adyacencia o conectividad (como pasillos y puertas). Los mapas topológicos le permiten al robot planear sus trayectorias sin necesidad de información precisa del mapa.

3. *Mapas de configuraciones*. Representan los movimientos posibles que puede realizar un robot. Una configuración es una abstracción del robot que representa en forma única su pose y tiene un parámetro por cada uno de sus grados de libertad. Por ejemplo, un robot móvil sin manipulador tiene usualmente tres, dos para desplazamiento y uno para orientación, y una configuración es una combinación única de estos tres parámetros. Estos mapas se representan como grafos cuyos nodos son configuraciones y sus aristas secuencias de configuraciones para transitar entre los nodos. La construcción de estos mapas es intratable computacionalmente por lo que se obtienen mediante muestreo ya sea en forma global (como los *Probabilistics Roadmaps* o PRMs) o incremental (como los *Rapidly Exploring Random Trees* o RRTs)

4. *Mapas semánticos*. Expresan información de más alto nivel, como el tipo de ambiente (interiores, exteriores) o la clase de habitación (sala, comedor, recámara, etc.) que permiten dar comandos en forma más natural a los robots, por ejemplo: "robot, tráeme una manzana de la cocina".

La construcción del mapa por el propio robot permite que la representación sea coherente con sus capacidades sensoriales; esto ha motivado un desarrollo significativo de las técnicas de SLAM, las cuales se estudian en México desde los noventa.[61]

[61] Por ejemplo, Romero, L., Sucar, L.E., Morales, E. (2000). **Learning Probabilistic Grid–Based Maps for Indoor Mobile Robots Using Ultrasonic and Laser Range Sensors**. En *MICAI 2000 - LNAI 1793,* Springer-Verlag, pp. 158–169.

Además de contar con conocimiento del mundo físico en el que habita, el robot puede tener otros tipos de conocimiento, como:

- Conocimiento sobre los objetos en su ambiente y sus propiedades, incluyendo objetos pequeños (manipulables), objetos fijos, etcétera.

- Conocimiento sobre diversos agentes, como personas u otros robots (por ejemplo, para reconocerlas).

- Conocimiento de lenguaje que le permita comunicarse con personas y posiblemente con otros robots.

Para representar este conocimiento se utilizan los diversos formalismos que se han desarrollado en inteligencia artificial, incluyendo lógica de predicados, reglas de producción, representaciones estructuradas, modelos gráficos probabilistas, entre otros.

En particular el robot Golem-III cuenta con una base de conocimiento no-monotónica en la que se pueden representar clases de individuos en una estructura jerárquica, con sus propiedades y relaciones, así como individuos concretos de cada clase, también con sus propiedades y relaciones particulares.[62] El sistema permite la expresión de "valores por omisión" o *defaults*, así como excepciones, por lo que permite algunas formas de razonamiento no-monotónico.[63] La base de datos de conocimiento se embebe en la conducta perceptual y motora del robot, y permite una gran expresividad y flexibilidad. El robot Golem-III cuenta también con un sistema de inferencia que le permite hacer diagnósticos, tomar decisiones (en relación a un conjunto de valores o preferencias preestablecidas) así como planes para satisfacer dichos objetivos. Por ejemplo, si en la tarea de asistente del supermercado Golem-III

[62] Pineda, L. A., Rodríguez, A., Fuentes, G., Rascón, C., Meza, I. (2017). **A Light non-monotonic knowledge-base for service robots.** *Intel Serv Robotics* 10(3):159-171. doi:10.1007/s11370-017-0216-y. Ver también el capítulo 1.

[63] Idem.

no encuentra un objeto en el estante en el que debería estar es capaz de diagnosticar que el proveedor lo puso en un estante incorrecto, de tomar la decisión de ordenar los estantes y/o atender los requerimientos del cliente, y de construir un plan y llevarlo a cabo para lograr este objetivo.[64] Este ciclo de razonamiento se pude utilizar de manera directa como parte de la estructura de una tarea, o dinámicamente, cuando las expectativas del robot no se cumplen, por lo que se sale de contexto; en este caso el ciclo inferencial tiene por objetivo identificar qué pasó y cómo contextualizarse nuevamente para completar la tarea.

4.7. Nivel implementacional

Este nivel implementacional está constituido por la plataforma de hardware del robot, incluyendo las partes mecánicas y electrónicas, así como por los programas de apoyo que permiten el acceso a los dispositivos de hardware (sensores y actuadores) y la interacción entre programas y datos (sistema operativo). Presentamos primero algunas plataformas robóticas orientadas a robots de servicio, y luego los principales programas de apoyo, en particular los sistemas operativos robóticos.

4.7.1. Plataformas de robots de servicio

Las plataformas más comunes para robots de servicio son los robots móviles.[65] Éstas se desplazan sobre ruedas utilizando diversas estructuras (comúnmente la llamada "diferencial") e incluyen en general los siguiente elementos:

- Motores para el control de las ruedas que permiten el desplazamiento del robot sobre el piso.

- Baterías que proveen la energía eléctrica a los diversos dispositivos.

[64] Ver nota 14.

[65] En la figura 2 se ilustran diversas plataformas robóticas de algunos grupos en México.

- Sensores internos que permiten conocer el estado del robot, como el nivel de la batería y su desplazamiento (odometría).
- Sensores externos para percibir su ambiente, como sonares, láser, infrarrojos, cámaras de video, cámaras de profundad y micrófonos.
- Equipo de cómputo interno para el control de sensores, actuadores y la ejecución de los programas que proveen la funcionalidad al robot.
- Equipo de comunicación que permite conectarse a la red (Internet) y usar otro equipo de cómputo externo si es necesario.
- Brazos manipuladores (normalmente uno o dos) para poder tomar y transportar objetos.
- Otros elementos como el simular una cara que dé cierta apariencia humanoide al robot y le facilite la interacción con las personas.

La mayoría de estos dispositivos se importan en México, aunque ya hay compañías mexicanas que han logrado desarrollos de calidad de brazos, manos y torsos[66] así como desarrollos experimentales de cabezas y rostros con la capacidad de hacer gestos y expresar emociones, como alegría, tristeza o sorpresa.[67]

[66] Por ejemplo, el Laboratorio de Innovación y Desarrollo Tecnológico (LAIDETEC), surgida del Departamento de Probabilidad y Estadística del IIMAS, UNAM, por iniciativa de Hernando Ortega Carrillo, con el apoyo de la Incubadora de Empresas de Base Tecnológica, InnovaUNAM. Sus líneas principales son el diseño y construcción de prótesis de mano robóticas y el desarrollo de una plataforma para robots de servicio con fines educativos y de investigación. Asimismo, los brazos, el torso, las pinzas y manos de Golem-III se diseñaron y construyeron por Hernando Ortega dentro del contexto de su participación y colaboración con el Grupo Golem.

[67] Por ejemplo la cabeza y rostro del robot Golem-III con la capacidad de expresar emociones se desarrollaron por Mauricio Reyes Castillo del Centro de Investigación en Diseño Industrial (CIDI) de la Facultad de Arquitectura de la UNAM en el contexto de su investigación doctoral en el Posgrado de Ciencia e Ingeniería de la Computación de la UNAM y su participación en el Grupo Golem.

Otro clase de robots de servicio son los robots humanoides que utilizan piernas robóticas para su desplazamiento. Tienen la ventaja de una mayor flexibilidad con respecto a los robots móviles ya que no están limitados a superficies planas. Sin embargo, son menos estables y en general se mueven más lentamente por lo que su uso como robots de servicio es aún muy limitado.

Recientemente han surgido los robots autónomos aéreos (drones), en particular aquellos basados en estructuras de tipo helicóptero como los cuadrópteros. Estos tienen también aplicaciones de servicio, como la de la entrega de paquetes a domicilio y la de vigilancia. En México hay algunos grupos que empiezan a explorar este campo, incluyendo grupos en el CINVESTAV, el ITESM y el INAOE.

4.7.2. Programas de apoyo

Los programas de apoyo proveen la interfaz entre los algoritmos de alto nivel y el hardware del robot, en particular los sensores y actuadores. Consisten de un conjunto de manejadores (*drivers*) y librerías que facilitan el acceso a cada tipo de dispositivo del robot y normalmente dependen de la plataforma robótica y dispositivo específico.

Adicionalmente, los programas de apoyo deben incluir los servicios análogos a los que proveen los sistemas operativos en las computadoras, como el manejo de la memoria, la ejecución y comunicación de procesos, la administración de usuarios, etc. Para ello se han utilizado sistemas operativos desarrollados para equipo de cómputo genérico, en particular sistemas basados en UNIX como LINUX orientados a los sistemas que funcionan en tiempo real. Recientemente han surgido sistemas operativos enfocados a robots, dentro de los cuales ROS (*Robot Operating System*)[68] se ha convertido en un estándar de facto en la comunidad robótica, en especial entre académicos. ROS facilita la interoperabilidad de los programas en diferentes plataformas y además per-

[68] http://www.ros.org/

mite compartir algoritmos entre los grupos de investigación, promoviendo la colaboración y el desarrollo de la robótica en el mundo.

4.8. Resumen y retos futuros

Los robots son máquinas que pueden percibir y actuar en el mundo controlados por procesos computacionales para realizar diversas tareas. Se pueden distinguir diferentes tipos de robots de acuerdo a su autonomía y la complejidad de su entorno, destacando los robots de servicio de media o alta autonomía que operan en entornos de complejidad baja o moderada. Estos robots se orientan a asistir a las personas en diversas actividades y ambientes, los cuales se pueden conceptualizar en tres niveles de sistema: i) funcional, ii) de dispositivos y algoritmos y iii) de implementación. El nivel funcional corresponde a la especificación y coordinación de las capacidades del robot; se han desarrollado diferentes esquemas a nivel funcional incluyendo las máquinas de estado, los lenguajes de especificación de tareas y los procesos de decisión de Markov. El nivel de algoritmos incluye los elementos computacionales que proveen las principales capacidades al robot: i) percepción y conducta motora, ii) audición y lenguaje, iii) conocimiento y razonamiento. El nivel de implementación está constituido por los elementos mecánicos y electrónicos del robot, y por los programas de apoyo y sistema operativo.

En México hay varios grupos activos en investigación y desarrollo en robótica, que han hecho aportaciones en los diferentes niveles y aspectos, con énfasis en los robots de servicio. Una forma de impulsar y evaluar los desarrollos en robótica es la participación en competencias, donde destaca RoboCup[69] a nivel internacional, en la cual participan regularmente los grupos mexicanos, y el Torneo Mexicano de Robótica[70] en nuestro país.

El desarrollo de los robots de servicio está en sus inicios y se espera que estos artefactos en el futuro sean ubicuos en nuestras casas, oficinas, hospita-

[69] http://www.robocup.org/
[70] https://femexrobotica.org/

les, etc., como los televisores y las computadoras. Sin embargo, para lograrlo todavía existen retos importantes por conquistar, incluyendo los siguientes:

- En cuanto a los aspectos mecánicos y electrónicos, se necesitan robots que puedan desenvolverse en ambientes como los que habitamos los humanos, para lo que se requiere robots humanoides que sean más robustos y rápidos que los actuales, con capacidades de desplazarse en ambientes complejos y manipular diferentes tipos de objetos. Deben mejorarse los sistemas de almacenamiento de energía (baterías) para que tengan una mayor autonomía.

- Aunque las capacidades de percepción, planeación y manipulación han evolucionado en los últimos años todavía falta mucho por hacer; por ejemplo que los robot puedan doblar prendas de ropa como una persona, abrir cajones y tomar objetos ocluidos, o desplazarse en ambientes desconocidos y dinámicos, entre otros.

- Dado que los robots de servicio tendrán que resolver tareas muy diversas, una capacidad fundamental es que puedan aprender de su experiencia o con ayuda del usuario, de forma que puedan realizar nuevas tareas o mejorar las que ya conocen. También deberán poder utilizar, como nosotros, los recursos disponibles en Internet, incluyendo textos, fotos y videos.

- La interacción con los humanos es fundamental para los robots de servicio, incluyendo la interacción a través del lenguaje natural hablado apoyada por ademanes. Además, para que logren una mayor empatía con sus usuarios deberán ser capaces de reconocer su estado emocional y simular ciertas emociones.

5. Ingeniería de Software

La Ingeniería de Software es una disciplina que tiene como propósito desarrollar programas de cómputo que brinden soluciones automatizadas a necesidades expresadas por personas con intereses en común; para tal fin dispone de un conjunto de técnicas, herramientas, métodos y procesos que se utilizan para la creación y mantenimiento de programas de cómputo.

5.1. Origen de la disciplina

Desde la aparición de las primeras computadoras en la década de los cincuenta, la necesidad de crear programas para solucionar problemáticas inicialmente vinculadas a la realización de numerosas operaciones de cálculo y, más tarde, al procesamiento de grandes volúmenes de datos, ha estado presente como parte de la actividad profesional de nuestra sociedad; dichos programas de computadora, conocidos como Programas de Software[1] consisten de un conjunto de instrucciones organizadas —de acuerdo a una sintaxis y con un estilo definido— que controlan y coordinan las operaciones que ejecuta el hardware de una computadora; para este efecto se utilizan los datos que se reciben a través de algún medio o que previamente se han almacenado en algún dispositivo, o incluso una combinación de ambos, con el objetivo de brindar solución a problemáticas previamente definidas u ofrecer algún tipo de servicio.

[1] Abran, A. et al. (2004). **SWEBOK: Guide to the Software Engineering Body of Knowledge Version,** IEEE Computer Society, Los Alamitos, California.

A finales de la década de los sesenta, ante el crecimiento en las capacidades de los equipos de cómputo, las exigencias de solución a problemas cada vez menos triviales, así como la proliferación de lenguajes de programación, se reconoció la necesidad de contar con un nuevo enfoque —sistemático, disciplinado y cuantificable— que permitiera a la programación evolucionar como disciplina profesional y dar respuesta a los nuevos desafíos del proceso de desarrollo de dichos programas de computadora; es decir, se comenzó a reconocer la necesidad de aplicar el enfoque ingenieril al proceso de desarrollo de software.

El término "Ingeniería de Software" se puede atribuir al menos a tres pioneros de la computación. Meyer[2] lo atribuye a Anthony Oettinger —Presidente de la ACM entre 1966 y 1968— quien al reflexionar sobre la actividad de los profesionistas en el emergente campo de la industria computacional, reconoce la existencia de nuevas profesiones de naturaleza ingenieril, como la Ingeniería del Hardware y la Ingeniería del Software. Por su parte, Margaret Hamilton —galardonada por el presidente de los Estados Unidos en noviembre de 2016— al ser entrevistada sobre el proyecto Apolo, comenta que durante el tiempo en que se desarrolló el software de navegación "*on-board*" comenzó a utilizar el término "Ingeniería de Software" para distinguirlo del hardware y de otras ingenierías; sin embargo, en aquel entonces se consideró una broma por lo divertido que les resultaba.[3] Aunque probablemente no es el primero en haber citado dicho término, Friedrich Ludwic Bauer es seguramente el más reconocido por haberlo utilizado al referirse a la necesidad urgente de aplicar la ingeniería al proceso de fabricación del software. Su intervención en la reunión del Comité de Ciencia de la Organización del Tratado del Atlántico Norte (OTAN) a finales de 1967 generó un impacto mediático que dio lugar a un par de conferencias celebradas en Alemania en 1968[4] y en

[2] Meyer, B. (2013). **The origin of software engineering**. https://bertrandmeyer.com/2013/04/04/the-origin-of-software-engineering/

[3] Hamilton, M. **The engineer who took the Apollo to the moon**. https://medium.com/@verne/margaret-hamilton-the-engineer-who-took-the-apollo-to-the-moon-7d550c73d3fa

[4] Naur, P., Randell, B. (1969). S**oftware Engineering: Report of a conference sponsored by the NATO Science Committee**. Garmisch, Germany, 7-11 Oct. 1968, NATO.

Italia en1969,[5] en las cuales se analizaron aspectos relevantes como el diseño, producción, especificación y calidad del software, entre otros temas.

5.2. Importancia del Software y necesidad de una disciplina ingenieril

En la actualidad resulta prácticamente imposible llevar a cabo cualquier tarea cotidiana sin el uso o la aplicación de software. La fabricación industrial, las infraestructuras y organizaciones nacionales, los servicios públicos y los sistemas financieros se controlan mediante sistemas de software. La industria del entretenimiento como la música, los videojuegos, la televisión y el cine utilizan también software de forma intensiva.

De acuerdo con Sommerville la Ingeniería de Software resulta especialmente importante por dos razones:[6] i) cada vez es más frecuente que las personas y las sociedades se apoyen en sistemas de software más avanzados y complejos por lo que es necesario producir software confiable, de manera económica y rápida; y ii) generalmente resulta más barato el uso de métodos y técnicas específicas de ingeniería de software a largo plazo que sólo diseñar y codificar software, como si fuese un proyecto de desarrollo personal.

Adicionalmente, diversos estudios han argumentado la necesidad de la disciplina ingenieril para los proyectos de Tecnologías de la Información (TI); por ejemplo *Standish Group*[7] señala que menos del 40% de los proyectos de TI resultan exitosos; McKinsey[8] señala que en promedio los grandes proyectos en el área de TI sobrepasan en un 45% los presupuestos estimados, utilizan

[5] Buxton, J., Randell, B. (1970). **Software Engineering Techniques: Report of a conference sponsored by the NATO Science Committee**, Rome, Italy, 27-31 Oct. 1969, NATO.

[6] Sommerville, I. (2010). **Software Engineering** 9ª. Ed. Pearson.

[7] Standish Group International (2012). **CHAOS Manifesto 2012 Coll. Research Reports**. The Standish Group International, Inc.

[8] McKinsey&Company (2012). **Study on large scale IT projects**, McKinsey & Company and the University of Oxford.

un 7% de tiempo adicional para su ejecución y sus entregables poseen un valor por debajo del acordado hasta en un 56%; KPMG[9] reporta que el 70% de las organizaciones han sufrido al menos un fallo en sus proyectos de TI.

Por otro lado las tendencias en áreas tecnológicas como *Big Data* o *IoT* (*Internet of Things*) demandarán la generación de una mayor cantidad de software; por tanto, si no se fomenta el enfoque ingenieril en el software no se logrará la madurez requerida para explotar la información existente.

Por lo anterior la Ingeniería del Software resulta imprescindible para el desarrollo y mantenimiento de software de calidad y aunque aquellas problemáticas que dieron origen a la "crisis del software" —costos imprecisos, plazos de entregas incumplidos y requisitos no satisfechos— aún no se han resuelto totalmente, la disciplina apenas cumplirá medio siglo de existencia, por lo que es de esperarse que en el siglo XXI se logren importantes avances en la consolidación de su cuerpo de conocimientos.

5.3. El cuerpo de conocimientos de la Ingeniería de Software

El cuerpo de conocimientos que sustenta a la Ingeniería de Software comenzó a construirse desde finales de la década de los sesenta precisamente en las reuniones de la OTAN en las que comenzaron a analizarse los síntomas de la crisis del software.

El desarrollo de software, analizado desde la óptica de la dualidad Proceso-Producto, consiste de cinco fases: i) requisitos, ii) diseño, iii) codificación, iv) pruebas y v) mantenimiento. Estas fases integran las actividades y tareas organizadas para un proyecto específico, en función del ciclo de vida, como se ilustra en la Figura 5.1. Este ciclo establece el orden de las fases y procesos involucrados, así como los criterios de transición de una fase a otra. Las

[9] KPMG (2010). **Survey of 100 businesses across a broad cross section of industries**, New Zeland, 2010.

actividades se seleccionan de acuerdo a la problemática, la claridad de los requisitos por parte del cliente/usuario, la madurez del equipo de desarrollo y la novedad de la tecnología, entre otros aspectos. Dichas tareas y actividades transforman la descripción de las necesidades expresadas por el cliente o usuario en un conjunto de artefactos a saber: i) Especificación de Requisitos Software, ii) Diseño y iii) Código, que derivan finalmente en un software con ciertas funcionalidades y restricciones de operación acordadas.

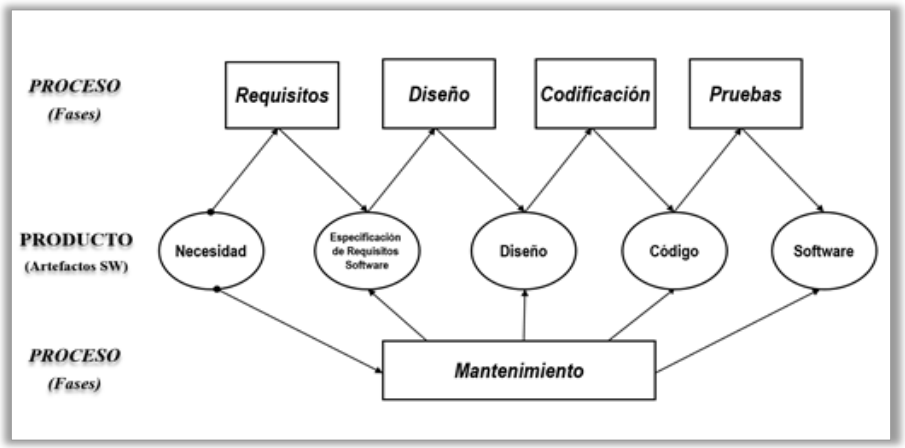


Figura 5.1. Dualidad Proceso-Producto en la Ingeniería de Software

El rápido desarrollo de la disciplina originó un acervo de conocimientos desde sus primeras tres décadas; la Sociedad de Computación del Instituto de Ingeniería Eléctrica y Electrónica (*IEEE-CS*, por sus siglas en inglés) y la Asociación para la Maquinaria Computacional (*ACM,* por sus siglas en inglés) en su proyecto denominado SWEBOK compilaron el conocimiento generalmente aceptado en la disciplina hasta 2004[10] y establecieron dos subconjuntos de áreas de conocimiento, cinco vinculadas con los procesos de desarrollo y otras cinco con las áreas de gestión. Junto con estas áreas se reconocieron otras disciplinas directamente relacionadas con la actividad de la disciplina;

[10] Ver nota 1.

una década después, dicho cuerpo se revisó y actualizó; el SWEBOK V3.0[11] representa el próximo paso en la evolución de la disciplina. En las siguientes subsecciones se describen dichas áreas así como el trabajo de investigación que se ha desarrollado en cada una de éstas por la comunidad en México.

5.3.1. Requisitos de Software

Se entiende por requisitos de software al conjunto de funcionalidades y restricciones expresadas respecto a un producto de software; los requisitos contribuyen a la solución de un problema, a la mejora de un servicio o a la automatización de un proceso específico; esta área de conocimiento integra un conjunto de procesos vinculados con la obtención de los requerimientos, el análisis, la negociación, la especificación y la validación.

La primera etapa del proceso, denominada *obtención*, tiene como propósito identificar aquellas necesidades reales y restricciones de operación —criterios de calidad— expresadas por un conjunto de personas o entidades que son afectadas por el sistema de información, servicio o proceso en cuestión que se desea automatizar; dichas personas o entidades denominadas *stakeholders* pueden ser usuarios, clientes, proveedores, decisores o reguladores externos; en fin, todo aquel que tiene algún interés y que, por ende, puede describir, desde una óptica distinta el funcionamiento actual, así como las oportunidades de mejora del sistema de información, servicio o proceso. Para la obtención de información, el Analista de Requisitos (AR)[12] —papel del especialista en el proceso de requisitos— diseña una estrategia en la que combina un conjunto de técnicas de educción como la entrevista, la encuesta estructurada/no estructurada, la observación de tareas habituales, los escenarios de uso, modelado del negocio, el uso de prototipos, por mencionar algunas de las más conocidas.

[11] Bourque, P., Firley, R. (2014**). Guide to the Software Engineering Body of Knowledge**. SWEBOK V3.0. IEEE Computer Society Press.

[12] Young, R. (2004). **The Requirements Engineering Handbook**, Artech House Inc., Capítulo 2, 3 y 7.

La etapa de *análisis y negociación* tiene como objetivo asegurar la calidad de los requisitos antes de incorporarlos al documento de especificación. En esta etapa se precisan los límites del sistema software y su interacción con el entorno y se traducen los requisitos del usuario a requisitos de software; es decir, se deben transformar las descripciones —desde la perspectiva del problema— obtenidas de los *stakeholders* respecto a los servicios que esperan del sistema, en descripciones detalladas —en términos de la solución— acerca de las funcionalidades y restricciones de operación de dichos servicios o prestaciones. Para logar lo anterior se realizan tres tareas principales con los requisitos: i) Clasificación, ii) Modelización y iii) Negociación. La tarea de Clasificación consiste en separar las descripciones de aquellos requisitos que expresan la funcionalidad o servicios que el sistema debe brindar —requisitos funcionales— de las descripciones que expresan las restricciones de operación o propiedades que debe cumplir el sistema —requisitos no funcionales. La segunda tarea, la Modelización, tiene como propósito identificar la arquitectura del sistema mediante una representación gráfica, típicamente semiformal, en la que se puedan identificar los subsistemas y requisitos asociados a cada uno de éstos. Finalmente, la tarea de Negociación consiste en identificar y eliminar los conflictos entre requisitos, con la intención de obtener finalmente un conjunto de requisitos factibles y mutuamente satisfactorios.

La etapa de *especificac*ión consiste en documentar los requisitos de software acordados, en un nivel apropiado de detalle; para ello se suele utilizar un modelo de documento redactado en términos comprensibles para los *stakeholders*; para este efecto el IEEE propuso el estándar 830-1998;[13] sin embargo, éste es sólo una recomendación. Un gran reto de la especificación consiste en expresar con claridad lo que los clientes desean y necesitan en términos de requisitos de software, por lo que se recomienda el uso de plantillas que permitan estandarizar el lenguaje utilizado; no obstante, en ocasiones la especificación textual es insuficiente por lo que es conveniente acompañar las

[13] IEEE Std 830-1998. **IEEE recommended practice for software requirements specifications**. *IEEE Computer Society,* Los Alamitos, 1998.

descripciones con modelos gráficos que ilustren algún aspecto que resulte de interés para el sistema; por ejemplo, para ilustrar la interacción del usuario con el sistema y el flujo de datos entre los procesos se suelen utilizar Diagramas de Casos de Uso y Diagramas de Flujo de Datos respectivamente; en ocasiones una aplicación (o parte de ella) se interpreta mejor si se piensa que está en uno o varios estados, para lo cual se suelen utilizar los Diagramas de Transición de Estado.

Finalmente, en la última etapa del proceso de requisitos, la fase de *validación*, se realiza una revisión cuidadosa de la consistencia, completitud y otros aspectos específicos de interés particular, vinculados con la calidad del documento; el objetivo es identificar problemas en el documento de especificación de requisitos de software antes de que sea usado como base para el diseño del sistema.

5.3.2. Diseño de Software

Se entiende por diseño al proceso de definición de la arquitectura, componentes, interfaces y otras características del sistema software, así como del producto de dicho proceso.[14] El diseño de software es una tarea que se lleva a cabo en una etapa temprana de los procesos vinculados con la construcción de una solución de software —diseño, codificación y pruebas— y a diferencia de la fase de requisitos en la que se define qué debe hacer el sistema, en el diseño se decide cómo debe hacerlo; para ello, el ingeniero de software utiliza como guía un conjunto de principios, métodos y técnicas.

Los principios como la abstracción, acoplamiento, cohesión, modularización, entre otros, representan nociones clave que proporcionan la base para muchos enfoques y conceptos de diseño de software diferentes; por su parte, los métodos y técnicas han ido evolucionando a la par con los diferentes paradigmas de desarrollo. A través de los métodos, se generan representaciones

[14] IEEE Std. 610.12-1990 (1990). **IEEE Standard Glossary of Software Engineering Terminology**.

—generalmente gráficas— de diferentes aspectos vinculados con el software, principalmente, aspectos relacionados con la estructura y el comportamiento que debe tener para satisfacer los requisitos acordados.

En una analogía con la ingeniería civil, para la construcción de una casa se debe pensar cómo la construcción podrá satisfacer las necesidades expresadas por sus futuros habitantes. Parte de la tarea de diseño consiste en elaborar los planos en los que se representan la ubicación y distribución de los espacios físicos, como la sala, el comedor y las habitaciones, y de los ductos y tuberías para el suministro de electricidad y agua. En el caso del diseño del software, los métodos y las técnicas permiten modelar los aspectos estructurales del software, como la distribución de los cuartos, y su dinámica de operación o comportamiento, como el servicio eléctrico y de agua, independientemente del paradigma utilizado.

En la sección correspondiente a Métodos y Modelos de la Ingeniería de Software se ofrece una mayor descripción de los métodos para diseño de software, los cuales se han desarrollado a la par del paradigma estructurado y del orientado a objetos. Desde la perspectiva del proceso de software, el diseño comprende tres actividades principales: i) la definición de la arquitectura, ii) la definición de sus componentes y iii) la definición de la interfaz del sistema.

La definición de la arquitectura tiene como propósito descomponer y organizar el software en componentes e interfaces, entendiendo por componente una clase, un módulo e incluso una base de datos. La arquitectura de alto nivel es un modelo que sirve como insumo a la fase de pruebas; se utiliza para diseñar la estrategia de integración de los diferentes componentes del sistema, así como para verificar su funcionamiento integral.

El objetivo de la definición de componentes o diseño detallado es especificar las estructuras de datos, algoritmos, características de la interfaz y mecanismos de comunicación con suficiente detalle para implementarlos. El proceso incluye una serie de tareas que permiten representar al software a

un nivel de abstracción cercano a la codificación. Uno de los componentes usualmente requeridos es el modelo de persistencia de los datos, mejor conocido como el modelo de la Base de Datos; para el diseño de dicho componente, independientemente del paradigma, se suele utilizar el modelo relacional.

La definición de interfaz de usuario tiene como propósito modelar el funcionamiento e interacción del usuario con el software, donde los modelos de usuario, mental y de implementación representan la interacción entre el usuario y el sistema.[15] Para que este proceso sea efectivo, el diseñador debe realizar un análisis de las tareas de los usuarios, del entorno de trabajo y cómo los usuarios interactúan con otras personas o con otro software y, de ser posible, desarrollar un prototipo; es importante realizar una evaluación de la interfaz propuesta, la cual permita retroalimentar al equipo de desarrollo, así como otros aspectos vinculados con la experiencia del usuario. En resumen, la interfaz de usuario comprende dos componentes principales: el lenguaje de presentación —conjunto de objetos como íconos, menús y formularios, entre otros— y el lenguaje de acción —como el lenguaje de comandos y tiempos de respuesta, por mencionar algunos. Ambos componentes se corresponden con la forma y el contenido en la definición de la interfaz de usuario.

5.3.3. Construcción de Software

Cuando se habla del proceso de construcción del software generalmente se piensa en las tareas vinculadas con las tres fases centrales del desarrollo: el diseño, la codificación y las pruebas. La codificación utiliza las salidas del diseño y proporciona el insumo de la fase de pruebas; no obstante, los límites entre el diseño, la codificación y las pruebas no siempre son claros y suelen variar en función del ciclo de vida seleccionado para el desarrollo de un proyecto

[15] Pressman, R. (2010). **Ingeniería de Software un Enfoque Práctico**, McGraw-Hill, Séptima Edición. Ver también el capítulo de Interacción Humano Computadora en este texto.

software. En este apartado nos enfocamos al proceso de codificación y más adelante se abordan la terminología y las actividades de diseño y pruebas.

La codificación, como actividad socio-técnica, involucra actividades dependientes del lenguaje, las cuales requieren del conocimiento de la sintaxis, la semántica y la pragmática del lenguaje de programación. Este es uno de los motivos por los que el aprendizaje de la programación es uno de los principales problemas en las primeras etapas de la formación de los ingenieros de software. Una de las formas de clasificar a los lenguajes de programación es en términos de su expresividad; si el lenguaje utiliza expresiones cercanas a las de una computadora, como el ensamblador, es de más bajo nivel que aquellos que permiten generar expresiones más cercanas al lenguaje humano, como Pascal. Por lo anterior, la elección del lenguaje de programación no es tan sólo la manera en la que se traduce el diseño detallado a código, sino que influye significativamente en la complejidad de la abstracción utilizada durante el proceso de construcción.

Los principios fundamentales que se utilizan en la construcción de software y que guían el proceso de codificación son: i) minimizar la complejidad, ii) anticiparse a los cambios, iii) construir para verificar, iv) reutilizar y v) utilizar estándares.

La complejidad se refiere al grado en que un sistema o componente tiene un diseño o código difícil de entender o de verificar.[16] Este primer principio consiste en generar código que sea fácil de entender, probar y mantener; para ello se han definido un conjunto de buenas prácticas, como técnicas de legibilidad o el uso estándares de documentación, que ayudan al ingeniero de software a simplificar el código. El segundo principio se relaciona con la utilidad del software a lo largo de su vida ya que éste se someterá a un proceso de mantenimiento en varias ocasiones por lo que es necesario anticiparse a los cambios desde el proceso de codificación; en este sentido es recomendable

[16] IEEE Std. 610.12-1990, **IEEE Standard Glossary of Software Engineering Terminology**

identificar y aislar[17] aquellas áreas que son más proclives de ser modificadas, como las reglas del negocio, las dependencias del hardware y las entradas y salidas, para que en lo posible los cambios afecten a módulos y/o rutinas ubicadas en el código de antemano. El tercer principio, construir para verificar, tiene como filosofía el uso sistemático de pruebas de unidad mediante la automatización y el uso de métodos estandarizados para las revisiones del código. La reutilización se refiere al uso de los activos existentes —módulos, bibliotecas, componentes— para la resolución de diferentes problemas, y en términos generales, para mejorar tanto la productividad como la calidad. Finalmente, el uso de estándares tiene como propósito brindar interoperabilidad entre artefactos desarrollados por diferentes equipos de desarrollo y/o por diferentes individuos asignados a un mismo proyecto.

5.3.4. Pruebas de Software

Antes de describir los procesos relacionados con la fase de pruebas es conveniente aclarar los términos que se utilizan en el proceso de verificación. En general los comportamientos que difieren de los que se especifican por el usuario son defectos o errores del software; en particular, cuando el defecto se manifiesta en tiempo de ejecución se dice que ha habido un fallo en el software, el cual se puede deber a una falta cometida por el equipo de desarrollo, ya sea en el código, en el diseño o en algún artefacto previamente elaborado.

