

# El argumento de la autonomía y el caso de los agentes adaptativos

José Martín Castro-Manzano

Universidad Nacional Autónoma de México (Instituto de Investigaciones Filosóficas)  
E-mail: [jmcmanzano@hotmail.com](mailto:jmcmanzano@hotmail.com)

---

**Resumen:** Hacemos una breve revisión del Argumento de la Autonomía a partir de un caso de estudio de agentes adaptativos junto con una serie de distinciones entre agentes y programas, y entre orígenes de la autonomía y ejercicio de la autonomía. El objetivo es doble: i) dilucidar el sentido de la autonomía en el contexto de la IA y, a partir de eso, ii) sugerir algunas ideas para repensar la autonomía tanto en los agentes artificiales como en la agencia racional en general.

**Palabras clave:** agentes, aprendizaje, autonomía

**Abstract:** We make a brief revision of the Argument from Autonomy given a case of study about adaptive agents and by making a series of distinctions between agents and programs, and between the origins and exercises of autonomy. The goal is twofold: i) to clarify the meaning of autonomy in the context of AI and, given that, ii) to suggest some ideas to rethink autonomy in artificial agents as well as in rational agency in general.

**Keywords:** agents, learning, autonomy

---

## 1. Introducción

Dos metas básicas de la Inteligencia Artificial (IA) son la comprensión y la reproducción de la conducta inteligente. Así, es usual decir que la IA es *prima facie* científica e ingenieril (Genesereth & Nilsson 1987). Científica, en tanto que se encarga del entendimiento de la inteligencia y de otros fenómenos cognitivos; ingenieril, en tanto que se ocupa de la construcción y desarrollo de sistemas inteligentes y tecnologías emergentes.

Bajo este enfoque la IA se define como el estudio de agentes inteligentes (Russell & Norvig 1995; Nilsson 2006), de tal suerte que la agencia se vuelve un objeto legítimo de la IA en su doble modo de estudio. Por tanto, el problema de la agencia es central en la IA. Y en consecuencia, la cuestión de la autonomía es esencial para la misma, pues los agentes se definen como sistemas autónomos (Wooldridge 2001; Wooldridge & Jennings 1995). A pesar de esto, la investigación reciente sobre la autonomía en agentes artificiales, con todos sus aspectos interesantes, no ha sido sistemática (Verhagen 2004).

Esta última situación puede ser, en parte, la causa de una ligera confusión acerca del objeto de estudio de la IA. Y esta confusión puede ser, hasta cierto grado, el origen de uno de los argumentos más comunes en contra de la posibilidad de los agentes artificiales inteligentes. Este argumento, simple pero poderoso, se basa en la idea de que los agentes naturales, y en especial los agentes humanos, tienen una autonomía natural —y por tanto real— que no puede ser simulada ni replicada; y por ende, los agentes artificiales inteligentes no son posibles, pues carecen de autonomía real.

Hay diferentes modos de representar este argumento y, sin embargo, a pesar de ser tan famoso y recurrente, no ha sido formulado explícitamente ni criticado en forma. Sus orígenes los podemos rastrear incluso antes de los inicios de la IA como disciplina formal. Ciertamente es posible poner nombre y fecha a la proposición que representa esta idea: el *dictum* de Lady Ada Lovelace que, básicamente, sostiene que las computadoras pueden hacer sólo lo que los programadores les han dicho que hagan y, por tanto, son

incapaces de tener creatividad (Russell & Norvig 1995). Entonces la primera inferencia es clara: si los agentes artificiales sólo pueden hacer lo que sus programadores les han dicho que hagan, carecen de autonomía.

Es común, no obstante, responder o tratar de vulnerar estas ideas con la objeción del aprendizaje automático —el exitoso aprendizaje automático, deberíamos decir—, pero como Pollock (1999) nos advierte, los agentes que aprenden no están mágicamente haciendo su propio programa, ya que están corriendo un programa de aprendizaje. Así pues, en general, estas ideas se pueden resumir en la usual objeción de que el ser inteligente es el programador, no el programa; y en términos de autonomía, el Argumento de la Autonomía es el que declara que el ser autónomo es el programador, no el programa.

Nuestra meta en este trabajo es doble: i) clarificar el sentido de la autonomía en la IA —especialmente partiendo del campo de los sistemas multi-agente (MAS)— y en la agencia racional en general. Para lograr nuestra meta revisaremos, con cierta profundidad, el Argumento de la Autonomía con la ayuda de una serie de distinciones extraídas de un marco más amplio de discusión y, en especial, con la revisión de un caso de estudio. Con todo esto pretendemos, no sólo divulgar, sino también esclarecer el significado de la autonomía en los agentes artificiales. El resto del trabajo está organizado del siguiente modo. En la Sección 2 exponemos el concepto de agencia en IA y, específicamente, bajo el paradigma de los sistemas multi-agente. En la Sección 3 describimos una serie de distinciones a partir de unas consideraciones sobre la ontología de los agentes. En la Sección 4 discutimos la noción de autonomía con cierta profundidad. En la Sección 5 revisamos un caso de estudio sobre agentes adaptativos. En la Sección 6 tratamos el Argumento de la Autonomía a partir de las distinciones realizadas en las secciones previas. Y finalmente, en la Sección 7, desplegamos nuestros comentarios y observaciones finales.

## 2. Del sentido de “agencia”

En un sentido muy amplio y general un agente es cualquier cosa capaz de percibir y actuar. Bajo esta definición intuitiva y corriente casi cualquier cosa puede clasificarse como un agente. Sin embargo, una definición más interesante y filosófica

puede encontrarse en el *nous poietikós* de Aristóteles, esto es, en esa inteligencia que ejecuta acciones (Aristóteles 1978). Esta definición no sólo cierra más la clasificación, sino que se muestra increíblemente moderna para su tiempo al introducir un elemento cognitivo que es fundamental para estas discusiones. Afortunadamente para nuestros fines, esta definición clásica tiene una traducción especializada en el campo de la IA, en especial en el área de MAS.

Así pues, un agente suele identificarse como un sistema computacional situado en un ambiente (Wooldridge y Jennings 1995, Wooldridge 2001):

Un agente es un sistema de computadora que está situado en algún ambiente, y que es capaz de actuar autónomamente sobre él para alcanzar sus objetivos.

Mientras que el ambiente, a su vez, se entiende como un entorno físico o virtual donde se enfatizan las propiedades más generales del agente: su percepción y su acción sobre él.

