



CEU

*Universidad
San Pablo*

Escuela Politécnica Superior

¿Pueden pensar las Máquinas?

Mariano Fernández López

Profesor de la Universidad CEU San Pablo

Festividad de San José

Marzo 2010

¿Pueden pensar las máquinas?

Mariano Fernández López
Profesor de la Universidad CEU San Pablo

Festividad de San José
Marzo 2010

Escuela Politécnica Superior
Universidad CEU San Pablo

¿Pueden pensar las máquinas?

Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra solo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Dirijase a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.org) si necesita escanear algún fragmento de esta obra.

© 2010, Mariano Fernández López
© 2010, Fundación Universitaria San Pablo CEU

CEU *Ediciones*
Julián Romea 18, 28003 Madrid
www.ceuediciones.es

Depósito legal: M-XXXXX-2010

Agradecimientos:

*Excelentísimo y magnífico Sr. Rector de la Universidad CEU San Pablo.
Excelentísimas e ilustrísimas autoridades académicas, religiosas y civiles.
Ilustrísimo Director de la Escuela Politécnica Superior de la
Universidad CEU San Pablo
Estimados compañeros profesores, alumnos y amigos.
Señoras y señores*

En primer lugar, deseo agradecer el haber sido elegido para impartir esta lección del acto de celebración de la festividad de San José, no sólo por el honor personal que supone para mí, sino también porque entiendo que es un gesto de confianza de nuestro director, nuestra universidad y nuestra fundación hacia los profesores de la Escuela Politécnica Superior (EPS). También quiero agradecer la ayuda de algunos de los compañeros a afinar y verificar algunas de mis afirmaciones. No obstante, el responsable único de cualquier error soy yo.

1. Introducción

Cuando nos preguntamos: *¿pueden pensar las máquinas?*, nos tenemos que plantear: *¿qué significa pueden?* (Nilsson, 2000), *¿pueden las máquinas pensar en la actualidad?*, *¿podrán hacerlo en el futuro?*, *¿podrían ser las máquinas capaces de pensar* (incluso aunque nunca llegemos a construir ninguna que lo haga)? Estas variantes de la pregunta las consideraré a lo largo la exposición, donde también, basándome en los trabajos de Alan Turing, precisaré el significado de la palabra *máquina* (Turing, 1936-37) y *pensar* (Turing, 1950) en el contexto de la cuestión a analizar.

2. Las máquinas computadoras

2.1. Computadores naturales y humanos

La noción de computabilidad, aunque no expresada en los términos actuales, ha estado muy presente en las reflexiones y análisis de los sabios desde hace centurias. Durante siglos, la física clásica consideró el universo como un sistema determinista (Mainzer, 2004). De acuerdo con esta premisa, cada estado futuro del universo puede ser determinado de forma única por el estado presente. Concretamente, Laplace (1814) asumió la total computabilidad y predecibilidad de la naturaleza si todas sus leyes y todos los estados iniciales de los cuerpos celestiales fueran conocidos. Enfoques similares se han sostenido para el caso de la mente humana. De hecho, a lo largo de la historia, ha habido filósofos, como es el caso de Lull, Leibniz y quizás Boole (Laita, 2006), que han intentado reducir a computación el razonamiento humano.

Sin embargo, no fue hasta el siglo XX cuando se acuñó el término *computador*. A principios de dicha centuria, un computador era una persona, un ayudante matemático que calculaba de acuerdo con un algoritmo (una “receta”) suministrado por un superior (Copeland, 2004). Hubo miles de computadores humanos trabajando en el estado, centros de investigación, empresas, etc.

2.2. El concepto matemático de computación

La definición formal de *computación* se llevó a cabo en el contexto de las investigaciones que se realizaron a principios del siglo XX sobre las posibilidades de deducir toda la matemática a partir de una serie de axiomas y definiciones mediante un método efectivo, es decir, un método ejecutable en un número finito de pasos y que no requiriera ni la perspicacia ni el ingenio. Si el proyecto hubiera tenido éxito, una máquina que operara al modo de los primeros computadores humanos, en caso de disponer de suficiente tiempo y espacio, hubiera permitido deducir todo el conocimiento matemático. Ahora bien, tal intento fracasó, como mostró el profesor Laita (2006) en su lección magistral de Inauguración del Curso Académico 2006-2007 de la Universidad CEU San Pablo, porque tal máquina ni existe ni puede existir. No obstante, una de las consecuencias más importantes

de este proyecto fue la definición formal de *máquina de computación* que proporcionó Alan Turing (1936-37).

Una *máquina de Turing* se compone simplemente de una cinta potencialmente infinita dividida en recuadros y un cabezal lector-escritor que es capaz de leer, borrar y escribir, símbolos en la cinta, así como de moverse a izquierda y derecha de la cinta (véase la figura 1). La máquina, después de cada acción, pasa de estar en un cierto estado q_i , a estar en un estado q_j (Turing, 1936-1937; Isasi et al., 1997; Hopcroft et al., 2007). La definición se basa en la siguiente intuición:

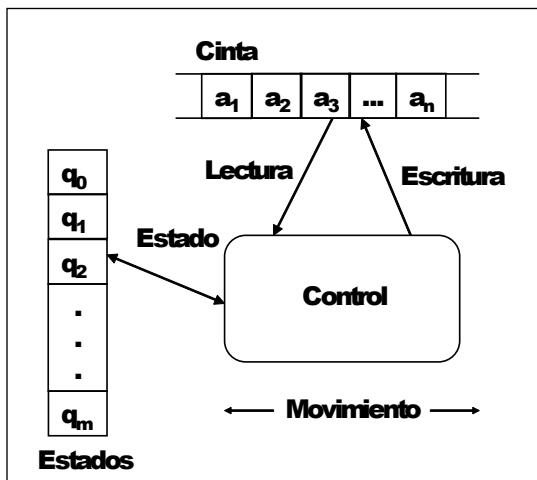


Figura 1. Máquina de Turing (Isasi et al., 1997)

“La computación se efectúa normalmente escribiendo ciertos símbolos en un papel. Podemos suponer que este papel se divide en cuadrados como el cuaderno de aritmética de un párvulo. En aritmética elemental se aprovecha a veces el carácter bidimensional del papel. Pero se puede prescindir de ese uso, y pienso que se estará de acuerdo conmigo en que el carácter bidimensional del papel no es esencial para la computación. Doy así por supuesto que la computación se lleva a cabo en papel unidimensional, es decir, en una cinta dividida en cuadrados. Supondré también que el número de símbolos que se puede imprimir es finito. El efecto de esta restricción no es muy serio. Siempre será posible usar símbolos aislados en lugar de secuencias de símbolos (...).