Las pruebas al software son procesos dinámicos que tienen como propósito verificar la calidad del código y evaluar su comportamiento; para este efecto se generan casos de prueba específicos y se intenta generar fallos. La actividad inmediata es la depuración, la cual consiste en descubrir las causas del defecto del software —por ejemplo, la falta en el código que al momento de ejecución generó el mal funcionamiento— y modificarlo para generar el comportamiento deseado. Aunque la evaluación tiene como propósito

[17] McConell, S. **Code Complete, A Practical Handbook of Software Construction**, Microsoft Press, 2nd. Edition, 2004.

generar información relacionada con los defectos, éstos no siempre se deben a faltas en el código y las faltas no siempre generan fallos. En términos generales, las pruebas no tienen como propósito obtener un código libre de faltas, sino eliminar faltas que permitan reducir al mínimo el tiempo medio entre fallos.

Para el diseño de las pruebas se conocen dos estrategias generales: las pruebas de caja negra, que se enfocan en los resultados generados al ejecutar el código en función de los datos de entrada, y las pruebas de caja blanca, enfocadas a seguir puntualmente el comportamiento del software durante la ejecución del código (por ejemplo, una estrategia es generar casos de prueba que obliguen a utilizar todos los caminos posibles para el flujo de los datos).

Existen cuatro niveles o tipos de pruebas: i) pruebas unitarias: se ejecutan a nivel de módulos o componentes y verifican el funcionamiento correcto de cada elemento; ii) pruebas de integración: se centran en diseño de alto nivel o diseño de la arquitectura; el objetivo es verificar la interacción entre todos los módulos y diseñar una estrategia para integrar cada uno de los componentes previamente probados de manera individual; iii) pruebas del sistema: verifican el comportamiento del software como una unidad; generalmente se enfocan a la evaluación de los requisitos no funcionales acordados con el usuario —aspectos como rendimiento, fiabilidad, velocidad, etc.— ya que los funcionales se evalúan en los dos niveles de prueba previos; y iv) pruebas de aceptación: se ejecutan en conjunto con el usuario; el documento de especificación de requisitos se utiliza como guía y el objetivo es validar que el software dé respuesta a las necesidades de funcionalidad, así como las restricciones acordadas al inicio del proceso de desarrollo. En resumen, los tres primeros niveles permiten verificar que el software funcione correctamente y el cuarto validar que disponga de la funcionalidad acordada con el usuario.

5.3.5. Mantenimiento de Software

El estándar IEEE 610.12[18] define al Mantenimiento del Software como la modificación de un sistema software, o de un componente, después de que se ha entregado a los usuarios o clientes con el fin de corregir defectos, mejorar su rendimiento u otros atributos, o adaptarlo a un cambio en el entorno.

A diferencia del hardware, el software no se avería con el tiempo, por lo que no requiere reparación o reemplazo de sus componentes; por lo mismo el mantenimiento del software es diferente al del hardware o al realizado a otros artefactos físicos. En términos prácticos, el mantenimiento es un proceso que permite al software brindar la utilidad para la cual fue concebido, a pesar de la volatilidad de los requisitos del usuario y de los cambios de los entornos tecnológicos; dependiendo del tipo de modificación requerida, el mantenimiento puede clasificarse en una de cuatro categorías: i) correctivo, ii) perfectivo, iii) adaptativo y iv) preventivo.[19]

A pesar de la estrategia utilizada para la verificación en la fase de pruebas siempre existe la posibilidad de encontrar defectos una vez que se libera el software; por ende se tiene la necesidad de continuar con la fase de mantenimiento correctivo a lo largo de su ciclo de vida. Los defectos se pueden deber a faltas cometidas en alguna de las fases del proceso de desarrollo —requisitos, diseño, programación, pruebas— y en función de la urgencia, el mantenimiento correctivo se planifica y ejecuta en el tiempo. Otro motivo para hacer modificaciones al software es la volatilidad en las necesidades de los *stakeholders*; el mantenimiento perfectivo atiende solicitudes para ampliar las funcionalidades o mejoras en aspectos vinculados con la calidad del software (por ejemplo, eficiencia). Por su parte, el mantenimiento adaptativo se refiere a las modificaciones provocadas por cambios en el entorno tecnológico y de

[18] IEEE Std. 610.12-1990, **IEEE Standard Glossary of Software Engineering Terminology.**

[19] Piattini, M. *et al.* **Mantenimiento del Software**, Alfaomega, 2001.

los datos o en el de los procesos. Finalmente, el mantenimiento preventivo se refiere a las modificaciones al software que mejoran sus propiedades, pero sin alterar las prestaciones originales.

Independientemente del tipo de modificaciones realizadas, durante el mantenimiento se ejecutan procesos y actividades relacionadas con las fases correspondientes al desarrollo del software; no obstante, suelen usarse métodos específicos como la ingeniería inversa o la reingeniería.

5.3.6. Gestión de la configuración

Los procesos vinculados con el desarrollo de software generan un conjunto numeroso de artefactos que en algunos casos contienen características volátiles; esta área de conocimiento se refiere a las actividades de gestión relacionadas con la identificación, documentación y control de todos los elementos de configuración[20] acordados en un proyecto de desarrollo; en sentido estricto, es una actividad que se realiza a lo largo de todo el ciclo de vida del software, tanto en los procesos de desarrollo como en las actividades de mantenimiento.

La gestión de la configuración es una disciplina de control que tiene por objetivos establecer y mantener la integridad de los elementos de configuración generados a lo largo de la vida del software; evaluar y controlar los cambios sobre dichos elementos y facilitar la visibilidad del producto. Un concepto clave en estas actividades es el de "línea base", que se refiere a un punto de referencia en el procesos de desarrollo del software que queda marcado por la aprobación de uno o varios elementos de configuración mediante una revisión técnica formal.

Para el logro de sus objetivos, la gestión de la configuración establece cuatro actividades principales:[21] i) identificación de la configuración, ii) con-

[20] Un elemento de configuración es un artefacto software generado como parte del proceso de desarrollo; puede tomar la forma de un: i) código, ii) documento, o iii) conjunto de datos.

[21] IEEE Std. 828-1998, **IEEE Standard for Software Configuration Management Plans**.

trol de la configuración, iii) generación de informes de estado, y iv) auditorias de la configuración. La actividad i) consiste en identificar la estructura del software, sus componentes, el tipo de cada uno de éstos, además de hacerlos únicos y accesibles; ii) se refiere al control de versiones y entregas del software y a los cambios que se producen a lo largo de su ciclo de vida; iii) consiste en informar el estado de los componentes del software y de las solicitudes de cambio, y generar estadísticas de su evolución; finalmente, iv) consiste en validar la completitud de un producto software y la consistencia entre sus componentes y asegurar que el software que se entrega es el que el usuario requiere.

5.3.7. Gestión de la Ingeniería de Software

Para la obtención exitosa de un producto de software, los procesos de gestión son tanto o más importantes que los procesos orientados al desarrollo, es por ello que los primeros deben ser considerados incluso desde antes de la obtención del primer producto y hasta la liberación del producto final.

El área relacionada con la gestión de la Ingeniería de Software involucra las actividades de medición, estimación, planificación, seguimiento y control. La estimación tiene como objetivo determinar los recursos humanos, económicos, así como el tiempo requerido para el desarrollo de un proyecto de software.[22] Para esto es fundamental dimensionar apropiadamente el sistema de manera previa. Una de las primeras métricas claras y precisas para este efecto es el número de líneas de código; sin embargo, su utilización es inviable para la estimación ya que esta medida se conoce hasta una fase avanzada de la codificación. Otra manera de dimensionar el software consiste en determinar los puntos de función[23] y se obtiene desde la primera fase del proceso de desarrollo mediante el análisis de varios tipos de funcionalidad en el documento

[22] Con la asistencia de algún modelo matemático como, por ejemplo, COCOMO (*Constructive Cost Model: COCOMO*) propuesto por Barry Boehm a finales de los setenta.

[23] El método para el análisis de los puntos de función fue propuesto inicialmente por Allan Albrecht en 1979.

de requisitos. El dimensionamiento del software es la base para estimar los recursos necesarios para su desarrollo, el conocimiento de las características del cliente, la madurez del equipo de desarrollo, el conocimiento del dominio del problema, así como los riesgos relacionados con la tecnología considerada, entre otros aspectos; la primera decisión previa a la planificación de las actividades consideradas es la selección de un modelo apropiado de ciclo de vida de desarrollo de software y, consecuentemente, de la metodología de desarrollo.

La planificación genera una relación ordenada de las actividades del proyecto, incluyendo plazos, responsables y recursos necesarios; se identifican claramente las dependencias entre las actividades y los tiempos de holgura para su finalización. Por su parte, el seguimiento permite recolectar y acumular datos sobre los recursos utilizados, costes generados e hitos asociados; éstos a su vez permiten generar los informes de estado y, en su caso, tomar decisiones de control.

Finalmente, las actividades de gestión para la Ingeniería de Software se realizan en tres niveles: i) gestión organizativa y de infraestructura, ii) gestión a nivel de proyectos y iii) gestión del programa de medición.

5.3.8. Procesos de la Ingeniería de Software

El área de procesos se ocupa de las actividades realizadas por ingenieros de software para desarrollar, mantener y operar el software; particularmente es el conjunto de actividades y tareas interrelacionadas que transforman los productos de entrada en productos de salida. Esta área se relaciona con las actividades de trabajo y no con la ejecución del proceso; es decir, especifica lo que se debe hacer, preo no lo que realmente se hace.

Los tópicos de interés en esta área se relacionan con: i) la definición del proceso del software, su gestión y la infraestructura requerida; ii) el análisis

de los diferentes modelos de ciclos de vida; iii) los modelos y métodos de evaluación de procesos de software, iv) los modelos de mejora de procesos de software y v) la medición de procesos y productos de software, así como la calidad de los resultados de dicha medición.

5.3.9. Métodos y modelos de la Ingeniería de Software

Los métodos y modelos relacionados con los procesos de software les imprimen un carácter ingenieril ya que promueven que la actividad se realice de manera sistemática, repetible, pero sobre todo con una orientación al éxito. El uso de modelos proporciona un enfoque particular en la resolución de problemas, así como una notación y procedimientos para la construcción y el análisis del modelo; por su parte, los métodos proporcionan un acercamiento a los procesos relacionados con la especificación, diseño, construcción, prueba y verificación del software, así como con los artefactos generados en dichas fases.

En esta área de conocimiento se hace énfasis en cuatro temáticas: i) la práctica del modelado, ii) la tipología de los modelos, iii) las técnicas de análisis para el modelado y iv) los métodos asociados a los procesos del desarrollo de software.

La aplicación del enfoque ingenieril al proceso de desarrollo de software con el paradigma estructurado, orientado principalmente a sistemas de información, originó los primeros métodos, modelos y técnicas para los procesos de análisis y diseño, actualmente conocidos como procesos de requisitos y diseño; éstos generan representaciones tanto del aspecto algorítmico como del estructural de los datos.[24] En la actualidad, el paradigma orientado a objetos ha generado el desarrollo de métodos, modelos y técnicas que permiten generar software para diversos dominios —comercio, transporte, servicios financieros, etc.— en los que los aspectos estructurales y del comportamien-

[24] DeMarco, T. **Structured Analysis and System Specification**, Prentice Hall, 1979.

to asociado al software requieren mayor variedad de representación, como las que ofrece UML.[25]

5.3.10. Calidad del Software

La calidad del software se define como la capacidad del producto para satisfacer las necesidades declaradas e implícitas bajo condiciones especificadas;[26] esta área se enfoca a las prácticas, herramientas y técnicas para definir la calidad del software, así como para evaluar su estado durante el desarrollo, mantenimiento y despliegue.

Debido a que el software es un producto intangible difiere de la mayoría de los productos construidos bajo otras disciplinas ingenieriles; de hecho, el software no se construye sino más bien se desarrolla. Asimismo, no se avería a lo largo de su vida útil. Más bien puede dejar de ser útil, a pesar de los procesos de mantenimiento a los que se someta; por lo tanto, su curva de fallos tiene un comportamiento distinto a los dispositivos físicos. Ante estas características surge la interrogante de si es realmente posible encontrar un conjunto de propiedades en un producto de software que nos den un indicativo de su calidad. La respuesta se intenta dar a través de los modelos de calidad. En éstos la calidad se define de manera jerárquica: en el nivel más alto se encuentran los factores de calidad, los cuales representan las características que desde el punto de vista del usuario debe reunir el software; se les conoce también como atributos de calidad externos. Cada uno de los factores tiene un conjunto de criterios asociados, cuya presencia contribuye al aspecto de calidad que dicho factor representa. En el nivel más bajo se encuentran las métricas, las cuales indican el grado en que el software posee determinado atributo de calidad. Por ejemplo, un factor de calidad asociado al software es

[25] *Unified Modeling Language* (UML, por sus siglas en inglés) es uno los lenguajes de modelado de aplicaciones de software que se ha convertido en un estándar.

[26] ISO/IEC 25010:2011 **Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Systems and Software Quality Models**, ISO/IEC, 2011.

la corrección, entendiéndola como el grado en que el software satisface las necesidades del usuario; un criterio asociado es la completitud, la cual evalúa que el software implemente todas las funciones requeridas y acordadas; finalmente, una métrica asociada a la completitud es el porcentaje de requisitos cubiertos, lo cual se puede determinar mediante una prueba de validación.

Otra interrogante es cómo es posible medir el grado de calidad de un producto software. Este problema se enfrenta con los procesos de evaluación; en términos generales se pueden identificar dos tipos de procesos de evaluación: verificación y validación.[27] La verificación permite determinar si los artefactos generados al final de una fase del proceso de desarrollo cumplen con los requisitos establecidos durante la fase anterior. La validación, por su parte, tiene como propósito asegurar el cumplimiento de las necesidades del cliente al final del proceso de desarrollo. Las técnicas asociadas a los procesos de verificación y validación se pueden clasificar en dos categorías: estáticas y dinámicas. Las primeras tienen por propósito buscar faltas sobre el sistema en reposo, por lo que se pueden aplicar al final de cada una de las fases del proceso de desarrollo (por ejemplo, revisiones y auditorias); por su parte, las técnicas dinámicas son aplicables al sistema en ejecución, es decir, una vez que el sistema se ha codificado. Las técnicas dinámicas se conocen también como pruebas del software (por ejemplo, unidad e integración) como se ha descrito previamente.

5.4. Aportaciones en investigación

La comunidad de Ingeniería de Software en México se ha orientado principalmente a consolidar la calidad en los programas de formación de recursos humanos,[28] a fortalecer a la industria del software[29] y, en tercer lugar, a la

[27] IEEE Std. 610.12-1990, **IEEE Standard Glossary of Software Engineering Terminology.**

[28] Aguilar, R., Díaz, J. (2015). **La Ingeniería de Software en México: Hacia la consolidación del primer programa de Licenciatura**. *Revista Tecnología Educativa.* 2(2):6–17.

[29] **Resultados completos de la Consultoría "Estrategia de calidad para el crecimiento de la industria de software en México"**, 4to. Entregable, Select, Cd. México, Marzo 2015.

investigación.[30] En esta sección se describen brevemente las principales temáticas que se han abordado en la investigación tomando como referencia las áreas de conocimiento establecidas en el SWEBOK.

En el área de requisitos de software (AC_01) se han desarrollado propuestas de especificación formal de las necesidades del software de manera precisa; destaca el desarrollo de una propuesta para el modelado formal de flujos de tareas,[31] la cual permite verificar, por ejemplo, los flujos detallados en los casos de uso; este modelo ofrece la posibilidad de corregir errores antes de que se llegue a etapas donde sería más costoso enfrentarlos. También se han explorado áreas no tradicionales donde la ingeniería de requisitos debe aplicarse, como en el software para la industria automotriz.[32]

En el área de diseño de software (AC_02) se ha utilizado la ingeniería Web para el diseño de aplicaciones colaborativas en entornos educativos; en particular, se ha explorado el uso de la Metodología UWE para modelar la funcionalidad y las características de dicho tipo de aplicaciones;[33] también en este ámbito se ha trabajado en plataformas de comunicación y educación ambiental soportada por una arquitectura de software para atender requerimientos no funcionales, particularmente, de seguridad y mantenimiento.[34] En el caso de la seguridad se ha elaborado un modelo de calidad que permite determinar las tácticas arquitectónicas y los mecanismos correspondientes para

[30] Juárez-Ramírez, R. et al, **Estado Actual de la Práctica de la Ingeniería de Software en México**. En R. Juárez-Ramírez et al (Eds.), Tendencias de la Práctica de la Ingeniería de Software en México en el ámbito Académico; Tijuana, México, Editorial UABC, pp. 3–14, 2013.

[31] Fernandez, C., Simons, A. (2014). **An implementation of the task algebra, a formal specification for the task model in the Discovery Method**. *J. Appl. Res. Technol.*, 12(5):908–918.

[32] Aguilar, J. Fernández-y-Fernández, C., (2015). **Especificación de Requerimientos para el Desarrollo de Software Automotriz en México**. *Rev. Latinoam. Ing. Softw.*, 3(6):250–258.

[33] Uicab, O., Ucán, J., Aguilar, R. (2016). **Una Herramienta para el Análisis de la Colaboración diseñada con UWE**. *Rev. Latinoam. de Ing. de Soft.*, 4(6):235–242.

[34] Íñiguez, F., Cortes, K., Pérez, J., Contreras, G., Maldonado, A. (2015). **The development of a software architecture for an environmental education platform**. *International Conference on Computing Systems and Telematics (ICCSAT).* Xalapa, Ver., México.

distintos escenarios de seguridad.[35] También se han realizado algunas investigaciones de tipo exploratorio acerca de SOA (*Service Oriented Architecture*) que han llevado a propuestas como la de una arquitectura orientada a servicios para una línea de productos de software.[36]

El área de pruebas de software (AC_03) ha sido poco atendida; se reportan algunos trabajos en los que se ha dado continuidad a la familia de experimentos propuesta inicialmente por Basili[37] en la que se pretende contrastar diferentes tipos de prueba para la detección de faltas en el código; se han realizado réplicas diferenciadas en ambientes académicos que incorporan factores como el uso de un entorno colaborativo virtual inteligente para asistir a los aprendices durante el proceso de identificación de faltas en el código.[38]

El área de construcción de software (AC_04) se ha enfocado a incorporar buenas prácticas así como a la inclusión de diversos patrones arquitectónicos y de diseño que permiten obtener software de calidad, incluyendo la investigación, evaluación y prueba de marcos de trabajo o *frameworks* y estilos arquitectónicos; por ejemplo, el desarrollo de una red social con el propósito de apoyar los estudios de seguimiento de egresados[39] con una aplicación móvil[40] apoyada precisamente en el estilo arquitectónico referido.

---

[35] Figueroa-Gutiérrez, S., Cortés, K., Contreras, G., Pérez, J. (2016). **Desarrollo de un Catálogo de Mecanismos de Seguridad**. *Research in Computing Science*. 128:57–67.

[36] Gómez, R., Cortés, K., Pérez, J., Arenas, A. (2014). **Desarrollo de una arquitectura orientada a servicios para un prototipo de una línea de productos de software**. *Research in Computing Science*, 79:75–86.

[37] Basili, V., Shull, F., Lanubile, F. (1999). **Building Knowledge through Families of Experiments**. *IEEE Transactions on Software Engineering*, 25 (4): 456–473.

[38] Ucán, J., Gómez, O., Aguilar, R. (2016). **Assessment of Software Defect Detection Efficiency and Cost through an Intelligent Collaborative Virtual Environment**. *IEEE Latin America Transactions*, 14(7):3364–3369.

[39] González-Jiménez, B., Contreras-Vega, G., Cortés-Verdín, K. (2012). **Redes Sociales para el Seguimiento de Egresados**. *Research in Computing Science*, 60:239–250.

[40] Zavaleta-Ibarra, L. A., Contreras Vega, G., Cortés Verdín, K., Pérez-Arriaga, J. C. (2014). **Desarrollo de la versión móvil para la red social FEIBook**. *Research in Computing Science*, 79:63–74.

En el área de procesos de software (AC_08) se ha implantado un modelo de referencia y evaluación de procesos en las normas mexicanas NMX-I-059/NYCE-2011 y NMX-I-1554-2010 que responden a las demandas de la industria de software.[41] Entre los resultados resaltan la identificación de factores de éxito y el desarrollo de herramientas para la autoevaluación de procesos. Asimismo, se han definido estrategias para la transferencia de conocimiento en equipos de mejora de procesos y desarrollo de software y se han identificado y caracterizado los tipos y flujos de conocimiento para promover procesos de innovación centrados en la mejora de procesos de software.[42] En el ámbito de la experimentación en entornos académicos se ha trabajado en la definición de métodos que incorporan una descripción ordenada, clara y concisa de lo que deben realizar los alumnos a lo largo de los cursos; cada método toma en cuenta los objetivos del curso y define su conjunto de prácticas descritas con Kuali-Beh;[43] dichos métodos tienen sus prácticas basadas en estándares como ISO/IEC 29110 y SCRUM.

En el área de métodos y modelos de la Ingeniería de Software (AC_09) se ha investigado la creación de modelos y estándares con el objetivo de ayudar a la industria de software a ser más competitiva. Uno de los proyectos más difundidos, con impacto a nivel internacional, ha sido la creación del MoProSoft[44] para las pequeñas organizaciones de software, el cual fue pu-

[41] Flores, B., Astorga, M., Rodríguez, O., Ibarra Esquer, J. E., Andrade, M. (2014). **Interpretación de las Normas Mexicanas para la implantación de procesos de software y evaluación de la capacidad bajo un enfoque de Gestión de conocimiento.** *Revista Facultad Ingeniería Universidad de Antioquia,* pp. 81–100, Colombia..

[42] Flores-Ríos, B., Pino, F., Ibarra-Esquer, J. E., González-Navarro, F. F., Rodríguez, O. (2014). **Análisis de Flujo de conocimiento en Proyectos de Mejora de Procesos software bajo una perspectiva multi-enfoque**. *Revista Ibérica de Sistemas y Tecnologías de la Información*, 14:51–66, Portugal.

[43] Ibargüengoitia, G., Oktaba, H. (2014). **Identifying the scope of Software Engineering for Beginners course using ESSENCE**. En *Software Engineering: Methods, Modeling, and Teaching, vol. 3*. Eds. C. M. Zapata, L. F. Castro. Universidad Nacional de Colombia. Medellín, Colombia 2014, pp. 67-76. lases.cidenet.org/index.php/es/descargas/libro-2014

[44] Oktaba, H. (2005). **Moprosoft: A Software Process Model for Smail Enterprises**. *Proceedings of International Research Workshop for Process Improvement in Small Settings,* 19-20 de Octubre, Pittsburg, EEUU, Special Report CMU/SEO-2006-SR-001, pp. 93-101.

blicado como norma mexicana MNX-I-059-NYCE; este modelo se aceptó por la ISO/IEC JTC1/SC7 *Software and System Engineering* como base para la creación del estándar ISO/IEC 29110 para *Very Small Entities* (VSEs) de la industria de software. Adicionalmente, se definió un método de evaluación de procesos de software denominado EvalProSoft y poco después se trabajó en un tercer proyecto orientado al diseño de pruebas controladas cuyo objetivo fue demostrar que las empresas pueden elevar sus niveles de capacidad adoptando MoProSoft en un tiempo relativamente corto. Otro proyecto de la comunidad en esta área fue el desarrollo de un nuevo estándar llamado *RFP A Foundation for the Agile Creation and Enactment of Software Engineering Methods*, con la propuesta de *KUALI-BEH: Software Project Common Concepts*,[45] la cual se integró en la propuesta de ESSENCE 1.0 publicada como estándar de OMG.

5.5. Tendencias de la Ingeniería de Software

Algunos pioneros de la Ingeniería de Software han proyectado el futuro de esta disciplina, como Barry Boehm,[46] quien enunció las principales características que presentan actualmente los sistemas software y las que presentarán en el futuro. Boehm sostiene que la evolución de esta disciplina se caracterizará por un incremento considerable del tamaño, complejidad, diversidad en contenido y apertura a la interacción con otros sistemas. También augura que para 2020 habrá tendencias computacionales muy variadas, como nuevos tipos de plataformas inteligentes (materiales inteligentes, nanotecnología, dispositivos micro mecánico-eléctricos, componentes autónomos para sensado y comunicación -MEMS) y nuevos tipos de aplicaciones (redes de sensores, materiales configurables o adaptativos, adaptación de prótesis humanas) así como el desarrollo de la bioinformática.

[45] Morales, M., Oktaba, H., Piattini, M. (2015). **Making-Of an OMG Standard**. *Computer Standards & Interfaces Journal*, 42:84–94, Elsevier.

[46] Boehm, B. (2006). **A view of 20th and 21st century software engineering**. En *Proceeding of the 28th International Conference on Software Engineering,* pp. 12–29.

En el contexto de la bioinformática, los sistemas a construir serán complicados ya que incluirán:

1. Computación basada en la biología, la cual utiliza fenómenos biológicos o moleculares para resolver problemas computacionales más allá del alcance de la tecnología basada en el silicio.

2. Incremento de las capacidades físicas o mentales del humano basadas en la computación, con dispositivos quizás integrados o conectados a los órganos humanos o sirviendo como hospedaje de los cuerpos humanos (o partes de éste).

Todas estas tendencias computacionales, principalmente vistas como sistemas y aplicaciones a construir, presentan retos para la Ingeniería del Software tales como i) la especificación de configuraciones y comportamientos, ii) la generación de aplicaciones y sistemas con base en las especificaciones, iii) la verificación y validación de la capacidad, rendimiento y fiabilidad, y iv) la integración de los sistemas en otros aún más complejos (sistemas de sistemas).

La diversidad y complejidad de los sistemas que se desarrollarán en los próximos años impondrán grandes retos a la Ingeniería de Software. El análisis presentado por Boehm permite visualizar los campos a los que se deberá orientar la Ingeniería de Software, ya que tradicionalmente se ha enfocado más a temas específicos de la misma disciplina y de la Ciencia de la Computación.[47] La globalización de los sistemas es una realidad y hay al menos tres paradigmas emergentes que plantean grandes retos a la Ingeniería de Software, como sigue:

1. Computación en la nube ("Cloud Computing"): Éste es un tipo de computación basada en Internet para habilitar el acceso por demanda a recursos informáticos compartidos, configurables y ubicuos (por

[47] Wang, Z., Li, B., Ma, Y. (2106). **An Analysis of Research in Software Engineering: Assessment and Trends**. https://arxiv.org/ftp/arxiv/papers/1407/1407.4903.pdf.

ejemplo, redes de computadoras, servidores, almacenamiento, aplicaciones y servicios).[48] El reto es la identificación de la calidad de los servicios; sin embargo, las capacidades y herramientas para enfrentarlo son limitados y los problemas en la concepción de sistemas y generación de soluciones son complejos.

2. Computación social ("Social Computing"): Este tipo de computación se refiere a los sistemas que permiten obtener, representar, procesar, usar y difundir información que se distribuye a través de colectividades sociales, tales como equipos, comunidades, organizaciones y mercados electrónicos.[49] Estos sistemas deberán acceder a funciones móviles tales como correo electrónico, mensajes, conocimientos y soluciones de administración de contenido y tener acceso a las aplicaciones transaccionales y sistemas de información, lo que involucra considerar arquitecturas complejas.

3. Datos masivos ("Big Data"): El término "Datos Masivos" se refiere a conjuntos de datos complejos de grandes dimensiones cuyos requerimientos de procesamiento y almacenamiento superan las capacidades de las aplicaciones tradicionales.[50] Se requiere encontrar patrones repetitivos dentro de los datos. Los retos para el tratamiento de los datos masivos incluyen el análisis, captura, búsqueda, intercambio, almacenamiento, transferencia, visualización, consultas, minería, privacidad y actualización de la información.

---

[48] Mell, P., Grance, T. **The NIST Definition of Cloud Computing (Technical report)**. National Institute of Standards and Technology: U.S. Department of Commerce, September 2011.

[49] Doug, Schuler, D. (1994). **Social Computing, introduction to Social Computing.** *Special edition of the Communications of the ACM*, 7 (1):28 –108.

[50] Chris Snijders, Ch., Matzat, Reips, U. D**.** (2012). **Big Data: Big gaps of knowledge in the field of Internet**. *International Journal of Internet Science*. 7:1–5.

Con las tendencias antes citadas, seguramente dos temas propios de la Ingeniería de Software estarán involucrados:

1. Interfaces de usuario adaptativas. La diversidad actual de los paradigmas de computación, así como las capacidades de las tecnologías de la información y las redes de conectividad, junto con los datos recopilados por los sensores disponibles en los dispositivos inteligentes, posibilitan la creación de experiencias personalizadas e inmersivas para el usuario, así como rastrear las interacciones, registrarlas y analizarlas en tiempo real. Estas capacidades proporcionan una oportunidad para diseñar la adaptabilidad y ofrecer una mejor experiencia de usuario discreta y transparente. Todo sistema requiere una interfaz de usuario, la cual generalmente es gráfica. Una interfaz gráfica de usuario adaptativa (GUI, por sus siglas en Inglés) tiene el potencial de optimizar el desempeño y la satisfacción de usuario mediante la adecuación automática de la funcionalidad a las capacidades de cada usuario específico.[51] El desarrollo de software "Front-end" es tan importante como el software "Back-end", así que las compañías necesitan tener ingenieros de interfaces de software que puedan crear aplicaciones móviles y aplicaciones Web intuitivas y enfocadas al usuario. Ya sea para una aplicación empresarial o para usuarios individuales, los ingenieros de software, en conjunto con el diseñador de la interfaz, deben construir una experiencia satisfactoria para el usuario final. No sólo se requiere software funcional también intuitivo y fácil de utilizar.[52]

2. Administración de proyectos. Debido a la diversidad y complejidad de los sistemas y aplicaciones es necesaria la formación de equipos de trabajo, los cuales frecuentemente deben ser multidisciplinarios. Gestionar en forma adecuada un proyecto es un reto que seguirá presente en el campo de la Ingeniería de Software. Los principales problemas

[51] Findlater, L., Gajos, K. (2009**). Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces**. *AI MAGAZINE*, 30(4):68–73.

[52] Ver capítulo de Interacción Humano-Computadora en este texto.

que se tienen en la administración de proyectos actualmente son la planificación incompleta de proyectos, la estimación de costos de software y la baja precisión en los criterios de selección de las mejores técnicas de análisis, diseño y pruebas.

Aunado a estos problemas y retos se tiene la apertura cada vez más grande al trabajo distribuido en tiempo y en espacio. La globalización conduce a una fuerza laboral diversa en términos de lengua, cultura, resolución de problemas, filosofía de gestión, comunicación de las prioridades y de interacción persona a persona. Para cuestiones de gestión en un contexto distribuido se requiere contar con dos elementos vitales: conectividad y colaboración.[53] La conectividad se debe habilitar por comunicación de alto ancho de banda y la colaboración se debe dar entre equipos de software que no ocupan el mismo espacio físico y que requieren de medios electrónicos y del empleo a tiempo parcial en un contexto local y no solamente en tareas pequeñas de trabajo por "pares".

[53] Vijiyan, G. (2015). **Current Trends in Software Engineering Research**. En *Proc. 3rd International Conference on Emerging Trends in Scientific Research,* At Peal International Hotel, Kuala Lumpur, Malaysia, https://www.researchgate.net/publication/282059722_Current_Trends_in_Software_Engineering_Research

6. Interacción Humano-Computadora

6.1. Introducción

Los orígenes de la Interacción Humano-Computadora (IHC) pueden remontarse a la época de la posguerra, cuando Vannevar Bush introduce en el artículo *As we may think*[1] muchos de los conceptos que han inspirado investigaciones y desarrollos tecnológicos en el área, tales como hipertextos e hipermedios, interfaces gráficas, interfaces basadas en voz, ambientes de colaboración e interfaces naturales. En la década de los sesenta se produjeron avances importantes en la investigación y desarrollo de prototipos, así como sistemas de demostración de concepto que hoy son componentes fundamentales de sistemas interactivos. Los sistemas de ventanas, la videoconferencia, los hipertextos y el ratón como dispositivo de interacción se implementaron y presentaron por primera vez en 1968 por Douglas Engelbart. En los setenta, los investigadores de IHC produjeron las primeras interfaces gráficas de usuario, las cuales representaron un avance significativo para acercar las tecnologías de información a comunidades amplias de usuarios particularmente cuando, ya en los ochenta, fueron la base de computadoras personales disponibles comercialmente.

Hasta antes del surgimiento formal de IHC como disciplina, mucho del avance en computación se centra en el desarrollo de hardware. En su trabajo

[1] https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/

seminal Jonathan Grudin habla de cinco etapas en el desarrollo de interfaces de usuario.[2] En la primera, antes de IHC, se tiene como interfaz al hardware, cuyos principales usuarios son ingenieros especializados. En la segunda, con el nacimiento de los lenguajes de programación de alto nivel, las interfaces se centran en la tarea de programar eliminando la necesidad de conocer detalles del hardware. En la tercera, a través del monitor y el teclado el usuario se comunica por medio de comandos que debían ser comúnmente memorizados. En la cuarta aparecen los diálogos interactivos con los sistemas por medio de interfaces gráficas de usuario o *Graphical User Interfaces* (GUIs), cuyo diseño e implementación requiere a su vez del desarrollo de marcos teóricos que permitan entender la ejecución de tareas complejas. Finalmente, en la última etapa, se vislumbra una computadora que va más allá del individuo, que tiene impacto en lo grupal y donde la colaboración, diligencia, cargos o autoridad se tienen que considerar de forma explícita. En este sentido, de la mano de esta evolución, se tiene cada vez más una interrelación con otras disciplinas que van desde la ingeniería eléctrica y electrónica, las ciencias de la computación, los factores humanos, la psicología cognitiva, hasta la sociología, la antropología y la psicología social. Más allá de las etapas que plantea Grudin,[3] la evolución tecnológica y de investigación en el área se ha centrado en las interfaces naturales en las cuales los usuarios no necesariamente tienen que aprender a usar una computadora ya que dichas interfaces aprovechan las conductas naturales de los seres humanos.

IHC estudia el diseño, implementación y evaluación de sistemas interactivos en el contexto de las actividades del usuario. En esta disciplina el término *humano* no se refiere necesariamente a un individuo, sino que puede referirse a un grupo de individuos con un perfil determinado o trabajando de manera colectiva, en secuencia o en paralelo. A su vez, el término *computadora* se refiere a una amplia gama de sistemas que pueden ir desde una

[2] Grudin, J. (1990). **The computer reaches out: the historical continuity of interface design**. En *Proc. of the SIGCHI Conference on Human factors in Computing Systems (CHI'90),* ACM Press, pp. 261–268.