Esta definición especializada, si bien sigue siendo general, provee un nivel de abstracción que presenta las siguientes ventajas (Ferber 1995):

- 1) Permite observar las facultades cognitivas de los agentes al realizar sus acciones.
- 2) Permite considerar diferentes tipos de agente, incluyendo aquellos que no se supone tengan facultades cognitivas.
- 3) Permite considerar diferentes especificaciones sobre los subsistemas que componen los agentes.
- 4) Muestra que un agente no es tal sin un ambiente correspondiente.

El ambiente, por su parte, también puede caracterizarse. Brooks considera que el medio ambiente por antonomasia es el mundo real, que el mundo es el mejor modelo del mundo, por lo que un agente debe tener una implementación material, en este caso, robótica (Brooks 1999). No obstante, como Etzioni ha argumentado, no es necesario que los agentes tengan implementaciones robóticas dado que los ambientes virtuales, como los sistemas operativos y la web, son tan reales como el

mundo real (Etzioni 1993). A lo largo de este trabajo asumimos la postura de Etzioni, sin rechazar la de Brooks, haciendo énfasis en que lo importante es que la interacción del agente con su ambiente se da en los términos de la definición especializada: de forma autónoma. Russell y Norvig (1995) señalan que, independientemente de la postura anterior, los ambientes pueden clasificarse del siguiente modo:

- 1) Accesible o inaccesible. Si un agente puede percibir a través de sus sensores los estados completos del ambiente donde se encuentra, se dice que el ambiente es accesible. Esta propiedad depende no sólo del ambiente, sino de las capacidades de percepción del agente. Como se puede ver, mientras más accesible sea el ambiente, más sencillo será de construir.
- 2) Determinista o no-determinista. Si el siguiente estado del ambiente está determinado por la acción del agente, se dice que el ambiente es determinista. Si otros factores influyen en el próximo estado del ambiente, se dice que éste es no-determinista. El no-determinismo implica dos nociones importantes: i) que los agentes tienen un control parcial sobre el ambiente, y ii) que las acciones del agente pueden fallar.
- 3) Episódico o no-episódico. Si la experiencia del agente puede evaluarse a través de episodios o rondas, decimos que el ambiente es episódico. Las acciones se evalúan en cada episodio. Dada la persistencia temporal de los agentes, estos tienen que hacer continuamente decisiones locales que tienen consecuencias globales. Los episodios reducen el impacto de estas consecuencias, y por lo tanto es más sencillo construir agentes en ambientes episódicos.
- 4) Estático o dinámico. Si el ambiente puede cambiar mientras el agente se encuentra deliberando, se dice que el ambiente es dinámico; de otro modo es estático. Si el ambiente no cambia con el paso del tiempo, pero la evaluación de las acciones del agente si lo hace, se dice que el ambiente es semi-dinámico.
- 5) Discreto o continuo. Si hay un número limitado de posibles estados del ambiente, distintos y claramente definidos, se dice que el ambiente es discreto; de otro modo se dice que es continuo. Como también se puede notar, es más sencillo construir agentes en

ambientes discretos, porque las computadoras también son sistemas discretos.

Esta categorización sugiere que es posible encontrar diferentes clases de ambientes, diferentes "mundos". Cada ambiente, o clase de ambientes, requiere de alguna forma agentes diferentes para que estos tengan éxito. La clase más compleja de ambientes corresponde a aquellos que son inaccesibles, no-episódicos, dinámicos y continuos. Por ejemplo, es discutible concebir a un *daemon* de sistema operativo, como *xbiff*, como un agente. Pero tal sistema cumple con la definición de agente: se las arregla para identificar a su usuario, encontrar su buzón electrónico en la red (su ambiente), buscar mensajes nuevos y comunicar al usuario la presencia de éstos. El resultado es que podemos aproximar la definición de *xbiff* de una manera más comprensible para el usuario: el agente *xbiff*, un *daemon* del sistema X Windows situado en un ambiente UNIX, vigila constantemente el buzón de su usuario para avisarle cuándo llegan mensajes nuevos a través de una interfaz gráfica (Wooldridge 2001).

## 2.1. Aproximación formal al concepto de agencia

Tanto la definición intuitiva como la especializada tienen una afortunada formalización, de tal modo que el concepto de agente y ambiente quedan bien definidos en una arquitectura abstracta (Wooldridge 2001). En primer lugar introducimos un conjunto de estados:

**Definición 1** (Estados) *Un conjunto finito de estados discretos*  $E = \{e_0, \dots, e_n\}$ .

Según la clasificación de ambientes que vimos previamente, los ambientes no necesariamente han de ser discretos, pero en esta aproximación se asume que  $E$  es discreto. Posteriormente se define un conjunto de acciones:

**Definición 2** (Acciones) *Un conjunto de acciones que un agente ha de ejecutar*  $A = \{a_0, \dots, a_n\}$ .

Con esto definimos una corrida:

**Definición 3** (Corrida) *Dado un conjunto de estados*  $E = \{e_0, \dots, e_n\}$  *y un conjunto de acciones*  $A = \{a_0, \dots, a_n\}$ , *una*

corrida se define como una secuencia  $c$  tal que:

$$c = e_0 \vec{a}_0 e_1, \dots, \vec{a}_{n-1} e_n$$

donde  $e_k \vec{a}_m e_{k+1}$  expresa que el estado  $k$  se transforma en el estado siguiente mediante la acción  $a_m$ .

Diremos que  $C = \bigcup_{i=1}^n c_i$  es el conjunto de todas las corridas. Así  $C_{A_i}$  es el subconjunto de las corridas que terminan en una acción  $i$  y  $C_{E_n}$  el subconjunto  $C$  que termina en un  $n$  estado del ambiente. Y con esto podemos definir:

**Definición 4** (Acción en el ambiente) Dado el subconjunto de las corridas que terminan en una acción  $i$   $C_{A_i}$  y un conjunto de estados  $E = \{e_0, \dots, e_n\}$ , la acción en un ambiente es una función  $\tau$  que va de las corridas que terminan en una acción  $i$  a todos los estados posibles:  $\tau: C_{A_i} \rightarrow \wp(E)$

De este modo, el ambiente es sensible a su historia, por lo que las acciones ejecutadas por el agente en el pasado también afectan la transición a estados futuros. Y con esto estamos ya en condiciones de definir formalmente las definiciones intuitiva y especializada de ambiente y agente:

**Definición 5** (Ambiente) Un ambiente es una 3-tupla  $Env = \langle E, e_0, \tau \rangle$  donde  $E$  es un conjunto de estados,  $e_0$  es un estado inicial y  $\tau$  es una función de acción sobre el ambiente.