El comportamiento del computador¹ está en todo momento determinado por los símbolos que está observando y por su ‘estado mental’ en ese momento. Podemos suponer que hay un límite L en el número de símbolos o cuadrados que el computador puede observar en un momento. Si desea observar más, habrá de hacerlo en observaciones sucesivas. Supondremos también que el número de estados mentales a tener en cuenta es finito (...). Tampoco esta restricción afecta seriamente a la computación, porque el uso de estados mentales más complicados se puede suplir escribiendo más símbolos en la cinta.” (Turing, 1936-1937, págs. 135-136)

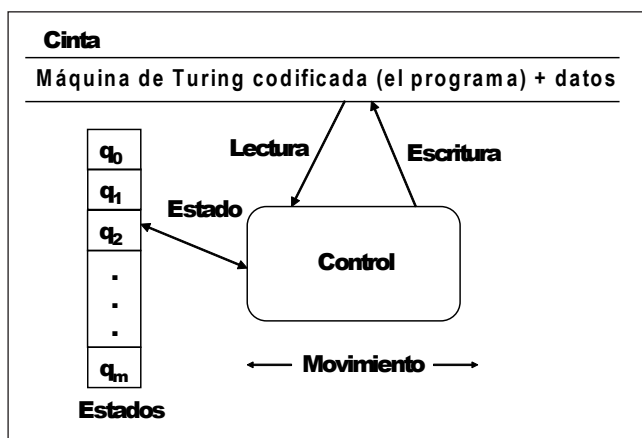


Figura 2. Máquina universal de Turing

La *máquina universal de Turing* es una abstracción teórica de computador con programa almacenado, pues ésta puede contener en su cinta otra máquina de Turing que puede ir ejecutando (véase la figura 2). De acuerdo con una de las formulaciones de la (hipó)tesis² de Church-Turing, cualquier método efectivo puede ser llevado a cabo por una máquina universal de Turing (Copeland, 2004). En consecuencia, cualquiera de los algoritmos que ejecutaban los antiguos computadores humanos, de acuerdo con esta hipótesis, puede ser ejecutado por una de estas máquinas.

¹ Aquí Turing se refiere a la persona que realiza la computación, de hecho, Garrido (2001, pág. 377) lo traduce como “*persona que calcula*”

² Se le suele llamar “tesis de Church-Turing”, aunque en realidad no está validada.

2.3. Computadoras electrónicas

El término *máquina computadora* se utilizó para referirse a máquinas que automatizaban elementos del trabajo de los computadores humanos. Las primeras máquinas computadoras eran algo parecido a las calculadoras no programables de mano que se utilizan actualmente. Para cálculos complejos, se podían necesitar varias decenas de computadores humanos. En los años 40, la escala de algunos cálculos que requerían físicos e ingenieros había llegado a ser tan grande que el trabajo no podía ser realizado en un tiempo razonable por computadores humanos, ni aun ayudados por máquinas. A finales de los años 40 y principios de los 50, con la llegada de las máquinas computadoras electrónicas, el término *máquina computadora* dejó lugar al término actual *computadora* o *computador*.

A pesar de que Alan Turing ya había definido formalmente una computadora con programa almacenado, las primeras computadoras electrónicas (salvo una de las construidas por Conrad Zuse en Alemania) no poseían esta característica y requerían cambiar la configuración del cableado cada vez que tenían que realizar una nueva tarea (Copeland, 2002). Ahora bien, una vez terminada la II Guerra Mundial, liberados de las tareas orientadas directamente a ganar la contienda, hubo científicos, por ejemplo tanto el propio Alan Turing como John von Neumann, que investigaron sobre las posibilidades de materializar la idea de computadora con programa almacenado. Un trabajo con especial impacto en la resolución de este problema fue el publicado por von Neumann (1945). La arquitectura (estructura) propuesta en el documento donde aparece von Neumann como autor ha sido el fundamento de la arquitectura de las computadoras durante décadas (von Neumann, 1945). El borrador propone las siguientes partes para la computadora:

Primera: puesto que el dispositivo es ante todo una computadora, tendrá que realizar las operaciones elementales de aritmética con más frecuencia. Éstas son: adición, sustracción, multiplicación y división (...). Es, por tanto, razonable que deba contener órganos especializados para sólo estas operaciones.

Segunda: El control lógico del dispositivo, es decir, la secuenciación apropiada de sus operaciones puede ser llevada a cabo de forma más eficiente por un órgano central de control. (...).

Tercera: cualquier dispositivo que lleve a cabo una secuencia larga y complicada de operaciones (especialmente de cálculos) debe tener una memoria considerable. (...)

El dispositivo debe estar dotado de la capacidad de mantener el contacto de entrada y salida (sensorial y motor) con un medio de este tipo (soporte magnético, teclado, etc.). (...)”(von Neumann, 1945, págs. 1 y 2)

En el día de hoy, la primera parte se conoce como *unidad aritmético lógica*; la segunda, como *unidad de control*; y la tercera, como *memoria principal* (véase la figura 3). Los sensores y los actuadores son los *periféricos*. Los registros que aparecen en la figura 3 son dispositivos de almacenamiento de acceso más rápido que la memoria principal.

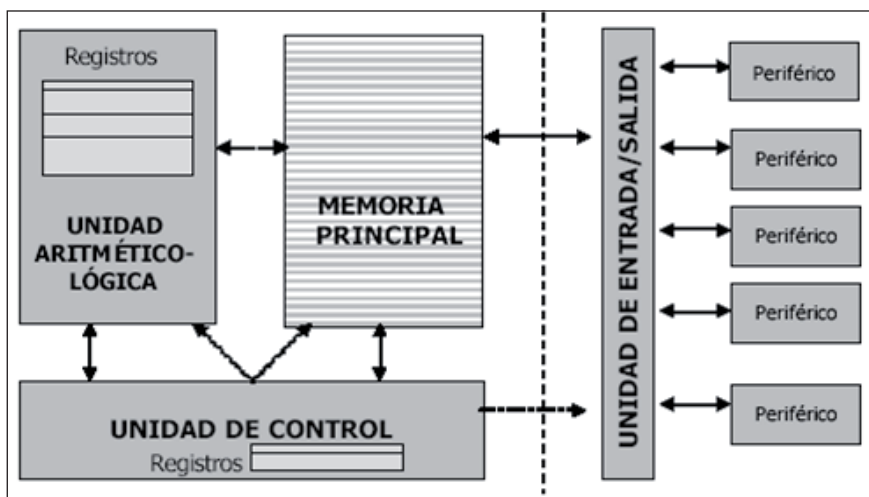


Figura 3. Arquitectura von Neumann (Fernández-Saavedra, 2009)³

Tal y como muestran Hopcroft, Motwani y Ullman (2007), por una parte, una máquina de Turing puede ser simulada mediante una computadora electrónica. Por otra parte, cualquiera de las computadoras electrónicas actuales puede ser simulada mediante una máquina de Turing. Sin embargo, el hecho de saber que una máquina de Turing resuelve un determinado problema en un número finito de pasos no resulta práctico si dicho tiempo es excesivo en una computadora electrónica (Alfonseca-Cubero et al., 2007). Así, por ejemplo, si disponemos de una máquina que realiza un millón de instrucciones por segundo (un mips)

³ Según comunicación personal de Gianluca Cornetta y Luis Azorín, el primer nivel de memoria caché sigue una arquitectura Harvard. No obstante, la figura 3 les puede dar a ustedes una idea aproximada de cuál es la estructura de una computadora moderna. (Cualquier error que pueda haber en esta afirmación, al igual que en el resto de la lección, es responsabilidad exclusiva del autor, que es quien está traduciendo e interpretando las fuentes).

(lenta para las computadoras personales actuales), entonces un algoritmo que requiera n^2 instrucciones, para una entrada de tamaño $n = 500$, necesitará un cuarto de segundo. Incluso si se necesitaran n^3 instrucciones, el procesamiento de una entrada de tamaño 500 requeriría no más de 2 minutos y 5 segundos. Sin embargo, un algoritmo que requiriera 2^n instrucciones, en el peor caso necesitaría 35 años para una entrada de tamaño 50 (Urquhart, 2004).

3. La palabra pensar en el contexto de la presente lección

Cuando Alan Turing en su trabajo presentado en la revista *Mind* en 1950 se planteó la pregunta que ahora nos hacemos, escribía:

“Propongo considerar la cuestión, ‘¿pueden pensar las máquinas?’ Esto debería empezar con las definiciones del significado de los términos ‘máquina’ y ‘pensar’. (...). En lugar de intentar tal definición reemplazaré la pregunta por otra, que está íntimamente relacionada con ella y está expresada en palabras relativamente no ambiguas.