[3] Idem.

computadora de escritorio, un teléfono celular, un vehículo, un horno de microondas, una tostadora de pan, un sistema embebido, hasta sistemas que incluyan elementos no necesariamente computarizados como personas o procesos. Finalmente, el término *interacción* involucra todo lo que se relaciona con el diálogo entre el humano y la computadora, utilizando dispositivos de entrada y salida, ya sea de manera implícita o explícita. Por estas razones, cuando se habla de computadoras en IHC se habla en realidad de sistemas interactivos.

Un concepto central en IHC es el de "usabilidad". Éste se refiere a que el humano pueda llevar a cabo la actividad que desea realizar con la ayuda de la computadora. El sistema debe brindar soporte para que los seres humanos realicen satisfactoriamente diversas actividades; por ejemplo, de trabajo, entretenimiento o ejercicio. Para este efecto el sistema debe ser *útil* (hacer lo que se desea: escuchar música, cocinar), usable (fácil de manipular, fácil de aprender, sin errores) y usado de manera frecuente (aceptado, de manera que la gente desee usarlo porque es útil, divertido o atractivo). Por el contrario, un sistema no usable es aquel donde el humano es forzado por la computadora a realizar una actividad de manera diferente.

Los tres conceptos mencionados —humano, computadora e interacción— ayudan a entender el desarrollo de IHC así como su naturaleza transdisciplinaria; la estructura de este capítulo se basa en estos tres conceptos. Se parte de marcos teóricos que se han utilizado ampliamente para ayudar a conocer al humano. Asimismo, la evolución constante y vertiginosa de la computadora ha ido de la mano con el desarrollo del área. Finalmente, para tratar de tener una interacción adecuada se resalta la importancia del proceso de diseño de nuevas formas de la computadora, así como el proceso de evaluación como una retroalimentación al proceso de diseño. En cada subsección se presentan ejemplos de los proyectos más relevantes del área, así como del trabajo de la comunidad de IHC en México.

6.1.1. El humano

El humano tiene limitaciones naturales para el procesamiento de información que impactan en el diseño de sistemas interactivos (por ejemplo, un cirujano no puede controlar una cámara mientras realiza una operación). Estas limitaciones o características se consideran y a menudo se estudian para el diseño de productos en IHC. Además, hay factores inherentes a la condición humana, como el cansancio, el aburrimiento y el enojo, que también se deben tomar en cuenta. Esto es particularmente relevante cuando las computadoras se usan en condiciones extremas o críticas (por ejemplo, por un piloto de avión).

De igual manera la información que proviene de las computadoras y del ambiente se captura por diversos canales como el auditivo, el visual, el háptico y el movimiento, y se guarda de manera temporal o definitiva en nuestra memoria de corto o largo plazo. Todos los datos que recibimos a través de los canales mencionados se procesan de manera consciente o inconsciente y a través de mecanismos muy complejos, nos permiten resolver problemas, razonar, adquirir habilidades y conocimiento, y hasta cometer errores. De igual manera las emociones impactan en nuestras actividades mentales y/o físicas, e incluso, si son muy fuertes, pueden llevar nuestras capacidades al límite. Aún cuando los humanos compartimos diferentes habilidades y capacidades las diferencias individuales se deben considerar para diseñar productos usables.

6.1.2. La computadora

La evolución de la computadora ha ido de la mano de la competencia entre compañías productoras de hardware. Sin embargo, cuando surge la computación personal en los años ochenta, el software se convierte en un diferenciador para las compañías que contaban con interfaces gráficas de usuario, como Apple, Microsoft y Xerox; por estas mismas razones la IHC adquiere una mayor relevancia.

Los dispositivos clásicos de entrada, como el teclado y el ratón, y de salida, como el monitor o proyector, el audio y la impresora, fueron objeto de intenso estudio en los ochenta, pero desde entonces ha habido innovaciones notables, como el reconocimiento del habla, las pantallas táctiles, las plumas digitales y, recientemente, el auge de los sensores o bio-señales en el caso de los dispositivos de entrada, y las pantallas públicas y el papel digital en los de salida. De igual forma, también ha habido un desarrollo importante, aunque en menor grado, en otro tipo de dispositivos de entrada/salida, como los controles físicos, la realidad virtual y aumentada, así como la retroalimentación háptica, olfativa y algunos actuadores.

En resumen, la computadora ha impactado significativamente la vida profesional y cotidiana tanto que algunas actividades comunes serían impensables hoy en día sin este artefacto tecnológico.

6.1.3. La interacción

La interacción se entiende como un diálogo entre la computadora y el humano. Debido a la gran diversidad de personas y de contextos en los que se usan las computadoras, continuamente se proponen métodos y técnicas para entender mejor cómo es o cómo debería ser este proceso. El diseño de computadoras es un proceso inherente al IHC en el que intervienen diversos factores como: i) las personas para las que se diseña (por ejemplo, sus habilidades, capacidades y limitaciones); ii) la actividad que se desempeñará con la computadora (por ejemplo, una cirugía a corazón abierto); y iii) el contexto en el cual se realizará la actividad (por ejemplo, un quirófano, sentado en la sala o en una oficina o un vehículo de carreras). Generalmente, la interacción se da en un lugar donde los aspectos sociales y el contexto organizacional tienen un efecto importante tanto en la persona (humano) como en el sistema (computadora).

De igual manera, una vez que se ha diseñado un sistema interactivo, se tiene que verificar que efectivamente el producto cumple con su propósito.

Por ejemplo, si el sistema es un software para aprendizaje de matemáticas se tiene que evaluar si éste permite aprender igual o mejor que con el apoyo de un profesor tradicional o con el de otro software diseñado para el mismo propósito. La evaluación no sólo abarca la efectividad del sistema interactivo sino también otros aspectos como eficiencia, eficacia, satisfacción al momento de usarlo e incluso su adopción final.

6.2. Modelos cognitivos y de interacción

La IHC no cuenta con una teoría unificada para describir, entender y predecir acciones del humano, por lo que adopta teorías y modelos de la psicología, la sociología y la antropología, entre otras; incluso no hay certeza de que sea posible establecer una teoría general de la IHC debido a la naturaleza compleja y diversa del área. No obstante, mucho del esfuerzo que se ha realizado en esta disciplina en términos de modelos y teorías se enfoca al estudio de las interacciones entre el humano y la computadora, las cuales, a la postre, sirven para el diseño de computadoras adecuadas para los humanos.

La tendencia en los años ochenta fue tratar de modelar de manera cognitiva el desempeño de una persona al utilizar una computadora para posteriormente optimizarlo mediante mejores diseños. Dichos modelos, denominados genéricamente "Modelo Humano Procesador" tenían como objetivo ayudar a los desarrolladores de sistemas a aplicar principios de psicología cognitiva. Estos modelos también se utilizaron para evaluar la usabilidad de un producto.

La evolución de estos modelos se facilitó en parte por desarrollos similares en áreas de ingeniería y diseño, muy cercanas y a menudo traslapadas con la IHC, como la ingeniería del factor humano y el desarrollo de documentación.[4] La primera había desarrollado algunas técnicas empíricas de análisis de tareas para medir las interacciones entre el humano y algunos

[4] Carroll, J. M., **Human Computer Interaction - Brief intro**. https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro

sistemas, notablemente en dominios como la aviación y la manufactura, y se extendía al área de sistemas interactivos en los que los operadores humanos realizaban labores de resolución de problemas cotidianamente. Por otro lado, el desarrollo de documentación se movía más allá de su papel tradicional de producir descripciones técnicas hacia un enfoque más cognitivo en el que se incorporaban teorías de escritura, lectura y medios, con evaluación empírica con usuarios, ya que los documentos escritos y otros medios también necesitaban ser usables.

6.2.1. Modelos de comportamiento motor

Los modelos de comportamiento motor se inspiran en las capacidades, limitaciones y potencial del cuerpo humano, especialmente en la armonía entre estas características humanas y los diversos dispositivos de entrada/salida. Es conveniente imaginar este tipo de modelos como un continuo que va de las analogías y metáforas a los modelos matemáticos. La mayoría de los modelos se ubican en algún punto intermedio, donde los modelos descriptivos se cargan hacia el lado de las metáforas y los modelos predictivos hacia el de las ecuaciones matemáticas, como se ilustra en la Figura 6.1.

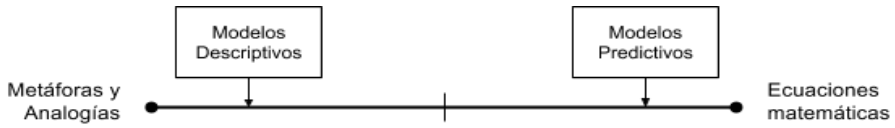


Figura 6.1. Tipos de modelos basados en el comportamiento motor del humano

Los modelos descriptivos proveen un marco teórico para caracterizar un contexto o un problema. Generalmente consisten en una serie de categorías interrelacionadas de manera gráfica que sirven al diseñador como una guía para la creación de sistemas computacionales adecuados para la interacción

del usuario con el sistema. Por ejemplo el *Key-Action Model* (KAM),[5] o Modelo Tecla-Acción, describe al teclado como un conjunto de teclas que pertenecen a tres categorías: simbólicas, ejecutivas y modificadoras. Las teclas simbólicas envían un símbolo a la pantalla, como letras, números o símbolos de puntuación; las ejecutivas indican una acción para el sistema computacional o a nivel del sistema operativo, como F1 o ESC; finalmente, las teclas modificadoras cambian el comportamiento de otras teclas como SHIFT o ALT. La utilidad de este modelo es que a pesar de su sencillez permite pensar cómo sería un teclado con una forma diferente, tomando en consideración tales categorías. Existen otros modelos como el Modelo de los 3 Estados para Entradas Gráficas de Bill Buxton,[6] en el que se describen las transiciones de estado de los dispositivos que apuntan, como el ratón.

Por otra parte, los modelos predictivos tienen un corte ingenieril y se usan en una gran diversidad de disciplinas. Una de sus ventajas es que permiten determinar analíticamente algunas métricas de rendimiento del humano sin la necesidad de recolectar datos empíricos, lo cual suele ser costoso en tiempo y dinero. Uno de los más populares consiste en la aplicación de la ley de Hick-Hyman para estimar el tiempo de reacción al elegir opciones. Este modelo tiene la forma de una ecuación: dado un conjunto de estímulos, cada uno asociado con respuestas, el tiempo de reacción (*RT*) entre el estímulo y la respuesta está dado por:

$$RT = a + b \log_2(n)$$

donde a y b son constantes que se obtienen empíricamente. Este modelo se ha utilizado, por ejemplo, para estimar el tiempo que le toma a una operadora telefónica seleccionar entre 10 botones después de que se enciende una luz

[5] Carroll, J. M. (Ed.). **HCI models, theories, and frameworks: Toward a multidisciplinary science**. Morgan Kaufmann, 2013.

[6] Buxton, W. (1990). **A Three-State Model of Graphical Input**. En D. Diaper et al. (Eds), *Human-Computer Interaction - INTERACT '90*. Amsterdam: Elsevier Science Publishers B.V. (North-Holland), pp. 449-456.

detrás de uno de ellos.[7] La Ley de Hick-Hyman se ha utilizado también para predecir el tiempo que toma seleccionar elementos en un menú jerárquico.[8]

Otros modelos predictivos han surgido específicamente en IHC como el modelo *Keystroke-Level Model* (KLM)[9] que tiene por objetivo predecir el tiempo que toma ejecutar una tarea en un sistema computacional; particularmente el tiempo para completar las tareas realizadas por expertos sin considerar errores, dados los siguientes parámetros:

- Tareas o una serie de sub-tareas
- Método utilizado
- Lenguaje de comandos del sistema
- Parámetros motor-habilidad del usuario
- Parámetros tiempo-respuesta del sistema

Una predicción KLM es la suma de los tiempos de las sub-tareas y el tiempo en general (overhead). Este modelo incluye cuatro operadores de control-motor (*K* = Key stroking, *P* = Pointing, *H* = Homing, *D* = Drawing), un operador Mental (*M*) y un operador Respuesta-del-sistema (*R*):

$$T_{EXECUTE} = t_K + t_P + t_H + t_D + t_M + t_R$$

Algunas de las operaciones se pueden omitir o repetir dependiendo de la tarea. Por ejemplo, si una tarea requiere presionar el teclado *n* veces se convierte en $n \times t_K$. A cada operación t_K se le asigna un valor de acuerdo con la habilidad del usuario, con valores que van desde $t_K = 0.08$ para los que

[7] Card, S. K., Moran, T. P., Newell. A. **The psychology of human-computer interaction**. Hillsdale, NJ: Erlbaum, 1983.

[8] Landauer, T. K., Nachbar, D. W. (1985). **Selection from alphabetic and numeric menu trees using a touch screen: Breadth, depth, and width**. *Proceedings of the ACM Conference on Human Factors in Computing Systems—CHI '85*, New York: ACM Press, pp.73–77.

[9] Card, S. K., Moran, T. P., Newell, A. (1980). **The Keystroke-Level Model for user performance time with interactive systems**. *Communications of the ACM*, 23:396–410.

son muy hábiles hasta $t_K = 1.20$ para quienes no lo son o utilizan un teclado que no les es familiar. Desde su introducción este modelo se ha utilizado en diversos contextos en IHC para predecir el rendimiento de usuarios con menús jerárquicos[10] o el rendimiento de personas con discapacidades físicas al ingresar un texto.[11]

6.2.2. Modelos de procesamiento de información

Aún cuando los modelos basados en comportamiento motor fueron exitosos, a medida que aumentó la complejidad de las interfaces, se requerieron modelos que tomaran en cuenta las interacciones entre humanos y computadoras de manera integral y no solamente en interacciones discretas. De igual forma, se requería que los modelos se centraran en el contenido de los monitores o pantallas, más allá de la manera en que estaban organizadas. En este contexto surgieron otros modelos basados en procesamiento de información que, tomando como analogía un programa de computadora, se describen en términos de mecanismos locales, pero que tomados en conjunto realizan comportamientos de alto nivel.

En la Figura 6.2 se ilustra un modelo generalizado del humano como procesador de información. El Procesador en el centro del diagrama recibe información de los Receptores y controla a los Efectores. Asimismo, envía y recibe información de la Memoria. Con el paso de los años, esta manera de ver la interacción entre humanos y computadoras llevó a la creación de modelos que analizan tareas como *Goals*, *Operators*, *Methods* y *Selection rules* (GOMS). Estos modelos son importantes para áreas orientadas a la ingeniería de

[10] Lane, D. M., Napier, H. A., Batsell, R. R., Naman, J. L. (1993). **Predicting the skilled use of hierarchical menus with the keystroke-level model**. *Human-Computer Interaction*, 8(2):185–192.

[11] Koester, H., Levine, S. P. (1994). **Validation of a keystroke-level model for a text entry system used by people with disabilities**. *Proceedings of the First ACM Conference on Assistive Technologies*. New York: ACM Press, pp. 115–122.

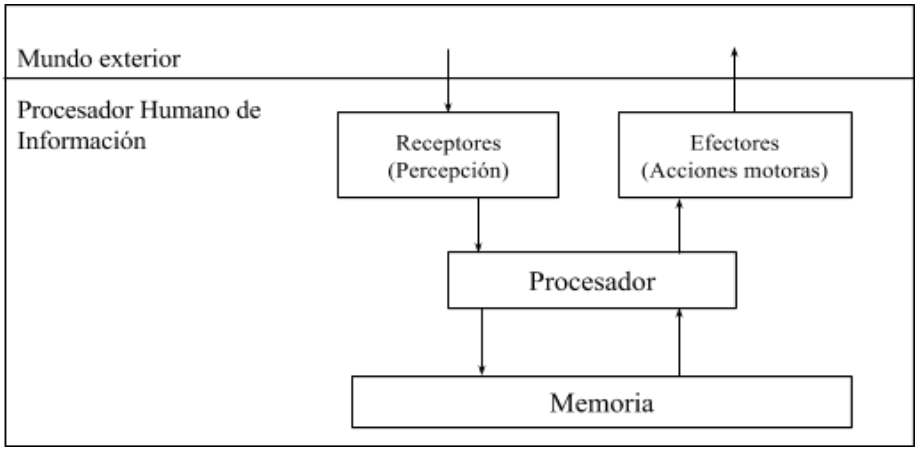


Figura 6.2. Representación esquemática del humano como sistema de procesamiento de información.[12]

software, en la que se pueden utilizar para el diseño de productos sin necesidad de realizar estudios empíricos que muchas veces resultan ser costosos. Aún cuando los modelos tipo GOMS no permitían predecir comportamientos muy complejos, sí fueron exitosos para predecir tareas muy específicas; por ejemplo, la velocidad de una persona para ingresar datos a través del teclado en diferentes teclados.

En particular, GOMS es un modelo cognitivo en el que se utiliza una estrategia de análisis basada en dividir las metas del usuario en sub-metas de manera recurrente hasta llegar a las metas que se pueden realizar directamente.[13] Por ejemplo, para realizar un reporte de ventas del libro "Interacción Humano-Computadora en México" se puede dividir la tarea en varias sub-metas, como recolectar los datos, analizarlos, producir gráficas adecuadas como histogramas y escribir el reporte. Asimismo, la sub-meta recolectar datos se

[12] Carroll, J. M. (Ed.). **HCI models, theories, and frameworks: Toward a multidisciplinary science**. Morgan Kaufmann, 2003.

[13] Card, S., Moran, T., Newell, A. **The model human processor- An engineering model of human performance**. Handbook of perception and human performance, Vol. 2, pp. 45-1, 1986.

puede dividir en varias sub-metas como contactar editorial, solicitar información de volúmenes vendidos, etc. Las metas se pueden subdividir hasta un nivel muy bajo, como mover las manos o los ojos, por lo que es importante tener en consideración el nivel de descripción apropiado para la tarea. GOMS consiste de cuatro elementos básicos:

- Metas *(Goals):* Describen qué es lo que el usuario quiere realizar. Deben representar un "punto en la memoria" del usuario en el que se puede analizar qué tiene que realizar y qué tarea se puede retomar en caso de que algo salga mal.

- Operadores *(Operators):* Constituyen el nivel de análisis más básico y consisten en las acciones concretas que debe realizar un usuario para operar un sistema. Hay mucha flexibilidad en cuanto al nivel que se requiere, y pueden incorporar acciones que afecten al sistema (e.j. presionar la tecla *Enter*) o al estado mental del usuario (e.j., leer una ventana de diálogo en la que se indica el error).

- Métodos *(Methods):* Constituyen las diferentes maneras como se puede realizar una meta. Por ejemplo, para cerrar la ventana actual el usuario puede i) seleccionar la *X* en la esquina superior derecha; ii) mostrar el menú emergente y seleccionar la opción *Cerrar Ventana* o iii) presionar la combinación de teclas ALT+F4. En GOMS estos métodos se pueden representar como el METODO-X, *METODO-ME*, o *METODO-F4*.

- Reglas de selección *(Selection Rules):* Estas reglas tienen por objetivo predecir cuál o cuáles métodos serán empleados por los usuarios, lo cual depende en muchos casos del mismo usuario, así como del estado del sistema. En el siguiente ejemplo se tienen tres métodos diferentes para llevar a cabo la misma Meta.

META: *CERRAR VENTANA*
[SELECCIONAR
META: *USAR-METODO-X*
MOVER CURSOR HACIA LA ESQUINA SUPERIOR
CLIC EN ICONO X
META: *USAR-METODO-ME*
MOVER CURSOR HACIA ENCABEZADO DE VENTANA
CLIC DERECHO EN RATON
SELECCIONAR OPCION CERRAR VENTANA
META: *USAR-METODO-F4*
PRESIONAR TECLA F4]

El método GOMS ha servido de base para muchos otros y ha sido sustento importante para métodos que se han dedicado al análisis de tareas rutinarias por parte de los usuarios.

Los métodos ejemplificados ilustran la manera en que se puede analizar sistemáticamente una serie de tareas que componen a una actividad humana. De esta manera es posible tomar en cuenta las características particulares de cada tarea para que un sistema interactivo le brinde el soporte adecuado. Es decir, el diseño de un sistema interactivo puede tomar como base las características no sólo de la tarea sino las particularidades del ser humano permitiendo que la persona sea efectiva al realizar la tarea al interactuar con dicho sistema.

6.3. Diseño centrado en el humano

El Diseño Centrado en el Humano (DCH) tiene por objetivo diseñar productos interactivos que sean fáciles de usar, efectivos en su uso y con una experiencia de uso que se disfrute, así como optimizar las interacciones de un usuario con un sistema y su ambiente o producto. El DCH se enfoca en entender el espacio del problema para proponer tecnología innovadora

y potencialmente disruptiva.[14] En contraste con la Ingeniería de Software,[15] donde los requerimientos de un programa de software se obtienen mediante entrevistas con el usuario y se especifican en un documento con carácter de contrato entre el desarrollador y el cliente, el DCH enfatiza involucrar a los usuarios potenciales en el proceso de diseño para ayudarlos a establecer requerimientos que de otra manera serían muy difíciles de identificar.

A menudo los usuarios no saben lo que quieren y les es difícil concebir un sistema innovador que facilite sus tareas. Por lo mismo el DCH apoya a los diseñadores a entender mejor las necesidades y problemas de los usuarios y a establecer sus requerimientos reflexionando sobre las estrategias actuales (las prácticas y herramientas del usuario en el espacio del problema). De manera formal el DCH se define como:

- El diseño de productos interactivos que apoyan la manera en que las personas se comunican e interactúan en su vida diaria.[16]

- El diseño de espacios para la comunicación e interacción humana.[17]

Sin embargo, diseñar productos usables y con una buena experiencia de usuario no es tárea fácil. Por ejemplo, los buzones de voz 01 800 de atención a cliente son generalmente ineficientes y frustrantes. Más aún, con frecuencia se encuentran diseños que a pesar de ser útiles presentan otro tipo de problemas y que, desafortunadamente, se utilizan en la vida cotidiana. En este rubro hay diseños útiles pero socialmente inaceptables como los llamados "chindogu" en Japón.[18] Estos diseños existen principalmente porque los diseñadores de

[14] Rogers, Y. (2011). **Interaction design gone wild: striving for wild theory**. *Interactions* 18, 4:58–62.

[15] Ver Capítulo 7 de Ingeniería de Software

[16] Preece, J., Sharp, H., Rogers, Y., (2001). **Interaction design: Beyond human–computer interaction**. *Univers. Access Inf. Soc.* 3, 3:289–289.

[17] Winograd, T. (1997) **From computing machinery to interaction design**. En P. Denning and R. Metcalfe (Eds.) Beyond Calculation: the Next Fifty Years of Computing. Amsterdam: Springer-Verlag, pp. 149-162.

[18] Como los mamelucos con cerdas inspirados en el gateo diseñandos en Japón, que le

sistemas no se preocupan por las interacciones ni por la interfaz de usuario, sino que se centran en la funcionalidad del sistema y los algoritmos necesarios para la optimización de los recursos computacionales.

Los diseños de sistemas interactivos de baja calidad que se utilizan en la actualidad son frustrantes, confusos e ineficientes, pero un mal diseño puede tener consecuencias aún más graves. Por ejemplo, existen reportes de muertes a raíz del mal uso de un equipo de radiación que utilizaba como entrada una serie de comandos complejos y confusos: "un mal diseño te puede matar".[19] La forma de interacción basada en escribir comandos desde una consola provocó que la computadora se percibiera como un dispositivo difícil de operar e incluso dio lugar a una cultura de los "gurús" que memorizaban la mayoría de los comandos de un sistema operativo. Sin embargo, en los últimos años se han propuesto nuevos dispositivos que buscan imitar cómo los humanos interactúan con el mundo real utilizando interfaces naturales.

A pesar de esto, el modelo de interacción más popular y al que estamos más acostumbrados continúa siendo el ratón-teclado-monitor. Sin embargo, este modelo frecuentemente inhibe nuestras capacidades de interacción innatas.[20] Por ejemplo, el ratón es un dispositivo que provee solo 2 grados de libertad, lo cual resulta marginal si se compara con los 23 grados que tenemos en nuestros dedos. A pesar de que el ratón fue una invención revolucionaria y de que es un buen dispositivo de entrada no es el más natural. Aunque muchos lo encuentran fácil de usar, requiere de aprendizaje y muchas personas se sienten desorientadas cuando lo utilizan por primera vez —principalmente niños, adultos mayores o individuos con capacidades diferentes y con el sistema motor comprometido. Es por ello que una área importante de estudio en IHC consiste en entender el espacio de diseño de diferentes productos y proponer nuevos diseños potencialmente disruptivos pero útiles y con una buena

permiten a un bebé trapear el piso mientras gatea http://www.chindogu.com/

[19] https://www.nngroup.com/articles/medical-usability/; http://www.nbcnews.com/id/28655104/

[20] Malizia, A., Bellucci, A. (2012). **The artificiality of natural user interfaces**. *Communication of the ACM* 55(3):36–38.

experiencia de uso. La investigación en México se ha abocado a entender el espacio de diseño de sistemas interactivos en contextos específicos: adultos mayores,[21] personal hospitalario,[22] comunidades rurales,[23] trabajadores de la información[24] y niños con autismo.[25]

Por estas razones en la actualidad los diseñadores de sistemas se preocupan por la interfaz y la interacción permitiendo la evolución de la IHC. Como consecuencia, el DCH se ha convertido en un gran negocio. En particular, los consultores de diseño de sistemas, compañías *start up* de computación y la industria de cómputo móvil se han dado cuenta del papel crucial que el DCH juega en el desarrollo de sistemas. La interacción ha logrado ser un excelente diferenciador para destacar en un campo altamente competitivo. Un ejemplo claro fue la aparición del *iPod* cuya forma de interacción novedosa e intuitiva logró eliminar a su competencia del mercado.[26] Esto subraya que la interacción debe de ser el centro del diseño de sistemas y no un aspecto secundario. El poder decir que el producto es "fácil de utilizar, efectivo en su uso y con una experiencia de uso que se disfrute" se ha convertido en el slogan oficial de las compañías de desarrollo de sistemas.

---

[21] Navarro, R. N., Rodríguez, M. D., Favela, J. (2016). **Use and Adoption of an Assisted Cognition System to Support Therapies for People with Dementia**. *Comp. Math. Methods in Medicine*. doi: 10.1155/2016/1075191

[22] Muñoz, M.A., Rodríguez, M., Favela, J., Martinez-Garcia, A. I., González, V.M. (2003). **Context-Aware Mobile Communication in Hospitals**. *Computer* 36(9):38–46.

[23] Moreno, M. A., Martínez, C.A. (2014). **Designing for sustainable development in a remote Mexican community**. *Interactions* 21:76-79.

[24] González, V.M., Mark, G. 2004. **Constant, constant, multi-tasking craziness": managing multiple working spheres**. En *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '04). ACM, New York, NY, USA, pp. 113-120.

[25] Tentori, M., Escobedo, L., Balderas, G. (2015). **A Smart Environment for Children with Autism**. *IEEE Pervasive Computing* 4(2):42-50.

[26] http://gizmodo.com/5671670/sony-kills-the-cassette-walkman-on-the-ipods-birthday

6.3.1. Principios básicos de diseño de interacción

La literatura en DCH ha propuesto abstracciones acerca de diferentes aspectos del diseño que se conocen como "principios de diseño". Éstos funcionan como guías de lo que se debe y no se debe hacer al diseñar un sistema. Los principios de diseño se derivan de una mezcla de teorías basadas en conocimiento, experiencia de uso y sentido común. Si bien existen muchos principios específicos para una población en particular, de manera general, los principios básicos del diseño de interacción son los siguientes: [27]

- **Visibilidad.** Hacer visibles las interacciones de los humanos con la computadora en la medida de lo posible. Mientras más visibles sean las funciones, más probable será que los humanos realicen la acción apropiada. En contraste, cuando las funciones estén fuera del campo visual más difícil será imaginar cómo se puede utilizar el producto. La investigación en DCH en México se ha enfocado en proponer una visibilidad adecuada al diseñar interfaces para la comprensión de grandes volúmenes de información[28] así como para videojuegos basados en movimiento, kioskos y superficies interactivas.[29] Por ejemplo, FroggyBobby[30] es un videojuego serio basado en movimiento donde los niños utilizan sus brazos para controlar la lengua de una rana y ayudarle a comer moscas. El juego utiliza instrucciones claras y cortas, y mini-películas que funcionan como una especie de tutorial. Además, la interfaz del juego muestra dos boto-

[27] Norman, D. A. **The design of everyday things: Revised and expanded edition,** New York; Basic Books; London: MIT Press (British Isles only), 2013.

[28] Sánchez, J. A. (2013). **Understanding collections and their implicit structures through information visualization.** En *Innovative Approaches of Data Visualization and Visual Analytics.* Huang, M. L., Huang, W. (eds.). Information Science Reference, 151-175.

[29] http://www.edis.mx/

[30] Caro, K., Tentori, M, Martinez-Garcia, A. I., Zavala-Ibarra, I. (2015) **FroggyBobby: An exergame to support children with motor problems practicing motor coordination exercises during therapeutic interventions**. *Computers in Human Behavior.* doi: 10.1016/j.chb.2015.05.055

nes que indican dónde inicia y termina el movimiento, y el patrón de vuelo de las moscas les proporciona a los niños una guía visual del tipo de movimiento que el niño debe de practicar. También se han realizado estudios desde una perspectiva teórica para clasificar la experiencia de usuario en el diseño de la interacción basada en movimiento.[31]

- **Retroalimentación.** Proporcionar al usuario información inmediata acerca de la acción que se está ejecutando o que se acaba de ejecutar. En DCH existen diferentes tipos de retroalimentación que involucran el uso de sonidos, animaciones, vibraciones y combinaciones de dichos estímulos sensoriales. La retroalimentación adecuada puede también proporcionar una buena visibilidad del producto. Para apreciar la importancia de este principio se puede imaginar tratar de partir un pan utilizando un cuchillo sin ver cómo se corta o escribir utilizando una pluma sin ver el papel. Las primeras interfaces de usuario desarrolladas en México enfocadas a proponer una buena retroalimentación involucraron el diseño de sistemas colaborativos de acuerdo a la filosofía de "lo que yo veo es lo que tú ves" en especial para la edición colaborativa[32] de documentos o la programación en pares.[33] Recientemente se ha explorado el uso de otros estímulos sensoriales como la háptica para proporcionar retroalimentación vibro táctil durante las terapias de rehabilitación.[34]

[31] Cruz Mendoza, R., Bianchi-Berthouze, N., Romero, P., Casillas Lavín, G. (2015). **A classification of user experience frameworks for movement-based interaction design.** *The Design Journal* 18:393-420. doi: 10.1080/14606925.2015.1059606

[32] Morán, A. L., Favela, J., Martínez, A. M., Decouchant, D. (2001). **Document Presence Notification Services for Collaborative Writing**. *CRIWG 2001*, pp. 125-135.

[33] Vizcaíno, A., Contreras-Castillo, J., Favela, J., Prieto, M. (2000). **An Adaptive, Collaborative Environment to Develop Good Habits in Programming**. *Intelligent Tutoring Systems* 2000:262-271.

[34] Ramírez-Fernández C. et al. (2016) **GoodVybesConnect: A Real-Time Haptic Enhanced Tele-Rehabilitation System for Massage Therapy**. En García C., Caballero-Gil P., Burmester M., Quesada-Arencibia A. (eds) *Ubiquitous Computing and Ambient Intelligence. (UCAmI 2016)*. Lecture Notes in Computer Science, Vol 10069. Springer.

- **Restricciones.** Limitar los caminos o las opciones que los usuarios pueden elegir al ejecutar una acción. Ejemplo, sombrear opciones del menú que no se permiten al utilizar un procesador de texto. Proveer restricciones adecuadas previene la selección de opciones incorrectas y reduce la posibilidad de cometer errores. En México este principio de diseño se ha explorado mayormente para el diseño de objetos tangibles[35] y prótesis de brazos,[36] y ha permitido la organización de recursos computacionales en términos de actividades. Por ejemplo, el Malabarista de Esferas[37] es un sistema que permite asociar documentos, aplicaciones y contactos a una esfera de trabajo que representa una actividad. De esta manera todos los recursos digitales se restringen a lo que es relevante a la esfera de trabajo que el usuario seleccione.

- **Consistencia**. Utilizar operaciones o elementos similares para tareas similares. Por ejemplo, el uso de *shortcuts* del teclado como ctrl+C o ctrl+Z independientemente del sistema operativo. La consistencia permite que los sistemas sean más fáciles de utilizar porque los usuarios deben de aprender un sólo camino aplicable a varios objetos. En México, se ha explorado el concepto de consistencia al diseñar sistemas de sensado para la recolección de datos de comportamiento utilizando teléfonos celulares.[38] Por ejemplo, InCense[39] es una herra-

[35] Escobedo, L., Ibarra, C., Hernandez, J., Alvelais, M., and Tentori, M. (2013) **Smart objects to support the discrimination training of children with autism**. *Personal and Ubiquitous Computing* (PUC), 18(6):1485-1497.

[36] Cruz, K., Cornejo, R., Martinez, F.. **Human-Computer Interaction: Anthropomorphic prosthetics using EMG signals**. *Memorias extendidas de MexIHC 2016*. Colima, Mexico. Septiembre 2016.

[37] Morteo, R., González, V. M., Favela, J., Mark, G., **Sphere Juggler: Fast Context Retrieval in Support of Working Spheres**. *ENC 2004*, pp. 361–367.

[38] Castro, L. A., Favela, J., Quintana, E., Pérez, M. **Behavioral data gathering for assessing functional status and health in older adults using mobile phones.** *Personal and Ubiquitous Computing* 19(2): 379–391 (2015)

[39] Perez, M., Castro, L., Favela, J. (2011). **InCense: A research kit to facilitate behavioral data gathering from populations of mobile phone users.** En *5th International Symposium of Ubiquitous Computing and Ambient Intelligence* (UCAml 2011), Riviera Maya, Mexico. Dec. pp 5-9.

mienta que permite a los usuarios con bajas habilidades técnicas diseñar campañas de sensado a través de una interfaz que incluye los sensores disponibles en un celular, como el GPS, el acelerómetro, los actuadores y las encuestas. Estos elementos se representan mediante íconos que son consistentes con la nomenclatura que se utiliza en los diagramas de flujo; además se utilizan "estándares" para los algoritmos y mecanismos de almacenamiento de datos para facilitar su integración con otras herramientas de sensado como Funf (Behav.io).[40]

- **Asequibilidad.**[41] Indicar o dar pistas acerca de las acciones que se pueden realizar sobre un objeto. Por ejemplo, un ícono debe de invitar a presionarlo, una barra de desplazamiento (*scroll*) debe invitar a moverla hacia arriba o hacia abajo y los botones deben invitar a presionarlos. Este principio de diseño se ha utilizado mayormente en el diseño de objetos físicos ya que el mapeo es natural e involucra el uso de metáforas basadas en interacciones reales. Por ejemplo, para persuadir a los trabajadores de la información a llevar una vida menos sedentaria se diseñó un ratón, inspirado en el mecanismo de defensa del puercoespín, que saca picos de manera gradual hasta imposibilitar que el usuario continúe trabajando y tenga forsozamente que levantarse. Pocas investigaciones en México e incluso en el extranjero han estudiado este concepto debido, principalmente, a que encontrar características únicas de los objetos no es fácil y generalmente las metáforas del mundo real son poco mapeables a servicios digitales.