**Definición 6** (Agente) Un agente es una función  $ag: C_{E_n} \rightarrow A$ .

De este modo un:

**Definición 7** (Sistema agente) Es una tupla  $\langle Env, ag \rangle$ .

La aproximación formal a la agencia, por tanto, define un sistema agente como una tupla  $\langle Env, ag \rangle$  donde  $ag$  es un agente y  $Env$  el ambiente (Wooldridge 2001). Esta arquitectura abstracta muestra la relación entre el agente y el ambiente. La idea es

que no hay sistema agente sin ambiente. Más allá de toda duda, a nuestras mentes vienen otras referencias que, motivadas por reflexiones diferentes, alcanzan conclusiones muy similares: *yo soy yo y mi circunstancia* (Ortega y Gasset 1968).

El algoritmo 1 muestra la función que implementa un agente de este tipo. También es posible implementar un programa básico de ambiente que ilustre la relación entre éste y los agentes situados en él. El algoritmo 2 muestra el programa Ambiente. La historicidad del ambiente queda oculta en las percepciones del agente. La semántica de la función *busca* es clara; la función *percibir* mapea estados del ambiente a alguna entrada perceptual. Los agentes con formas robóticas pueden implementar esta función mediante el *hardware* (cámaras de vídeo, sensores infrarrojos, sonares, etc.). Los agentes *software* pueden implementar esta función usando comandos del sistema operativo, por ejemplo:

Algoritmo 1: *Agente basado en mapeo ideal*

```
function Agente-Mapeo-Ideal(p)
  percepciones ← percepciones ∪ p
  acción ← busca(percepciones, mapeo)
  return acción
end function
```

Algoritmo 2: *Ambiente*

```
procedure Ambiente(e, τ, ags, fin)   e : Estado inicial
repeat
  for all ag ∈ ags do               ags : Agentes
    p(ag) ← percibir(ag, e)
  end for
  for all ag ∈ ags do
    acción(ag) ← ag(p(ag))
  end for
e ← τ (ag ∈ ags acción(ag)) : Transición del ambiente
until fin(e)                       fin de corrida
end procedure
```

## 2.2. Atributos de los agentes

Bajo la anterior arquitectura un sistema agente, en sentido débil, tiene las siguientes propiedades:

- 1) *Autonomía*. Una vez activo, un agente opera en su ambiente sin intervención directa externa y tiene cierto control sobre sus acciones y su estado interno (Wooldridge 2001).

- 2) *Reactividad*. Un agente percibe su ambiente y responde a él (Wooldridge 2001).
- 3) *Proactividad*. Un agente exhibe una conducta orientada a metas (Wooldridge 2001).
- 4) *Habilidad social*. Un agente puede interactuar con otros sistemas a través de ciertos lenguajes (Genesereth & Kephthcel 1994).

El concepto de agencia fuerte, por otro lado, incluye las propiedades definidas para el concepto de agencia débil, más una serie de elementos que son usualmente aplicados o asociados con humanos. Así es común caracterizar a los agentes usando nociones o supuestos mentalistas como conocimiento, creencia, intención y obligación (McCarthy 1979; Shoham 1990). Más aún, algunos investigadores han llegado a considerar el desarrollo de agentes emocionales (Bates 1994). Finalmente, algunos autores sugieren otras propiedades que pueden o no incluirse en la especificación formal del agente:

- 1) *Movilidad*. Un agente tendrá la habilidad para moverse en su entorno (White 1994).
- 2) *Veracidad*. Un agente no comunicará información falsa (Galliers 1998).
- 3) *Benevolencia*. Un agente tratará de alcanzar sus metas (Rosenschein 1985).
- 4) *Racionalidad*. Un agente actuará de tal modo que alcance sus objetivos (Galliers 1998).

Una discusión sobre varios atributos de agencia aparece en (Goodwin 1993). Lo relevante de esta revisión de propiedades es que permite ver que el tema de la autonomía es esencial para la IA, pues los agentes son sistemas autónomos.

### 3. Ontología de un programa agente

Si ahora volvemos al Argumento de la Autonomía (AA) podremos ver que una de sus fortalezas viene, irónicamente, de una debilidad. Su aparente solidez proviene de la falta de claridad sobre una cuestión que es fundamental para esta discusión: el *status* ontológico de los programas computacionales y, consecuentemente, de los agentes.

Para describir este *status* seguimos una taxonomía ontológica particular (Eden & Turner 2006). Esta taxonomía comienza con dos grupos: tipos lingüísticos estáticos (*scripts*) y procesos dinámicos (*threads*). Los *scripts* se definen como entidades

atemporales que consisten de instrucciones bien formadas de una clase de máquinas digitales, comúnmente representados como archivos de texto. Por otro lado, los procesos son entidades temporales creadas por la ejecución de un *script* particular en un ambiente físico, en este caso, en un sistema operativo.

Por tanto, los programas son codificados como secuencias bien formadas de alguna signatura. Pero tomando en cuenta que los *scripts* dependen de la existencia de compiladores comerciales, la noción de *script* depende del lenguaje de programación. Así, siguiendo las ideas Eden & Turner (2006), mantenemos la noción de *Turing-completeness* de tal modo que un lenguaje de programación soporte un conjunto no-trivial de instrucciones. De este modo, esta propiedad garantiza una independencia ontológica de los programas. Así pues, un programa *script* se define como la categoría de entidades  $s_L$  tal que  $L$  es un lenguaje de programación *Turing-complete* y  $s$  es una fórmula bien formada de  $L$ .

Por otro lado, la noción de un programa como proceso corresponde a la noción de proceso como es reconocido por un sistema operativo. Por ejemplo, Ubuntu 10.4 y Windows 7 se refieren a sus programas-procesos, propiamente, como procesos. Cada proceso es una entidad temporal generada por la ejecución de un *script* codificado en un lenguaje. Cuando el proceso es generado, las instrucciones del *script* son copiadas y alojadas en un segmento particular de la memoria, donde el proceso espera por más instrucciones o datos. Estos procesos incluyen procesos simples —como mover el cursor en un monitor— hasta procesos complejos de mensajería, aritmética y lógica.

Usando este marco de categorías comenzamos a mostrar la primera distinción:

**Distinción 1** *Al hablar de agentes necesitamos distinguir entre el agente-script y el agente-proceso.*

Debería ser claro ahora que un agente-script no es igual a un agente-proceso, pues ambos tienen características ontológicas distintas. Sin embargo, aunque esta distinción es necesaria, no es suficiente, ya que no hemos dado definiciones relevantes para establecer la diferencia entre un programa regular y un agente. Así, la siguiente distinción a trabajar es aquella entre un programa sin más y un agente

pues, ciertamente, un agente es por definición un programa, pero para que un programa alcance el *status* de agente tiene que satisfacer ciertas condiciones.