La nueva forma del problema se puede describir en términos de un juego que se puede llamar el ‘juego de imitación’. Lo juegan tres personas, un hombre (A), una mujer (B), y un interrogador (C) que puede ser de cualquiera de los dos sexos. El interrogador permanece en una habitación aparte de la de los otros dos. El objeto del juego para el interrogador es determinar cuál de las otras dos personas es el hombre y cuál es la mujer. Los conoce por las etiquetas X e Y, y al final del juego dice bien ‘X es A e Y es B’ o ‘X es B e Y es A’. Al interrogador se le está permitido hacer preguntas a A y B así:

C: ¿Podrá X, por favor, decirme la longitud de su pelo?

Ahora suponga que X es realmente A, entonces A deberá responder. El objetivo de A en el juego es intentar y causar que C se equivoque en la identificación. Su respuesta puede ser por tanto:

‘Mi pelo está cortado a lo garçon y los mechones más largos tienen en torno a nueve pulgadas de largo.’

Con el propósito de que los tonos de voz no puedan ayudar al interrogador las respuestas deberían ser escritas, o mejor aún, mecanografiadas. La disposición

ideal es tener un teletipo entre ambas habitaciones. Otra alternativa es que las preguntas y las respuestas sean repetidas por un intermediario. El objeto del juego para el tercer jugador (B) es ayudar al interrogador. La mejor estrategia para ella es probablemente proporcionar respuestas verdaderas. Puede añadir cosas como ‘soy la mujer, no le escuches a él’ en sus respuestas, pero no le servirá de nada dado que el hombre puede hacer comentarios similares.

Ahora preguntamos: ‘¿qué ocurrirá cuando una máquina tome la parte de A en este juego? ¿Decidirá erróneamente el interrogador tan a menudo como cuando juegan un hombre y una mujer? Estas cuestiones reemplazan la nuestra original: ‘¿pueden pensar las máquinas?’” (Turing, 1950, pág. 433).

Aquí hay que decir que Turing establece un límite de 5 minutos de interrogatorio.

El propio Turing reconocía que la inteligencia es algo tan complejo, que este test era muy limitado, pero que intentando métodos similares a éste, alguna vez se llegaría a atribuir la capacidad de pensar a las máquinas.

Es especialmente interesante, por una parte, observar que el test marca el nivel de dificultad del problema que debe resolver una máquina para ser considerada inteligente: engañar a una persona bajo ciertas circunstancias. Por otra parte, marca el nivel de eficacia mínima: hacerlo tan bien como una persona. No obstante, hay una interrogante importante que queda abierta: ¿qué persona? El hecho de que un sistema sea capaz de recomendar un tratamiento para la migraña mejor que alguien no vinculado al mundo sanitario que desconoce la enfermedad no es significativo sobre su validez. Sin embargo, el proponer tratamientos tan buenos, o mejores, que los de los especialistas en la enfermedad sí es una prueba contundente de tal validez.

4. ¿Pueden pensar las computadoras actuales?

4.1. ¿Qué son capaces de hacer las computadoras actuales?

Según Urquhart (2004), existe una máquina de Turing que representa la resolución del test. Dado que en el mundo formal se prescinde de los detalles materiales, no hay transcurso del tiempo; sin embargo, el límite de tiempo que marca el juego puede transformarse en un límite del número de símbolos intercambiado entre sus participantes. La máquina tiene una tabla que asocia a cada interrogatorio (con sus preguntas y respuestas) si revela, o no, la identidad del entrevistador (véase la figura 4). Ahora bien, puesto que la tabla es enorme y en el mundo material sí transcurre el tiempo y sí hay espacio, y tanto el uno como el otro imponen una serie de restricciones, se plantean dos problemas prácticos que en la actualidad son inabordables: (1) construir la tabla; y (2) realizar la búsqueda en un tiempo razonable. De hecho, ninguna computadora electrónica ha logrado, hasta ahora, pasar el test de Turing.

Sin embargo, hay una serie de problemas cuya resolución se ha considerado tradicionalmente que requiere de inteligencia, en que las computadoras actuales han mostrado una competencia igual (a veces mayor incluso) que la de las personas. Así, por ejemplo, en algunos juegos, la computadora es claramente superior. En el caso concreto de las damas, en 1994, el programa informático Chinook se proclamó campeón del mundo. En 1996, era nítido que la máquina era superior a la persona en este juego y, por tanto, Chinook se retiró⁴. En ajedrez, en el año 1997, el entonces campeón mundial, Gari Kasparov, fue derrotado por la computadora Deep Blue (Campbell et al., 2002). Cinco años después, a finales de noviembre y principios de diciembre de 2006, se enfrentaron quien acababa de proclamarse campeón del mundo, Vladimir Kramnik, y el programa de ajedrez Deep Fritz⁵. Cuatro de las seis partidas terminaron en tablas, y las otras dos las ganó Deep Fritz. Es decir, el programa obtuvo la victoria por un contundente 4 a 2.

⁴ <http://webdocs.cs.ualberta.ca/~chinook/project/>

⁵ <http://www.chessbase.com/newsdetail.asp?newsid=3524>

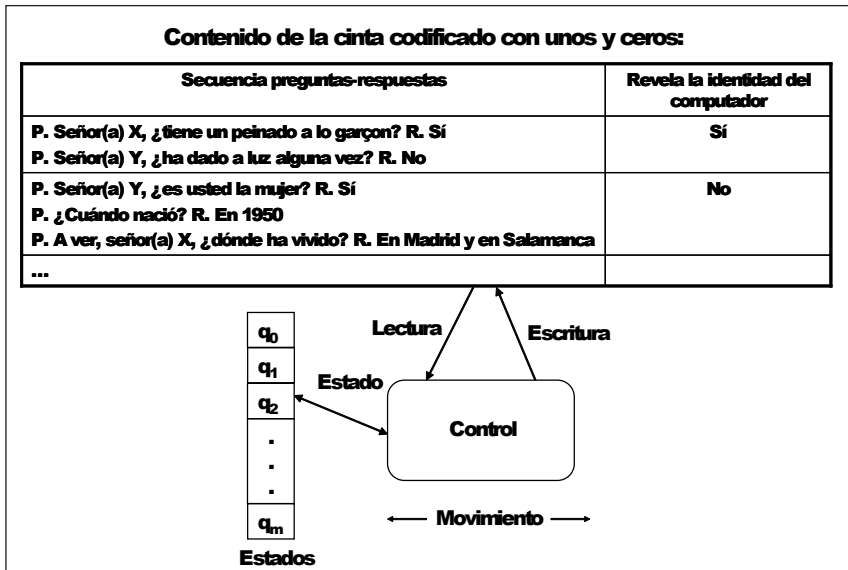


Figura 4 . Máquina que, según Urquhart (2004), resuelve el test de Turing

Hace pocos años, un juego que se estaba mostrando especialmente difícil para las computadoras era el Go, bastante popular en Asia oriental. De hecho, Jonathan Schaeffer, director del proyecto Chinook, afirmaba (Schaeffer, 2004, pág. 16): *“no tenemos ni idea de cómo abordar este dominio (el del Go) eficazmente con computadoras”*. Sin embargo, en agosto de 2008, el programa Mogo derrotó al jugador profesional Myungwan Kim⁶.

El propio Jonathan Schaeffer afirma que en los enfrentamientos máquina-hombre los puntos fuertes de la máquina frente a la persona son su capacidad de: (1) cálculo y de enumeración exhaustiva; (2) realización de tareas repetitivas; y (3) memoria de gran tamaño en infalible. Por otra parte, los puntos fuertes de la persona frente a la máquina son su capacidad de: (1) procesamiento de lenguaje y de visión; (2) generalización; y (3) razonamiento por analogía.