6.3.2. Proceso de diseño de interacción

El proceso de DCH es altamente empírico lo cual permite a los diseñadores tomar decisiones basadas en el conocimiento que se tiene de los usuarios y del contexto en el que se utilizará el producto. Durante este proceso se debe escuchar qué quieren los usuarios, tomar en cuenta sus habilidades y consi-

[40] http://www.funf.org
[41] Traducción al español del concepto de *affordances*

derar qué los puede ayudar a mejorar la forma como realizan sus tareas. De manera general, el proceso de diseño involucra cuatro actividades.

1. *Identificar las necesidades y establecer los requerimientos para la experiencia de usuario.* Esto se realiza mediante estudios empíricos que involucran entrevistas con los usuarios, observar sus interacciones en su práctica diaria y aplicar encuestas para verificar la representatividad de los datos. La información recabada se representa en modelos conceptuales y en narrativas que describen las necesidades, estrategias y metas de los usuarios potenciales.

2. *Desarrollar diseños alternativos que satisfagan los requerimientos*. Se proponen alternativas de diseño tomando en cuenta los datos empíricos. Estas ideas iniciales se plasman generalmente en bosquejos en papel para discutirse con los usuarios potenciales. Durante esta etapa los diseñadores y los usuarios reflexionan sobre las ventajas y desventajas de cada alternativa y seleccionan la idea que mejor satisfaga sus necesidades.

3. *Construir versiones interactivas de los diseños para ser comunicados y evaluados.* Se especifica el diseño mediante la creación de escenarios de uso que muestren cómo el producto se utilizará en la práctica y se construyen prototipos a diferentes niveles de fidelidad, los cuales permiten al usuario final "interactuar" con diferentes versiones del diseño e imaginar su uso en la práctica y en escenarios concretos.

4. *Evaluar el prototipo a través del proceso y la experiencia de usuario.* Finalmente, se evalúa la usabilidad y experiencia de uso del prototipo. Generalmente se utilizan técnicas cualitativas para realizar estudios exploratorios evaluados en el campo o técnicas cuantitativas que involucran la realización de experimentos en laboratorios de usabilidad, como se describe más adelante en la sección 6.3.4.

Existen diferentes ciclos de vida que indican el orden de estas actividades y cómo se relacionan unas con otras. Los más comúnmente utilizados son dirigidos por modelos conceptuales y son altamente iterativos o secuenciales. Por ejemplo, el modelo simple de DCH consta de las cuatro actividades antes mencionadas, las cuales se pueden visualizar como los nodos de un grafo completo cuyo estado inicial es la actividad *1* (establecer los requerimientos). Las actividades se realizan iterativamente tantas veces como sea necesario. En contraste, el diseño contextual rápido[42] propone pasos que se realizan de manera secuencial. Los primeros cuatro ayudan al diseñador a entender mejor el espacio del problema, las necesidades de los usuarios finales y el contexto de uso del sistema, mientras que los últimos tres involucran las propuestas de alternativas de diseño, su especificación mediante escenarios y el desarrollo de prototipos a diferentes niveles de fidelidad que se discuten con los usuarios potenciales (*2*, *3* y *4*). Los investigadores mexicanos siguen estos ciclos pero, además, en nuestro país se propuso una metodología conocida como "de la guitarra",[43,44] en la que cada ciclo de vida se dirige por escenarios de uso y por una comprensión inicial de la literatura.

Una vez que se tiene el diseño del sistema interactivo se busca identificar la tecnología más adecuada para realizarlo. Para este efecto se han propuesto diferentes tecnologías y modelos de interacción que facilitan la implementación de prototipos robustos. A continuación se describen los avances de IHC en esta dirección.

---

[42] Holtzblatt, K.; Wendell, J., Woods, S. **Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design**, Morgan-Kaufmann, San Francisco, 2005.

[43] Muñoz, M. A., González, V.M., Rodríguez, M., Favela, J. **Supporting Context-Aware Collaboration in a Hospital: An Ethnographic Informed Design**. *CRIWG 2003*: 330-344.

[44] Martínez-García, A. I., Tentori, M., Rodriguez, M. (2015) **Aplicaciones Interactivas para Salud**. En, Muñoz Arteaga, J., González-Calleros, M., Sánchez, A. (Eds). La Interacción Humano-Computadora en México, Pearson Educación, México.

6.3.3. Tecnologías y modelos de interacción

Identificar cuál es el modelo de interacción apropiado del producto o sistema involucra visualizar su funcionalidad tomando en cuenta las necesidades y habilidades de los usuarios, el contexto de uso y los requerimientos del producto. Para lograrlo es necesario tomar decisiones sobre:[45,46]

- **El modo de interacción**. Determina las actividades de interacción que el usuario podrá realizar para comunicarse con el sistema y viceversa. Los principales modos de interacción son i) instruir, ii) conversar, iii) manipular y navegar, y iv) explorar.

- **El estilo de la interfaz**. Define la apariencia *(look)* y el comportamiento *(feel)* de la interfaz de usuario. Se toma en cuenta el modo de interacción para elegir los tipos específicos de interfaces y sus componentes; por ejemplo, si se utilizará un sistema basado en menús, comandos o entrada por voz, o una combinación de éstos.

- **El paradigma de interacción**. La forma de interacción (modo de interacción y estilo de la interfaz*)* se implementa mediante soluciones tecnológicas concretas. Éstas a su vez, implementan paradigmas de interacción que podemos clasificar en dos tipos: i) *"*de escritorio*"* que permite interacciones explícitas mediante dispositivos como el ratón, el teclado y el monitor; y ii) paradigmas que van más allá del escritorio, tal como la realidad virtual, los robots, el cómputo vestible, el cómputo tangible, los visualizadores ambientales y la realidad aumentada. Estos últimos utilizan tecnologías que permiten una interacción natural e implícita, tal como cámaras y sensores inerciales que implementan nuevos estilos de interacción basados en gestos, movimientos y posturas del cuerpo.

[45] Preece, J., Rogers, Y., Sharp, H. **Interaction Design: Beyond Human Computer Interaction**, John Wiley and Sons, Inc., 2002.

[46] Stone, D., Jarrett, C.,Wodroffe, M., Minocha, S. **User Interface Design and Evaluation**, Morgan Kaufmann Publishers, Elsevier, 2005.

Las decisiones sobre el modo de interacción son de más alto nivel de abstracción que las realizadas para seleccionar el estilo de la interfaz, ya que las primeras determinan la naturaleza de las actividades del usuario que serán apoyadas, mientras que las segundas se refieren a la selección de tipos específicos de interfaces de usuario. Por otro lado, seleccionar desde un principio el paradigma de interacción ayuda a informar el diseño conceptual del sistema interactivo ya que influye en la selección de los modos y estilos de interacción. A continuación se describen los cuatro modos de interacción más comunes. Se describe cómo diversas soluciones tecnológicas ayudan a implementarlos siguiendo alguno de los paradigmas de interacción mencionados:

Instruir. El usuario indica al sistema lo que debe hacer. Por ejemplo es cuando el usuario da instrucciones al sistema para que almacene, borre o imprima un archivo. Este modo de interacción no sólo ha evolucionado sino que también varía dependiendo del paradigma de interacción y el estilo de la interfaz. Así, la interacción con la computadora personal mediante sistemas basados en comandos (ej., DOS, UNIX) cambió al incluir sistemas basados en interfaces de usuario gráficas que reciben instrucciones mediante teclas de control o menú de opciones, hasta sistemas que reciben instrucciones mediante reconocimiento de voz y gestos en 2-D y 3-D, lo cual no sólo ha simplificado este modo de interacción sino también ha facilitado la accesibilidad de los dispositivos computacionales por quienes padecen alguna discapacidad. Por ejemplo, Google Assistant[47] y Siri[48] se han utilizado por invidentes y débiles visuales para dar instrucciones mediante voz a las aplicaciones de sus dispositivos móviles (ej., "llama a José", "cerrar Facebook").[49]

Conversar. El usuario y el sistema mantienen un diálogo, posiblemente en lenguaje natural. En este modo el sistema actúa más como un compañero que como una máquina que obedece órdenes; es útil en aplicaciones en que

[47] https://assistant.google.com/

[48] Apple Inc., Siri. http://www.apple.com/ios/siri/

[49] Wong, M. E., Tan, S. S. (2012). **Teaching the benefits of smart phone technology to blind consumers: Exploring the potential of the iPhone**. *Journal of Visual Impairment & Blindness*, 106(10): 646.

el usuario necesita encontrar algún tipo específico de información o discutir algún aspecto de la misma, como en los sistemas tutores, las máquinas de búsqueda y los sistemas de ayuda. Desde los años sesenta del siglo pasado se mostró que es posible construir sistemas de diálogo con modelos muy simples de la conversación[50] y actualmente podemos encontrar sistemas llamados *chatbot*, capaces de aprender de su entorno para entablar conversaciones informales, tal como el chatbot Tay diseñado para conversar por Twitter con los jóvenes y que con base al contexto de la conversación responde de forma agradable o agresiva.[51] Los agentes inteligentes de software también se han utilizado para implementar nuevos paradigmas de interacción.[52] De esta línea surgieron los Agentes Relacionales[53] que se diseñan para construir relaciones socio-emocionales con las personas emulando la interacción cara a cara. Los agentes no sólo hablan al conversar sino también emiten gestos y expresiones faciales con el fin de generar empatía con el humano. El beneficio principal de esta interacción es que permite a las personas (especialmente a los novatos) interactuar con el sistema de una forma que les resulta familiar, pero existe el riesgo de que el sistema no responda como el humano espera, lo confunda y en consecuencia se interrumpa la conversación.[54]

En México, se ha investigado sobre modelos cognitivos de la IHC que incluyen protocolos para que aplicaciones como sistemas de diálogo y robots de servicio puedan comprender el contexto y entablar conversaciones con los humanos, tal como el robot de servicio Golem.[55]

---

[50] Weizenbaum, J. **Computer Power and Human Reason: From Judgment to Calculation**. New York: W.H. Freeman and Company. pp. 2,3,6,182,189, 1976. Ver también el capítulo de Lingüística Computacional en este texto.

[51] Hope Reese. **Why Microsoft's 'Tay' AI bot went wrong**. *Tech Republic* (24 de marzo de 2016). Ver capítulo 3.

[52] Ver el Capítulo 1 de este texto.

[53] http://relationalagents.com

[54] Cafaro, A., Vilhjálmsson, H. H., Bickmore, T. (2016). **First Impressions in Human–Agent Virtual Encounters**. *ACM Trans. Comput.-Hum. Interact.* 23(4), Art. 24 (August 2016).

[55] Pineda, L. A. et al. (2008). **Specification and Interpretation of Multimodal Dialogue Models for Human-Robot Interaction.** En *Artificial Intelligence for Humans: Service Robots and Social Modeling,* G. Sidorov (Ed.), SMIA, México pp 33-50. Ver también capítulos 3 y 4.

Por otro lado se ha investigado el efecto que tiene este modo de interacción en los humanos y se ha evaluado cómo los agentes de software representados como avatares logran comunicar emociones mediante expresiones faciales.[56] De manera similar la evaluación del avatar EMI, desarrollado para asistir a una comunidad de Oaxaca a elegir rutas de transporte rápidas y seguras, mostró que personas analfabetas se beneficiaron al acceder a esta información fácilmente.[57]

Manipular y navegar. El usuario manipula objetos y navega a través de espacios virtuales utilizando su propio conocimiento sobre el mundo físico. Por ejemplo, para mover, seleccionar, abrir, cerrar y aumentar[58] objetos virtuales. La manipulación directa (MD)[59] es un estilo de interacción en que los usuarios actúan sobre los objetos mostrados utilizando acciones físicas que tienen un efecto visible e inmediato en la pantalla. Éste es uno de los conceptos centrales de las GUIs y esta tecnología permitió capitalizar la comprensión de lo que sucede con los objetos físicos del mundo real, ya que las acciones físicas de los usuarios se emulan por el sistema mediante pistas auditivas y visuales, como cuando se arrastra un archivo al ícono de la basura. La primera compañía en diseñar un sistema basado en GUIs fue Xerox PARC.[60] Posteriormente surgieron otros paradigmas tecnológicos que ayudan a implementar este modo de interacción, como la realidad virtual, en la que los usuarios interactúan y navegan por un mundo físico simulado en 3-D y las aplicaciones del cómputo ubicuo, en las que se interactúa con objetos físicos aumentados digitalmente, los cuales se integran de forma natural a las actividades del usuario.

[56] Sánchez, J. A., Medina, P., Starostenko, O., Cervantes, O., Wan, W. (2014). **Affordable development of animated avatars for conveying emotion in intelligent environments. Ambient Intelligence and Smart Environments**. *2nd International Workshop on Applications of Affective Computing in Intelligent Environments*, Shanghai, China, pp. 80-91. Disponible en http://www.ebooks.iospress.nl/volumearticle/36664.

[57] Banos, T., Aquino, E., Sernas, F., Lopez, Y., Mendoza. R. (2007). **EMI: a system to improve and promote the use of public transportation**. *CHI '07 extended abstracts on Human factors in computing systems*, pp. 2037-2042. New York, NY, USA, ACM.

[58] para visualizar su información/contenido.

[59] https://www.nngroup.com/articles/direct-manipulation/

[60] https://www.parc.com/

México se ha destacado por el desarrollo de aplicaciones de cómputo ubicuo para asistir a personas con capacidades diferentes. Por ejemplo, el Visualizador Ambiental para la Medicación,[61] despliega pictogramas para indicarle al adulto mayor si tomó el medicamento correcto, lo que se detecta mediante sensores pasivos (NFC).[62] También se han desarrollado tecnologías ambientales para fomentar el envejecimiento activo, las cuales promueven la interacción basada en gestos o movimientos del cuerpo para manipular los elementos de un juego, tal como en los juegos de Kinect.[63] En esta línea se desarrolló un dispositivo de interacción para detectar la fuerza de agarre de la mano, que se interpreta como un parámetro de la acción a ejecutar; por ejemplo, la fuerza que se aplica al golpear una bola de billar o al lanzar a un pájaro en el popular juego "Angry Birds".[64] Adicionalmente, se ha favorecido la rehabilitación física de adultos mayores con enfermedad cerebrovascular mediante videojuegos que proveen retroalimentación háptica acerca de los movimientos realizados con la mano.[65] Por otro lado, se ha investigado cómo potenciar la integración social de personas con autismo dándoles apoyo visual durante interacciones cara a cara. Por ejemplo, MOSOCO es un sistema de realidad aumentada que utiliza el teléfono móvil para proporcionar pistas visuales que guíen al niño con autismo durante su interacción social con niños neuro-típicos.[66] Además, se ha evaluado el potencial de los lentes inteligentes (como Google Glass) para dar retroalimentación visual

[61] Zárate-Bravo, E., García-Vázquez, J. P., Rodríguez, M. D. (2015). **An Ambient Medication Display to Heighten the Peace of Mind of Family Caregivers of Older Adults: A Study of Feasibility**. *MindCare* 2015:274–283.

[62] http://nearfieldcommunication.org/

[63] http://www.xbox.com/

[64] Zavala-Ibarra, I; Favela, J. (2012). **Ambient Videogames for Health Monitoring in Older Adults**. *8th International Conference on Intelligent Environments (IE)*, 2012, pp. 27–33.

[65] Ramirez-Fernandez, C., García-Canseco, E., Morán, A. L., Orihuela-Espina, F. (2014). **Design Principles for Hapto-Virtual Rehabilitation Environments: Effects on Effectiveness of Fine Motor Hand Therapy.** *REHAB 2014*, pp. 270–284.

[66] Escobedo, L., Nguyen, D. H., Boyd, L. E., Hirano, S., Rangel, A., Garcia-Rosas, D., Tentori, M., Hayes. G. (2012). **MOSOCO: A Mobile Assistive Tool to Support Children with Autism Practicing Social Skills in Real-Life Situations**. *In ACM Proceedings of the Conference on Human Factors in Computing Systems*, pp. 2589–2598.

que ayude a adultos con autismo a regular las alteraciones en la entonación y el ritmo del lenguaje.[67]

Explorar. En este modo las personas pueden buscar y explorar información conforme se la presenta el sistema, tal como cuando se hojea una revista o se sintoniza la radio. Las páginas Web y portales de venta de productos aplican este modo de interacción. Este modo se ha utilizado también en Sistemas Colaborativos, por ejemplo, PIÑAS es un sistema que facilita que una comunidad de co-autores distribuidos utilicen la Web para colaborar en la edición de documentos compartidos. El sistema permite resaltar las secciones modificadas, indicar quién y cuándo las modificó y de esta forma dar conciencia sobre las actividades de edición que realiza el grupo así como facilitar la exploración del documento.[68] También se han utilizado mecanismos de conciencia de colaboración para apoyar la programación por pares distribuidos en diferentes localidades y facilitar la exploración del código.[69] En esta línea el concepto de "Esferas de Trabajo" explica la forma inherente en que las personas organizan "unidades de trabajo"; éstas involucran el manejo de diversos recursos informativos (ej. documentos, aplicaciones, correos, etc.), además de que pueden fragmentarse —dado que las personas suelen cambiar de una tarea (unidad de trabajo) a otra. Lo anterior motivó el desarrollo de un sistema que incluye mecanismos que ayudan a identificar las esferas de trabajo activas y explorar los recursos que contienen.[70]

---

[67] Boyd, L. E., Rangel, A., Tomimbang, H., Conejo-Toledo, A., Patel, K., Tentori, M., Hayes, G. (2016). **SayWAT: Augmenting Face-to-Face Conversations for Adults with Autism**. *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, pp. 4872–4883.

[68] Morán, A. L., Decouchant, D., Favela, J., Martínez Enríquez, A. M., González-Beltrán, B., Mendoza, S. **PIÑAS: Supporting a Community of Co-authors on the Web**. *DCW 2002*, pp. 113–124.

[69] Morán, A. L., Favela, J., Romero, R., Natsu, H., Pérez, C. B., Robles, O., Martínez Enríquez, A. M. (2008). **Potential and Actual Collaboration Support for Distributed Pair-Programming**. *Computación y Sistemas* 11(3).

[70] González, V. M., Gloria Mark, G. **"Constant, constant, multi-tasking craziness": managing multiple working spheres**. *CHI 2004*, pp. 113–120.

6.3.4. Evaluación

Hasta mediados de los años ochenta la investigación en IHC, como en otras áreas de la computación, tenía una orientación ingenieril o de construcción (*build methodology*) en la que se proponían nuevos dispositivos de entrada o salida, sistemas, o nuevos modos de interacción; en esta época la evaluación se reducía a probar la funcionalidad, lo que era suficiente para que el trabajo fuese publicable. Sin embargo, gradualmente fue permeando en la comunidad la necesidad de aportar mayor evidencia de que el dispositivo o técnica propuesta fuese, además de factible, efectiva, eficiente y/o usable. Hoy en día, tanto en la investigación, como en la práctica profesional en IHC, el uso de técnicas de evaluación es una constante cuya competencia es indispensable en la currícula de un profesional del área.

La evaluación del trabajo en IHC ha contribuido a la formación de un cuerpo de conocimiento en el área. Evaluar la interacción entre el humano y la computadora permite generalizar resultados y establecer principios de diseño generales. También da pie a la creación de teorías del comportamiento humano relevantes a su interacción con tecnologías de información. Teorías que permiten explicar o predecir el resultado de dichas interacciones; por ejemplo, para entender por qué la voz puede resultar adecuada para interactuar con el dispositivo de navegación GPS de un automóvil, pero no en un ambiente de oficina en el que se encuentran otras personas. Evaluar también permite probar ideas, visiones o hipótesis. Operar los aparatos electrodomésticos en un hogar por medio de gestos puede parecer una idea interesante, pero si hacer el gesto toma más tiempo o genera más errores que presionar el botón del dispositivo puede no resultar práctico. Sólo por medio de la evaluación es posible responder con precisión a estas preguntas. Finalmente, resultados de evaluación en IHC han ayudado a establecer principios y guías de diseño que permiten a profesionales del área diseñar nuevas aplicaciones utilizando las mejores prácticas sin tener que recurrir a prueba y error.

Las técnicas de evaluación en IHC pueden utilizarse en distintas tareas. Durante el desarrollo de sistemas se pueden utilizar para entender a los usuarios potenciales así como las tareas que realizan. El diseño de un sistema que será utilizado por un médico, un controlador aéreo o un estudiante de primaria debe tomar en cuenta las capacidades y limitaciones de cada tipo de usuario. Por ejemplo, si se tiene atención dividida por estar realizando varias tareas a la vez, si se tiene que tomar decisiones en poco tiempo o si se requiere una explicación detallada. Además de los usuarios directos, otros individuos pueden verse afectados por la interacción con el sistema, como un estudio de uso de expediente electrónico que fue evaluado positivamente por los médicos; sin embargo, los pacientes percibían que la calidad de la consulta se veía afectada por el tiempo que el médico dedicaba a la computadora, lo que interpretaban como falta de interés del médico sobre su caso.

La evaluación se puede hacer al inicio del proyecto, durante la etapa de diseño y/o una vez que se concluye el sistema. En el primer caso el propósito es concebir al sistema o informar su diseño; en el segundo obtener retroalimentación temprana acerca del diseño, incluso con prototipos de baja fidelidad, como dibujos o maquetas no funcionales; también se pueden utilizar otras técnicas con prototipos más avanzados para identificar problemas concretos de usabilidad. Finalmente, después de liberado un sistema se pueden hacer pruebas de aceptación e identificar recomendaciones de cambios a realizar en una segunda versión.

Los estudios de usuarios también se pueden utilizar para identificar problemas en la forma en que un individuo realiza una tarea, ya sea con o sin el apoyo de un sistema computacional. En el primer caso pueden ayudar a diseñar un mejor sistema y en el segundo se abre la oportunidad de desarrollar nuevos sistemas para apoyar la tarea. Evaluar un sistema en uso permite identificar las fuentes comunes de errores y las operaciones que demandan más tiempo al usuario, de manera que un rediseño de la interface permita hacer más eficiente su uso.

En investigación en IHC la evaluación permite probar hipótesis y descubrir principios generales de diseño. Los avances vertiginosos en nuevas formas de interacción dan origen a nuevas preguntas de investigación y a plantear constantemente nuevas hipótesis. Algunas de estas preguntas son cómo se debe dar retroalimentación por voz a un robot de servicio —para facilitar su aceptación— cuando no entiende el comando que recibe; qué capacidad tiene el ser humano de distinguir distintos patrones de vibración producidos por un reloj inteligente de manera que pueda asociarlos a la persona que lo llama; cómo perciben terceros los riesgos a su privacidad cuando un individuo utiliza unos lentes inteligentes que son capaces de tomar fotografías de su entorno; o qué estrategia de comunicación debe adoptar un agente inteligente para motivar a un individuo a cambiar de comportamiento, por ejemplo, para que deje de fumar.

En IHC se han desarrollado y adaptado distintas técnicas de evaluación usadas en otras áreas de investigación. Dado que la persona es el elemento fundamental de la interacción muchas de las técnicas de evaluación en IHC tienen su origen en la psicología, la sociología y la antropología.

Las técnicas para recabar información de usuarios incluyen i) el uso de cuestionarios, ii) las entrevistas, iii) grupos focales y iv) la observación. Los cuestionarios permiten obtener información específica de muchos sujetos; las entrevistas ayudan a establecer el contexto del usuario y pueden ser estructuradas, semi-estructuradas o no-estructuradas. Estas últimas se basan en preguntas abiertas y el flujo de la conversación depende de las respuestas que da el entrevistado. La entrevista no-estructurada permite generar datos cualitativos y entender de manera más profunda el contexto de uso de la tecnología. El grupo focal, por su parte, permite recabar opiniones de un grupo de personas relacionadas con sus necesidades; es particularmente útil para encontrar puntos de coincidencia y cuando hay visiones alternativas. Finalmente, la observación permite analizar al usuario y a las tareas en el contexto en que se llevan a cabo, así como recabar información que difícilmente se

puede obtener de otra forma. Incluso es posible que el sujeto no esté consciente de ciertos aspectos de la tarea o la forma como ésta se puede mejorar; un escenario ilustrativo es el del médico que utiliza un expediente electrónico sin darse cuenta que el paciente se siente ignorado, como en el ejemplo mencionado anteriormente.

La información recabada por las diferentes técnicas se puede analizar por métodos cuantitativos o cualitativos. Los primeros se basan en el planteamiento de hipótesis derivadas de preguntas de investigación. Para probar estas hipótesis se diseñan experimentos, los cuales se llevan a cabo con la participación de usuarios (ej., mediante dos interfaces de usuario diferentes); se obtienen datos y se analizan utilizando técnicas estadísticas, notablemente, pruebas de hipótesis. Los resultados se discuten y se contrastan con la literatura y, en lo posible, se generalizan.

Un ejemplo de métodos cuantitativos es la evaluación de protocolos de asistencia a llamadas de emergencia a través de diferentes medios.[71] En este estudio se compara el uso de protocolos de atención en papel con los disponibles en una aplicación desarrollada para teléfonos inteligentes en tres condiciones distintas: cuando el paciente y la enfermera que sigue el protocolo se encuentran en el mismo lugar, cuando la interacción se realiza por teléfono y cuando se hace por videoconferencia. La evaluación se realizó en condiciones controladas en laboratorio con la participación de 12 enfermeras familiarizadas con los protocolos. Para asegurar que todas las enfermeras recibieran el mismo estímulo en las distintas condiciones los pacientes fueron interpretados por actores que seguían un script al solicitar la asistencia. Las variables que se midieron en el estudio fueron el tiempo de la consulta, la ruta de navegación en el protocolo de atención, las pausas en la conversación, el número de veces en que la enfermera y el paciente hacían contacto visual (excluyendo la condición de teléfono) y la eficacia de la consulta, en

[71] Castro, L. A., Favela, J., García Peña, C. (2014). **Effects of communication media choice on the quality and efficacy of emergency calls assisted by a mobile nursing protocol tool**. *CIN-Computers informatics nursing,* 32(11), 550–558.

términos de si la enfermera siguió el protocolo adecuadamente y si realizó la recomendación correcta. El estudio cuenta con dos variables independientes: Tipo de protocolo (papel o teléfono inteligente) y Medio de comunicación (presencial, teléfono y videoconferencia). El análisis se basó en una prueba de hipótesis utilizando análisis de varianza (ANOVA). Entre los resultados se encontró que en el protocolo por teléfono se cometieron menos errores de navegación, que hubo menos contacto visual en las sesiones presenciales que por videoconferencia, y que no hubo diferencia significativa entre lo adecuado de la recomendación que realiza la enfermera siguiendo el protocolo en las distintas condiciones.

Por otra parte, los métodos cualitativos se basan en el análisis de información recabada principalmente de observación y entrevistas semi-estructuradas o no-estructuradas. En contraste con los métodos cuantitativos, una evaluación cualitativa generalmente involucra a pocos sujetos. Se parte de una pregunta de investigación abierta y el análisis es de carácter exploratorio, a diferencia de las técnicas cuantitativas que buscan probar una hipótesis concreta. Las técnicas cualitativas permiten comprender de manera más amplia el problema de estudio, sin partir de un sesgo o una idea preconcebida. Asimismo, la aplicación de un método cualitativo puede generar nuevas hipótesis que den lugar a una evaluación cuantitativa posterior, lo que a su vez sugiere el uso de Métodos Mixtos que complementen las fortalezas de ambos tipos y disminuyan sus debilidades.

Los métodos cualitativos se pueden ejemplificar con un estudio sobre la percepción del envejecimiento y el uso de tecnología realizado en México.[72] Éste consistió en una intervención en la que seis adultos mayores utilizaron cuatro paradigmas tecnológicos novedosos para ellos (un celular inteligente, un sistema de red social, un dispositivo para leer libros electrónicos y un dispositivo vestible para monitorear actividad física). Se realizaron entrevistas semi-estructuradas cada dos semanas con los participantes durante todo

[72] Juárez, M. R., González, V. M., Favela, J. (2016). **Effect of technology on aging perception**. *Health Informatics Journal*. doi: 10.1177/1460458216661863

el estudio. El análisis de la información se realizó mediante la técnica de la "Teoría Fundamentada". Como resultado del análisis se produjo un modelo para explicar el fenómeno del efecto del uso de la tecnología en la percepción de envejecimiento que fue comparado con otros marcos teóricos reportados en la literatura. Se encontró que el uso de la tecnología produce una serie de efectos positivos en la auto-percepción de los adultos mayores. Los informantes consideraron que los hacía sentirse más jóvenes, activos e independientes. También reportaron que percibieron el uso de tecnología como una ayuda para mantenerse socialmente activos.

Los distintos grupos que realizan investigación en IHC en México han desarrollado infraestructura para apoyar la evaluación de dispositivos y modos de interacción. Uno de los primeros esfuerzos en este sentido es el Laboratorio de Tecnologías Interactivas y Cooperativas, en la Universidad de las Américas Puebla[73] establecido en 1996. Destaca también el *Usability Laboratory* (UsaLab) en la Universidad Tecnológica de la Mixteca.[74] Estos laboratorios incluyen una cámara Gessel, una área de observación, una área de uso así como equipo y software especializado. Además de actividades de investigación se han utilizados para hacer evaluaciones de usabilidad en la industria. Otro caso a resaltar es el laboratorio viviente *Life at a Pie (Living at a Pervasive Interaction Environment)*.[75] Un laboratorio viviente consiste en un entorno de uso diario que tiene equipo embebido que facilita la integración y evaluación de nuevas tecnologías. *Life at a Pie* es una escuela-clínica inteligente para niños con autismo establecida en Tijuana en el 2012. Tiene por objetivo desarrollar y evaluar intervenciones innovadoras por medio de tecnología de cómputo ubicua en apoyo a los niños con autismo y al personal de la escuela. Varios salones y laboratorios de la escuela se han equipados con sensores y pantallas situadas en apoyo a intervenciones basadas en tecnología. Para gra-

[73] **Laboratorio de Tecnologías Interactivas y Cooperativas,** http://lct.udlap.mx

[74] Moreno Rocha, M. A., Hernández Martínez, D. (2008). **UsaLab: the experience of a usability lab from the Mexican perspective.** *BCS HCI* (2), pp. 171–172.

[75] Tentori, M., Escobedo, L. Balderas, G. L. (2015). **A Smart Environment for Children with Autism**. *IEEE Pervasive Computing* 14(2):42–50. http://www.pasitos.org/

bar y monitorear la conducta de los niños se utiliza un registro electrónico de comportamientos que hace posible evaluar la eficacia de las intervenciones. La integración de la tecnología en la escuela ha facilitado la participación de las maestras y los niños en el co-diseño de las tecnologías así como en su evolución.

La propuesta de nuevas técnicas de evaluación o su adecuación a nuevos entornos o circunstancias es un área de investigación activa en IHC. Un ejemplo de dicho trabajo es la propuesta metodológica llamada *Naturalistic Enactment*[76] que propone un método para la evaluación formativa de interacción en condiciones naturales, en ambientes críticos, como los de cuidado a la salud y en condiciones controladas. En esta línea se han realizado también investigaciones de carácter teórico relacionadas con la tecnología y los sistemas biológicos[77] así como el desarrollo interactivo de la conciencia social.[78]

6.4. Conclusiones

Interacción Humano-Computadora es un área relativamente joven que ha tenido un gran impulso desde el nacimiento de la computación personal. A partir de entonces y derivado de un gran avance tecnológico ha tomado un papel central en el diseño de nuevos productos tecnológicos. La IHC es un área multidisciplinaria por naturaleza y es precisamente esta característica la que hace que sea interesante y ofrezca un reto a quienes trabajan en ella.

---

[76] Castro, L. A., Favela, J., García-Peña, C. (2011). **Naturalistic enactment to stimulate user experience for the evaluation of a mobile elderly care application**. *Mobile HCI*, pp. 371–380.

[77] Froese, T. (2014). **Bio-machine hybrid technology: A theoretical assessment and some suggestions for improved future design.** *Philosophy & Technology,* 27(4): 539-560

[78] Froese, T., Iizuka, H. & Ikegami, T. (2014). **Using minimal human-computer interfaces for studying the interactive development of social awareness.** *Frontiers in Psychology,* 5(1061). doi: 10.3389/fpsyg.2014.01061

Uno de estos retos es el estudio y diseño de interfaces naturales entre humanos y computadoras. Aunque se ha avanzado en el tema queda aún mucho por hacer. Recientemente se ha hablado de que la relación entre humanos y computadoras está en un punto de quiebre en el que la máquina ya no estará necesariamente supeditada a los deseos y necesidades del humano —y en la que no se utilizará meramente como una herramienta— sino que se logrará una integración casi simbiótica entre ambos agentes.[79] Con el término "integración" se pone un mayor énfasis en una era en la que los humanos cohabitan con las computadoras de manera coordinada y cooperativa, con grandes beneficios para la sociedad.

Es en esta integración donde radican muchos de los principales retos y oportunidades del área y también donde convergen múltiples áreas de investigación de las ciencias de la computación y disciplinas afines, como la inteligencia artificial, el cómputo colaborativo, la robótica, la telemática, el procesamiento de señales y el procesamiento del lenguaje natural, entre otras. Más aún, la transdisciplinariedad será aún más importante ya que se tendrá que comprender las maneras en que individuos, grupos y sociedad entienden, interpretan y se adaptan a estas nuevas realidades a través de áreas de estudio como la ciberpsicología y otras.