La literatura sobre MAS está llena de estas condiciones, como hemos visto en la sección anterior. Los agentes, decíamos, son reactivos, proactivos, autónomos y tienen cierta habilidad social. Un sistema reactivo es aquel que percibe su ambiente y responde a él. Lo distintivo de un sistema reactivo es que no puede ser descrito o implementado por métodos del enfoque funcional, el cual considera a todos los programas como programas regulares, es decir, como funciones que van de un estado a otro sin más. Ciertamente, en un sentido los agentes son funciones que mapean percepciones a acciones sobre el ambiente, pero la diferencia estriba en que el enfoque funcional requiere que los programas se detengan dada cierta entrada, mientras un agente entra en un ciclo infinito que no se detiene y, por esta razón, se suele decir que los agentes son programas mal implementados.

La proactividad es la propiedad de un agente para tener una conducta orientada a metas en lugar de orientada a eventos. La habilidad social es la habilidad para interactuar con otros sistemas agentes mediante ciertos lenguajes. Y finalmente, la autonomía se define, usualmente, como la interacción del agente con su ambiente sin la intervención directa externa de otras entidades. Ciertamente esta es la definición problemática, por lo que la trataremos con más profundidad en la siguiente sección.

Para ilustrar estas condiciones consideremos un programa para el pago de nóminas. Tal programa no puede considerarse como un agente, en primer lugar, porque no está inmerso en un ambiente. No obstante, si somos generosos, podríamos decir que el ambiente del programa de nómina son sus entradas y salidas. Sin embargo, aún así, sus acciones no muestran influencia en su conducta posterior y, por tanto, no se puede tomar como un agente. Pero si somos más caritativos y permitimos que el programa sea un agente, tendríamos que rechazar esa suposición, pues el programa de nómina carece de habilidad social y continuidad temporal: corre una vez y no vuelve a correr hasta que se activa de nuevo.

Una clara taxonomía de estas propiedades o condiciones aparece en la Tabla 1.

Propiedad	Significado
Reactivo	Responde de manera rápida a los cambios en el ambiente
Autónomo	Ejerce control sobre sus propias acciones
Orientado a metas	Pro-activo, no meramente reactivo
Continuamente temporal	Es un proceso continuo
Comunicativo	Capaz de comunicarse con otros agentes
Aprendiz	Cambios adaptativos en su conducta dadas experiencias previas
Móvil	Capaz de transportarse a sí mismo
Flexible	Sus acciones no están escritas
Carácter	Personalidad creíble y estado emocional

**Tabla 1:**

Una taxonomía de propiedades de agentes.  
Adaptada de Franklin & Graesser (1996).

Es cierto que la falta de estas propiedades en un programa motiva a negarle el *status* de agente al mismo. No obstante, también es cierto que exigir todas estas propiedades a un agente, propiamente hablando, dejaría de lado a muchos programas agentes que tienen derecho a ser llamados así. Por tanto, requerimos un compromiso entre propiedades, un mínimo de propiedades. Y para determinar este mínimo número de propiedades esenciales podemos seguir la literatura usual: si un programa no cumple con, al menos, las propiedades de reactividad, proactividad, continuidad temporal, comunicación y autonomía, no puede considerarse un agente.

Así pues, podemos hacer las siguientes distinciones:

**Distinción 2** *Todo agente es un programa pero no todo programa es un agente* (Franklin y Graesser 1996).

**Distinción 3** *Si un programa no cumple con al menos las propiedades de reactividad, proactividad, continuidad temporal, comunicación y autonomía, no puede considerarse un agente.*

En efecto, la propiedad que aún sigue en juicio es la autonomía, sobre la cual pretendemos hablar ahora.

#### 4. Del sentido de “autonomía”

Usualmente la autonomía es entendida como autonomía moral. Quizás la mejor exposición de este tipo de autonomía es la kantiana, la cual sostiene que la auto-imposición de la ley moral universal es la autonomía (Kant 1993). Esta ley moral es creada por el agente mismo en lugar de ser impuesta por una entidad externa o fuerza superior. En el fondo, como puede apreciarse, el AA basa su crítica a los agentes artificiales partiendo de estas ideas.

Sin embargo, en el contexto de la IA el término *autonomía* no se usa para hablar autonomía moral sino, mejor dicho, de una autonomía que etiquetaremos como funcional: cómo es que el agente se las ve con su ambiente. Estas observaciones son relevantes porque la premisa principal del AA, la idea de que los agentes artificiales carecen de autonomía, se infiere de la proposición de que los agentes son programados y, por tanto, no se dan a sí mismos sus normas de funcionamiento, es decir, no crean su propia autonomía funcional. Así, es claro que aunque hay diferentes tipos de autonomía, el AA ataca únicamente a la autonomía funcional. De este modo alcanzamos la siguiente distinción, típica, pero importante:

**Distinción 4** *Hay diferentes tipos de autonomía.*

Después de hacer una breve revisión sobre este segundo tipo de autonomía hallamos que no hay un consenso absoluto sobre la naturaleza exacta de la misma. Así, por ejemplo, en el campo de la ingeniería y la robótica la autonomía se usa para hablar de la auto-suficiencia de una máquina para llevar a cabo ciertas tareas (Brooks 1991, Pfeifer 1996); en el área de la vida artificial se usa para referirse a la auto-organización de ciertos sistemas (Wheeler 1997, Nolfi y Floreano 2000); y en el campo de la biología se usa para hablar de la auto-constitución de un sistema vivo (Weber & Varela 2002).

Cierto, aunque no hay un acuerdo absoluto, es muy evidente que hay un lugar común: la autonomía parece referirse al funcionamiento de un sistema por sí mismo. Pero a pesar de esta idea común que refleja la naturaleza de la autonomía, el ataque del

AA sigue vigente en un nivel más bajo, por falta de una distinción más básica:

**Distinción 5** *El origen de la autonomía funcional es diferente del acto de ejercer la autonomía funcional.*

Los orígenes de la autonomía funcional y el ejercicio de la autonomía funcional son cosas diferentes. El AA carece de esta distinción pues, inicialmente, el ataque a la autonomía artificial tiene que hacerse al nivel del ejercicio y no al nivel del origen, del mismo modo en que la evaluación de una teoría científica tiene que hacerse al nivel del contexto de justificación y no al nivel del contexto de descubrimiento (Reichenbach 1938) (en efecto, estas categorías epistemológicas pueden estar fuera de moda y el contexto de descubrimiento puede dar luz sobre el contexto de justificación, pero en un sentido muy estricto, la valuación veritativa de una teoría tiene que hacerse al nivel del primer contexto, no del segundo).