En el día de hoy, dado que las facultades del hombre y de la máquina son complementarias, en problemas prácticos, más que entrar ambos en competencia, ésta última sirve de apoyo al primero en la toma de decisiones (Ríos-Insúa et al., 2001). Algunas aplicaciones en que se da esta colaboración son relativas a sistemas de diagnóstico de enfermedades (Rodríguez-Solano et al., 2005), supervisión de

⁶ http://www.usgo.org/index.php?%23_id=4602

enfermos en unidades de cuidados intensivos (Otero et al., 2009), reconstrucción de imágenes 3D a partir de microscopías electrónicas (Sorzano, 2009), sistemas de ayuda a la decisión en entornos financieros (Power, 2007), conducción automática de vehículos (Pomerleau, 1992; Jonathan, 2004), recomendaciones de lugares de interés en el Camino de Santiago sintetizando la experiencia de otros peregrinos⁷ etc.

4.2 ¿Cómo funcionan los sistemas inteligentes artificiales?

Tradicionalmente, dos de las formas de funcionamiento que han seguido los sistemas inteligentes han sido: la imitación de la estructura del cerebro (McCulloch y Pitts, 1943; McLaughlin, 2004) y procesamiento de símbolos mediante el razonamiento lógico-matemático (Craik, 1943; Newell y Simon, 1976; Haugeland, 1985), conocido como *enfoque simbólico*.

El caso más conocido de imitación de la estructura del cerebro es el de las redes de neuronas artificiales. Cada neurona es un elemento simple que interactúa, a través de unas conexiones llamadas *sinapsis*, con las demás que forman la red. De acuerdo con este modelo, la inteligencia emerge a partir de la interacción de una cantidad mayor o menor (dependiente de la dificultad del problema) de componentes sencillos (las neuronas). Así, por ejemplo, el sistema de conducción automática de vehículos ALVINN (Pomerleau, 1992) utilizaba una red de 994 neuronas y 3994 sinapsis para procesar las imágenes de la carretera que le llegaban desde una cámara. Según fueran estas imágenes, la red ordenaba girar suavemente el volante a la izquierda, bruscamente a la derecha, seguir recto, etc. El proceso de aprendizaje consistió fundamentalmente en que un conductor humano llevaba el coche, y la red iba simulando órdenes e iba reforzando unas sinapsis y debilitando otras cuando tales órdenes simuladas discrepaban de las acciones que realizaba el conductor. Después de este aprendizaje, la máquina ya fue capaz de conducir de manera autónoma a velocidades superiores a 110 km/h a lo largo de más de 140 km en autopistas públicas en presencia de otros vehículos⁸.

⁷ <http://www.geobuddies.net/web/guest;jsessionid=17CFB6E30CE0BF4B76BF1D8A7BDDDC4C>

⁸ La red de neuronas artificiales no es el único componente *inteligente* de ALVINN, pero sí es el principal.

Cuando el sistema funciona de acuerdo con el enfoque simbólico, la máquina tiene representados una serie de principios básicos del dominio de la tarea a realizar y, utilizando un motor de inferencias, se generan todas las afirmaciones que pueden ayudar en la resolución de problemas. Así, por ejemplo, los ajedrecistas automáticos tienen codificados principios como:

- o El rey está en jaque si está amenazado en la casilla en que se encuentra.*
- o El alfil mueve en diagonal.*
- o En la evaluación de una posición, el peso del número de casillas donde puede mover la computadora vale 900.*
- o Etc.*

Luego, la generación y elección de movimientos, así como la comprobación de movimientos del contrario se realiza mediante deducción automática a partir de tales principios.

La práctica ha demostrado que el enfoque simbólico es adecuado para llevar a cabo una descripción computacional de tareas para las que se dispone de documentación de la que extraer y personas de las que educir los pasos de razonamiento que se siguen para resolver los problemas. El razonamiento puede estar basado en la experiencia de personas particulares o/y en teorías más o menos formalizadas. La imitación de la estructura del cerebro, por otra parte, es adecuada para problemas para los que se tiene un conjunto de casos que permitan entrenar el sistema (Mira, 2008).

4.3 ¿Cómo se construyen los sistemas inteligentes artificiales?

A quienes nos dedicamos a la computación, con frecuencia se nos imagina trabajando exclusivamente con bits, circuitos lógicos y microprocesadores. Sin embargo, uno de los aspectos fascinantes del trabajo en esta disciplina es el de los niveles de abstracción (Newell, 1982; Colburn, 2004). De hecho, uno de los conceptos clave en computación, al igual que en matemáticas, es la abstracción. Ahora bien, mientras que la primera de estas disciplinas elimina contenido de los términos, la segunda elimina detalles formales. En el caso de las matemáticas, por ejemplo, el concepto de espacio euclídeo en \mathbb{R}^3 prescinde del hecho de que

los vectores representen velocidades o fuerzas. Incluso, el concepto de espacio vectorial hace inútil la representación gráfica de vectores mediante flechas. Sin embargo, en el desarrollo de sistemas computacionales los niveles más abstractos están más cerca de la realidad en que vivimos las personas (mientras que se alejan del sustrato físico y de los detalles de implementación). Así, por ejemplo, cuando se construye un sistema de diagnóstico de enfermedades, el ingeniero diseña representando conceptos como *síntoma*, *fiebre*, *dolor de cabeza*, etc. Luego, el programador codifica estos conceptos utilizando un lenguaje formal. Es más, hay partes del diseño que son transformadas en código de manera automática por un programa. Finalmente, un programa compilador, transforma el código obtenido a partir del diseño en las instrucciones que puede ejecutar la máquina. De esta manera, la parte más creativa del proceso se realiza en un nivel próximo a los expertos del dominio.

4.4. Funcionamiento de un sistema que juega el ajedrez

La resolución del ajedrez de forma perfecta, debido a la gran cantidad de posibilidades de juego que se plantean, se considera hoy en día inviable (véase Shannon, 1950 así como Russel y Norvig, 2004, pág. 182). En consecuencia, el juego por computadora debe incluir métodos de aproximación.

La figura 5 muestra la arquitectura del sistema que juega al ajedrez que estamos construyendo en la EPS de la Universidad CEU San Pablo. En su desarrollo han colaborado hasta ahora profesores, alumnos de 5º de Ingeniería Informática y un alumno de Ingeniería Técnica Telemática. Los objetivos que nos hemos planteado en el desarrollo de este sistema son: (1) facilitar a los estudiantes de ingeniería el conocimiento de los fundamentos de los sistemas inteligentes y (2) validar ideas que son potencialmente aplicables en otros sistemas. Como se puede observar, no tenemos en mente pagarle un millón de euros al campeón del mundo de ajedrez para que juegue contra nuestra máquina; sino que estamos pensando más en cuestiones docentes y de investigación.

El ajedrecista artificial imita en cierta medida a un jugador humano. Así, el órgano de visión del ajedrecista computacional es una cámara (*webcam*) que está controlada por el *reconocedor de movimientos*, que a partir de las capturas de la imagen del tablero, después de mover la persona, calcula la diferencia con

respecto a la imagen anterior al desplazamiento de la pieza. De esta forma, la computadora puede conocer qué movimiento ha realizado su adversario (véase la figura 6). Ésta es una solución simple que no considera, por ejemplo, problemas con la iluminación. Para resolverlos, posiblemente utilicemos en el futuro una red de neuronas artificiales.