En el caso particular de México la investigación en IHC puede ser poca en números pero ha tenido un impacto importante a nivel nacional. Un indicador es la publicación del libro "La Interacción Humano-Computadora en México",[80] un esfuerzo colectivo que ofrece una introducción para quienes se inician en el área así como referencias a avances realizados en el país. Otro indicador es el Congreso Mexicano de Interacción Humano-Computadora (Mex IHC),[81] el cual se organiza cada dos años desde 2006. Esto represen-

[79] Umer Farooq, U., Grudin, J. (2016). **Human-computer integration**. *Interactions* 23(6):26–32. doi: http://dx.doi.org/10.1145/3001896.

[80] Muñoz, J., González, J. M., Sánchez, J. A. **La Interacción Humano-Computadora en México,** Pearson, 2016.

[81] http://mexihc.org

ta un gran reto, pero también una gran oportunidad para intercambiar (e incluso cuestionar) puntos de vista fuertemente establecidos en sociedades más industrializadas. En particular, hay retos que dependen de problemas o fenómenos prevalecientes en sociedades latinoamericanas, relacionados con los índices de población analfabeta, la marginación, la criminalidad y las comunidades vulnerables. Asimismo, los aspectos sociales, geo-políticos y demográficos, como la gran proporción de poblaciones indígenas, así como el acceso a infraestructura adecuada, son distintivos de nuestra región y requieren soluciones especificas. Todas estas características representan, sin duda, un panorama de oportunidades en los que IHC puede incidir de manera muy efectiva a través del diseño de tecnologías adecuadas.

7. Análisis de Señales y Reconocimiento de Patrones

Este capítulo cubre la intersección entre dos áreas de investigación y desarrollo tecnológico en computación: el análisis de señales digitales y el reconocimiento automático de patrones. Los términos *señales*, *patrones*, *análisis* y *reconocimiento* son extensos y presentan un amplio margen de interpretación. Por esta razón, se inicia con una introducción a dichos conceptos; se describen los problemas científicos principales y algunas de las metodologías para enfrentarlos y se aborda el trabajo de la comunidad científica mexicana. Finalmente, se discuten las perspectivas y retos de esta área del conocimiento. Esta temática es ubicua y tiene aplicaciones en robótica, aprendizaje e inteligencia computacional, interacción humano-computadora y tecnologías de lenguaje entre otras, como se discute en los capítulos correspondientes de este texto.

7.1. Definiciones y problemática

Una *señal digital* es un conjunto de magnitudes físicas medidas en un punto o intervalo de tiempo o espacio determinado. Estas magnitudes se representan de forma más o menos congruente y finita en una computadora u otro dispositivo digital, de tal forma que se puedan analizar. Por ejemplo, la temperatura, la presión, la corriente eléctrica, así como otros fenómenos, se representan como señales digitales. Las fotografías, termografías, radiografías y tomografías se representan también como señales de dos o más dimensiones

(las llamadas "imágenes digitales"). Las señales pueden tener un origen físico, como los fenómenos mecánicos o electromagnéticos, o uno social, como los económicos, demográficos o los procesos de la administración pública. También hay señales que resultan de procesos biológicos, como la voz, la presión arterial, la actividad del cerebro, las secuencias de ADN y la expresión de genes. El análisis tiene por objetivo entender el contenido informativo de las señales a fin de aprovechar, mejorar, corregir o simplemente profundizar el conocimiento del fenómeno, proceso o evento que se estudia. A menudo se miden varias señales al mismo tiempo, lo que permite un análisis más profundo pero a la vez más complejo.

Por su parte, un *patrón* es la representación de una relación estocástica entre señales y se obtiene mediante el análisis matemático de ejemplos de señales adquiridas previamente. Por lo mismo, un patrón representa a una clase de señales, que a su vez representa una clase de entidades individuales (objetos, acciones, eventos, procesos, etc.). Normalmente se asocia una etiqueta a cada patrón que indica el nombre de la clase correspondiente. Los patrones se obtienen a partir de la extracción de características distintivas o atributos de dichas entidades, normalmente del medio ambiente, por lo que las señales casi siempre contienen detalles irrelevantes y se requiere reducir la información sensada, de tal forma que se representen sólo los rasgos significativos de las clases. Los patrones se pueden usar para llevar a cabo tareas discriminatorias, predictivas o explicativas. Estas tareas se llevan a cabo usando algoritmos conocidos como *clasificadores*. Matemáticamente, la clasificación consiste en partir un espacio *n*-dimensional, con una dimensión por cada característica relevante, y cada región de este espacio corresponde a una clase determinada.[1]

El *reconocimiento* consiste en clasificar una o varias señales desconocidas en la clase correspondiente. Por ejemplo, la señal que representa a una radiografía y la radiografía propiamente se pueden clasificar como "crecimiento de la aurícula izquierda", "posible tumor maligno" o "paciente sano". Otro

[1] Tou, J.T., Gonzalez, R. C. **Pattern Recognition Principles, 2nd. edition**, Addison-Wesley, 1977.

ejemplo es el reconocimeinto de personas o acciones, como en un sistema de seguridad que analiza durante varios segundos el caminar de un sujeto para identificarlo y clasificarlo como "usuario autorizado" o "usuario no autorizado."

Un sistema de reconocimiento de patrones que se construye a partir de analizar y manipular instancias o ejemplos de un problema se conoce como *adaptivo*. La construcción de este tipo de sistemas tiene dos etapas principales: diseño y evaluación. En la primera se generan reglas de decisión para clasificar instancias del problema y en la segunda se verifica de forma intensiva que dichas reglas sean apropiadas para tomar buenas decisiones.[2] El diseño del clasificador tiene a su vez varias etapas. Primero es necesario determinar qué técnicas se deben emplear para limpiar las señales, es decir, eliminar el ruido u otros componentes que no son relevantes para la aplicación. Este proceso se conoce como pre-procesamiento de la señal. Posteriormente se seleccionan las transformaciones matemáticas para obtener las características que representarán a la señal y, consecuentemente, a los objetos o escenas que ésta representa, de tal manera que se maximice la posibilidad de realizar una buena clasificación. Finalmente se escoge el método de clasificación o reconocimiento propiamente. Frecuentemente la selección de las estrategias para extraer características y el método de clasificación están estrechamente relacionas y se influyen mutuamente. Este proceso se ilustra en la Figura 7.1.

El diseño de un clasificador es una actividad iterativa que implica probar y modificar varias veces las transformaciones hasta encontrar las que obtienen los mejores resultados. Este proceso depende en gran medida de la experiencia de los diseñadores; sin embargo, en los últimos años se ha incrementado la investigación de métodos basados en inteligencia artificial que permiten realizar automáticamente algunas decisiones de diseño.

---

[2] Shih, F.Y. **Image Processing and Pattern Recognition. Fundamentals and Techniques.** IEEE Press, Wiley, 2010.

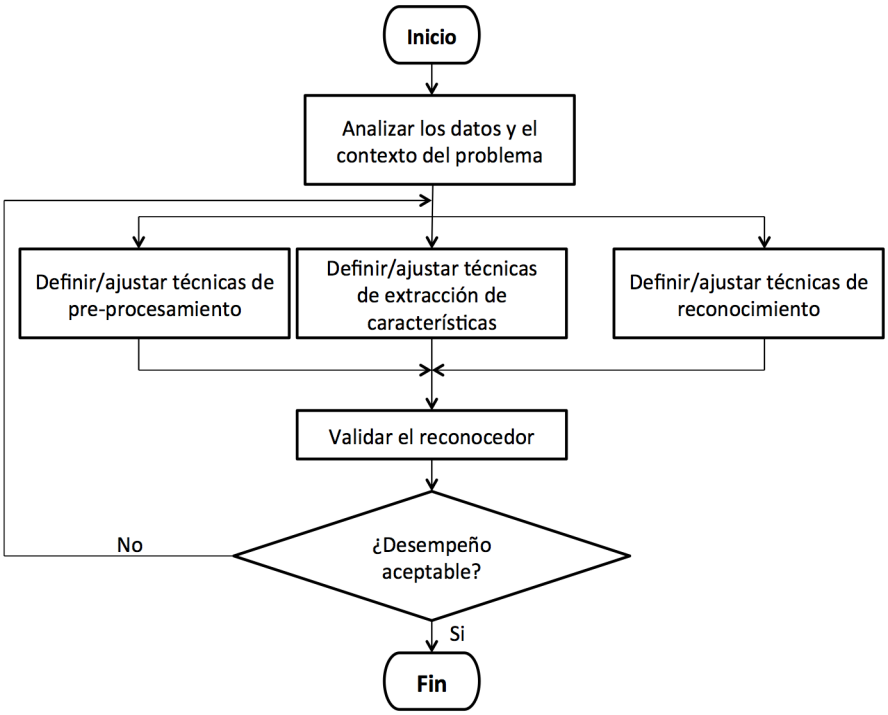


Figura 7.1. Etapas de diseño de un clasificador

Uno de los métodos más exitosos de diseño automático de clasificadores se basa en el aprendizaje profundo, el cual ha tenido muy buenos resultados, principalmente en problemas de visión y reconocimiento de voz.[3]

Por su parte, el uso de un sistema de reconocimiento involucra tres pasos: i) adquirir y pre-procesar la señal, ii) aplicar las transformaciones para obtener características y iii) aplicar las reglas de reconocimiento y decisión del clasificador. Este proceso se ilustra en la Figura 7.2.

[3] LeCun, Y., Bengio, Y., Hinton, G. (2015). **Deep learning**. *Nature*, *521*(7553): 436–444.

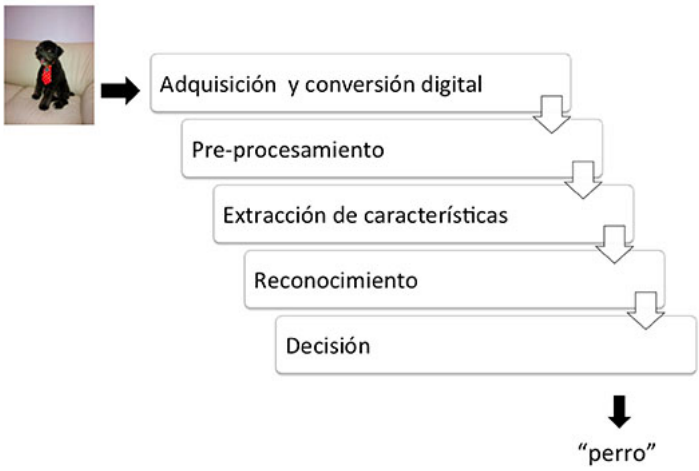


Figura 7.2. Pasos de un clasificador

El análisis de señales agrupa a una inmensa variedad de teorías matemáticas y algoritmos que permiten definir secuencias de transformaciones orientadas a la extracción de información de las señales, en particular para su reconocimiento. El proceso de análisis se inicia con la observación del fenómeno, el cual se registra a través de sensores y genera datos crudos. Existen varios tipos de transformaciones que pueden realizarse sobre una señal digital, como la convolución, el filtrado de frecuencia, el sub-muestreo, las funciones de transformación tiempo-frecuencia y multi-resolución, etc. Algunas de estas transformaciones se describen brevemente en la siguiente sección.

En resumen, el quehacer principal en el área de análisis de señales y reconocimiento de patrones (AS/RP) es el desarrollo de nuevas transformaciones y algoritmos, así como la comprensión de sus implicaciones matemáticas, bajo el contexto de asignación de clases a objetos.

Además de resolver problemas de clasificación, estas transformaciones también son útiles para resolver problemas de regresión, ordenamiento, regularización o asociación de patrones, como se explica a continuación:

- Clasificación/Regresión. Se asigna una señal a una clase representada por un modelo matemático a partir de una métrica apropiada.

- Inferencia. Consiste en hacer predicciones a corto, mediano o largo plazo. Se lleva a cabo de manera posterior a la regresión.

- Ordenamiento o indexación de conjuntos. Se realiza como parte del proceso de toma de decisiones, de forma posterior a la inferencia.

- Regularización. Se lleva a cabo, entre otras cosas, para apoyar el diseño de modelos de clasificación cuando se cuenta con pocas muestras o éstas están desbalanceadas; la regularización previene favorecer la clase o clases mayormente representadas.

- Asociación. Consiste en comparar las características de una muestra con las de una clase o patrón conocido.

AS/RP está íntimamente relacionado a otras ciencias, como la estadística, la física, la topología y la matemática, así como a varias ramas de la ingeniería, por lo que muchos investigadores provienen de ámbitos multidisciplinarios. Por lo mismo, aunque AS/RP toma técnicas y modelos de otras disciplinas científicas es una especialidad por sí misma. Por ejemplo, la transformada rápida de Fourier,[4] utilizada ampliamente por AS/RP, surgió en el ámbito de las matemáticas.[5] Asimismo, AS/RP es una disciplina ubicua y se aplica prácticamente a cualquier rama de las ciencias exactas, naturales o de la ingeniería. Esto ejerce una fuerte influencia en la generación de soluciones y promueve el trabajo interdisciplinario, aunque también ha provocado algunos inconvenientes, como la duplicidad de las metodologías, las ambigüedades en el vocabulario, la aplicación por inexpertos o su utilización de manera *ad hoc*,

[4] Cooley, J.W., J.W. Tukey (1965). **An algorithm for the machine calculation of complex Fourier series**. *Mathematics of Computation,* 19(2):297-301.

[5] Ramírez Cortés, J. M., Gómez Gil M. P., Baez López, D. (1998). **El algoritmo de la transformada rápida de Fourier y su controvertido origen.** *Ciencia y Desarrollo,* 24(139):70-77.

por lo que su importancia se minimiza frecuentemente. Por todo esto la AS/RP es una sub-disciplina de la Computación y los científicos dedicados a su estudio utilizan metodologías que van mucho más allá de la programación básica o del uso ciego de paquetería de software.

El crecimiento que ha experimentado la investigación y aplicación de AS/RP en los últimos años, gracias al avance del poder de cómputo y a la creciente vinculación entre investigadores de diferentes disciplinas, ha dado lugar a los siguientes desarrollos:

I *La especialización por sub-áreas*. Los problemas más comunes a los que se enfrenta AS/RP se han estudiados desde hace mucho tiempo y se han conseguido enormes avances en las propuestas de soluciones. Esto ha generado una especialización, por lo que es común encontrar expertos enfocados a metodologías particulares. En la siguiente sección se detallan algunas de las especialidades más populares entre los investigadores mexicanos.

II *La especialización en cuanto al enfoque de investigación*. Es posible encontrar dos tipos fundamentales de interacción con los dominios de aplicación: i) buscar nuevos dominios de aplicación para metodologías ya existentes, con el objetivo de abrir nuevos horizontes o atender alguna necesidad aún no cubierta, y ii) desarrollar alternativas metodológicas novedosas, óptimas en algún sentido, que enriquezcan el abanico de opciones existentes.

III *La producción de herramientas genéricas y especializadas*. Existen múltiples entornos y herramientas de carácter genérico que dan servicio a diferentes dominios de aplicación como LabView®,[6] Matlab®[7] o el software estadístico R.[8] Estas herramientas ofrecen librerías con

[6] http://www.ni.com/labview/why/esa/
[7] https://es.mathworks.com/products/matlab.html
[8] https://www.r-project.org/

módulos robustos que implementan los métodos y algoritmos más populares para la solución de aplicaciones de AS/RP. Hay también un sinnúmero de tutoriales, libros y cursos que apoyan a los diseñadores en la construcción de soluciones basadas en estas herramientas. Por otra parte, se han creado otras más especializadas, como Hércules, que se usa para desarrollar aplicaciones de visión computacional,[9] o WEKA[10] para tareas de aprendizaje y minería de datos. Además, existe una inmensa variedad de aplicaciones de software, que utilizan lenguajes de programación de propósito general, como C++, Phyton[11] o Java.[12] Como sucede en la construcción de cualquier otro sistema de software, el uso de técnicas formales de programación e ingeniería de software es de vital importancia para conseguir una implementación eficiente y robusta, capaz de aprovechar al máximo el poder de cómputo y los avances de la ciencia.

IV *El desarrollo de investigaciones con enfoques empírico y analítico.* El área AS/RP, como otras ramas de la ciencia, requiere y se beneficia tanto del conocimiento empírico o experimental como del analítico o demostrativo formal. Asimismo, encontrar una solución exacta u óptima, desde el punto de vista de la reducción de un sesgo para un problema determinado, no siempre supone el haber alcanzado el fin de una investigación. Otros tipos de optimización, como la reducción de los requerimientos de cómputo en tiempos de ejecución y/o espacio de almacenamiento, la construcción de heurísticas avanzadas o el desarrollo de métodos que generen soluciones aproximadas no exactas pero eficientes, conllevan a que el área esté en constante revisión y se genere un avance del conocimiento.

[9] http://www.mvtec.com/products/halcon/product-information/
[10] http://www.cs.waikato.ac.nz/ml/weka/index.html
[11] https://www.python.org/
[12] https://www.java.com/es/

7.2. Metodologías y herramientas

En esta sección se describen algunos de los métodos y herramientas más populares para resolver los tres problemas fundamentales que enfrenta el diseño de sistemas AS/RP:

- La representación adecuada de las señales de entrada (procesamiento de señales)
- La extracción de las características
- La determinación del proceso de decisión óptimo para estimar parámetros de separación del espacio de características (reconocimiento de patrones).

Para abordar los dos primeros problemas se utilizan técnicas de procesamiento de señales e imágenes y para el tercero hay una gran variedad de algoritmos apoyados en modelos matemáticos, estadísticos, de inteligencia artificial, inteligencia computacional, teoría de caos, teoría cuántica, etc. A continuación se explican brevemente algunos de los conceptos básicos en que se apoyan estas soluciones.

7.2.1. Análisis de Señales

Los conceptos y métodos básicos asociados al anásis de señales e imágenes están disponibles en libros clásicos como los de Proakis y Manolakís,[13] Gonzalez y Woods[14], Cuevas et al.[15] y Rodríguez y Sossa.[16]

[13] Proakis, J.G., Manolakis, D.G. **Digital Signal Processing. Third Edition**. Prentice Hall, 1996.

[14] Gonzalez, R.C., Woods, R. E. **Digital Image Processing. Fourth Edition.** Prentice Hall, 2016. Con contribuciones del Dr. Ernestro Bribiesca, del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), Universidad Nacional Autónoma de México (UNAM).

[15] Cuevas, E., Zaldívar, D., Pérez,M. **Procesamiento digital de imágenes con Matlab y Simulink**, AlfaOmega, 2010.

[16] Rodríguez, R., Sossa, H. **Procesamiento y análisis digital de imágenes**, Alfaomega-RAMA, 2012.

Algunos conceptos que comúnmente aparecen en la descripción de técnicas para procesar señales y extraer características son los siguientes:

- *Convolución.* Es una función u operación que superpone una función de referencia con la inversa de otra, con diferentes desplazamientos, y produce o genera una nueva función. En el contexto del procesamiento de señales digitales, dichas funciones son discretas, es decir, sólo tienen valores en instantes específicos y regulares de tiempo. La convolución se utiliza entre otras cosas para filtrar señales, obtener promedios ponderados en el tiempo, calcular el "eco" de un sonido, etcétera.

- *Transformada de Fourier*. Este es uno de los algoritmos más utilizados en AS/RP; se basa en la serie matemática definida por Jean-Baptiste Joseph Fourier, quien vivió en el siglo XIX. Fourier demostró que cualquier función se puede representar por una serie infinita de funciones trigonométricas sinusoidales. Esta teoría ha sido la base de muchas otras y, en particular, de algoritmos que permiten representar una función discreta definida en el tiempo como una serie de señales sinusoidales de diferentes frecuencias. Los coeficientes de los términos de dicha serie representan la contribución de cada frecuencia en la señal original. Existen varios algoritmos para calcular dicha transformación entre los que destaca la *Transformada Rápida de Fourier*.[17]

- *Filtrado de señales*. Consiste en transformar una señal en otra a través de modificarla mediante operaciones de multiplicación o de convolución. El filtrado puede realizarse en el dominio de frecuencia o en el dominio espacial. En el primer caso se restringen sus valores de frecuencia, ya sea eliminando un rango inferior, un rango superior o dejando solamente una banda posible de valores. En el segundo se realizan operaciones entre todos los pixeles de una imagen y una matriz

[17] Cooley, J.W., J.W. Tukey (1965). **An Algorithm for the Machine Calculation of Complex Fourier Series**. *Mathematics of Computation,* 19(2):297-301.

de coeficientes conocida como "kernel". El filtrado de señales tiene muchas aplicaciones, por ejemplo, eliminación de ruido, suavizado de la señal, detección de la presencia de fenómenos específicos, etcétera.

- *Análisis multi-resolución y transformadas Wavelets*. El análisis de Fourier informa sobre los componentes espectrales de frecuencia de una señal discreta, pero no especifica el tiempo exacto en que ocurrieron dichas frecuencias. Además, la resolución en las frecuencias que se obtiene depende de la resolución en el tiempo y del tamaño de la ventana que se utilice. El análisis wavelet permite solventar estas limitaciones ya que maneja diferentes tiempos y escalas de resolución a la vez. Las wavelets, conocidas en español como "ondeletas", son funciones matemáticas que satisfacen ciertos requerimientos; por ejemplo, son irregulares, asimétricas y con media cero. Las funciones wavelets más famosas incluyen a la Daubechies (llamada así en honor a su creadora), Morlet y Gaussiana.[18] Las características de las wavelet permiten su uso como funciones base, es decir, cualquier función se puede representar como una combinación lineal de funciones wavelets.

- *Transformación de color, escalas de grises y bitmaps*. Cada pixel de una imagen se puede representar —entre muchas formas— como: i) una terna de valores enteros, donde cada uno representa un componente del color (por ejemplo, rojo, verde y azul) que en conjunto representan al color, ii) un número entero que representa un nivel de gris, y iii) un número binario, donde 0 y 1 representan que el pixel es blanco y negro respectivamente. En la solución de problemas particulares es común hacer transformaciones entre estos formatos, como del color a su representación binaria o a niveles de gris, o asignar un color específico o "falso color" a un rango de niveles de gris, a fin de resaltar alguna propiedad.

[18] Daubechies, I. **Ten lectures on wavelets.** Society for industrial and applied mathematics, 1992.

- *Mejoramiento de imágenes*. A través del uso de operaciones matriciales se puede alterar cómo luce una imagen para apreciar mejor sus detalles.

- *Compresión de imágenes*. Dado que la representación de imágenes a través de pixeles puede ser muy grande, a veces es necesario reducir su tamaño pero sin perder detalles relevantes al problema.

- *Segmentación*. Consiste en separar los objetos que forman una imagen para poder analizarlos, agruparlos o clasificarlos.

7.2.2. Reconocimiento de Patrones

La tarea fundamental del reconocimiento de patrones es determinar el proceso de decisión para separar el espacio de características. Existen una gran cantidad de técnicas para realizar esta decisión. Los conceptos y métodos más importantes se pueden consultar en textos ampliamente conocidos.[19,20] Algunos de los más utilizados son los siguientes:

- *Estimación paramétrica*. Se estima la probabilidad condicional de ocurrencia de una clase, dado un conjunto de ejemplos para k clases determinadas de antemano. La forma de dicha probabilidad condicional no se conoce de antemano por lo que se asume y se calculan los parámetros que la describen. Este es un problema clásico de estadística que se aborda comúnmente de dos maneras: con estimación Bayesiana y con estimación de máxima probabilidad. La primera considera a los parámetros como variables aleatorias que tienen una distribución probabilística conocida, mientras que la segunda considera a los parámetros como cantidades fijas pero desconocidas.

- *Estimación no paramétrica*. Estas técnicas no requieren determinar el número de clases de antemano y se pueden usar sin la necesidad de

[19] Duda, R. O., Hart, P. E., Stork, D. G. **Pattern classification, Second Edition**. John Wiley & Sons, 2000.

[20] Bishop, C. M. **Pattern Recognition and Machine Learning**, Springer, 2006.

suponer la forma de las funciones de distribución. Se utilizan en varios tipos de reconocedores, por ejemplo, la estimación de funciones de densidad, las ventanas de Parzen o la técnica de estimación de *k* vecinos más cercanos.

- *Funciones discriminantes lineales*. Asumen que las funciones que separan a las clases son lineales y se conocen de antemamo, y utilizan ejemplos o instancias del problema para estimar sus parámetros. Estas técnicas incluyen al perceptrón, las máquinas de soporte vectorial y el procedimiento de Widrow-Hoff, también conocido como la regla de mínimos cuadrados promedio (LMS por sus siglas en inglés *Least-Mean-Squared*); son muy populares y fáciles de implementar.

- *Funciones discriminantes no lineales*. Incluyen a las redes neuronales de varios niveles (*Multilayer Perceptrons* o MLP por sus siglas en inglés) y las redes de funciones de base radial.

- *Métodos estocásticos*. Utilizan el azar (es decir, funciones aleatorias o *random*) para buscar los parámetros que definen a modelos de decisión complejos; se aplican cuando el espacio del problema es muy grande o cuando se cuenta con pocos datos de entrenamiento. Los modelos más comunes incluyen al de Boltzmann, inspirado en la mecánica estadística y a los métodos evolutivos, inspirados en la biología. Estos métodos requieren grandes recursos de cómputo.

- *Métodos no métricos*. Se utilizan cuando las características de las instancias del problema tienen valores discretos nominales (o "etiquetas"), por lo que no se aplican medidas de similitud, como distancias métricas, ni presentan algún orden. Por lo mismo, las descripciones se hacen con listas de los atributos con sus valores correspondientes. Los métodos más comunes incluyen a los árboles de decisión, árboles de Clasificación y Regresión (CART por sus siglas en inglés *Classification and Regression Trees*) y algunas de sus variaciones.

- *Algoritmos de aprendizaje supervisados*. Los clasificadores se construyen con algoritmos de aprendizaje de máquina que utilizan ejemplos de señales previamente etiquetadas con la clase a la que pertencen. En estos algoritmos las clases se conocen o determinan de antemano y dependen de que un "maestro", normalmente un experto humano, asigne los ejemplos a las clases a las que pertenencen.

- *Aprendizaje no supervisado y agrupamiento*. Hay problemas en los cuales no se conocen las clases de las señales de antemano, por lo que tampoco se pueden etiquetar a las señales de ejemplo; en este caso el diseño de clasificadores se basa en algoritmos que busquen la estructura inherente a los datos, sin la ayuda de un "maestro" —como sucede en el aprendizaje supervisado— para formar grupos de instancias del problema o *clusters* que correspondan a las clases. Una vez que se determinan estos grupos se les pueden asignar etiquetas apropiadas. El agrupamiento se basa en la intuición de que las instancias de cada clase son más similares entre sí y menos similares con las instancias de otras clases, por lo que depende en gran medida del tipo de métricas y distancias utilizadas. El agrupamiento y las técnicas de aprendizaje no supervisado son muy utilizados para la minería de datos. Entre los métodos más comunes se encuentra el de *k*-medias, el agrupamiento jerárquico, los métodos basados en grafos, los modelos de análisis de componentes principales (PCA por sus siglas en inglés de *Principal Component Analysis*) y los mapas auto-organizados (SOM por sus siglas en inglés *Self-Organizing Maps*).

7.3. La investigación de AS/RP en México

Según el Portal SCImago de clasificaciones de revistas y publicaciones por país,[21] el cual se basa en las bases de datos de SCOPUS-Elsevier,[22] México

[21] Portal SCImago Journal & Country Rank (2016). http://www.scimagojr.com/

[22] Elsevier. Scopus (2011) http://www.americalatina.elsevier.com/corporate/es/scopus.php#

se encuentra entre los primeros lugares de América Latina que realizan publicaciones científicas relacionadas a las áreas de procesamiento de señales, visión por computadora, reconocimiento de patrones e inteligencia artificial, todas ellas componentes fundamentales de AS/RP. A continuación se describen algunos de los trabajos desarrollados por la comunidad mexicana en este campo del conocimiento.

7.3.1. Reconocimiento de patrones en grandes bases de datos

La evolución actual de las tecnologías de adquisición y almacenamiento de datos masivos ha tenido como consecuencia que ya no sea posible analizar estos recursos con herramientas comunes. Esto ha generado una nueva tendencia para el análisis de información y la toma de decisiones conocida como *Big Data*; que consiste en los activos disponibles en grandes volúmenes de datos que se pueden originar en una gran variedad de fuentes como los sistemas GPS, dispositivos móviles, redes sociales, sensores digitales, ADN, colisionadores de partículas, etc. El volumen de información se mide en Gigabytes (GB) = 10^9, Terabytes (TB) = 10^{12}, Petabytes (PB) = 10^{15}, Exabytes (EB) = 10^{18} o Zettabytes (ZB) = 10^{21}. Un ejemplo de base de datos que maneja estos niveles de información es *Gen Bank* del Centro Nacional de Información sobre Biotecnología en los Estados Unidos (NCBI, por sus siglas en inglés), la cual contenía 1.67 Terabytes de bases de pares de nucleótidos o ADN[23] en octubre de 2016. Otros escenarios comunes donde se aplica el *Big Data* incluyen el análisis de sentimientos[24] y el seguimiento a preferencias de los usuarios,[25] el procesamiento de información de sensores e Internet de las

[23] National Center for Biotechnology Information. **GenBank and WGS Statistics**. https://www.ncbi.nlm.nih.gov/genbank/statistics/

[24] Uriarte-Arcia, A. V., López-Yáñez, I., Yáñez-Márquez, C., Gama, J., Camacho-Nieto, O. (2015). **Data stream classification based on the gamma classifier**. *Mathematical Problems in Engineering* 1(17), doi: 10.1155/2015/939175

[25] Ver el capítulo de Lingüística Computacional en este texto.

cosas,[26] el análisis de accesos a servidores, seguridad en dispositivos móviles,[27] el análisis de información médica[28] y el análisis semántico sobre Internet.[29] Estas grandes bases de datos normalmente no están estructuradas, por lo que la minería de datos[30] apoyada en sistemas de agrupamiento de datos automáticos[31] es una herramienta fundamental para la toma de decisiones en grandes bases de datos.

7.3.2. Redes Neuronales Artificiales

Las redes neuronales artificiales (RNA) son modelos computacionales capaces de adaptar su comportamiento en respuesta a ejemplos tomados del medio ambiente.[32] Estos modelos se inspiran en la anatomía del cerebro y de las neuronas biológicas, aunque están bastante lejos de representar exactamente dichos objetos o los procesos cerebrales de los seres vivos. Aún así, las RNA presentan características muy interesantes que las hacen apropiadas para la solución de algunos problemas asociados a AS/RP.

Como sucede con otras técnicas de la inteligencia computacional, la mayoría de los modelos artificiales de redes neuronales no buscan realmente ser

[26] Ríos, L. G., Diéguez, J. A. I. (2016). **A Big Data Test-bed for Analyzing Data Generated by an Air Pollution Sensor Network**. *International Journal of Web Services Research (IJWSR)*, *13*(4), 19-35.

[27] Rodríguez-Mota, A., Escamilla-Ambrosio, P. J., Morales-Ortega, S., Salinas-Rosales, M., & Aguirre-Anaya, E. (2016). **Towards a 2-hybrid Android malware detection test framework.** *Proc. of 2016 International Conference on Electronics, Communications and Computers (CONIELECOMP 2016), IEEE,* pp. 54-61.

[28] Ver el capítulo de Interacción Humano-Computadora.

[29] Guzmán-Arenas, A., Cuevas, A. D. (2010). **Knowledge accumulation through automatic merging of ontologies**. *Expert Systems with Applications*, 37(3):1991-2005.

[30] Witten, I.,Frank, E., Hall, M**. Data Mining: Practical Machine Learning Tools and Techniques. Third edition,** Morgan Kaufmann, 2011.

[31] Kuri-Morales, A., & Rodríguez-Erazo, F. (2009). **A search space reduction methodology for data mining in large databases**. *Engineering Applications of Artificial Intelligence*, 22(1):57-65.

[32] Haykin, S. **Neural Networks and Learning Machines. Third Edition**, Pearson Education, 2009.

una copia exacta del modelo original, sino más bien simular las actividades de procesamiento autónomo y distribuido de las neuronas, así como su conectividad y capacidad de influencia vecinal.

Las RNA son capaces de almacenar y utilizar el conocimiento adquirido de la experiencia, el cual se representa por números reales que modelan la fuerza sináptica que hay entre neuronas biológicas. Para una red neuronal, aprender significa modificar los valores de los pesos (números reales) con un algoritmo de aprendizaje, normalmente supervisado.

La investigación que se desarrolla en México en redes neuronales artificiales para AS/EP es extensa y en diversos dominios, por ejemplo en análisis de genes,[33] desbalance de clases,[34] segmentación de imágenes,[35] algoritmos de aprendizaje híbridos,[36] clasificación estelar,[37] entrenamiento de redes morfológicas para clasificación,[38] redes basadas en wavelets,[39] aproximación de

[33] Garro, B. A., Rodríguez, K., Vázquez, R. A. (2016). **Classification of DNA microarrays using artificial neural networks and ABC algorithm**. *Applied Soft Computing*, 38:548-560.

[34] Alejo, R., Valdovinos, R. M., García, V., & Pacheco-Sanchez, J. H. (2013). **A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios**. *Pattern Recognition Letters,* 34(4):380-388.

[35] Trujillo, M.C.R., Alarcón, T.E., Dalmau, O.S., Zamudio-Ojeda, A. (2017) **Segmentation of carbon nanotube images through an artificial neural network,** *Soft Computing*, 21: 611. doi:10.1007/s00500-016-2426-1

[36] Castro, J. R., Castillo, O., Melin, P., Rodríguez-Díaz, A. (2009). **A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks**. *Information Sciences, 179*(13), 2175-2193.

[37] Gulati, R. K., Gupta, R., Singh, H. P. (1998). **Analysis of IUE Low Resolution Spectra Using Artificial Neural Networks**. En *Ultraviolet Astrophysics Beyond the IUE Final Archive,* 413:711-711.

[38] Sossa, H., Guevara, E. (2014). **Efficient training for dendrite morphological neural networks**. *Neurocomputing*, *131*:132-142.

[39] Alarcon-Aquino, V., Barria, J. A. (2006). **Multiresolution FIR neural-network-based learning algorithm applied to network traffic prediction**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(2):208-220.

funciones para evaluación[40] y predicción[41] y memorias asociativas.[42,43] Una discusión más amplia de esta temática se presenta en el capítulo 2.

7.3.3. Aprendizaje Profundo

En el contexto de Inteligencia Artificial (IA), "aprendizaje profundo" (en inglés *Deep Learning* o DL) se refiere a la adquisición de conocimiento a través del uso de máquinas con varios niveles para la extracción.[44] El adjetivo "profundo" se aplica a la forma en que se adquiere el conocimiento y no al conocimiento en sí mismo. La gran ventaja de DL es que no se requiere definir de antemano las características o atributos que identifican a los patrones, sino que se generan de forma automática manipulando datos crudos. A este estilo se le conoce como "aprendizaje de representaciones" el cual construye características de alto nivel automáticamente, para lo cual utiliza una gran cantidad de niveles jerárquicos de extractores.