Parece claro, entonces, usando el esquema ontológico, que la relación entre el *status* ontológico de un agente y los aspectos de la autonomía funcional puede visualizarse del siguiente modo (Tabla 2):

Autonomía	Agente artificial
Origen	Script
Ejercicio	Proceso

**Tabla 2:**  
Relación entre el *status* ontológico y la autonomía funcional.

Como decíamos, las posibles interacciones entre el origen de la autonomía y el ejercicio de la misma no carecen de valor, pero en un sentido muy estricto, el AA funde y confunde estas categorías. Inmediatamente vienen a la mente algunas de las comparaciones más obvias con los agentes humanos, pues el criterio por *default* del AA es el agente humano: ¿No son los agentes humanos programados también? Ciertamente, este sentido de programación no es el mismo que en un sentido computacional. Y tal vez la comparación no sea de mucha utilidad, pero nos servirá para ilustrar con más precisión la necesidad de hacer nuestras distinciones.

El programa agente es un programa computacional, pero su ejercicio consiste de un proceso, una serie de acciones durante el tiempo sin la intervención externa para la selección de acción. La otra cuestión es

clara: ¿Es el caso que los agentes humanos se dan a sí mismos sus normas funcionales como especie? Es verdad que como individuos los agentes humanos tienen acceso a ciertos estados que dependen de su propia experiencia, pero tales estados son posibles gracias a un funcionamiento que, por falta de una mejor palabra, podríamos llamar innato: unas normas de funcionamiento que los agentes humanos no se han dado a sí mismos. Podríamos ilustrar toda esta situación mediante la siguiente analogía (Tabla 3):

Autonomía	Agente artificial	Agente humano
Origen	Script	¿Código genético?
Ejercicio	Proceso	Proceso

**Tabla 3:**  
La analogía ontológica.

Sí, en efecto, el origen de la autonomía de los agentes artificiales está en su programas. Sí, el ejercicio de la autonomía del agente es posible gracias al programa-script, pero el origen no constituye el ejercicio. Cuando el programador realiza un agente, le da su ser, su forma y, en algunos casos, un conocimiento de base (generalmente ningún agente comienza como una *tabula rasa*). Si el programa únicamente utiliza este conocimiento sin prestar atención a sus percepciones, si no selecciona por sí mismo sus acciones, tenemos que decir que el programa no fue

inculcado con la posibilidad de autonomía. Sin embargo, cuando el agente funciona con su programa, su conocimiento de base y adquiere su propia experiencia, decimos que el agente es autónomo: se las ve con su ambiente sin la intervención de otro elemento externo como el programador.

Finalmente, siguiendo con la obligada comparación, dos comentarios extras para atar algunos cabos sueltos. Primero, sería demasiado exigente pedir que los agentes artificiales exhiban la misma autonomía que nosotros tenemos, pero la analogía aquí es relevante: nuestra autonomía tiene cientos de años de desarrollo. En segundo lugar, la objeción de la conducta exitosa: esta, contrariamente a como se podría pensar, no implica autonomía, ni viceversa. La conducta exitosa de un programa que realiza pagos de nómina no implica autonomía; inversamente, la acción autónoma no garantiza conducta exitosa: la mayoría de nosotros estaríamos dispuestos a considerarnos funcionalmente autónomos y, sin embargo, aceptamos nuestros errores.

Pues bien, en el contexto de la construcción de agentes artificiales es común recurrir a la descripción de los mismos a través de arquitecturas y especificaciones. Una de estas especificaciones se puede hacer mediante la técnica de PAGE's (Russell & Norvig 1995), la cual consiste en una descripción de percepciones (*percepts*), acciones (*actions*), metas (*goals*) y ambiente (*environment*) (Tabla 4).

Agente	Percepciones	Acciones	Metas	Ambiente
Sistema de diagnóstico médico	Síntomas, respuestas del paciente	Preguntas, pruebas, tratamientos	Pacientes exitosos, minimizar costos	Paciente, hospital
Sistema de análisis de imágenes digitales	Píxeles de diferentes colores e intensidad	Imprimir una categorización	Categorizar correctamente	Imágenes de un satélite
Controlador de una refinería	Lecturas de temperatura y presión	Abrir y cerrar válvulas, ajustar temperatura	Maximizar pureza y seguridad	Refinería
Tutor interactivo de Español	Palabras introducidas mediante teclado	Imprimir ejercicios, sugerencias y correcciones	Maximizar la puntuación del estudiante en la prueba	Conjunto de estudiantes

**Tabla 4:**  
Ejemplos de PAGE's. Adaptado de Russell & Norvig (1995).

Lo que pretendemos ahora es argumentar a favor de otra distinción, una idea simple, pero que no ha sido explícitamente formulada:

**Distinción 6** *La autonomía funcional como ejercicio no es bivalente, hay grados de autonomía.*

La suposición es simple: si el número de descripciones por PAGE's incrementa y el agente es capaz de satisfacer tales descripciones, entonces la autonomía funcional del agente incrementa. Podemos acercarnos a una formalización de esta idea sugiriendo una función  $g$  que mapea las posibles descripciones que satisface un agente,  $P$ , a una escala numérica:

$$g: \wp(P) \rightarrow [0,1]$$

Así, en los extremos tenemos los casos por defecto y el caso límite. El caso por defecto es cualquier agente con un grado de autonomía 0, por ejemplo, un apagador de luz (en sentido estricto, un agente con grado de autonomía 0 no es un agente, pero lo incluimos por mor del ejemplo). El caso límite tendría que ser un agente que se da a sí mismo, desde su origen y para su ejercicio, sus propias reglas: no se nos ocurre, por el momento, otro ejemplo más original que el caso del Motor Inmóvil, el Primer Motor aristotélico.