La acción sobre el mundo exterior la realiza el ajedrecista artificial a través de un brazo robot, que, tal y como muestra la figura 7, a imitación del brazo humano, consiste en una base de rotación (que hace el papel del cuerpo al que va unido el brazo), un hombro, un codo, una muñeca y unas pinzas (que desempeñan el papel de dedos). El modelo que implementa el *controlador del brazo*, que permite llevarlo al lugar donde está la pieza a mover, y trasladar ésta a su destino, se ha elaborado planteando y resolviendo un sencillo sistema de ecuaciones trigonométricas. La complicación más importante ha surgido en la afinación del modelo comprobando los errores reales que cometía el robot. El desarrollo e integración de este sub-sistema de acción sobre el exterior ha sido el producto de un proyecto final de carrera de Ingeniería Técnica Telemática de la EPS (Márquez, 2009).

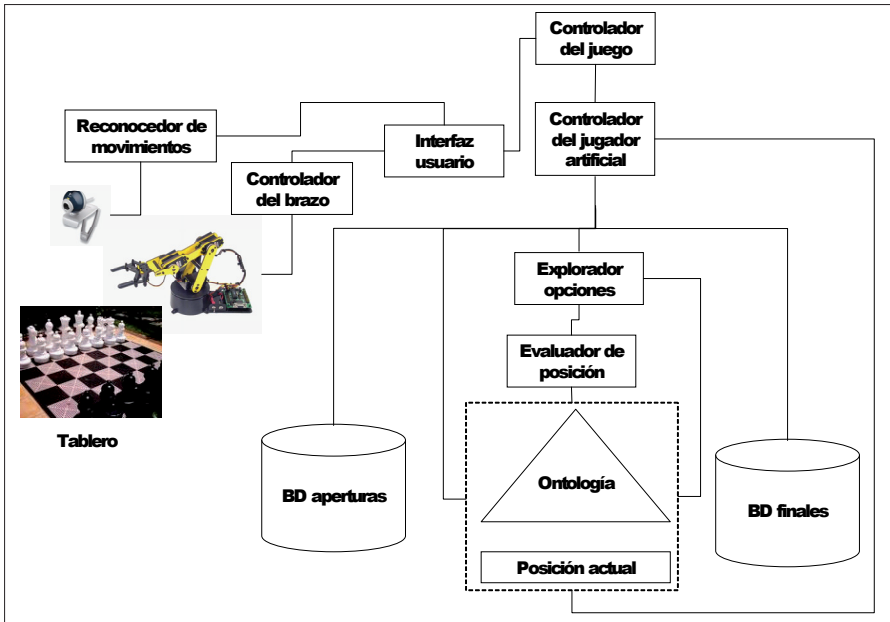


Figura 5 . Arquitectura de nuestro sistema que juega al ajedrez

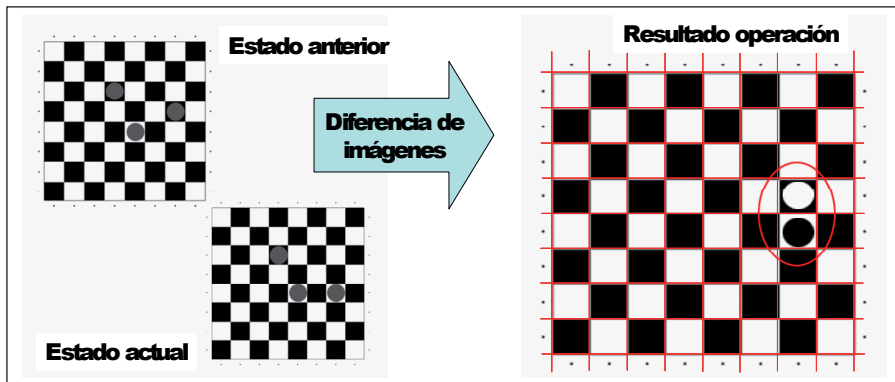


Figura 6 .Detección de movimiento mediante diferencia de imágenes (Márquez, 2009)

Nótese que de lo dicho anteriormente se puede deducir que las partes esenciales de la *interfaz de usuario* son el sistema de visión y el robot.

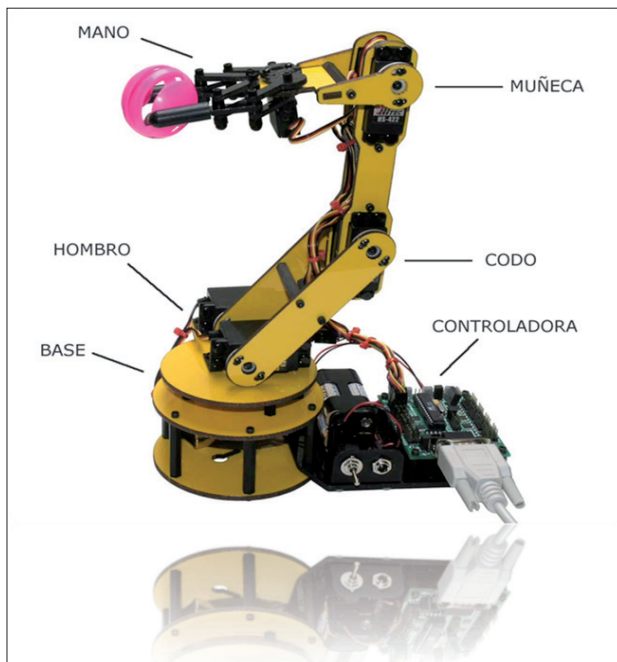


Figura 7 . Brazo Lynx 5 ® (Márquez, 2009)

También existe semejanza con los jugadores humanos en la toma de decisiones a la hora de elegir el movimiento a efectuar. De forma similar a los jugadores profesionales (o, hablando de forma más exacta, superándolos), la máquina sabe, gracias a una *base de datos de aperturas*, qué movimientos han dado mejores resultados en la apertura de partidas de otros jugadores o incluso en aquellas en que ha participado ella misma. Así, tal y como se puede comprobar en la figura 8, la salida *e4* llevó a la victoria en un 55,1 % de las partidas almacenadas que empezaron con este movimiento. En consecuencia, se considera una buena alternativa.

Con respecto a la fase final del juego, aunque, como se ha comentado anteriormente, no es viable expandir todo el árbol de búsqueda del ajedrez al principio de la partida, sí es posible hacerlo para posiciones donde quedan ya pocas piezas en el tablero. Para estos casos, se han elaborado *bases de datos de finales* que contienen cuáles son los movimientos ganadores. Por ejemplo, en la posición de la figura 9, la base de datos está indicando que si las blancas mueven su rey a *c6*, ganan la partida en 5 movimientos.

Como ocurre con los jugadores humanos, en aquellas fases del juego en que el ajedrecista automático no dispone de información adicional, debe realizar él mismo una exploración de las posibilidades que tiene. Es decir, cuando ni la base de datos de aperturas ni la de finales son útiles, ejecuta el *explorador de opciones*, que a su vez utiliza el *evaluador de posiciones* para valorar lo buena o lo mala que es cada posición para la máquina. La estrategia que sigue consiste en simular, a través de un algoritmo conocido como alfa-beta (véase, por ejemplo, Russell y Norvig, 2004), todas las posibilidades de una serie de movimientos de la máquina y contra-movimientos del humano. Alfa-beta evita la simulación de posiciones que se sabe a priori que no suponen una mejora sobre las ya simuladas.