7.3.4. Clasificación y control para interfaces cerebro/computadora

Las interfaces cerebro-computadora (*Brain Computer Interfaces* o BCI por sus siglas en inglés) tienen por objetivo permitir que un humano controle de forma voluntaria a un dispositivo —como una prótesis o manejar un carro— a

[40] Gómez-Gil P., Mendoza-Velázquez A. **Redes Neuronales Artificiales para calificar la capacidad de crédito de entidades Mexicanas de Gobierno**. *Komputer- Sapiens,* 2 (2):26-29.

[41] Fonseca-Delgado, R., Gómez-Gil, P. (2016). **Modeling Diversity in Ensembles for Time-Series Prediction Based on Self-Organizing Maps.** En Merényi, E., Mendenhall, M., O'Driscoll, P. (eds.) Advances in Self-Organizing Maps and Learning Vector Quantization, Springer, pp.119-128.

[42] Cruz,B., Sossa, H., Barrón, R. (2007). **A new two-level associative memory for efficient pattern restoration**. *Neural Processing Letters,* 25(1):1-16.

[43] Villegas, J., Sossa, H., Avilés, C., Olague, G. (2011). **Automatic Synthesis of Associative Memories through Genetic Programming: a co-evolutionary approach**. *Revista Mexicana de Física,* 57(2):110-116.

[44] Ver nota 3.

través de señales cerebrales. El principio de funcionamiento consiste en asociar una señal producida por el cerebro como resultado de fijar la atención o pensar en algo específico, de carácter arbitrario, con una acción de control concreta. Para producir la señal el usuario puede seguir varias estrategias, como la atención selectiva o la imaginación motora.

El proceso utiliza un algoritmo de aprendizaje de máquina supervisado, cuyas entradas son las características de la señal cerebral y las salidas las clases de acciones de interés. En la fase de entrenamiento el sujeto piensa en el objetivo asociado con la acción, al tiempo que se analiza y clasifica la señal cerebral. Este proceso se realiza para cada una de las acciones que se deberán controlar por la interfaz cerebral. En la fase de uso, una vez que se cuenta con el clasificador, el sujeto piensa en el objetivo, se procesa la señal, se alimenta al clasificador y se envía la señal de control correspondiente, que en el caso básico es simplemente habilitar o inhibir al dispositivo.

La adquisición de la señal cerebral puede ser de forma invasiva, a través de implantes cerebrales, o no invasiva a través de dispositivos externos colocados a los usuarios, tales como sistemas de adquisición de electroencefalogramas o diademas.

Es importante subrayar que estos dispositivos no leen los contenidos mentales o los pensamientos, ni controlan la acción motora intencional, sino simplemente asocian la señal que se correlaciona con un proceso mental con una acción particular.

La construcción de las interfaces cerebrales involucra tareas de procesamiento de señales y reconocimiento de patrones avanzadas. Como todo sistema de clasificación, un sistema BCI contiene actividades de adquisición de señales, pre-procesamiento, extracción de características, clasificación y ejecución de la acción deseada.

La comunidad mexicana AS/RP trabaja extensamente diseñando aplicaciones basadas en BCI, por lo que existe una gran cantidad de trabajos.[45,46,47] Algunos proyectos aprovechan las ventajas de las técnicas de aprendizaje de redes neuronales artificiales para clasificar las señales usando información temporal.[48]

7.3.5. Análisis y predicción de series de tiempo

Esta tarea consiste en predecir los valores de las variables que caracterizan a un fenómeno o el comportamiento de un sistema en base a sus valores en el pasado. Este es un problema complejo, sobre todo cuando se intenta aplicar a sistemas altamente no lineales o caóticos, los cuales son frecuentes en aplicaciones de AS/RP.[49] Debido a su gran variedad de aplicaciones, el pronóstico de series de tiempo ha sido de gran interés en los últimos años. Hay dos tipos básicos de predicción: a un paso (también conocida como a corto plazo) y a largo plazo. De éstos, el segundo es el más difícil y frecuentemente imposible de resolver, pero aún así es necesario contar con estimados de los posibles va-

[45] López-Espejel, J. N. (2015). **Control de movimiento de objetos a través del uso de electro-encefalogramas y redes neuronales artificiales con equipo de bajo costo,** Tesis de Licenciatura en Ingeniería en Ciencias de la Computación. Benemérita Universidad Autónoma de Puebla- Instituto Nacional de Astrofísica, Óptica y Electrónica.

[46] Alvarado-González, M., Garduño, E. Bribiesca, E., Yáñez-Suárez, O., Medina-Bañuelos, V. (2016). **P300 Detection Based on EEG Shape Features**. *Computational and Mathematical Methods in Medicine,* 14. doi: 10.1155/2016/2029791; ver también Alvarado-González, M. **Interfaces Cerebro Computadora con perspectivas a su aplicación en robots de servicio** (2015), Tesis de Doctorado, Universidad Nacional Autónoma de México, Ciudad de México.

[47] Neri, J. F. (2013). **Diseño y desarrollo de un sistema para el control mental de prótesis utilizando una interfaz cerebro-computadora (BCI),** Tesis de Licenciatura, Ingeniería en Computación, Facultad de Ingeniería, UNAM.

[48] Morales-Flores, E., Ramírez-Cortés, J.M., Gómez-Gil, P., Alarcón-Aquino V (2013). **Brain Computer Interface Development Based on Recurrent Neural Networks and ANFIS Systems**. En Melin, P., Castillo, O. (Eds.) Soft Computing Applications in Optimization, Control and Recognition, pp. 215-236, Springer.

[49] Gómez-Gil, P., Ramírez-Cortes, J. M., Pomares Hernández, S. E., Alarcón-Aquino, V. (2011). **A neural network scheme for long-term forecasting of chaotic time series**. *Neural Processing Letters*, 33(3):215-233.

lores a futuro de una serie. Hay una gran cantidad de técnicas que se utilizan para hacer las predicciones, como las redes neuronales artificiales.[50]

7.3.6. Visión Computacional[51]

La visión computacional es un área de gran amplitud en términos de investigación. Intuitivamente estudia el procesamiento de información visual en formato digital, como la que se genera por las cámaras de video, cámaras fotográficas, imágenes en la web, *scanners* digitales, entre otras. Desde sus inicios ha habido tres problemas fundamentales a resolver dada una imagen de entrada:[52] i) construir una representación de los elementos atómicos que componen a una imagen, como las esquinas o intersecciones que se forman en la escena, en ventanas, puertas, pisos, objetos, o bien los bordes o líneas que surgen de las formas geométricas contenidas en la escena; ii) describir las propiedades de la escena tales como luminosidad, rugosidad, textura, etc., y iii) inducir una representación en 3-D de los objetos que aparecen en la escena.

Si bien la visión computacional se motiva en la visión natural, su objetivo no es imitar el proceso biológico sino más bien desarrollar algoritmos que permitan resolver tareas que un humano resuelve de manera natural, como reconocer rostros, seguir con la mirada un objetivo, calcular distancias mediante la observación visual de la escena, analizar videos,[53] comprender el lenguaje de señas, reconocer la geometría de una habitación en una imagen, etc.[54] Éstas y muchas otras tareas visuales parecen sencillas para nosotros

[50] Gómez-Gil, P. (2007)**. Long Term Prediction, Chaos and Artificial Neural Networks. Where is the meeting point?** *Engineering Letters*, (15)1:1-5.

[51] Ver la sección correspondiente en el capítulo de Robótica de Servicio en este mismo texto.

[52] Marr, D. Vision: **A computational investigation into the human representation and processing of visual information,** The MIT Press. 1982.

[53] Campos, Y., Sossa, H., Pajares, G. (2016). **Spatio-temporal analysis for obstacle detection in agricultural videos.** *Applied Soft Computing,* 45:86-97.

[54] Osuna-Coutiño, J. A., Martínez-Carranza, J., Arias-Estrada, M., Mayol-Cuevas, W. (2016). **Dominant Plane Recognition in Interior Scenes from a Single Image**. En *Proc. of the International Conference on Pattern Recognition.* pp. 1923-1928.

pero pueden demandar grandes cantidades de recursos computacionales para un sistema artificial.

7.3.7. Audición Robótica[55]

La audición robótica es una sub-especialidad del análisis de escenas auditivas. De la misma forma que las escenas visuales se constituyen por individuos con propiedades y relaciones espaciales, las escenas auditivas contienen individuos que corresponden a las fuentes sonoras, con sus respectivas propiedades y relaciones acústicas. La audición robótica tiene como propósito construir una representación o imagen de la escena acústica de manera análoga al proceso de visión computacional cuyo objetivo es construir una representación de la escena visual. El primer paso es separar a las fuentes sonoras, lo que corresponde con indentificar a los individuos de la escena visual, para llevar a cabo, por ejemplo, reconocimiento de voz. Se le refiere como "robótica" porque normalmente se lleva a cabo por robots —como de servicio o de rescate— aunque también es posible incorporarla a dispositivos más pequeños como teléfonos móviles o dispositivos de ayuda auditiva. En México se ha implementado la localización de múltiples locutores como parte de una aplicación de mesero en un robot de servicio. Gracias a esta funcionalidad los comensales puede llamar al robot por voz para pedirle que tome la orden.[56]

7.3.8. Vehículos Autónomos

El crecimiento vertiginoso de las grandes ciudades así como la necesidad de movilidad de sus habitantes han llevado a los científicos en AS/RP y otras áreas a buscar la construcción de vehículos autónomos de transporte urbano, capaces de desplazarse sin conductores humanos por calles y carreteras. Éste es un proyecto que presenta retos científicos en el área de AS/RP muy impor-

[55] Ver el capítulo de Robótica de Servicio en este mismo texto.

[56] Rascon, C., Meza, I.V., Fuentes, G., Salinas, L., Pineda, L. (2015). **Integration of the Multi-DOA Estimation Functionality to Human-Robot Interaction.** *International Journal of Advanced Robotic Systems*, 12(8).

tantes, pues los autos autónomos deben procesar señales digitales y clasificarlas en tiempo real, a fin de mantenerse en los carriles asignados, respetar las reglas de tránsito, identificar peatones y otros obstáculos que puedan aparecer en su camino, localizar a otros vehículos cercanos, definir la ruta que seguirán para llegar a su destino, etcétera.

El departamento de Matemáticas y Ciencias de la Computación de la Universidad Libre de Berlín desarrolla desde 2006 automóviles autónomos. El más reciente, llamado *MadeInGermany*, es un automóvil VW Passat acondicionado con video cámaras, sistemas de escaneo basados en rayos láser, radares y una unidad de navegación GPS. Las medidas obtenidas por los sensores son procesadas por varias computadoras embebidas que transmiten esta información a una computadora principal. Una computadora de control contiene un mapa de la ciudad marcado con direcciones GPS que se usa para trazar la ruta del auto. El *MadeInGermany* se ha auto-conducido en varias ciudades del mundo, incluyendo a la Ciudad de México.[57] Varios investigadores y estudiantes mexicanos han participado en este proyecto en el marco del Año Dual Alemania-México 2016-2017 y el 80 aniversario del Instituto Politécnico Nacional, el cual lleva a cabo un proyecto bilateral titulado "Visiones de Movilidad Urbana". Este proyecto incluyó la donación por parte de Alemania de 10 vehículos a escala, similares a *MadeInGermany*, a varias instituciones de educación superior mexicanas, con el objetivo de entrenar a estudiantes en la programación de vehículos autónomos, promover intercambios académicos y realizar demostraciones y competencias.[58]

7.4. Ejemplos de aplicaciones desarrolladas en México

Desde el punto de vista de la aplicación, AS/RP ofrece nuevas aproximaciones algorítmicas en áreas muy diferentes. Por ejemplo, en la biomédica, desde

[57] Rojas, R. **Autonomous Cars (2006-2015)**, Freie Universitat Berlin, Department of Mathematics and Computer Science. http://dcis.inf.fu-berlin.de/rojas/autonomous-cars-2006-2015/

[58] Embajada Alemana Ciudad de México. **Visiones de movilidad Urbana**. http://www.mexiko.diplo.de/Vertretung/mexiko/es/01-DEJahr/20160627RaulRojasEntrega.html

la prevención y el diagnóstico hasta el monitoreo, seguimiento y tratamiento de una enfermedad; en el área industrial, desde energías renovables y basadas en combustibles fósiles hasta comunicaciones, ingeniería civil, materiales, etc; en el área de la genética, desde la identificación de las secuencias de ADN y ARN, su transcripción en genes y proteínas hasta la explicación de la transmisión de la herencia biológica de generación en generación; otras áreas incluyen la climatología, economía, política y ciencias sociales. Además, AS/RP se usa para observar fenómenos estáticos, como el reconocimiento de una placa de un coche en una imagen, o fenómenos dinámicos, como el desgaste por rozamiento de los materiales. También se pueden observar fenómenos muy rápidos mediante simulaciones estocásticas, como la resolución espectroscópica del tiempo de vuelo de los fotones a medida que viajan por un tejido biológico. Se pueden valorar cambios relevantes en conjuntos de datos de cardinalidad limitada como ensayos clínicos con una población de muestra pequeña, así como lidiar con enormes volúmenes de datos provenientes de muy diferentes fuentes, como en la predicción del tiempo meteorológico. Se pueden encontrar patrones sutiles entre una cantidad importante de ruido como el paso de un planeta por delante de su estrella (a través de la atenuación que produce su brillo en su observación desde la tierra) o "ver" patrones que exceden la parte observada de un fenómeno, como en la estimación de la evolución del proceso de sanado del agujero de la capa de ozono.

En ingeniería, en el Centro de Investigación en Computación del Instituto Politécnico Nacional (CIC-IPN) se han utilizado técnicas de AS/RP para identificar anillos defectuosos en los cilindros de un motor,[59] encontrar daños en las venas de la retina de un paciente diabético,[60] identificar los rostros en una fotografía antes de que la cámara se dispare,[61] clasificar imágenes

[59] López Cárdenas, R. **Método híbrido para el diagnóstico de fallas en motores de inducción trifásicos,** Tesis Doctoral en Ciencias de la Computación. Instituto Politécnico Nacional, Centro en Investigación en Computación, 2008.

[60] Villalobos-Castaldi, F. M., Felipe-Riverón, E. M., Sánchez-Fernández, L. P. (2010). **A fast, efficient and automated method to extract vessels from fundus images**. *Journal of Visualization,* 13(3):263-270.

[61] Guzman, E., Alvarado, S., Pogrebnyak, O., Yanez, C. (2007). **Image recognition pro-**

de la retina para diagnóstico de enfermedades,[62] así como identificar plantas en sembrados a través de análisis de textura.[63]

Los investigadores del grupo de visión de la Universidad Panamericana han diseñado nuevos algoritmos para detectar movimiento en imágenes médicas;[64] se han implementado técnicas de segmentación de imágenes para prevenir enfermedades cardiacas;[65,66] se han instrumentado robots con visión computacional para minimizar sensores,[67] y se han identificado nuevas técnicas de reconocimiento de actividades humanas mediante el uso de sensores vestibles.[68]

En la Universidad Autónoma de Baja California, el reconocimiento de patrones en el ADN se ha aplicado a la identificación de mamíferos superio-

---

cessor based on morphological associative memories. En *Proc. of Electronics, Robotics and Automotive Mechanics Conference, 2007*. CERMA 2007, IEEE. pp. 260-265.

[62] Vega, R.,Sánchez, G., Falcón, L. E. , Sossa, H., Guevara, E. (2015). **Retinal vessel extraction using Lattice Neural Networks with dendritic processing**. *Computers in Biology and Medicine,* 58:20-30.

[63] Campos, Y., Sossa, H., Pajares, G. (2017). **Comparative analysis of texture descriptors in maize fields with plants, soil and object discrimination**. *Precision Agriculture*. doi: 10.1007/s11119-016-9483-4, pp. 1-19.

[64] Moya-Albor, E., Escalante-Ramírez, B., Vallejo, E., (2013). **Optical Flow Estimation in Cardiac CT Images Using the Steered Hermite Transform**. *Signal Processing: Image Communication*, 28(3): 267–291.

[65] Barba-J, L., Moya-Albor, E., Escalante-Ramírez, B., Brieva, J., Vallejo Venegas, E., (2016). **Segmentation and Optical Flow Estimation in Cardiac CT Sequences Based on a Spatiotemporal PDM with a Correction Scheme and the Hermite Transform**. *Computers in Biology and Medicine*, 69: 189–202.

[66] Moya-Albor, E., Mira, C., Brieva, J., Escalante-Ramírez, B., Vallejo Venegas, E., (2017). **3D optical flow estimation in cardiac CT images using the Hermite transform**. En *Proc. of the SPIE 12th International Symposium on Medical Information Processing and Analysis, International Society for Optics and Photonics,* doi: 10.1117/12.2256777.

[67] Ponce, H., Moya-Albor, E., Brieva, J. (2016). **A Novel Artificial Organic Controller With Hermite Optical Flow Feedback for Mobile Robot Navigation.** En Ponce, P. Molina-Gutierrez, A. Rodriguez, J (Eds.), New Applications of Artificial Intelligence, InTech, pp. 145–169.

[68] Ponce, H., Martínez-Villaseñor, L., Miralles-Pechuán, L. (2016). **A Novel Wereable Sensor-Based Human Activity Recognition Approach Using Artificial Hydrocarbon Networks**. *Sensors*, 16(7): 1033.

res para la selección genómica.[69] Otras aplicaciones incluyen la construcción de prototipos para la detección de cáncer cérvico-intrauterino realizada en la Universidad Veracruzana,[70] y un sistema de corrección de errores de escritura en niños por fallas en la orientación espacial realizado en la Universidad Popular Autónoma del Estado de Puebla.[71]

En el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) se desarrolló un sistema de rehabilitación basado en el seguimiento de la mano de un paciente,[72] y un sistema para la definición de factores para la identificación de errores en el proceso de producción de tubos de acero.[73] Asimismo, en el INAOE se han realizado prototipos para la clasificación de señales de electroencefalogramas, lo que permitirá apoyar el diagnóstico de enfermedades cerebrales[74] y la clasificación de leucocitos para identificación

[69] Salomón-Torres, R., González-Vizcarra, V. M., Medina-Basulto, G. E., Montaño-Gómez, M. F., Mahadevan, P., Yaurima-Basaldúa, V. H., Villa-Angulo, R. (2015). **Genome-wide identification of copy number variations in Holstein cattle from Baja California, Mexico, using high-density SNP genotyping arrays**. *Genetics and Molecular Research,* 14(4):11848–11859.

[70] Acosta-Mesa, H. G., Rechy-Ramírez, F., Mezura-Montes, E., Cruz-Ramírez, N., & Jiménez, R. H. (2014). **Application of time series discretization using evolutionary programming for classification of precancerous cervical lesions**. *Journal of biomedical informatics*, 49:73–83.

[71] Castro-Manzano, J. M., Reyes-Meza, V., Medina-Delgadillo, J. (2015). **{dasasap}, an App for Syllogisms**. En *Proc. of the Fourth International Conference on Tools for Teaching Logic* (TTL2015), arXiv:1507.03664, pp. 1–8.

[72] Sucar, L.E., Orihuela-Espina, F., Velázquez, R.L., Reinkensmeyer, D.J., Leder, R., Hernández-Franco, J. (2014). **Gesture Therapy: An upper limb virtual reality-based motor rehabilitation platform**. *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* 22(3):634–643.

[73] Ibargüengoytia, P.H., García, U.A., Herrera-Vega, J., Hernández-Leal, P., Morales, E.F., Sucar, L.E., Orihuela-Espina, F. (2013). **On the estimation of missing data in incomplete datasets: autoregressive Bayesian networks**. En *Proc. of the Eighth International Conference on Systems* (ICONS 2013)*,* pp. 111–116 (2013)

[74] Juárez-Guerra, E., Alarcón-Aquino, V., Gómez-Gil, P. (2014). **Epilepsy Seizure Detection in EEG Signals Using Wavelet Transforms and Neural Networks.** En Elleithy, K., Sobh, T. (Eds.). New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering, Springer, pp. 261–269.

de leucemia.[75] La empresa Probayes Américas en Puebla desarrolló un sistema para reconocimiento de rostros.[76]

Asimismo en el Tecnológico de Chihuahua se construyó un sistema de interacción con objetos en escenarios virtuales para rehabilitación de miembros superiores e inferiores,[77] así como un sistema para el reconocimiento de nudos en hojas de madera.[78] En el Instituto Tecnológico de Tijuana se han realizado aplicaciones para biometría multimodal,[79] clasificación de imágenes[80] y reconocimiento de eco-cardiogramas.[81] En la universidad de las Américas, Puebla, se han desarrollado aplicaciones para segmentación automática y reconocimiento de CAPTCHA's (prueba automática y pública para diferenciar ordenadores de humanos).[82]

---

[75] Ramirez-Cortes, J. M., Gomez-Gil, P., Alarcon-Aquino, V., Gonzalez-Bernal, J., Garcia-Pedrero, A. (2010). **Neural networks and SVM-based classification of leukocytes using the morphological pattern spectrum**. En Melin, P., Kacprzyk, J. Pedrycz, W. (Eds.), Soft Computing for Recognition Based on Biometricsm Springer, pp. 19–35.

[76] Carro, R. C., Larios, J. M. A., Huerta, E. B., Caporal, R. M., Cruz, F. R. (2015). **Face Recognition Using SURF**. En *Proc. of International Conference on Intelligent Computing,* Springer International Publishing, pp. 316–326.

[77] Arias-Enriquez, O., Chacon-Murguia, M. I., Sandoval-Rodriguez, R. (2012). **Kinematic analysis of gait cycle using a fuzzy system for medical diagnosis**. En *Proc. of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS* 2012), IEEE, pp. 1–6.

[78] Ramírez Alonso, G. M. D. J., Chacón Murguía, M. I. (2005). **Clasificación de Defectos en Madera utilizando Redes Neurales Artificiales.** *Computación y Sistemas,* 9(1):17 –27.

[79] Hidalgo, D., Castillo, O., Melin, P. (2009). **Type-1 and type-2 fuzzy inference systems as integration methods in modular neural networks for multimodal biometry and its optimization with genetic algorithms**. *Information Sciences,* 179(13):2123 –2145.

[80] Valdez, F., Castillo, O., Melin, P. (2016, July). **Ant colony optimization for the design of Modular Neural Networks in pattern recognition**. En *Proc. of 2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE. (pp. 163–168).

[81] González, B., Valdez, F., Melin, P., Prado-Arechiga, G. (2015). **Fuzzy logic in the gravitational search algorithm enhanced using fuzzy logic with dynamic alpha parameter value adaptation for the optimization of modular neural networks in echocardiogram recognition**. *Applied Soft Computing*, 37:245–254.

[82] Starostenko, O., Cruz-Perez, C., Uceda-Ponga, F., Alarcon-Aquino, V. (2015). **Breaking text-based CAPTCHAs with variable word and character orientation**. *Pattern Recognition*, 48(4), 1101–1112.

Por su parte, en el IIMAS, UNAM, se ha mantenido una actividad constante en el estudio y análisis de imágenes biomédicas[83] y de imágenes provenientes de barredores multiespectrales.[84] Se ha trabajado también en la representación, análisis y reconocimiento de patrones estructurales con enfoque en la forma de los objetos. Entre algunos desarrollos se puede mencionar la Compacidad Discreta.[85] Esta medida se ha usado para analizar tumores cervicouterinos,[86] para calcular la compacidad en zonas ecológicas[87] y para clasificar tumores de mama,[88] entre otras aplicaciones, y se propuso como el nuevo estándar mundial para la clasificación de tumores cervicouterinos.[89] También se ha trabajado en el estudio de códigos de cadenas para representar curvas y árboles en dos y tres dimensiones.[90] Con este código se generaron 282,429,536,481 curvas 3-D para la clasificación de nodos,[91] superando al record anterior de 70,000,000,000 de curvas.[92] En procesamiento de imágenes, señales y visualización se han hecho contribuciones en el campo de recons-

[83] Martinez-Perez, M. E., Hughes, A. D., Stanton, A.V., Thom, S. A. Chapman, N., Bharath, A. A., Parker, K.H. **Retinal vascular tree morphology: a semiautomatic quantification**, *IEEE Transactions on Biomedical Engineering,* 49(8):912–917, 2002.

[84] Guzmán, A. **Reconfigurable Geographic Data Bases**. In Pattern Recognition in Practice. E. S. Gelsema and Laveen N. Kanal (eds), 99–112, North Holland, 1980.

[85] Bribiesca, E. (2008). **An Easy Measure of Compactness for 2D and 3D Shapes**. *Pattern Recognition,* 41(2): 543–554.

[86] Braumann, U. et al. (2005). **Three-dimensional reconstruction and Quantification of cervical carcinoma invasion fronts from histological serial sections**. *IEEE Transactions on Medical Imaging,* 24(10):1286–1307.

[87] Bogaert, J., Rousseau, R., Van Hecke, P., Impens, I. (2000). **Alternative area-perimeter ratios for measurement of 2D shape compactness of habitats**. *Applied Mathematics and Computation,* 11 (2000) 71–85.

[88] Moon, W. K. et al. (2011). **Computer-aided diagnosis for the classification of breast masses in automated whole breast ultrasound images**. *Ultrasound in Medicine and Biology* 37(4):539–548.

[89] Einenkel, J. et al. (2007). **Evaluation of the invasion front pattern of squamous cell cervical carcinoma by measuring classical and discrete compactness**. *Computerized Medical Imaging and Graphics,* 31:428–435.

[90] Bribiesca, E. (2000). **A Chain Code for Representing 3D Curves**. *Pattern Recognition*, Vol. 33, No. 5, pp. 755–765.

[91] Bribiesca, E. (2005). **A Method for Computing Families of Discrete Knots Using Knot Numbers.** *Journal of Knot Theory and Its Ramifications*, 14(4): 405–424.

[92] Hayes, B. (1997). **Square Knots**. *American Scientist,* 85(6):506–510.

trucción de imágenes a partir de proyecciones, en particular, proponiendo la metodología de Superiorización.[93,94] También, se han hecho contribuciones en el campo de segmentación de imágenes usando principios de lógica borrosa difusa.[95] Asimismo, en el área de robótica auditiva se ha trabajado en la localización de múltiples locutores móviles en un ambiente real con un número pequeño de micrófonos.[96]

El grupo de aplicaciones de IA del Instituto Nacional de Electricidad y Energías Limpias (INEEL, antes IIE) ha realizado diferentes proyectos de AS/RP en el sector energético. En todos ellos se utilizan redes Bayesianas como la técnica principal para modelar y reconocer patrones. Por ejemplo, en el proyecto de diagnóstico de transformadores de potencia mediante señales de vibración para la empresa Prolec-GE se implantaron sensores de vibración en los tanques de los transformadores y se desarrolló un sistema que aprende un modelo probabilista del espectro de frecuencias de la vibración del transformador que permite detectar desviaciones del comportamiento normal.[97]

Otro proyecto desarrollado en el INEEL consistió en el desarrollo de un sistema de diagnóstico de turbinas eólicas que utiliza señales históricas del Sistema de Control Supervisor y Adquisición de Datos de la turbina (SCADA) para aprender modelos de comportamiento de la turbina bajo diferentes contextos de operación, como con vientos de baja o alta intensidad. Con dichos modelos se identifican los patrones de valores de las variables que

[93] Herman, G. T., Garduño, E., Davidi, R., Censor, Y. **Superiorization: An optimization heuristic for medical physics**, por publicarse en Medial Physics, doi: 10.1118/1.4745566

[94] Garduño, E., Herman, G. T. (2017). **Computerized tomography with total variation and shearlets**. *Inverse Problems,* 33(4). doi: 10.1088/1361-6420/33/4/044011

[95] Carvalho, B., Garduño, E., Santos, T., Oliviera, L., Silva Neto, J. (2014). **Fuzzy segmentation of video shots using hybrid color spaces and motion information.** *Pattern Analysis and Applications* 17(2):249–264.

[96] Rascon, C., Fuentes, G., Meza, I.V. (2015). **Light weight multi-DOA tracking of mobile speech sources**. *EURASIP Journal on Audio, Speech, and Music Processing* 2015:11 .

[97] Ibargüengoytia, P.H., Liñan, R., Betancourt, E. (2009) **Transformer Diagnosis Using Probabilistic Vibration Models**. En: Aguirre, A.H., Borja, R.M., García, C.A.R. (Eds.) *MICAI 2009: Advances in Artificial Intelligence.* LNCS, 5845. Springer.

representan el comportamiento normal de la turbina que a su vez permiten identificar desviaciones tempranas a dicho comportamiento.[98]

7.5. La comunidad científica

La comunidad científica mexicana tiene una presencia importante relacionada a AS/RP, la cual es difícil de cuantificar dado que se reporta en una gran variedad de medios y en prácticamente todas las disciplinas de la computación nombradas en este libro. Algunas de las instituciones que generan conocimiento en esta área son: el Centro de Investigación en Matemáticas (CIMAT), el Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV), el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), el Instituto Nacional de Electricidad y Energías Limpias (INEEL) (antes Instituto de Investigaciones Eléctricas), el Instituto Politécnico Nacional (IPN), el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), la Universidad Nacional Autónoma de México (UNAM), la Universidad Autónoma de Baja California (UABC), la Universidad Autónoma de San Luis Potosí (UASLP) y la Universidad Panamericana (UP).

Los congresos que organiza la comunidad mexicana de AS/RP incluyen: la Conferencia Internacional en Ingeniería Eléctrica, Ciencias de la Computación y Control Automático (CCE), el Congreso Iberoamericano de Reconocimiento de Patrones (CIARP), el Congreso Mexicano de Inteligencia Artificial (COMIA), la Conferencia Internacional en Electrónica, Comunicación y Computadoras (CONIELECOM), el Encuentro Nacional en Ciencias de la Computación, el Congreso Iberoamericano de Inteligencia Artificial (IBERAMIA), el Congreso Mexicano/Internacional de Inteligencia Artificial (MICAI) y la Conferencia Internacional en Cómputo Ubicuo e Inteligencia Ambiental (UCAMI).

[98] Ibargüengoytia, P.H., Garcia, U.,Reyes, A., Borunda,M. (2016). **Anomalies detection in the behavior of process using the sensor validation theory.** *Advances in Artificial Intelligence – IBERAMIA 2016, Lecture Notes in Computer Science,* 10022.

Asimismo, las asociaciones científicas mexicanas que se relacionan con el análisis de señales y reconocimiento de patrones incluyen la Sociedad Iberoamericana de Inteligencia Artificial (IBERAMIA), la Sociedad Mexicana de Inteligencia Artificial (SMIA), la Sociedad Mexicana de Ciencias de la Computación (SMCC), la Asociación Mexicana para la Visión computacional, Neuro-Computación y Robótica (MACVNR) y la Academia Mexicana de Computación (AMEXCOMP). Algunas de las asociaciones internacionales relacionadas a AS/RP que cuentan con representatividad mexicana incluyen a la *International Association of Pattern Recognition* (IAPR), al *Institute of Electrical and Electronics Engineers* (IEEE) y a la *Association of Computing Machinery* (ACM), a través de secciones especiales y/o capítulos relacionados al AS/RP, Inteligencia Artificial e Inteligencia Computacional.

Por otra parte, la comunidad mexicana participa activamente en congresos internacionales, como por ejemplo: *The Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI)*, la Conferencia Iberoamericana de Inteligencia Artificial (IBERAMIA), *el IEEE International Instrumentation and Measurement Technology Conference* (*I2MTC)*, el *International Congress on Pattern Recognition* (ICPR), la *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, la *International Joint Conference on Neural Networks* (IJCNN), el *IEEE International Symposium on Biomedical Imaging* (ISBI) *y el World Congress on Computational Intelligence* (WCCI).

7.6. Perspectivas

El futuro de AS/RP en México es prometedor. La cantidad de información disponible aumenta cada día debido a las múltiples maneras de obtener datos a través de sensores, aplicaciones en dispositivos móviles, el Internet, los repositorios de información gubernamentales, etc. Esta inmensa riqueza de datos presenta el reto de encontrar algoritmos eficientes, así como nuevas maneras de programación para controlar los costos que conlleva su procesamiento. En este sentido, los avances recientes del cómputo de alto desempeño y del su-

per-cómputo permiten la ejecución de algoritmos de AS/RP que utilizan grandes cantidades de datos, lo cual era imposible hace apenas unos años.

Cada día surgen nuevas áreas asociadas con AS/RP, como el reconocimiento de actividades humanas a través de sensores portables que ayudan a cuantificar la actividad física, motivar y poner metas apropiadas a cada persona. Estos sistemas pueden contribuir significativamente a mejorar la calidad de vida, salud y seguridad de ancianos, niños y personas discapacitadas. Asimismo, la madurez alcanzada por las comunidades de investigación mexicanas ha dado lugar a que se desarrollen trabajos interdisciplinarios de trascendencia y aumenten las aplicaciones de la alta tecnología para beneficio de la comunidad.

Otra área de oportunidad surge de la necesidad de contar con grandes repositorios de información en diferentes ámbitos o dominios de aplicación para apoyar la toma de decisiones en tiempo real, mediante el aprovechamiento de las capacidades de clasificación, predicción e inferencia de las técnicas de inteligencia artificial, minería de datos y los modelos de aprendizaje profundo. Estos repositorios estarán disponibles en la medida en que se mejoren los sistemas de captura masiva de información y de protección de datos. Se requiere también contar con repositorios de datos típicos que permitan validar las soluciones propuestas y compararlas formalmente con otras.

Desde el punto de vista de la educación urge considerar los conocimientos básicos asociados a AS/RP y, en general de las Ciencias de la Computación, como parte de la currícula de las carreras de ingeniería y ciencias. Con respecto a la enseñanza en niveles básicos, desde hace varios años se ha promovido la formación del pensamiento computacional en la enseñanza media superior como una herramienta de solución de problemas basada en el pensamiento crítico, pero aún hay un largo camino que recorrer.[99]

[99] Instituto Nacional de Astrofísica, Óptica y Electrónica. **Programa de pensamiento computacional para la educación media y superior en México**. http://www.pensamientocomputacional.org/

En relación a la metodología de investigación, en los últimos años se ha madurado en la formalización y homogeneización de la terminología. Sin embargo, todavía es necesario alcanzar la precisión de otras áreas del conocimiento como la física, la química o las matemáticas. Por otra parte, es necesario establecer procedimientos experimentales óptimos y estándares de evaluación para aumentar la calidad de la investigación y fomentar la generación de publicaciones de alto impacto. Las oportunidades de desarrollo y aplicación del Análsis de Señales y Reconocimiento de Patrones seguirán creciendo en los próximos años con muy buenas perspectivas.