Resulta claro, entonces, que entre estos dos casos tenemos agentes cuya autonomía viene dada en función de su operación satisfactoria en una variedad de ambientes. Podemos visualizar esta situación del siguiente modo (Tabla 5):

Grado de autonomía	Ejemplo
1	Motor inmóvil
...	
0.6	Humanos
...	
0.2	Agentes artificiales
...	
0.1	Termostato
0	Apagador de luz, teclado de ordenador

**Tabla 5:**  
Ejemplos y grados de autonomía.

A pesar de todas estas distinciones, el argumento de la AA parece tener aún un punto a su favor: en efecto, el AA puede asumir que la autonomía como ejercicio es graduada, pero dicha autonomía viene dada por su origen: es el programador quien le da al agente la posibilidad de satisfacer diferentes PAGE's. Esta crítica es correcta, pero hay una observación extra que permite aceptar esta crítica y, sin embargo, aceptar que la autonomía como ejercicio puede originar nuevas formas de origen de tal suerte que no sólo el programador le da al agente dicha posibilidad. En la siguiente sección intentaremos seguir esta línea de argumentación.

## 5. Autonomía, aprendizaje y flexibilidad: agentes adaptativos

Las siguientes ideas que trataremos nos permitirán ver algunas relaciones entre autonomía, flexibilidad y aprendizaje para mostrar cómo un agente puede darse a sí mismo ciertas normas de funcionamiento no previstas —por los programadores, por supuesto— para satisfacer nuevos ambientes o situaciones.

Si volvemos por un momento a la taxonomía que sugieren Franklin y Graesser (1996), podemos ver que hay otras cualidades que, aunque no esenciales, son sumamente importantes para nuestra argumentación: aprendizaje y flexibilidad, respectivamente, cambios adaptativos en la conducta del agente dadas experiencias previas y capacidad para realizar acciones no escritas aún. En este sentido, es necesaria una forma de aprendizaje para generar flexibilidad. Y para mostrar esto recurrimos a un caso de estudio de agentes con estas propiedades: agentes adaptativos.

Hay estudios y experimentos (Guerra Hernández et al. 2007, 2008, 2010) diseñados para relacionar el aprendizaje con una forma de compromiso flexible (Rao 1991) como un caso de reconsideración basada en políticas (Bratman 1987). Los experimentos han sido implementado en *Jason* (Bordini et al. 2007).

En el experimento hay cuatro agentes en el mundo de los bloques: el agente *experimenter* propone a los otros agentes (*bold*, *learner* y *single-minded*) la tarea de poner un bloque  $b$  sobre un bloque  $c$  y, con cierta probabilidad, introduce ruido en el experimento al colocar un bloque  $z$  sobre  $c$  antes de su pedido. Posteriormente el agente *experimenter* recolecta la

información sobre la eficiencia de los otros tres agentes para cierto número de iteraciones.

Los otros agentes son, inicialmente audaces (*bold*), en el sentido de que los contextos de sus planes son vacíos. Esto es, todos ellos comparten el plan:

$$+!on(X,Y) \leftarrow put(X,Y)$$

El cual nos dice que si el agente adquiere la meta de colocar *X* sobre *Y*, entonces el agente realiza la acción de poner *X* sobre *Y*. Como puede verse, el agente audaz (*bold*) no puede aprender intencionalmente: es un agente por *default* del lenguaje AgentSpeak(L) que ha sido programado sin un módulo de aprendizaje.

El agente con aprendizaje intencional (*learner*), por otro lado, es capaz de aprender el contexto de sus planes. El agente flexible (*single-minded*), más aún, puede aprender tanto el contexto de los planes como razones de abandono. El agente con aprendizaje y el agente flexible pertenecen a una diferente clase de agentes debido a que usan acciones primitivas para lograr sus acciones de aprendizaje, y generan eventos especializados para usar su conocimiento generado por aprendizaje.

Tenemos entonces dos situaciones: la intención adoptada para colocar *b* sobre *c* falla o tiene éxito. El agente con aprendizaje intencional modifica el contexto de su plan del siguiente modo:

$$+!on(X,Y) : clear(Y) \leftarrow put(X,Y)$$

Así, el agente modifica su plan original con la regla que agrega una cláusula contextual: si el agente adquiere la intención de poner *X* sobre *Y* y es verdad que *Y* está libre, entonces el agente puede realizar la acción de poner *X* sobre *Y*.

Mientras tanto, el agente flexible, que es capaz de aprender razones de abandono, genera la regla:

$$abandon(on(X,Y)) :- intending(on(X,Y)) \& not clear(Y)$$

La cual, a manera de Prolog, nos dice que si el agente intenta poner *X* sobre *Y* pero *Y* no está libre, entonces tiene que abandonar la intención.

En el experimento el ruido puede aparecer antes, después o durante la adopción de la intención. Esto depende de la

organización de *Java* con respecto a los procesos de *Jason*. Aún cuando es posible controlar el momento donde el ruido aparece, es importante notar que los agentes en condiciones normales carecen de dicho control. Los resultados relevantes son los siguientes:

- 1) El agente audaz (*bold*), incapaz de aprender, siempre falla cuando hay ruido, ya que no puede aprender nada acerca de la adopción o abandono de las intenciones.
- 2) El agente con aprendizaje intencional (*learner*), por otro lado, reduce el número de fallos debidos al ruido, puesto que aprende cierto contexto: en este caso, poner *X* sobre *Y* requiere que *Y* esté libre; cuando esto no es el caso, el plan deja de ser aplicable.
- 3) El agente flexible (*single-minded*) reduce drásticamente el número de fallos al prevenir la adopción inconveniente de ciertos planes al abandonar intenciones cuando es necesario: cuando el agente adopta cierta intención pero el agente experimentador coloca un bloque *z* sobre *c*. En efecto, el agente flexible sólo falla cuando está listo para ejecutar su acción y el ruido aparece.

Para los casos exitosos, Guerra-Hernández y Ortiz-Hernández (2008) han considerado la conducta previsora del agente como un éxito. Así si el agente se rehúsa a adoptar cierto plan, se considera como un caso exitoso. Usualmente se espera que el agente tenga diferentes planes para cierto evento. El rechazo de un plan resultará en la generación de un plan diferente a ser adoptado para solventar el problema: un caso de verdadera reconsideración. Abandonar un plan también se considera como un caso de éxito: significa que el agente usó su reconsideración basada en políticas para prevenir un fallo real. En la práctica, esto resulta en la eliminación del evento asociado a la intención perteneciente a los eventos del agente.

Claramente, tanto el agente audaz como el agente con aprendizaje intencional no pueden abandonar intenciones. Por tanto, como se esperaba, el agente flexible es más exitoso que el agente con aprendizaje intencional cuando hay tazas altas de ruido; mientras que estos dos agentes son más exitosos que un agente audaz.