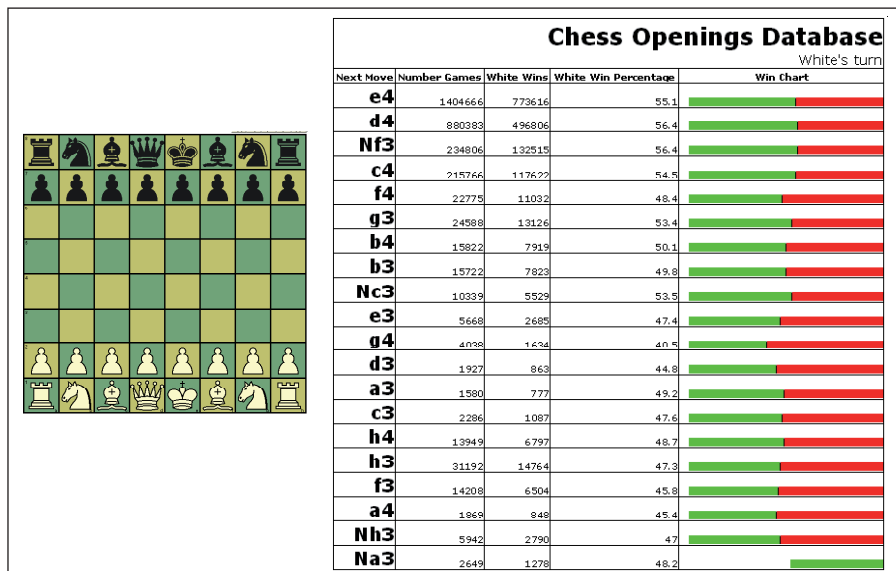


Figura 8. Visualización de una base de datos de aperturas *

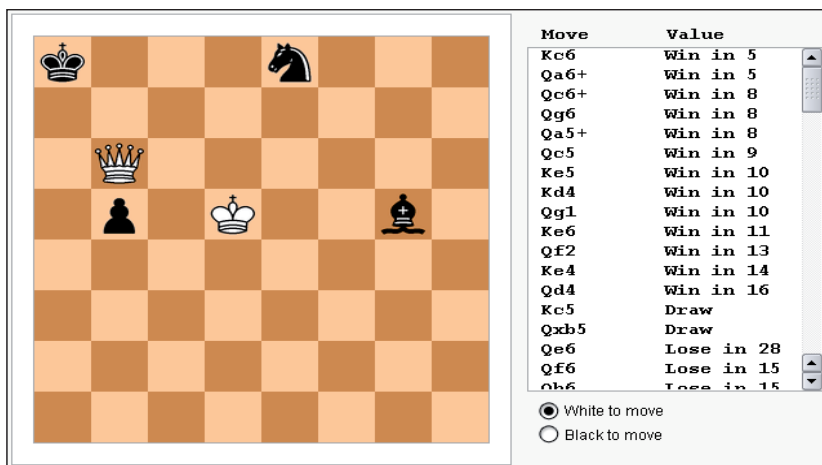


Figura 9. Visualización de una base de datos de finales **

* Véase <http://chessopeningsdatabase.com/>

** Véase http://upload.wikimedia.org/wikipedia/commons/a/af/Chess_tabquery.png

Si la capacidad de la memoria de la computadora lo permite, en la exploración de opciones, en caso de que alguna de ellas sea sospechosamente beneficiosa para la máquina, ésta continúa la simulación por si se tratara de una trampa de su adversario.

Pero... ¿cómo funciona el *evaluador de posición*? Básicamente realiza una suma ponderada de valores de características de la posición como: las piezas que tiene cada jugador en el tablero, cuántos movimientos puede realizar cada uno de ellos, los hilados que tienen sus peones, cuántos peones libres tienen, etc.

La definición computable del sistema de conceptos de un programa inteligente se llama *ontología* (Studer et al., 1998; Gruber, 1993; Smith, 2004). Para el caso que nos ocupa, algunos de estos conceptos son: *pieza, rey, movimiento legal, jaque, jaque mate*, etc., y están implementados en lenguaje Prolog (Sterling y Shapiro, 1994). Dado que estos conceptos, con sus definiciones correspondientes, están extraídos de la página oficial de la FIDE⁹, podemos afirmar que la ontología de nuestro sistema es compartida por los humanos que jugamos al ajedrez. Una ventaja importante de que sea compartida es que la ontología puede ser reutilizada en otros sistemas. Además, se facilita la posibilidad de que, en el futuro, nuestra computadora pueda dialogar con otras para tomar decisiones sobre el juego.

La ontología permite describir la *posición actual*, que representa la situación de las piezas en el tablero y quién posee el turno.

Otros componentes del ajedrecista artificial son: el *controlador del juego*, que va cambiando el turno entre la máquina y el humano, modificando la posición actual y verificando los movimientos del humano; y el *controlador del jugador artificial*, que decide, por ejemplo, si hay que ejecutar un movimiento de la base de datos de aperturas, uno de la de finales o uno obtenido del explorador de opciones.

Aunque es cierto que los problemas prácticos no suelen seguir unas reglas tan bien definidas como las del ajedrez, sí es verdad que algunas de las técnicas utilizadas en el desarrollo de un ajedrecista automático se pueden aplicar a otros problemas. Por ejemplo, el control del robot puede ser generalizable a tareas propias de entornos industriales. La idea de las bases de datos de aperturas puede ser aplicada al razonamiento basado en casos. Así, en medicina, se puede tener una base de datos de historiales de enfermos y utilizarlos en la resolución de problemas de salud de otras personas. El concepto de función de evaluación

⁹ <http://www.fide.com/component/handbook/?id=32&view=category>

puede ser extendido a la evaluación de soluciones candidatas en problemas de toma de decisiones en la empresa.

Es también importante mencionar que hemos construido la primera ontología de ajedrez. Dadas las características de este juego, esta experiencia nos ha permitido extraer conclusiones de interés sobre el proceso de diseño, implementación y evaluación de ontologías.

5 ¿Podrán pensar las máquinas en el futuro?

Para responder a esta pregunta, voy a ir considerando diferentes escenarios futuros que hipotéticamente podrían darse.

5.1 Escenario futuro 1: computadoras con cantidades ingentes de conocimiento

Uno de los motivos por el que la investigación en ontologías cobró gran vigor a partir de los años 90 fue precisamente la intención de transitar a este escenario (Gómez-Pérez et al., 2003). La idea era conseguir que las ontologías fueran componentes reutilizables y compartibles que hicieran que los sistemas inteligentes no fueran sólo útiles en problemas muy concretos (por ejemplo, diagnóstico de una determinada enfermedad), sino también en problemas más amplios (por ejemplo, el diagnóstico de una familia de enfermedades).

Un caso específico de esta idea es *la Web Semántica*. Se basa en la representación del contenido de la Web de tal forma que permita, no sólo a las personas, sino también a las computadoras hacer inferencias sobre él (Berners-Lee, 1999). Esta representación se lleva a cabo utilizando ontologías.

Suponiendo que la Web Semántica llegara a tener una extensión aproximada, o mayor, que de la Web actual, se podría contar con agentes autónomos artificiales inteligentes que nos ayudaran en decisiones de configuración de viajes (vuelos, hoteles, rutas, etc.), elaboración de discursos e informes de investigación, compra de productos básicos, etc. Es decir, parte de nuestras decisiones las delegaríamos en nuestros agentes artificiales.

Si se lograra implementar la Web Semántica, los sistemas informáticos, colaborando unos con otros, dispondrían de cantidades ingentes de información con las que obtener conclusiones y hacer recomendaciones. Sin embargo, uno de los problemas que hay planteados para llegar a culminar esta idea es que la cantidad de información que contiene la Web tradicional es tan enorme, que se necesita un procesamiento automático bastante complejo para realizar la transformación a la Web Semántica (Benjamins et al., 2002). Otro problema importante y complicado de resolver es que, para muchos dominios, es imposible conseguir una sola ontología acordada por absolutamente toda la comunidad. En consecuencia, es necesario establecer correspondencias entre conceptos de unas ontologías y otras (Euzenat y Shvaiko, 2007). Dado el tamaño que pueden llegar a tener algunas ontologías y los cambios que pueden sufrir éstas a lo largo de su tiempo de vida, la determinación de correspondencias debe estar automatizada.