8. Computación Evolutiva

La computación evolutiva es un área de las ciencias de la computación que se enfoca al estudio de las propiedades de una serie de metaheurísticas estocásticas (a las cuales se les denomina "Algoritmos Evolutivos") inspiradas en la teoría de la evolución de las especies formulada por Charles Darwin, según la cual los individuos más aptos a su ambiente tienen una mayor probabilidad de sobrevivir.

Una metaheurística es un procedimiento de alto nivel que aplica una regla o conjunto de reglas que se basa(n) en una fuente de conocimiento. El énfasis de las metaheurísticas es explorar el espacio de búsqueda de manera relativamente eficiente; es decir, mejor que una búsqueda exhaustiva, aunque no necesariamente óptima. Las metaheurísticas suelen considerarse también como técnicas de búsqueda y optimización de uso general pese a sus restricciones teóricas.[1] Además, suelen requerir de poca información específica del problema y son útiles cuando el espacio de búsqueda es muy grande, poco conocido y/o con dificultades para explorarlo. Cuando se usan para optimización, no pueden garantizar, en general, que convergerán a la mejor solución posible (es decir, al óptimo global del problema), si bien en la práctica suelen producir aproximaciones razonablemente buenas en tiempos razonablemente cortos.

[1] Wolpert, D. H., Macready, W. G. (1997). **No Free Lunch Theorems for Optimization**. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Los Algoritmos Evolutivos (AEs) son metaheurísticas estocásticas porque su comportamiento se basa en el uso de números aleatorios y, por lo tanto, no necesariamente generan el mismo resultado cada vez que se ejecutan. Los AEs se han aplicado exitosamente a diversos tipos de problemas.[2] A lo largo de los años, se han desarrollado al menos dos tipos de AEs con base en el tipo de problemas que pretenden resolver: i) los diseñados para resolver problemas de optimización (sobre todo no lineal y combinatoria) y ii) los diseñados para resolver problemas de aprendizaje de máquina (sobre todo de clasificación). Sin embargo, la investigación en computación evolutiva no se ha limitado al desarrollo de nuevos algoritmos, sino también al diseño de nuevos operadores, mecanismos de selección y de representación, así como al modelado matemático de los AEs, el estudio de las fuentes de dificultad en un espacio de búsqueda y su correlación con los operadores de un AE y la hibridación de los AEs con otro tipo de técnicas.

El resto de este capítulo está organizado de la siguiente manera. Primero se revisan los antecedentes históricos de los AEs. Posteriormente, se describen los tres principales tipos de AEs que existen (las estrategias evolutivas, la programación evolutiva y los algoritmos genéticos), así como una variante muy popular de los algoritmos genéticos conocida como "programación genética". En dicha descripción se incluyen algunas aplicaciones relevantes, así como una breve discusión de trabajos de investigación que se han realizado en México. En la parte final del capítulo se mencionan otras metaheurísticas bio-inspiradas con las que actualmente se trabaja en nuestro país. También se proporciona una breve discusión sobre los orígenes de esta área de investigación en México y se concluye con una breve discusión de sus perspectivas.

8.1. Antecedentes históricos

La computación evolutiva tiene una historia larga y fascinante. En sus orígenes, varios investigadores propusieron de manera totalmente independiente

[2] Fogel, D. B. **Evolutionary Computation: Toward a New Philosophy of Machine Intelligence**, IEEE Press, Piscataway, NJ, USA, 1995.

diversos modelos computacionales con objetivos específicos, los cuales tenían en común su inspiración en la evolución natural. Con el tiempo, las evidentes similitudes entre estas técnicas llevaron a acuñar el término genérico "Algoritmo Evolutivo" en la década de los noventa del siglo pasado.[3] En esta sección, se discuten brevemente los acontecimientos más importantes de su historia. Cabe indicar que muchas de las bases de la computación evolutiva se desarrollaron en los sesenta y los setenta, cuando se propusieron los tres enfoques más populares dentro de la computación evolutiva: i) la programación evolutiva, ii) las estrategias evolutivas y iii) los algoritmos genéticos. Si bien los tres enfoques se inspiraron en las mismas ideas, difieren en sus detalles, como el esquema para representar soluciones y el tipo de operadores adoptados. Adicionalmente se discuten otras propuestas que se consideran importantes por su valor histórico pese a que su uso no llegó a popularizarse.

Wright[4] y Cannon[5] fueron los primeros investigadores que vieron a la evolución natural como un proceso de aprendizaje en el cual la información genética de las especies se cambia continuamente a través de un proceso de ensayo y error. De esta forma, la evolución puede verse como un método cuyo objetivo es maximizar la capacidad de adaptación de las especies en su ambiente. Wright fue más lejos e introdujo el concepto de "paisaje de aptitud" como una forma de representar el espacio de todos los genotipos posibles junto con sus valores de aptitud. Este concepto se ha utilizado extensamente en computación evolutiva para estudiar el desempeño de los AEs.[6] Adicionalmente, Wright también desarrolló uno de los primeros estudios que relacionan a la selección con la mutación, y concluyó que éstas deben tener

[3] Fogel, D. B. **Evolutionary Computation: The Fossil Record**, The Institute of Electrical and Electronic Engineers, New York, USA, 1998.

[4] S. Wright, S. (1932). **The roles of mutation, inbreeding, crossbreeding and selection in evolution**. En *VI International Congress of Genetics*, Vol. 1, pp. 356-366.

[5] Cannon, W. B., **The wisdom of the body**, W. W. Norton & Company, inc., 1932.

[6] Richter, H. (2014). **Fitness landscapes: From evolutionary biology to evolutionary computation**. En H. Richter and A. Engelbrecht (eds.), *Recent Advances in the Theory and Application of Fitness Landscapes, Emergence, Complexity and Computation*, Springer Berlin Heidelberg, Vol. 6, pp. 3–31.

un balance adecuado para tener éxito en el proceso evolutivo. Las ideas de Wright se extendieron por Campbell,[7] quien afirmaba que el proceso de variación ciega combinado con un proceso de selección adecuado constituye el principio fundamental de la evolución.

Turing puede ser considerado también uno de los pioneros de la computación evolutiva, pues reconoció una conexión (que a él le parecía obvia) entre la evolución y el aprendizaje de máquina.[8] Específicamente, Turing afirmaba que para poder desarrollar una computadora capaz de pasar la famosa prueba que lleva su nombre (con lo cual sería considerada "inteligente"), podría requerirse un esquema de aprendizaje basado en la evolución natural. Turing ya había sugerido antes que este tipo de métodos basados en la evolución podrían usarse para entrenar un conjunto de redes análogas a lo que hoy conocemos como "redes neuronales".[9] De hecho, Turing incluso usó el término "búsqueda genética" para referirse a este tipo de esquemas. Sin embargo, debido a que su supervisor consideró que este documento lucía más como un ensayo juvenil que como una propuesta científica,[10] el trabajo se publicó hasta 1968,[11] cuando existían ya algunos algoritmos evolutivos y las ideas de Turing al respecto acabaron por ser ignoradas.

Hacia finales de los cincuenta y principios de los sesenta se realizaron varios avances relevantes que coincidieron con el período en que las computadoras digitales se volvieron más accesibles. Durante dicho período, se

[7] Campbell, D. T. (1960). **Blind Variation and Selective Survival as a General Strategy in Knowledge-Processes**, En M. C. Yovits, S. Cameron (eds.), *Self-Organizing Systems*, Pergamon Press, New York, USA, pp. 205–231.

[8] Turing, A. M. (1950). **Computing Machinery and Intelligence**. *Mind,* 59:433–460.

[9] Turing, A. M. (1948). **Intelligent Machinery**. Report, National Physical Laboratory, Teddington, UK.

[10] Burgin M., Eberbach, E. (2013). **Recursively Generated Evolutionary Turing Machines and Evolutionary Automata**. En X.-S. Yang (ed.), *Artificial Intelligence, Evolutionary Computing and Metaheuristics*, Studies in Computational Intelligence, Vol. 427, Springer, Berlin, Germany, pp. 201–230.

[11] Evans, C. R., Robertson, A. D. J. (eds.) **Cybernetics: key papers**, University Park Press, Baltimore, Maryland, USA, 1968.

llevaron a cabo varios análisis de la dinámica poblacional que aparece durante la evolución. Una revisión del estado del arte en torno a este tema se presentó por Crosby en 1967,[12] quien identificó tres tipos de esquemas:

1. Métodos basados en el estudio de formulaciones matemáticas que emergen en el análisis determinista de la dinámica poblacional. Estos esquemas usualmente presuponen un tamaño de población infinito y analizan las distribuciones de los genes[13] bajo distintas circunstancias.

2. Enfoques que simulan la evolución pero que no representan explícitamente una población.[14] En estos métodos, las frecuencias de los diferentes genotipos[15] potenciales se almacenan para cada generación y resultan en un esquema no escalable en términos del número de genes. Adicionalmente, se requiere una representación matemática del sistema evolutivo.

3. Esquemas que simulan la evolución y que sí mantienen una representanción explícita de la población. Fraser[16] fue pionero en este tipo de métodos, los cuales se describen en una serie de artículos publicados a lo largo de una década.[17] Las conclusiones principales que se derivaron de dichos estudios se reúnen en un libro seminal sobre el tema.[18] Desde

[12] Crosby, J. L. (1967). **Computers in the study of evolution**, *Science Progress Oxford*, 55:279–292.

[13] Un "gene" es una secuencia de ácido desoxirribonucleico (ADN) que ocupa una posición específica en un cromosoma. A su vez, un "cromosoma" es una estructura dentro del núcleo de una célula que contiene el material genético que conforma a un individuo.

[14] Crosby, J. L. (1960). **The Use of Electronic Computation in the Study of Random Fluctuations in Rapidly Evolving Populations**. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 242(697):550–572.

[15] Se llama "genotipo" a la información genética de un organismo. Esta información (contenida en los cromosomas del organismo) puede o no ser manifestada u observada en el individuo.

[16] Fraser, A. S. (1957). **Simulation of genetic systems by automatic digital computers. I. Introduction**, *Australian Journal of Biological Science*, 10:484–491.

[17] Fraser, A. S., Burnell, D. (1967). **Simulation of Genetic Systems XII. Models of Inversion Polymorphism**. *Genetics*, 57:267–282.

[18] Fraser, A. S., Burnell, D. **Computer models in genetics**, McGraw-Hill, New York, USA, 1970.

su creación, este modelo se aceptó ampliamente y se adoptó por varios investigadores. De hecho, aunque existen varios detalles de implementación en los que las simulaciones de Fraser difieren de los algoritmos evolutivos modernos, algunos autores consideran al trabajo de Fraser como la primera propuesta de un algoritmo genético.[19] Cabe mencionar que aunque a nivel de la implementación la diferencia principal entre la propuesta de Fraser y el algoritmo genético moderno radica en que la primera usaba cromosomas diploides (es decir, un par de cromosomas por individuo) mientras que el segundo usa cromosomas haploides (es decir, un solo cromosoma por individuo), conceptualmente, el trabajo de Fraser se centró en analizar la dinámica poblacional y no en resolver problemas, que es la aplicación típica de los algoritmos evolutivos.

En este mismo período otros investigadores propusieron esquemas similares a un algoritmo evolutivo. De entre ellos, destacan Friedman, Box y Friedberg por el impacto que su trabajo tuvo posteriormente. Friedman[20] propuso el diseño automático de circuitos de control usando un esquema inspirado en la evolución natural, al que denominó "retroalimentación selectiva." Su propuesta era muy diferente a los algoritmos evolutivos modernos. Por ejemplo, no manejaba la noción de población[21] ni de generación.[22] Además, no implementó su método y planteó ciertas presuposiciones que resultaron demasiado optimistas. No obstante, su trabajo se considera pionero dentro de un área que hoy se conoce como "hardware evolutivo", en la que se busca diseñar circuitos con algoritmos evolutivos (en los que se adopta frecuentemente implementaciones en hardware).

---

[19] David B. Fogel, **Evolutionary Computation: The Fossil Record**, The Institute of Electrical and Electronic Engineers, New York, USA, 1998.

[20] Friedman, G, J., **Selective Feedback Computers for Engineering Synthesis and Nervous System Analogy**, MSc Dissertation, University of California, Los Angeles, 1956.

[21] La mayor parte de los algoritmos evolutivos modernos operan sobre un conjunto de soluciones potenciales al problema que están resolviendo. A dicho conjunto de soluciones se le denomina "población".

[22] Los algoritmos evolutivos se ejecutan durante un cierto número de iteraciones, a las cuales se les denomina "generaciones".

Box[23] propuso una técnica a la que denominó "Operación Evolutiva" (*Evolutionary Operation*, o EVOP) para optimizar el manejo de procesos de la industria química. EVOP usa una solución base (el "padre") para generar varias soluciones adicionales (los "hijos"), modificando unos cuantos parámetros de producción a la vez. Posteriormente, un individuo se selecciona para pasar a la siguiente generación, a partir de cierta evidencia estadística. Este sistema no era autónomo, pues requería de asistencia humana en los procesos de variación y selección. De tal forma, EVOP puede considerarse como el primer algoritmo evolutivo interactivo.

La contribución más importante de Friedberg[24] fue su intento de generar automáticamente programas de computadora mediante un esquema evolutivo. En sus primeros trabajos,[25] no identificó ninguna relación específica entre su propuesta y la evolución natural; sin embargo, dicha relación se identificó en publicaciones posteriores de sus co-autores.[26] En sus primeros intentos, el objetivo era generar automáticamente programas pequeños con algunas funcionalidades simples basadas en un conjunto de instrucciones hechas a la medida. Para poder dirigir la búsqueda a regiones prometedoras, se incorporaba información dependiente del problema, mediante la definción de un esquema de variación diseñado específicamente para el problema a resolverse. Pese al uso de operadores especializados, el éxito de esta técnica fue limitado. Sin embargo, algunos de los análisis realizados por Friedberg y sus co-autores fueron particularmente importantes para algunos de los logros posteriores de la computación evolutiva. Entre sus contribuciones más importantes, destacan las siguientes:

---

[23] Box, G. E. P. (1957). **Evolutionary Operation: A Method for Increasing Industrial Productivity**. *Applied Statistics*, 6(2):81–101.

[24] Friedberg, R. M. (1958). **A Learning Machine: Part I**. *IBM Journal of Research and Development*, 2(1):2–13.

[25] Friedberg, R. M., Dunham, B., North, J. H. (1959). **A Learning Machine: Part II**. *IBM Journal of Research and Development*, 3(3):282–287.

[26] Dunham, B., Fridshal, D., Fridshal, R., North, J. H. (1963). **Design by natural selection**. *Synthese*, 15(1): 254–259.

- La idea de dividir las soluciones candidatas en clases es precursora de las nociones de paralelismo implícito[27] y de esquema[28] que propusiera años después John Holland.[29]

- Se probaron varios conjuntos de instrucciones, mostrándose, por primera vez, la influencia del mapeo del genotipo al fenotipo.

- Se utilizó un algoritmo de asignación de crédito para medir la influencia de las instrucciones, de manera aislada. Esta idea está muy relacionada con el trabajo que realizara años después Holland con los algoritmos genéticos y los sistemas de clasificadores.

Algunos artículos no llamaron mucho la atención cuando se publicaron originalmente, pero propusieron técnicas que se reinventaron varios años después. Por ejemplo, el trabajo de Reed, Toombs y Barricelli[30] proporcionó varias innovaciones que se asemejan mucho a los conceptos de auto-adaptación,[31] cruza[32] y coevolución[33] que se usan hoy en día en computación evolutiva.

[27] El "paralelismo implícito" es una propiedad atribuída a los algoritmos genéticos, la cual les permite explorar más eficientemente el espacio de búsqueda mediante un proceso de estimación de la calidad de las soluciones.

[28] Un "esquema" es una abstracción usada para representar las soluciones que procesa un algoritmo genético. Holland usó este concepto para analizar matemáticamente la forma en la que opera un algoritmo genético.

[29] Holland, J. H. **Adaptation in Natural and Artificial Systems**, University of Michigan Press, Ann Arbor, Michigan, USA, 1975.

[30] Reed, J., Toombs, R., Barricelli, N. A. (1967). **Simulation of biological evolution and machine learning: I. Selection of self-reproducing numeric patterns by data processing machines, effects of hereditary control, mutation type and crossing**". *Journal of Theoretical Biology*, 17(3):319–342.

[31] La "auto-adaptación" es un mecanismo que permite que un algoritmo evolutivo adapte por sí mismo los valores de sus parámetros.

[32] La "cruza" es un operador utilizado en los algoritmos evolutivos para hacer que dos individuos intercambien material genético y produzcan descendientes (o "hijos") que combinen dicho material.

[33] La "coevolución" es un modelo en computación evolutiva en el cual se usan dos poblaciones que interactúan entre sí de tal forma que la aptitud de los miembros de una de ellas depende de la otra población y vice versa. En otras palabras, en este modelo, la evolución de una especie está condicionada a la de otra especie.

8.2. Programación Evolutiva

Este tipo de algoritmo evolutivo se propuso originalmente por Lawrence Fogel a principios de los sesenta cuando realizaba investigación básica en torno a inteligencia artificial para la *National Science Foundation* en Estados Unidos.[34] En esa época, la mayor parte de los intentos existentes para generar comportamientos inteligentes usaban al humano como modelo. Sin embargo, Fogel consideró que, puesto que la evolución fue capaz de crear a los humanos y a otras criaturas inteligentes su simulación podría conducir a comportamientos inteligentes.[35]

Las ideas básicas de Fogel eran similares a las de algunos trabajos que describimos anteriormente: usar mecanismos de selección y variación para evolucionar soluciones candidatas que se adaptaran mejor a las metas deseadas. Sin embargo, dado que Fogel desconocía dichos trabajos, su propuesta puede considerarse una re-invención de los esquemas basados en la evolución. Filosóficamente, las estructuras de codificación utilizadas en la programación evolutiva son una abstracción del fenotipo[36] de diferentes especies.[37] Consecuentemente, la codificación de soluciones candidatas se puede adaptar libremente para satisfacer los requerimientos del problema (es decir, no se requiere codificar las soluciones en un alfabeto específico como ocurre, por ejemplo, con el Algoritmo Genético, en donde las soluciones normalmente se codifican usando un alfabeto binario). Dado que no es posible la recombinación sexual entre especies diferentes, la Programación Evolutiva no incorpora un operador de cruza o recombinación. La ausencia del operador de cruza, la libertad de adaptar la codificación y el uso de un operador de selección pro-

[34] Fogel, L. J. (1962). **Autonomous automata**. *Industrial Research,* 4:14–19.

[35] Fogel, L. J. (1999). **Intelligence through simulated evolution: forty years of evolutionary programming**, Wiley series on intelligent systems, Wiley, 1999.

[36] El "fenotipo" es un término que se usa en genética para denotar los rasgos físicos visibles de un individuo. En algoritmos evolutivos, el fenotipo se refiere al valor decodificado de una solución candidata.

[37] Fogel, D. B. (1994). **An introduction to simulated evolutionary optimization**. *IEEE Transactions on Neural Networks*, 5(1):3–14.

babilístico[38] (el cual no se incorporó en las primeras versiones del algoritmo) son los rasgos principales que distinguen a la programación evolutiva de los demás algoritmos evolutivos.

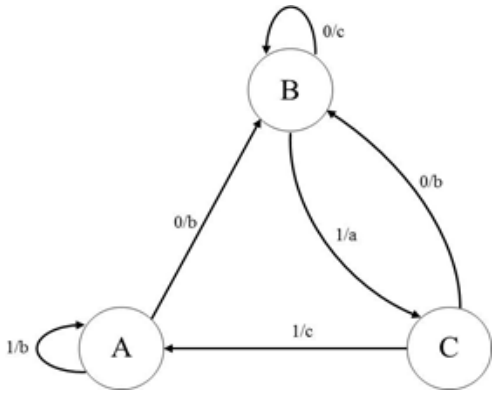


Figura 8.1 Máquina de estados finitos como las usadas por Lawrence Fogel en sus primeros experimentos con la Programación Evolutiva. Los símbolos a la izquierda y a la derecha del "/" son de entrada y de salida respectivamente. El estado inicial en este caso es el C.

En sus primeras versiones de la Programación Evolutiva, Fogel se percató de que uno de los principales rasgos que caracterizan el comportamiento inteligente es la capacidad de predecir el ambiente, junto con un mecanismo que permita traducir las predicciones a una respuesta adecuada de manera que se logre el objetivo deseado. Por otra parte, un transductor de estados finitos es una máquina que permite transformar una secuencia de símbolos de entrada en una secuencia de símbolos de salida, como la que se ilustra en la Figura 8.1. Dependiendo de los estados y transiciones, esta máquina se puede usar para modelar diferentes situaciones y producir diferentes transformaciones. Por tanto, Fogel la consideró como un mecanismo adecuado para lidiar con el problema de generar comportamiento inteligente. En sus primeros experimentos, su objetivo fue desarrollar una máquina capaz de predecir el comportamiento de un ambiente, el cual se representaba como una secuencia

[38] El uso de un operador de selección probabilístico implica que los individuos menos aptos de la población tienen una probabilidad distinta de cero de ser seleccionados.

de símbolos de un alfabeto de entrada; es decir, dados los símbolos generados previamente por el ambiente, predecir el siguiente símbolo que emergerá del ambiente. Pensemos, por ejemplo, que queremos obtener un transductor de estados finitos como el que se muestra en la figura 8.1. Para este efecto se requiere crear una tabla de transiciones como la que se muestra en la tabla 8.1. En ésta se indica, para cada estado, el siguiente estado según la entrada, y el símbolo de salida. Fogel intentó generar estas máquinas de manera automatizada (simulando la evolución), usando como entrada los símbolos de salida de la tabla de transiciones. El objetivo era que la Programación Evolutiva produjera una máquina de estados finitos que generara los símbolos de salida deseados, sin intervención humana.

Estado Actual	C	B	C	A	A	B
Símbolo de entrada	0	1	1	1	0	1
Estado siguiente	B	C	A	A	B	C
Símbolo de salida	b	a	c	b	b	a

Tabla 8.1 Esta tabla ilustra las transiciones de la máquina de estados finitos que se muestra en la figura 8.1.

El algoritmo original de Programación Evolutiva funcionaba de la manera siguiente: Primero, se producía de manera aleatoria una población con *N* máquinas de estados finitos. Posteriormente, cada miembro de la población era mutado, creándose una población de hijos del mismo tamaño de la de los padres. Fogel adoptó cinco tipos de operadores de mutación: i) agregar un estado, ii) borrar un estado, iii) cambiar el estado siguiente de una transición, iv) cambiar el símbolo de salida de una transición, o v) cambiar el estado inicial. Los operadores de mutación se elegían aleatoriamente[39] y, en algunos casos, los hijos se mutaban más de una vez. Finalmente, los hijos eran evaluados y se seleccionaban las *N* mejores máquinas (de entre los padres e hijos). Cada máquina de estados finitos era evaluada en términos de su capacidad para

[39] Fogel experimentó con diversos esquemas para elegir el operador de mutación más adecuado, pero al final decidió adoptar un método de selección aleatorio.

predecir correctamente los siguientes símbolos en las secuencias conocidas. Inicialmente, la aptitud[40] se calculaba considerando un número pequeño de símbolos, pero conforme la evolución progresaba, se agregaban más símbolos al conjunto de entrenamiento. En sus primeros experimentos Fogel realizó pruebas con diferentes tipos de tareas de predicción: secuencias periódicas de números, secuencias con ruido, ambientes no estacionarios, etc. Posteriormente, consideró tareas más complejas como el reconocimiento de patrones, la clasificación y el diseño de sistemas de control. Todo este trabajo originó el que se considera el primer libro de computación evolutiva de la historia.[41] Este trabajo fue muy importante porque presenta, entre otras cosas, una de las primeras aplicaciones de la coevolución. También es importante enfatizar que en la mayor parte de los primeros estudios en torno a la Programación Evolutiva, los resultados computacionales presentados no eran muy amplios, debido al limitado poder de cómputo de la época. La mayor parte de estos experimentos iniciales fueron recapitulados y extendidos varios años después, proporcionando una mayor comprensión de los verdaderos alcances de la Programación Evolutiva.[42]

En los años setenta la mayor parte de la investigación en torno a la Programación Evolutiva se realizó bajo la tutela de Dearholt. Una de las principales contribuciones de su trabajo fue la aplicación de Programación Evolutiva a problemas prácticos. Por ejemplo, al reconocimiento de patrones en expresiones regulares[43] y en caracteres manuscritos,[44] así como para clasificar

[40] La "aptitud" de un individuo es una medida que permite comparar una solución con respecto a las demás. Si el problema a resolverse es de optimización, normalmente la aptitud se define en términos de la función objetivo que se desea optimizar.

[41] Fogel, L. J., Owens, A. J., Walsh, M. J. **Artificial Intelligence Through Simulated Evolution**, John Wiley & Sons, USA, 1966.

[42] Fogel, L. J., Fogel, D. B. **Artificial intelligence through evolutionary programming**, Technical Report PO-9-X56-1102C-1, U.S. Army Research Institute, San Diego, California, USA, 1986.

[43] Lyle, M. R. **An Investigation Into Scoring Techniques in Evolutionary Programming**, MSc Thesis, Las Cruces, USA, 1972.

[44] Cornett, F. N. **An Application of Evolutionary Programming to Pattern Recognition**, MSc Thesis, Las Cruces, USA, 1972.

diferentes tipos de electrocardiogramas.[45] Estos trabajos incorporaron varias novedades algorítmicas. De entre ellas, tal vez las más importantes hayan sido el uso simultáneo de varios operadores de mutación y la adaptación dinámica de las probabilidades asociadas con los diferentes esquemas de mutación.

A principios de los ochenta, la Programación Evolutiva se diversificó usando otras representaciones arbitrarias de las soluciones candidatas a fin de resolver diferentes tipos de problemas. Eventualmente se utilizó, por ejemplo, para resolver problemas de ruteo mediante una codificación basada en el uso de permutaciones de enteros,[46] y para generar programas de computadora de manera automática, adoptando una codificación de árbol.[47] Asimismo, se usaron vectores de números reales para resolver problemas de optimización continua[48] y para entrenar redes neuronales.[49] Durante este período, se aceptaba comúnmente que el diseñar representaciones específicas para un problema, junto con operadores especializados, permitía realizar una búsqueda más eficiente.[50] De hecho, varios investigadores defendían la posición de que el diseño de esquemas de mutación inteligentes podía evitar el uso de los operadores de recombinación.[51]

En los noventa, el uso de la Programación Evolutiva en problemas de optimización continua se extendió considerablemente. Durante este perío-

[45] Dearholt, D. W. (1976). **Some experiments on generalization using evolving automata**. En *9th Hawaii International Conference on System Sciences*, Western Periodicals, Honolulu, Hawaii, USA, pp. 131–133.

[46] Ver nota 42.

[47] Chellapilla, K. (1997). **Evolving Computer Programs Without Subtree Crossover**. *IEEE Transactions on Evolutionary Computation*, 1(3):209–216.

[48] Yao, X., Liu, Y., Lin, G. (1999). **Evolutionary programming made faster**. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102.

[49] Porto, V. W., Fogel, D. B. (1995). **Alternative neural network training methods [active sonar processing]**. *IEEE Expert*, 10(3):16–22.

[50] Fogel, L. J. **Intelligence through simulated evolution: forty years of evolutionary programming**, Wiley series on intelligent systems, Wiley, USA, 1999.

[51] Fogel, D. B., Atmar, J. W. (1990). **Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems**. *Biological Cybernetics*, 63(2):111–114.

do se hicieron enormes esfuerzos por desarrollar esquemas auto-adaptativos. Cuando se usa auto-adaptación, algunos parámetros del algoritmo se codifican junto con las variables del problema, de tal manera que los operadores evolutivos actúen sobre ellos y definan sus valores. Bajo este tipo de esquema, las variables del problema son llamadas parámetros objeto, mientras que los parámetros del algoritmo evolutivo se denominan parámetros de la estrategia. En los primeros intentos por incorporar auto-adaptación a la Programación Evolutiva, se adaptaba el factor de escalamiento de la mutación Gaussiana.[52] En otras variantes más avanzadas se consideran varios operadores de mutación simultáneamente.[53] Puesto que los esquemas auto-adaptativos resultaron tener un desempeño superior a las variantes tradicionales, este mecanismo se volvió muy popular en muchas de las implementaciones usadas para resolver problemas del mundo real, particularmente cuando se adoptaba codificación mediante vectores de números reales.[54] Cabe destacar que el mecanismo de auto-adaptación de la Programación Evolutiva para optimización continua presenta varias similitudes con respecto al mecanismo adoptado en las Estrategias Evolutivas. La diferencia más significativa es que la Programación Evolutiva no incorpora recombinación. Adicionalmente, algunos detalles de implementación eran también diferentes en las primeras variantes auto-adaptativas de la Programación Evolutiva.[55] Por ejemplo, el orden en el que se sometían a mutación los parámetros objeto y los de la estrategia era diferente en la Programación Evolutiva y las Estrategias Evolutivas, si bien estas diferencias acabaron por desvanecerse con el tiempo. Además, existen casos documentados en los que se ha podido mostrar que el uso exclusivo del operador de mutación produce mejores resultados que cuando se incorpora recombinación y viceversa.[56] Hay evidencia que indica que es benéfico adap-

[52] Fogel, D. B., Fogel, L. J., Atmar, J. W. (1991). **Meta-Evolutionary Programming**. En *Asilomar Conference in Signals Systems and Computers*, pp. 540–545.

[53] Ver nota 48.

[54] Fogel, D. B. **Evolutionary Computation: Toward a New Philosophy of Machine Intelligence**, IEEE Press, Piscataway, New Jersey, USA, 1995.

[55] Eiben, A. E., Smith, J. E. **Introduction to Evolutionary Computing**, Natural Computing Series, Springer, 2003.

[56] Richter, J. N., Wright, A., Paxton, J. **Ignoble Trails - Where Crossover Is Provably**

tar también los parámetros asociados con el operador de cruza usado por las Estrategias Evolutivas,[57] lo cual hace que disminuyan las diferencias entre la Programación y las Estrategias Evolutivas. Se hace notar, sin embargo, que el primer mecanismo de auto-adaptación de parámetros para optimización continua se produjo en las Estrategias Evolutivas, por lo cual este tema se discutirá a mayor detalle cuando hablemos de dicho tipo de algoritmo evolutivo.

En la actualidad, la Programación Evolutiva es el tipo de algoritmo evolutivo menos usado a nivel mundial. Aunque su uso original fue en problemas de predicción hoy en día se suele usar más para optimización en espacios continuos. Esto se refleja también en México, en donde existe muy poca evidencia de su uso para optimización mono-objetivo[58] y multi-objetivo[59] en espacios continuos.

8.3. Estrategias Evolutivas

A mediados de la década de los sesenta, tres estudiantes de la Universidad Técnica de Berlín (Bienert, Schwefel y Rechenberg) estudiaban problemas prácticos de mecánica de fluídos y otras áreas similares con el fin de construir robots que pudieran resolver de manera automatizada problemas de

---

Harmful. En G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume (eds.), *Parallel Problem Solving from Nature PPSN X*, pp. 92-101, Lecture Notes in Computer Science, Vol. 5199, Springer, Berlin, Germany, 2008.

[57] Jain, A. Fogel, D. B. (2000). **Case studies in applying fitness distributions in evolutionary algorithms II: Comparing the improvements from crossover and Gaussian mutation on simple neural networks**. En *2000 IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, pp. 91–97.

[58] Coello Coello, C. A., Landa Becerra, R. (2002). **Adding Knowledge and Efficient Data Structures to Evolutionary Programming: A Cultural Algorithm for Constrained Optimization**". En W.B. Langdon et al. (Editors), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*, Morgan Kaufmann Publishers, San Francisco, California, USA, pp. 201–209.

[59] Coello Coello, C. A., Landa Becerra, R. (2003). **Evolutionary Multiobjective Optimization using a Cultural Algorithm**. En *2003 IEEE Swarm Intelligence Symposium*, IEEE Service Center, Indianapolis, Indiana, USA, pp. 6–13,

ingeniería.[60] Para este efecto, formularon a dichas tareas como problemas de optimización y desarrollaron máquinas autónomas que se modificaban automáticamente bajo la aplicación de ciertas reglas. También intentaron aplicar algoritmos clásicos de optimización. Sin embargo, debido a que los problemas que querían resolver tenían ruido y eran multimodales, estos algoritmos fueron incapaces de producir soluciones aceptables. Para enfrentar esta problemática decidieron aplicar métodos de mutación y selección análogos a la evolución natural. Rechenberg publicó el primer reporte en torno a la nueva técnica, denominada "Estrategia Evolutiva", aplicada a la minimización del arrastre total de un cuerpo en un túnel de viento.[61] Posteriormente, resolvieron otros problemas tales como el diseño de tubos y de boquillas hidrodinámicas.[62]

Filosóficamente, la codificación de soluciones adoptada por las Estrategias Evolutivas es una abstracción del fenotipo de los individuos.[63] Por esta razón se puede adoptar libremente la codificación de soluciones candidatas, a fin de adaptarse mejor a los requerimientos del problema. Además, en las Estrategias Evolutivas se permite la recombinación de individuos, si bien este operador no fue implementado en la versión original del algoritmo. A la primera versión de este algoritmo se le conoce como la (1+1)-EE (EE significa "Estrategia Evolutiva"). Su funcionamiento es el siguiente: primero, se crea una solución inicial de manera aleatoria y se considera como "el padre". Esta solución es mutada (es decir, el valor de una de sus variables es modificado aleatoriamente) y al individuo mutado se le denomina "el hijo". La mutación se aplicaba siguiendo una distribución binomial, a fin de aplicar mutaciones pequeñas más frecuentemente que las mutaciones grandes, como ocurre en la naturaleza. La solución mutada se vuelve el padre si es mejor o igual que el

[60] Fogel, D. B. **Evolutionary Computation: The Fossil Record**, Wiley-IEEE Press, USA, 1998.

[61] Rechenberg, I. **Cybernetic solution path of an experimental problem**, Technical Report, Library Translation 1122, Royal Air Force Establishment, 1965.