Se sabe que en ambientes dinámicos un agente muy cauto tiene más eficiencia que

uno audaz e, inversamente, en ambientes estáticos la audacia resulta más eficiente (Kinny y Georgeff 1991). Como muestra este caso de estudio, el aprendizaje es relevante ya que permite que los agentes se comporten “más autónomamente” en su ambiente; sin embargo, el grado de audacia o cautela es algo difícil de definir *a priori*. El mecanismo de aprendizaje permite que los agentes sean adaptativos a sus ambientes en lugar de ser rígidamente establecidos por sus programadores. Y esta capacidad de adaptación automática, a nivel del ejercicio, es una forma de autonomía funcional que permite expandir las posibilidades iniciales del agente dadas por su programa-script, pues este ejercicio permite que el programa-script se modifique sin la intervención externa del programador. Así, podemos agregar esta nueva distinción:

**Distinción 7** *El ejercicio autónomo del agente-proceso es capaz de modificar su origen como agente-script.*

Esta última distinción permite tener una regulación entre ambos aspectos de la autonomía funcional sin caer en la carencia de distinciones que hacen al AA un argumento incorrecto, como veremos más adelante. Así, con estas distinciones, la objeción de que estos agentes adaptativos no están mágicamente haciendo su propio programa, ya que están corriendo un programa de aprendizaje (Pollock 1999), pierde fuerza y se desvanece pues, en efecto, están corriendo un programa-script, pero aún así los agentes, como procesos, están ejerciendo una adaptación autónoma.

## 6. El argumento de la autonomía de nuevo

Usando estas distinciones y el caso de estudio estamos en una buena posición para resumir las objeciones al AA. Nuestra meta ahora es mostrar que el AA no es un argumento sólido. Así que empezamos por formularlo explícitamente:

- Premisa 1: Los agentes artificiales carecen de autonomía (porque fueron programados por programadores verdaderamente autónomos).
- Premisa 2: La autonomía es una condición necesaria para la inteligencia (porque la inteligencia depende de la adaptación autónoma al ambiente).

- Conclusión: Los agentes artificiales carecen de inteligencia.

Ciertamente no queremos justificar la negación de tal conclusión —que los agentes artificiales no carecen de inteligencia—, ni siquiera pretendemos pronunciarnos sobre la veracidad de dicha conclusión; y tampoco diremos nada sobre la segunda premisa. Lo que buscamos es más simple: señalar una serie de objeciones a la premisa 1 con el propósito de mostrar que el AA es un argumento incorrecto. Y al hacer esto comenzaremos a alcanzar nuestros objetivos principales: (i) clarificar el sentido de la autonomía en el contexto de la IA, (ii) sugerir algunas ideas para repensar la autonomía en la agencia racional en general.

Como debería ser claro ahora, usando estas distinciones el AA no sólo pierde fuerza, sino que también parece muy forzado. El AA hace su ataque a la autonomía de los agentes artificiales basado en una carencia de distinciones importantes, y por tanto es incorrecto al violarlas:

- *Violación a la distinción 1:* el AA no diferencia entre agente-script y agente-proceso, pues critica el ejercicio de la autonomía de los agentes artificiales basándose en que son programados. La evaluación de la autonomía de un agente tiene que hacerse por su condición presente y no por su origen. Por tanto, el ataque del AA es ilegítimo al cometer una falacia genética. Para que el ataque del AA fuera correcto tendría que hablar con criterios del origen para criticar el nivel del origen (Tabla 6).

Crítica del AA	Origen	Ejercicio
Origen	Válida	Falacia genética
Ejercicio	Falacia genética	Válida

**Tabla 6.** El ataque del AA.

- *Violación a las distinciones 2 y 3:* el AA trata el problema de la autonomía en los agentes artificiales haciendo una equivalencia entre cualquier programa regular y cualquier agente, pero un agente tiene un *status* ontológico diferente al de un programa regular.
- *Violación a la distinción 4:* el AA hace su ataque a la autonomía sin considerar los diferentes tipos de autonomía y trata el

- problema en bloque basándose en ideas de un solo tipo de autonomía.
- *Violación a la distinción 5:* dada la violación a la distinción 1, el AA viola esta distinción por la estrecha relación entre el *status* ontológico de un agente y los aspectos de la autonomía funcional.
  - *Violación a la distinción 6:* el AA pretende que la autonomía sea una propiedad tipo Hamlet: o se tiene o no se tiene; cuando, al contrario, la autonomía es una propiedad tipo Zadeh: admite grados.
  - *Violación a la distinción 7:* el AA explota indiscriminadamente las posibles interdependencias legítimas entre el origen y el ejercicio de la autonomía al violar también las distinciones 1 y 5.

La violación de estas distinciones muestra que el Argumento de la Autonomía está mal fundado. La idea de que los agentes artificiales carecen de autonomía porque los verdaderos seres autónomos son los programadores no es sólo parte de un argumento falaz, sino también una proposición falsa. Así, la autonomía funcional en el campo de la IA, y especialmente en MAS, es una autonomía funcional que puede ser definida como un ejercicio de conducta auto-suficiente. Y debería ser claro que el origen ontológico del agente no excluye la posibilidad de autonomía (el clásico contra-argumento de que es el programador el autónomo y no el programa agente se cae y en consecuencia parece que el *dictum* de Lovelace se vulnera). Por otro lado, la exposición de las distinciones nos ha permitido repensar el sentido de la autonomía no sólo en el contexto de los MAS, sino en la agencia en general.

## 7. Conclusiones

Revisamos el argumento de la autonomía a partir de una serie de distinciones que consideramos importantes. El objetivo ha sido doble: (i) dilucidar el sentido de la autonomía en el contexto de la IA y, a partir de eso, (ii) sugerir algunas ideas para repensar la autonomía tanto en agentes artificiales como en la agencia racional en general.

Algunas de estas ideas pueden resumirse en una forma generalizada de las distinciones y otras observaciones de tal modo que la agencia y la autonomía no sólo se piensen al interior de la IA:

- 1) La agencia es parte de un sistema que incluye al agente y su ambiente: no hay agentes sin ambientes.
- 2) En la agencia, tanto como en la autonomía, se tiene que distinguir entre la agencia como *script* y la agencia como proceso; e.d. se tiene que distinguir entre el origen y el ejercicio.
- 3) Así como un programa computacional no puede considerarse un agente si no cumple ciertas condiciones, cualquier otro sistema no puede considerarse como un agente si no cumple, al menos, la reactividad, proactividad, continuidad temporal, comunicación y autonomía.
- 4) Así como hay diferentes tipos de autonomía, también hay diferentes tipos de agencia que pueden apreciarse mediante taxonomías.
- 5) El origen de la autonomía funcional es diferente del acto de ejercer la autonomía funcional.
- 6) La autonomía funcional como ejercicio no es bivalente, hay grados de autonomía.