Mi experiencia trabajando en este escenario, que es donde se desarrolla fundamentalmente mi actividad investigadora, me plantea una cuestión de carácter filosófico. Según Julián Marías (2001), desde la Antigua Grecia hasta aproximadamente la época de Galileo, los sabios estuvieron preocupados sobre qué son las cosas, qué permanece en las cosas incluso cuando cambian. Es decir, estaban interesados en extraer la esencia de las cosas. Sin embargo, la posición preponderante del nominalismo al final de la Edad Media llevó a una nueva ciencia, que, utilizando símbolos, estaba más enfocada a cómo codificar las características de las cosas (tamaño, velocidad, peso, etc.). Ahora bien, actualmente, quienes construimos ontologías, siguiendo el enfoque de la nueva ciencia, codificamos las características de las cosas y, por otra parte, intentamos buscar la esencia de las cosas, pues cuanto más capaces seamos de extraerla, más reutilizables y compartibles serán nuestros productos y, por tanto, en mejor medida habremos conseguido nuestros objetivos. Por tanto, ¿no estaremos en el camino de la síntesis de la nueva ciencia y la más tradicional ontología (como disciplina filosófica)?

5.2 Escenario futuro 2: la máquina de Turing sigue “en vigor”, pero el silicio, no

Las computadoras basadas en la arquitectura von Neumann o la de Harvard no son la única opción de emulación de la máquina de Turing. Dos de las alternativas más pujantes a estos modelos, al menos en los laboratorios de investigación, son las computadoras con ADN (Adleman, 1994; Lipton 1995) y las basadas en computación cuántica (Gruska, 1999). Ambas se caracterizan por el altísimo grado

de paralelismo en la computación. En el primer caso, los símbolos de entrada al sistema, es decir, el equivalente a los símbolos escritos en la cinta de la máquina de Turing, son las bases nitrogenadas citosina, guanina, timina y adenina. En estas máquinas, la computación se lleva a cabo mediante reacciones químicas. Así, por ejemplo, si el problema a resolver es el encontrar el camino óptimo entre dos ciudades en un mapa de carreteras, la ciudad de Madrid se podría codificar como *adenina-guanina-guanina-timina-adenina-citosina* y, de forma análoga, el resto de las ciudades. El hallazgo de la solución se realizaría introduciendo en un tubo de ensayo ADN que contuviera estas codificaciones así como las de los enlaces entre ciudades y sus distancias, y sometiénolo a reacciones químicas. Como podemos comprobar a través del ejemplo, cuando digo computación con ADN es que es con ADN, es decir, no se trata de una mera metáfora.

En el caso de la computación cuántica, los estados de la máquina son estados de las moléculas. Ahora mismo hay implementados prototipos con una cantidad de moléculas inferior a la decena. Sin embargo, el potencial futuro de este enfoque es impresionante. Con unos pocos centenares de moléculas podrían llegar a hacerse hasta 10^{75} operaciones simultáneas (Laita, 2006). Obsérvese que, dada la facilidad de aprendizaje que tendrían estas máquinas, es bastante probable que una culminación de este escenario llevara también a la culminación del escenario 1. Sin embargo, todavía no está descartada la posibilidad de que el aumento de partículas en el sistema para llevar a cabo los cálculos provoque en estos una ralentización de orden exponencial, de tal forma que lo que se gana debido al número de unidades de computación se pierda debido al tiempo que el conjunto tarda en responder (Alfonseca-Cubero et al., 2007).

5.3 Escenario futuro 3: conviven las máquinas de Turing y los hipercomputadores

La máquina de Turing lleva a cabo tareas realizables (si tuvieran suficiente tiempo y espacio) por los primeros computadores humanos, que eran los que conocía el londinense cuando inventó su máquina abstracta; pero, al menos teóricamente, puede haber otras máquinas que funcionen sin imitar a estas personas. De hecho se han elaborado modelos de tales máquinas (Copeland, 2002), conocidas como *hipercomputadores*.

Los hipercomputadores “resuelven” problemas que no resuelven las máquinas de Turing. Pongo “resuelven” entre comillas porque en realidad lo que quiero decir

es que los resuelven teóricamente, como la máquina propuesta por Urquhart resuelve el test de Turing, pero no se ha podido implementar hasta ahora una máquina material que lo haya conseguido. De hecho, hay investigadores (Davis, 2005) que son escépticos sobre la posibilidad real de sistemas materiales que no puedan ser emulados por una máquina de Turing y que éstos a su vez sean capaces de emular tales hipercomputadores.

6. Conclusion personal

En mi Socovos natal, hay noches en que todavía somos capaces de obligar a replegarse a la televisión, e incluso acaba apagada y fuera de combate durante unas horas, dejando lugar a historias de cementerios y aparecidos. Así, retomando una de las costumbres de nuestros abuelos, cada uno intenta que haya alguna víctima entre quienes le escuchan que recorra el camino que le lleva a su dormitorio asegurándose de que no apaga una luz hasta que no ha encendido la siguiente y, una vez haya llegado a su destino, se meta en la cama deprisa y corriendo antes de encontrarse con algún fantasma. Esta competición no declarada suele terminar cuando alguien, bastante desenvuelto dice:

“Yo no temo a los muertos. Me preocupa lo que me puedan hacer los vivos”

Pues bien, a mí no me dan miedo las máquinas, que simulan la vida, pero no la tienen. No me da miedo pensar que alguna vez las máquinas, por sí mismas, lleguen a dominarnos. Mi preocupación es lo que puedan hacer unas personas contra otras utilizando estas máquinas.

Centrándome en un caso, suponiendo que la Web Semántica llegara a tener una extensión aproximada, o mayor, a la que tiene la Web actual, la incógnita que queda por despejar es si llegaría a haber un grupo de gran tamaño de agentes artificiales inteligentes gratuitos al servicio de quien los quisiera utilizar o si, más bien, más allá del legítimo beneficio que ha de obtener la empresa, los servicios que aportararan valor añadido real serían de pago y, por tanto, servirían para aumentar la brecha entre quienes podemos permitirnos el lujo de jugar al ajedrez contra las computadoras y quienes no poseen ni los bienes materiales fundamentales.

Es decir, asumo que en el interior de las personas crecen juntos el trigo y la cizaña, y, si bien por una parte me ilusiona que continuemos la obra creadora de Dios, también me preocupa la cizaña que vaya a crecer y con cuánto vigor si alguna vez nos encontramos en alguno de los escenarios futuros que he planteado.