[62] Ver nota 60.

[63] Fogel, D. B. (1994). **An introduction to simulated evolutionary optimization**. *IEEE Transactions on Neural Network*s, Vol. 5, No. 1, pp. 3–14.

individuo que la originó. El nuevo padre se vuelve a mutar, repitiéndose este proceso hasta alcanzar un cierto criterio de paro. Los experimentos realizados por los creadores de esta técnica mostraron que resultaba mejor que los métodos de optimización clásicos en problemas de alta complejidad.

En 1965, Hans-Paul Schwefel implementó por primera vez una Estrategia Evolutiva en una computadora de uso general y la aplicó a problemas de optimización continua.[64] La contribución más importante de Schwefel fue que incorporó una mutación Gaussiana con media cero, y este operador se sigue usando a la fecha. El operador de mutación se podía controlar mediante el ajuste de las desviaciones estándar (es decir, el tamaño de paso) de la distribución Gaussiana. Los estudios realizados por Schwefel en torno al tamaño de paso originaron los primeros análisis teóricos de la Estrategia Evolutiva.

Hacia finales de los sesenta se introdujo una versión de la Estrategia Evolutiva que usaba una población con varios individuos. Esta primera versión poblacional fue denominada $(\mu+1)$-EE, y adoptaba μ individuos, los cuales se usaban para crear un nuevo individuo por medio de operadores de recombinación y mutación. Posteriormente, los μ mejores individuos de entre los $\mu+1$ existentes se seleccionaban para sobrevivir. Este esquema es muy similar a la selección de estado uniforme[65] que se volvió popular años más tarde en los algoritmos genéticos. Con el tiempo, las Estrategias Evolutivas permitieron adoptar cualquier número de padres y de hijos. Además, se comenzaron a utilizar dos esquemas de selección. Así surgió la $(\mu+\lambda)$-EE y la (μ,λ)-EE. En la primera, μ padres generan λ hijos y sobreviven los μ mejores de la unión de ellos (padres e hijos). En la segunda, μ padres generan λ hijos y sobreviven los μ mejores hijos.

---

[64] Schwefel, H. P. **Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik**, Dipl.-Ing. Thesis, Technical University of Berlin, Hermann Föttinger, Institute for Hydrodynamics, 1965.

[65] Whitley D., J. Kauth, J. **GENITOR: A different genetic algorithm**, Technical Report, ColoradoState University, Department of Computer Science, 1988.

Las primeras publicaciones sobre las Estrategias Evolutivas fueron escritas en alemán, lo cual limitó su acceso. Sin embargo, en 1981, Schwefel publicó el primer libro sobre estas estrategias en inglés.[66] Desde entonces, un número importante de investigadores en el mundo las han estudiado y aplicado extensamente. En México, se han utilizado para resolver problemas de procesamiento de imágenes[67] y para optimización no lineal con restricciones.[68]

El libro de Schwefel se enfoca en el uso de las Estrategias Evolutivas para optimización continua, que es de hecho el área en la que esta técnica ha tenido mayor éxito. En éste se discute, entre otros temas, el uso de diferentes tipos de operadores de recombinación y la adaptación de las distribuciones Gaussianas utilizadas para el operador de mutación.

En los estudios iniciales del operador de recombinación se propusieron cuatro esquemas posibles, combinando dos propiedades diferentes. Primero, se proporcionaron dos opciones para el número de padres involucrados en la creación de un hijo: i) recombinación bisexual (o local) y ii) recombinación multisexual (o global). En la primera se seleccionan dos padres y éstos se recombinan entre sí, produciendo así la solución candidata del hijo. En la segunda, se seleccionan diferentes padres para ir generando cada variable de la solución candidata del hijo. También hay dos opciones para combinar los valores de los padres: la denominada "recombinación discreta", en la que uno de los valores se selecciona aleatoriamente, y la denominada "recombinación

[66] Schwefel, H. P. **Numerical Optimization of Computer Models**, John Wiley & Sons, Inc., New York, New York, USA, 1981.

[67] Gómez García, H. F., González Vega, A., Hernández Aguirre, A., Marroquín Zaleta, J. L., Coello Coello, C. A. (2002). **Robust Multiscale Affine 2D-Image Registration through Evolutionary Strategies**, en Juan Julián Merelo Guervós, Panagiotis Adamidis, Hans-Georg Beyer, José-Luis Fernández-Villacañas y Hans-Paul Schwefel (editors), *Parallel Problem Solving from Nature VII*, Lecture Notes in Computer Science Vol. 2439, Springer-Verlag, Granada, Spain, September, pp. 740–748.

[68] Mezura Montes, E., Coello Coello, C. A. (2005). **A Simple Multi-Membered Evolution Strategy to Solve Constrained Optimization Problems**. *IEEE Transactions on Evolutionary Computation*, 9(1):1–17.

intermedia", en la que se calcula el promedio de los valores de los padres. En estos esquemas de recombinación no se puede especificar la cantidad de padres que toman parte en cada recombinación. Rechenberg[69] propuso una generalización que cambia esta restricción. En este caso, se usa un nuevo parámetro, denominado ρ para especificar el número de padres que tomarán parte en la creación de cada individuo. Esto originó las variantes denominadas (μ/ρ, λ)-EE y (μ/ρ+ λ)-EE. Posteriormente, se propusieron otras versiones que asignaban pesos a la contribución de cada uno de los individuos que participaban en la recombinación.[70]

Pese a utilizar recombinación, el operador principal de las Estrategias Evolutivas es la mutación. Como se indicó anteriormente, este operador se suele basar en una distribución Gaussiana con media cero. En la mayor parte de los casos, los individuos se perturban mediante la adición de un vector aleatorio generado de una distribución Gaussiana multivariada. En otras palabras, se usa la expresión siguiente:

$$x_i' = x_i + N_i\,(0,\, C)$$

donde x_i' es el nuevo individuo, generado a partir de x_i, C representa una matriz de covarianza y $N_i\,(0,\, C)$ devuelve un número aleatorio con media cero. De tal forma, el nuevo individuo (es decir, el "hijo") se crea a partir del anterior (el "padre") mediante una perturbación aleatoria a una de sus variables. Desde los orígenes de las Estrategias Evolutivas, resultó claro que C podría tener un efecto significativo en el desempeño del algoritmo. Consecuentemente, se han desarrollado diversos métodos para adaptar C durante la ejecución del algoritmo. Los primeros esquemas de este tipo se originaron de los estudios realizados por Rechenberg,[71] quien analizó las propiedades de

[69] Rechenberg, I. (1978). **Evolutionsstrategin**. En B. Schneider and U. Ranft (eds.), *Simulationsmethoden in der Medizin und Biologie*, Springer-Verlag, Berlin, Germany, pp. 83–114.

[70] Bäck, T., Schwefel, H. P. (1993). **An Overview of Evolutionary Algorithms for Parameter Optimization**. *Evolutionary Computation*, 1(1):1–23.

[71] Rechenberg, I. **Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution**, Frommann-Holzboog, Stutgart, Germany, 1973.

convergencia de las Estrategias Evolutivas en relación a la frecuencia relativa de las mutaciones exitosas (si el hijo es mejor que su padre en términos del valor de la función objetivo). Este trabajo condujo al primer esquema adaptativo en el cual el tamaño de paso global se ajusta mediante un procedimiento en línea con el objetivo de producir mutaciones exitosas con una tasa igual a 1/5. Este procedimiento se conoce como la "regla de éxito 1/5". Esta regla hace ajustes a la desviación estándar cuando la tasa de éxito no es exactamente 1/5, bajo la premisa de que esta tasa de mutaciones exitosas es la ideal. Es importante hacer notar que dicha regla no es general, lo cual originó las versiones auto-adaptativas de las Estrategias Evolutivas. En el primer esquema de este tipo, el único parámetro adaptado fue el tamaño de paso global. Sin embargo, con los años, se desarrollaron esquemas auto-adaptativos mucho más sofisticados.

Otro punto interesante en torno a las Estrategias Evolutivas es que si se adapta toda la matriz de covarianza pueden inducirse correlaciones entre las mutaciones realizadas en las cuales se involucran diferentes parámetros, lo cual hace más apropiado su uso para lidiar con funciones no separables. El primer esquema en el cual se adapta toda la matriz de covarianza fue propuesto por Schwefel.[72] En este caso, el sistema coordenado en el cual se realiza el control del tamaño de paso, así como los tamaños de paso mismos, se auto-adaptan. Se hace notar, sin embargo, que dicho esquema no ha sido muy exitoso porque requiere de tamaños de población muy grandes para operar adecuadamente. Una alternativa más eficiente es la Estrategia Evolutiva con un esquema de adaptación de la matriz de covarianza propuesta por Hansen y Ostermeier conocida como CMA-ES.[73] En este esquema se combinan dos técnicas diferentes de auto-adaptación con el objetivo de construir matrices de covarianza que maximicen la creación de vectores de mutación que fueron exitosos en generaciones anteriores. Esta técnica es muy compleja y requiere

[72] Schwefel, H. P., **Numerical Optimization of Computer Models**, John Wiley & Sons, Inc., New York, New York, USA, 1981.

[73] Hansen, N., Ostermeier, A. (2001). **Completely Derandomized Self-Adaptation in Evolution Strategies**. *Evolutionary Computation*, 9(2):159–195.

de varios parámetros, lo cual ha dado pie a variantes más simples y con menos parámetros.[74]

Finalmente, existen también esquemas que favorecen ciertas direcciones de búsqueda sin adaptar toda la matriz de covarianza. Por ejemplo, Poland y Zell[75] propusieron utilizar la dirección principal de descenso para guiar la búsqueda. La mayor ventaja de este tipo de esquemas es que reducen la complejidad computacional en términos de tiempo y espacio. Sin embargo, dado que el tiempo para calcular las matrices de covarianza no es significativo con respecto al de la evaluación de la función objetivo (sobre todo, si el problema es costoso en términos de tiempo de cómputo) este tipo de esquemas no ha sido muy popular, aunque puede resultar útil en problemas con un gran número de variables, pues en ese caso, el costo de calcular matrices de covarianza se incrementa de manera significativa.

Las Estrategias Evolutivas son una opción muy buena para resolver problemas de optimización continuos en los que todas las variables son números reales, aunque en años recientes han sido desplazadas por metaheurísticas como la evolución diferencial, que resultan más fáciles de implementar y usar.

8.4. Algoritmos Genéticos

John Holland[76] identificó la relación entre el proceso de adaptación que existe en la evolución natural y la optimización, y conjeturó que dicho proceso podría jugar un papel crítico en otras áreas tales como el aprendizaje, el control

[74] Beyer, H. G., Sendhoff, B. (2008). **Covariance Matrix Adaptation Revisited – The CMSA Evolution Strategy**. En G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume (eds.), *Parallel Problem Solving from Nature - PPSN X*, pp. 123–132, Lecture Notes in Computer Science, Vol. 5199, Springer, Berlin, Germany.

[75] Poland, J. Zell, A. (2001). **Main Vector Adaptation: A CMA Variant with Linear Time and Space Complexity**. En L. Spector et al. (eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, Morgan Kaufmann, San Francisco, California, USA, pp. 1050–1055.

[76] Holland, J. H. **Adaptation in Natural and Artificial Systems**, University of Michigan Press, Ann Arbor, Michigan, USA, 1975.

automático o las matemáticas. Su objetivo inicial era estudiar de manera formal la adaptación y propuso un entorno algorítmico inspirado en estas ideas, al cual denominó originalmente "planes reproductivos y adaptativos", pero que después renombró como "Algoritmo Genético". El funcionamiento de estos algoritmos se basa en la modificación progresiva de un conjunto de estructuras de datos con el objetivo de adaptarlas a un cierto ambiente. La forma específica de realizar dichas adaptaciones está inspirada en la genética y en la evolución natural. No obstante, en poco tiempo, los Algoritmos Genéticos se volvieron solucionadores de problemas en áreas tales como el aprendizaje de máquina[77] y el control automático. Sin embargo, su uso más extendido ha sido en optimización, tanto en espacios discretos como continuos.

Filosóficamente, la codificación utilizada en los Algoritmos Genéticos es una abstracción del genotipo de los individuos. En este caso, sin importar el tipo de variables que tenga el problema, éstas se deben codificar en binario. Aunque hoy en día es posible usar otras codificaciones, Holland argumentó que el alfabeto binario es universal y presenta varias ventajas, tanto de tipo práctico como teórico. Sin embargo, pese a sus ventajas tiene la desventaja evidente de que se requiere de un proceso de decodificación para convertir la cadena binaria en la variable que codifica. Asimismo, el uso de la codificación binaria puede en algunos casos introducir problemas debido a la discretización de un espacio de búsqueda que originalmente no era discreto (por ejemplo, cuando las variables son números reales). Tales problemas van desde usar una precisión inadecuada hasta el manejo de cadenas binarias de gran tamaño con las cuales es más ineficiente operar dentro del Algoritmo Genético.

A cada cadena binaria que codifica una variable se le denomina *gene*, y cada bit dentro de un gene se conoce como *alelo*. A la concatenación de todos los genes (es decir, al conjunto de todas las variables del problema) se le denomina *cromosoma*, como se ilustra en la Figura 8.2. Un individuo está for-

[77] Los Algoritmos Genéticos dieron pie a los **Sistemas de Clasificadores,** en los cuales se evolucionaron conjuntos de reglas del tipo IF-THE-ELSE para resolver problemas de clasificación.

mado normalmente por un cromosoma (es decir, se considera una estructura haploide, pese a que en la naturaleza es muy común que las especies tengan una estructura diploide, es decir, con dos cromosomas).

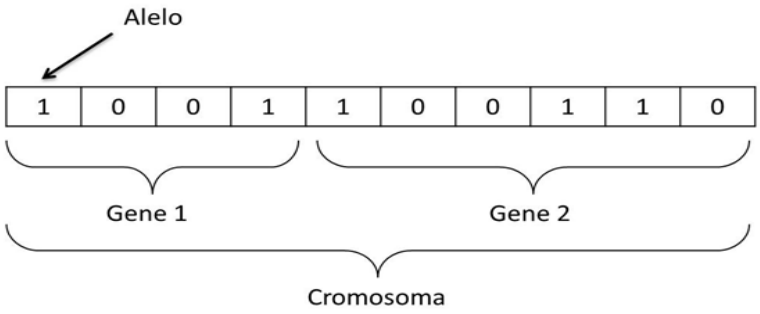


Figura 8.2 Estructura de una cadena cromosómica para un problema con dos variables ("gene 1" y "gene 2" corresponden a dichas variables).

La forma de operar de un Algoritmo Genético es bastante simple. Se parte de un conjunto de soluciones que se generan aleatoriamente al que se conoce como la "población inicial". Para cada una de estas soluciones (o "individuos") se debe calcular un valor de aptitud. Dicho valor está en función del objetivo que se quiere optimizar, aunque por lo general, requiere algún proceso de normalización a fin de evitar valores de aptitud negativos o divisiones entre cero. Las parejas de padres que se reproducirán se seleccionan de acuerdo a su valor de aptitud (los mayores de toda la población). En la versión original del Algoritmo Genético se utiliza un esquema de selección proporcional, lo que significa que los individuos más aptos tienen mayor probabilidad de sobrevivir. Sin embargo, los menos aptos tienen una probabilidad distinta de cero de ser seleccionados y podrían volverse padres también. Los individuos seleccionados se recombinan mediante un operador de cruza para generar la población de hijos, a la cual se aplica el operador de mutación. Los hijos mutados conforman la nueva población que reemplaza por completo a la población anterior. El proceso se repite hasta cumplirse un cierto criterio de paro, que normalmente es un número máximo de gene-

raciones (o de iteraciones). La tabla 8.2 muestra una población hipotética de cinco individuos de un algoritmo genético, incluyendo los cromosomas y las aptitudes correspondientes.

Individuo No.	Cromosoma	Aptitud	% del Total
1	11010110	254	24.5
2	10100111	47	4.5
3	00110110	457	44.1
4	01110010	194	18.7
5	11110010	85	8.2
Total		1037	100.0

Tabla 8.2 Ejemplo de una población de cinco individuos, cuyos cromosomas contienen 8 bits cada uno. En la tercera columna se muestra la aptitud (calculada mediante una función hipotética).

La figura 8.3 ilustra el funcionamiento de un método de selección proporcional conocido como "la ruleta", que se usa con los algoritmos genéticos. El triángulo negro es un pivote. En la selección por el método de la ruleta, se simula el giro aleatorio de un disco que contiene a los individuos de la población y en cada giro se selecciona como padre al individuo que marca el pivote al detenerse la ruleta.

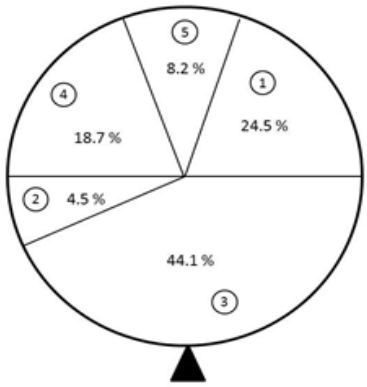


Figura 8.3 Método de selección de la ruleta. Los cinco individuos de la tabla 8.2 se colocan en una rueda, en la cual cada subdivisión corresponde

a su proporción de aptitud. En esta técnica de selección se simula el giro de una ruleta, de tal forma que seleccionamos como padre al individuo donde la ruleta se detiene. Evidentemente, los individuos más aptos tienen mayor probabilidad de ser seleccionados, pero los menos aptos también podrían ser seleccionados como padres, aunque su probabilidad es muy baja.

La porción que cada individuo ocupa en la ruleta corresponde a su aptitud, de tal forma que los individuos más aptos tengan mayor probabilidad de seleccionarse y recombinarse mediante alguno de los varios tipos de cruzas disponibles. Las figuras 8.4 y 8.5 ilustran el funcionamiento de la cruza de uno y dos puntos respectivamente.

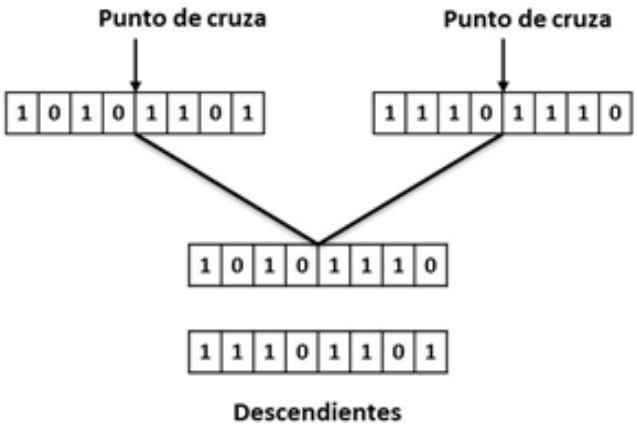


Figura 8.4 Cruza de un punto. Nótese cómo la cruza entre dos padres produce dos hijos. El hijo 1 tiene los primeros 4 bits del padre 1 y los otros 4 del padre 2. El hijo 2 contiene los primeros 4 bits del padre 2 y los otros 4 del padre 1.

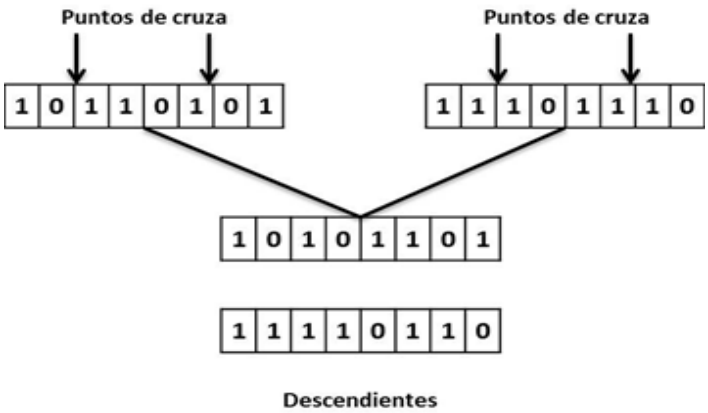


Figura 8.5 Cruza de dos puntos. El hijo tiene los 2 primeros y 2 últimos bits del padre 1 y los bits complementarios del padre 2. El hijo 2 tiene los 2 primeros y 2 últimos bits del padre 2 y los bits complementarios del padre 1.

Finalmente, cada uno de los hijos que se producen mediante la cruza es mutado, como se ilustra en la Figura 8.6.

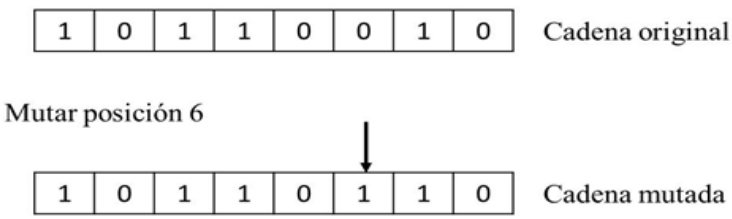


Figura 8.6 Ejemplo del operador de mutación de un algoritmo genético. En este caso, se muta la posición 6 de la cadena original. El operador aplica un "NOT" a la posición original, con lo cual se cambia el cero a uno.

Con los años, se desarrollaron distintas variantes del Algoritmo Genético que involucran el uso de esquemas de selección deterministas y de estado uniforme (en los cuales se reemplaza a un solo individuo de la población en cada iteración, en vez de reemplazarlos a todos), codificaciones no binarias, operadores de cruza y de mutación muy sofisticados y/o altamente especia-

lizados para un dominio en particular, cromosomas diploides y multiploides, así como otro tipo de operadores tales como la inversión, la traslocación y la segregación.[78]

Un aspecto distintivo de los Algoritmos Genéticos es que se ha considerado tradicionalmente que el operador de cruza es su fuente de mayor poder y que el operador de mutación es sólo complementario, como un esquema para mantener diversidad y para mantener el espacio de búsqueda completamente conectado. Esta filosofía se contrapone a la de las Estrategias Evolutivas en las que la mutación es el operador principal y la cruza es sólo un operador secundario y a la Programación Evolutiva en la que sólo se usa la mutación.

Holland en su libro de 1975 trató de modelar el funcionamiento del Algoritmo Genético a través de su "Teorema de los Esquemas" en el que deriva una fórmula que proporciona la probabilidad de que un cierto patrón de bits (denominado "esquema") sobreviva a los efectos de la selección, la cruza y la mutación.[79] Con el paso de los años, se desarrollaron modelos matemáticos mucho más sofisticados para intentar explicar el funcionamiento de un Algoritmo Genético, incluyendo aquellos que se basan en cadenas de Markov y los que se basan en mecánica estadística.[80] La primera demostración de convergencia de un Algoritmo Genético ordinario[81] (llamado "canónico") mostró que debe usarse un operador conocido como "elitismo" para poder garantizar convergencia. Este operador consiste en retener al mejor individuo de cada generación y pasarlo intacto a la generación siguiente, sin cruzarlo ni mutarlo.

---

[78] Goldberg, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**, Addison-Wesley, USA, 1989.

[79] Holland, J. H. **Adaptation in Natural and Artificial Systems**, University of Michigan Press, Ann Arbor, Michigan, USA, 1975.

[80] Eiben, A. E., Smith, J. E. **Introduction to Evolutionary Computing**, Natural Computing Series, Springer, Germany, 2003.

[81] Rudolph, G. (1994). **Convergence Analysis of Canonical Genetic Algorithms**. *IEEE Transactions on Neural Networks*, 5:96–101.

En México, se ha hecho investigación muy relevante en torno a los Algoritmos Genéticos, tanto en el contexto de aprendizaje de máquina[82] como en el de optimización mono-objetivo y multi-objetivo.[83] También se hace investigación en torno al uso de algoritmos genéticos y sus variantes en electrónica (el denominado "hardware evolutivo"),[84] visión,[85] bioinformática,[86] finanzas,[87] criptografía,[88] procesamiento de imágenes,[89] y diseño.[90] Asimismo, hay grupos de investigación que han abordado el estudio de los operadores evolutivos[91] y de aspectos teóricos de los algoritmos genéticos.[92] En nuestro país se ha abordado también el estudio de la programación genética,[93] una variante del algoritmo genético propuesta por John Koza[94] en la que se usa una codificación de árbol para evolucionar programas.

8.5. Otras Metaheurísticas Bio-Inspiradas

En México se han estudiado otras metaheurísticas bio-inspiradas entre las que destacan las siguientes:

- **Cúmulos de Partículas:** Propuesta por Kennedy y Eberhart a mediados de los noventa[95] simula el movimiento de un conjunto de par-

[82] Por ejemplo, hay grupos de investigación en el ITESM Campus Monterrey y en el INAOE.

[83] Hay grupos de investigación que han abordado estos temas en la Universidad Veracruzana, la Universidad de Guadalajara, el CINVESTAV-IPN, la Universidad Autónoma de Querétaro, el CICESE, el CIMAT y la UNAM.

[84] En el INAOE y el CIMAT.

[85] En el CICESE.

[86] En el CICESE y el ITESM Campus Estado de México.

[87] En el ITESM Campus Monterrey, el CINVESTAV-IPN y el Banco de México.

[88] En el CIC-IPN y el CINVESTAV-IPN.

[89] En el ITESM Campus Guadalajara, el CIMAT, el CICESE y el CINVESTAV-IPN.

[90] En el Instituto Tecnológico Autónomo de México (ITAM).

[91] En la Universidad de Guadalajara y el CINVESTAV-IPN.

[92] En la UNAM y el CINVESTAV-IPN

[93] En la UNAM, INFOTEC, CINVESTAV-IPN y el CICESE.

[94] Koza, J. R. **Genetic Programming. On the Programming of Computers by Means of Natural Selection**. The MIT Press, Cambridge, Massachusetts, 1992.

[95] Kennedy, J., Eberhart, R.C. **Swarm intelligence**, Morgan Kaufmann, San Francisco, California, USA, 2001.

tículas en un fluído (agua o aire). Las trayectorias se definen a partir de una fórmula muy simple que involucra la velocidad de cada partícula, un factor de inercia y la influencia de la mejor solución que se haya obtenido para una partícula individual y para todo el cúmulo. Hay grupos de investigación que han usado la optimización mediante cúmulos de partículas, sobre todo para optimización mono-objetivo y multi-objetivo.[96]

- **Sistemas Inmunes Artificiales:** Busca simular el comportamiento de nuestro sistema inmune, el cual nos defiende de las enfermedades que invaden nuestro cuerpo en forma de "antígenos". Para ello, el sistema inmune genera células denominadas "anticuerpos", que tienen la capacidad de acoplarse a los antígenos a fin de anularlos. Los primeros modelos computacionales del sistema inmune datan de los ochenta y se popularizaron por Stephanie Forrest, quien realizó aportaciones clave en esta área.[97] Hoy en día existen diversas propuestas de modelos computacionales basados en sistemas inmunes, de entre los que destacan la selección negativa, la selección clonal y el modelo de red inmune. De ellos, la selección clonal ha sido el más utilizado en la literatura especializada (sobre todo para resolver problemas de optimización). Los sistemas inmunes artificiales se han usado por grupos de investigación en México para abordar problemas de criptografía y de optimización mono-objetivo y multi-objetivo.[98]

- **Evolución Diferencial:** Propuesta por Kenneth Price y Rainer Storn a mediados de los noventa,[99] se puede ver más bien como un método de búsqueda directa en el que se trata de estimar el gradiente en una

[96] En CINVESTAV-IPN, unidades Tamaulipas, Guadalajara y Zacatenco.

[97] De Castro, L., Timmis, J. **Artificial Immune Systems: A New Computational Intelligence Approach**, Springer, 2002.

[98] En el ITESM Campus Estado de México, el CIC-IPN y el CINVESTAV-IPN.

[99] Storn, R., Price, K. (1997). **Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces**. *Journal of Global Optimization*, 11(4):341–359.

región (y no en un punto) mediante el uso de un operador que recombina a 3 individuos. Pese a que su diseño luce muy simple, esta técnica es sumamente poderosa para realizar optimización en espacios continuos y produce resultados equiparables e incluso superiores a los obtenidos por las mejores Estrategias Evolutivas que se conocen, pero usando menos parámetros y con implementaciones mucho más sencillas. Hay grupos de investigación en México que han utilizado la evolución diferencial en diferentes problemas de optimización mono-objetivo y multi-objetivo.[100]

- **Colonia de Hormigas:** Propuesta por Marco Dorigo a principios de los noventa[101] simula el comportamiento de un grupo de hormigas que sale de su nido a buscar comida. Durante su recorrido segregan una sustancia llamada "feromona" la cual pueden oler las demás hormigas. Al combinarse los rastros de feromona de toda la colonia la ruta resultante es la más corta del nido a la comida. En el modelo computacional desarrollado por Dorigo se considera un factor de evaporación de la feromona y se debe poder evaluar una solución parcial al problema. En su versión original, esta metaheurística se podía utilizar sólo para problemas que se mapearan al problema del viajero. Sin embargo, con los años se desarrollaron versiones más sofisticadas que se pueden aplicar a otros tipos de problemas combinatorios e incluso a problemas continuos. En México, existen grupos de investigación que han utilizado esta metaheurística para resolver problemas de diseño de circuitos y de optimización mono-objetivo y multi-objetivo en espacios continuos.[102]

[100] En la Universidad Veracruzana y el CINVESTAV-IPN.

[101] Dorigo, M., Di Caro, G. (1999). **The ant colony optimization meta-heuristic**. En D. Corne, M. Dorigo y F. Glover (eds), *New Ideas in Optimization*, McGraw-Hill, UK, pp. 11–32.

[102] En CINVESTAV-IPN.

8.6. La Computación Evolutiva en México

Los orígenes de la investigación en computación evolutiva en México se remontan a finales de los años ochenta en el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) Campus Monterrey y en la Universidad Nacional Autónoma de México (UNAM). En el ITESM se creó el Centro de Inteligencia Artificial en 1989 (hoy llamado Centro de Sistemas Inteligentes) el cual, casi desde su inicio, tuvo interés en estudiar Sistemas Inspirados en la Naturaleza. En la UNAM existen tesis de licenciatura y doctorado de finales de los ochenta y principios de los noventa dirigidas por Germinal Cocho Gil, que utilizan el método de Monte Carlo y algoritmos genéticos,[103] si bien las primeras tesis que utilizan el término "algoritmo genético" en su título se remontan a 1993.[104]

Hacia la segunda mitad de los noventa existían ya cursos sobre computación evolutiva en instituciones tales como el Centro de Investigación en Computación del Instituto Politécnico Nacional (CIC-IPN), la Maestría en Inteligencia Artificial de la Universidad Veracruzana y la de la UNAM.

Actualmente, existen diversos posgrados en México en los cuales es posible obtener una maestría y un doctorado en computación con especialidad en Computación Evolutiva. Por ejemplo, en el Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN), el Centro de Investigación en Matemáticas (CIMAT), el Instituto de Investigación en Matemáticas Aplicadas y Sistemas de la UNAM (IIMAS-UNAM),

[103] Miramontes Vidal, O. R. **Algunos Aspectos de la Teoría de Autómatas Celulares y sus Aplicaciones en Biofísica**, Tesis de Licenciatura (Física), Facultad de Ciencias, UNAM, 1988.

Arce Rincón, J. H. **Estados de Mínima Energía en una Red a Temperatura Cero**, Tesis Doctoral (Doctor en Ciencias (Física)), Facultad de Ciencias, UNAM, 1990.

Miramontes Vidal, P. E. **Un Esquema de Autómata Celular como Modelo Matemático de la Evolución de los Ácidos Nucleicos**, Tesis Doctoral (Doctor en Ciencias (Matemáticas)), Facultad de Ciencias, UNAM, 1992.

[104] Comunicación personal de la Dra. Katya Rodríguez Vázquez, quien realizó diversas consultas al sistema TESISUNAM para proporcionar este dato.

el CIC-IPN, la Universidad de Guadalajara, el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), el Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación (INFOTEC) y en el ITESM.

De manera paralela a los trabajos que condujeron a la escritura de este capítulo, la sección de Computación Evolutiva de la Academia Mexicana de Computación creó un directorio (https://www.cs.cinvestav.mx/~compevol/) el cual pretende complementar el contenido de este capítulo con información de las áreas en las que trabajan los investigadores así como sus páginas web, donde puede encontrarse más detalle sobre el trabajo que han realizado y que llevan a cabo actualmente.

8.7. Perspectivas

La investigación en torno a la Computación Evolutiva ha tenido un crecimiento importante en México en los últimos años. Hoy en día existe un número considerable de investigadores que trabajan con distintos tipos de algoritmos evolutivos y con otras metaheurísticas bio-inspiradas, no sólo en torno a aplicaciones, sino también en investigación básica (por ejemplo, diseño de nuevos algoritmos, análisis teórico y estudio de operadores). Como se indicó anteriormente, existen también varios programas de maestría y doctorado reconocidos en el Padrón Nacional de Posgrado de CONACyT en los cuales es posible especializarse en Computación Evolutiva. Asimismo, existen grupos de investigación que colaboran con investigadores de varios países[105] y que tienen amplia visibilidad internacional.

En años recientes ha habido también una creciente participación de estos grupos en los congresos internacionales más importantes del área: la

[105] Todos los grupos de investigación de México mencionados a lo largo de este capítulo han tenido colaboraciones con investigadores de países tales como Estados Unidos, Escocia, Australia, España, Chile, Francia, Holanda, Alemania, China y Japón.

Genetic and Evolutionary Computation Conference, el *IEEE Congress on Evolutionary Computation* y *Parallel Problem Solving from Nature*. Sin embargo, son todavía muy pocos los grupos de investigación de México que publican en las revistas más reconocidas del área: *IEEE Transactions on Evolutionary Computation*, *Evolutionary Computation* y *Genetic Programming and Evolvable Machines*. Esto puede deberse a que varios de los grupos de investigación están conformados por doctores jóvenes en proceso de consolidación y también porque algunos de los grupos actuales están orientados al desarrollo de aplicaciones y prefieren publicar en revistas no especializadas en computación evolutiva.

Finalmente, es deseable que los grupos de investigación en Computación Evolutiva existentes en México que son relativamente recientes se consoliden y que su visibilidad aumente; sin duda hay el potencial para que esto ocurra dada la cantidad de colaboraciones internacionales con las que ya se cuenta.

La Computación en México por especialidades académicas,
se terminó de imprimir en septiembre de 2017 en
Agys Alevín S. C. Retorno de Amores No. 14-102.
Col. Del Valle. C. P. 03100, Ciudad de México.
En su composición se utilizó tipo Garamond.
Impreso en papel couché mate de 115 grs.
La edición consta de 240 ejemplares.