7. Referencias

- ADLEMAN, LM (1994) *Molecular Computation of Solutions to Combinatorial Problems*. Science 266:1021-1024. http://koehllab.genomecenter.ucdavis.edu/teaching/frs03/pdf_files/adelman_94.pdf
- ALFONSECA-CUBERO E, ALFONSECA-MORENO M, MORIYÓN-SALOMON R (2007) *Teoría de autómatas y lenguajes formales*. McGraw-Hill, Aravaca, Madrid
- BENJAMINS VR, CONTRERAS J, CORCHO O, GÓMEZ-PÉREZ A (2002) *Six Challenges for the Semantic Web*. En: Cristiani M (eds) Proceedings of the Semantic Web workshop held at Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR2002). Toulouse, Francia
- BERNERS-LEE T (1999) *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. HarperCollins Publishers, New York
- CAMPBELL M, HOANE AJ, HSU F-h (2002) *Deep Blue*. Artificial Intelligence 134:57-83
- COLBURN T (2004) Methodology of computer science. In: Floridi L (ed) *Philosophy of Computing and Information*. Blackwell Publishing, Massachussets, pp 318-326
- COPELAND BJ (2002) *Hypercomputation*. Minds and Machines 12:461-502. http://research.cs.queensu.ca/home/akl/cisc879/papers/PAPERS_FROM_MINDS_AND_MACHINES/VOLUME_12_NO_4/NV6361035557Q678.pdf
- COPELAND BJ (2004) Computation. In: Floridi L (ed) *Philosophy of Computing and Information*. Blackwell Publishing, Massachussets, pp 3-17
- CRAIK RJW (1943) *The nature of explanation*. Cambridge University Press, Cambridge, United Kingdom
- DAVIS M (2005) *Why there is no such discipline as hypercomputation*. Applied Mathematics and Computation 178:4-7
- EUZENAT J, Shvaiko P (2007) *Ontology matching*. Springer-Verlag, Berlin, Germany
- FERNÁNDEZ-SAAVEDRA R (2009) *La máquina de von Neumann*. Transparencias de la asignatura de Estructura de Computadores II, 2º de Ingeniería Informática, Universidad CEU San Pablo, Madrid

GÓMEZ-PÉREZ A, FERNÁNDEZ-LÓPEZ M, CORCHO O (2003) *Ontological Engineering*. Springer-Verlag, Londres

GRUBER TR (1993) *Toward principles for the design of ontologies used for knowledge sharing*. In: Guarino N, Poli R (eds) International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation. Padova, Italy. (Formal Ontology in Conceptual Analysis and Knowledge Representation) Kluwer Academic Publishers, Deventer, The Netherlands. <http://citeseer.nj.nec.com/gruber93toward.html>

GRUSKA J (1999) *Quantum Computing*. McGraw- Hill, Nueva York

HAUGHELAND J (1985) *Artificial intelligence: the very idea*. MIT Press, Cambridge, Massachusetts

HOPCROFT JE, Motwani R, Ullman J (2007) *Introducción a la teoría de autómatas, lenguajes y computación*, 3ª edn. Pearson Addison-Wesley

ISASI P, MARTÍNEZ P, BORRAJO D (1997) *Lenguajes, gramáticas y autómatas*. Addison Wesley

JONATHAN JBS, CHANDRASEKHAR A, SRINIVASAN T (2004) *Sentient Autonomous Vehicle using Advanced Neural net Technology*. En: Ge SS (ed) IEEE Conference on Cybernetics and Intelligent Systems. Singapur. IEEE Computer Science, Piscataway, New Jersey

LAITA LM (2006) *Algunas consideraciones acerca de la pregunta: ¿llegarán a ser algún día las máquinas más inteligentes que los seres humanos?* Clase magistral de inauguración del curso 2006-2007 en la Universidad San Pablo CEU. CEU Ediciones, Madrid, España

LAPACE PS (1814) *Essai philosophique sur les probabilités*. París

LIPTON RJ (1995) *DNA solution of hard computational problems*. Science 268:542-545

MAINZER K (2004) *System: an introduction to systems science*. In: Floridi L (ed) *Philosophy of Computing and Information*. Blackwell Publishing, Massachusetts, pp 28-39

Marías J (2001) *Historia de la filosofía*, 4ª edición, Filosofía y Pensamiento, Alianza Editorial, Madrid, Spain

MÁRQUEZ L (2009) *Diseño e implementación de un robot ajedrecista*. Proyecto Final de Carrera. Ingeniería Técnica Telemática. Escuela Politécnica Superior. Universidad CEU San Pablo, Madrid

MCCULLOGH WS, PITTS W (1943) *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics 5:115-133

MIRA J (2008) Aspectos conceptuales de la inteligencia artificial y la ingeniería del conocimiento. En: Palma Méndez JT, Marín Morales R (eds) *Inteligencia Artificial*, McGraw-Hill

NEWELL A (1982) *The Knowledge Level*. Artificial Intelligence 18(1):87-127

NEWELL A, SIMON HA (1976) *Computer science as empirical inquiry: symbols and search*. In: Haugeland J (ed) *Mind design: philosophy, psychology, artificial intelligence*. MIT Press, Cambridge, Massachusetts

NILSSON N (2000) *Inteligencia artificial. Una nueva síntesis*. McGraw-Hill, Madrid

OTERO A, FÉLIX P, BARRO S, PALACIOS F (2009) *Addressing the flaws of current critical alarms: a fuzzy constraint satisfaction approach*. Artificial Intelligence in Medicine 47(3):219-238

POMERLEAU DA (1992) *ALVINN: Autonomous Land Vehicle In a Neural Network*. PhD Thesis. Carnegie Mellon University

POWER DJ (2007) *A Brief History of Decision Support Systems*. DSSResources.COM, World Wide Web, <http://DSSResources.COM/history/dsshistory.html>, version 4.0, March 10, 2007

RÍOS-INSÚA S, BIELZA-LOZOYA C, MATEOS-CABALLERO A (2001) *Sistemas de ayuda a la decisión*. Ed. Ra-Ma, Madrid

RODRÍGUEZ-SOLANO C, SERRANO V, LAITA DE LA RICA LM, LAITA L, ROANES-LOZANO E (2005) *Construcción, basada en la Lógica y el Álgebra Computacionales, de un Sistema Experto para diagnóstico de la Depresión*. Boletín de la Sociedad Puig Adam de profesores de matemáticas 69:40-52

SCHAEFFER J (2004) *The games computers (and people play)*. Course on Heuristic Search. University of Alberta, Canada. <http://webdocs.cs.ualberta.ca/~jonathan/Courses/657/Notes/2.Games.pdf>

SMITH B (2004) Ontology. In: FLORIDI L (ed) *Philosophy of computing and information*. Blackwell Publishing, 155-166

SORZANO COS, RECARTE E, ALCORLO M, BILBAO-CASTRO JR, SAN-MARTÍN C, MARABINI R, CARAZO JM (2009) *Automatic particle selection from electron micrographs using machine learning techniques*. Journal of Structural Biology, 167:252-260. <http://biocomp.cnb.csic.es/~coss/Articulos/Sorzano2009e.pdf>

Mariano Fernández López nació en Socovos (Albacete) en 1971. Es Ingeniero en Informática, Máster en Ingeniería del Software, Máster en Ingeniería del Conocimiento y Doctor en Informática (con premio extraordinario de tesis) por la Universidad Politécnica de Madrid (UPM). Ha sido profesor de la Universidad Pontificia de Salamanca y de la UPM y, desde el año 2002, lo es de la Universidad CEU San Pablo, donde ha sido director del Departamento de Ingeniería del Software y del Conocimiento entre los años 2004 y 2008.

Su labor investigadora ha estado dirigida fundamentalmente a la reutilización y compartición de conocimientos en sistemas informáticos. Es co-autor del libro *Ontological Engineering*, publicado por Springer Verlag y asumido como texto de referencia en universidades como la de Jena, Edimburgo y Milán. También ha publicado en *AI-Magazine* (editada por la *American Association for Artificial Intelligence*) y otras revistas JCR. En concreto, su artículo en *IEEE-Intelligence Systems* está entre los 15 más referenciados de toda la historia de dicha revista. Ha sido miembro del equipo investigador en proyectos financiados por empresas y por organismos como la Comisión Europea o el Ministerio de Ciencia y Tecnología.