Estas ideas generales, simples como son, ofrecen un marco teórico para poder hablar de manera más ordenada y sistemática de la autonomía y de la agencia no sólo en la IA, sino también en agencia en general y, en consecuencia, se pueden evitar los problemas semánticos que hacen al AA un argumento incorrecto.

Antes de terminar nos gustaría atraer la atención sobre otro aspecto que podría seguir discutiéndose: recordemos que se argumenta que el ser inteligente es el programador, no el programa; y en términos de autonomía, el Argumento de la Autonomía nos dice que el ser autónomo es el programador, no el programa. Pero a lo largo del trabajo hemos mostrado que esta última idea es incorrecta. La consecuencia, entonces, no es difícil de conjeturar: la idea que el ser inteligente es el programador y no el programa parece también ser incorrecta bajo la luz de las distinciones previas: el origen ontológico de un agente no excluye la posibilidad de inteligencia real, dado que podemos distinguir entre el origen y el ejercicio de la inteligencia, y podemos reconocer que la inteligencia admite grados, tal y como lo hicimos con la autonomía.

Finalmente, un trabajo pendiente es aclarar cómo el proceso creativo de construir un agente ilustra aspectos acerca de la naturaleza de la autonomía en programas agentes y en la agencia en general.

## Referencias

- Aristóteles (1978) *Acerca del alma*. Madrid: Gredos.
- Bates, J. (1994) The role of emotion in believable agents. *Communications of the ACM*, 37(7), pp. 122-125.
- Bordini, R. H., Hübner, J. F. y Wooldridge, M. (2007) *Programming Multi-Agent Systems in AgentSpeak using Jason*. West Sussex (England): Wiley.
- Bratman, M. (1987) *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.
- Brooks, R. A. (1991) Intelligence without reason. In J. Myopoulos y R. Reiter (eds.), *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence*. (pp. 569-595) San Mateo, CA: Morgan Kaufmann.
- Brooks, R. A. (1999) *Cambrian Intelligence: the Early History of the New AI*. Cambridge, MA: The MIT Press.
- Eden, A. H., Turner, R. (2007) Problems in the Ontology of Computer Programs. *Applied Ontology*, 2(1), pp. 13-36. Amsterdam: IOS Press.
- Etzioni, O. (1993) Intelligence without robots. *AI Magazine*, 14(4).
- Ferber, J. (1995) *Les Systemes Multi-Agents: vers une intelligence collective*. Paris: InterEditions.
- Franklin, S. y Graesser, A. (1996) Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer.
- Galliers., J. R. (1998) *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict*. PhD thesis, Open University, UK.
- Genesereth, R. y Ketchpel., S. P. (1994) Software agents. *Communications of the ACM*, 37(7), pp. 48-53.
- Genesereth, R. y Nilsson, N. J. (1987) *Logical Foundations for Artificial Intelligence*. Palo Alto, CA: Morgan Kauffman Publishers, Inc.
- Goodwin., R. (1993) *Formalizing properties of agents*. Technical Report CMU-CS-93-159, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.
- Guerra-Hernández, A., González-Alarcón, C. A., Seghrouchni, A. E. F. (2010) Jason Induction of Logical Decision Trees: A Learning Library and Its Application to Commitment. In V. Sidorov y A. Hernández (eds.), *MICAI 2010, Part I*. Volume 6437 of Lecture Notes in Artificial Intelligence. (pp. 374 -385) Berlin & Heidelberg: Springer-Verlag.
- Guerra-Hernández, A. y Ortíz-Hernández, G. (2008) Toward BDI sapient agents: Learning intentionally. In Mayorga y R. V., Perlovsky, L. I. (eds.), *Toward Artificial Sapience: Principles and Methods for Wise Systems*. (pp. 77-91) London: Springer.
- Guerra-Hernández, A., Ortíz-Hernández, G. y Luna-Ramírez, W. A. (2007) Jason smiles: Incremental BDI MAS learning. In *MICAI 2007 Special Session*, IEEE, Los Alamitos.
- Kant, I. (1993) *Critique of Practical Reason*. Prentice-Hall.
- Kinny, D. y Georgef, M. (1991) Commitment and effectiveness of situated agents. In *Proceedings of the twelfth international joint conference on artificial intelligence (IJCAI-91)*, Sydney, Australia.
- Maturana, H. R. y Varela, F. J. (1980) *Autopoiesis and Cognition: The Realization of the Living*. Dordrecht (The Netherlands): Kluwer Academic Publishers.
- McCarthy. J. (1979) *Ascribing mental qualities to machines*. Technical report, Computer Science Department, Stanford University, Stanford, CA (USA).
- Nilsson, N. (2006) *Inteligencia artificial. Una nueva síntesis*. México: McGraw-Hill.

- Nolfi, S. y Floreano, D. (2000) *Evolutionary Robotics: The biology, intelligence, and technology of self-organizing machines*. Cambridge, MA: The MIT Press.
- Ortega y Gasset, J. (1094) *Meditaciones del Quijote*. Madrid: Cátedra.
- Pfeifer, R. (1996) Building Fungus Eaters: Design Principles of Autonomous Agents. In P. Maes et al. (eds.), *Proc. of the 4<sup>th</sup> Int. Conf. on the Simulation of Adaptive Behavior*. (pp. 3-12) Cambridge, MA: The MIT Press.
- Pollock, J. L. (1999) Planning agents. In A. Rao y M. Wooldridge (eds.), *Foundations of Rational Agency*. Dordrecht (The Netherlands): Kluwer Academic Publishers.
- Rao, A. S. y Georgeff. M. P. (1998) Modelling Rational Agents within a BDI-Architecture. In M. N. Huhns y M. P. Singh (eds.), *Readings in Agents* (pp. 317-328). San Mateo, CA: Morgan Kaufmann.
- Reichenbach, H. (1938) Meaning. In *ibid.*, *Experience and prediction*. Chicago: University of Chicago Press.
- Rosenschein, J. S. y Genesereth., M. R. (1985) Deals among rational agents. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pp. 91-99, Los Angeles, CA.
- Russell, S. J. y Norvig, P. (1995) *Artificial Intelligence, a modern approach*. New Jersey, USA: Prentice Hall.
- Shoham, Y. (1990) *Agent-oriented programming*. Technical Report STAN-CS-1335-90, Computer Science Department, Stanford University, Stanford, CA (USA).
- Verhagen, H. (2004) Autonomy and Reasoning for Natural and Artificial Agents. In M. Nickles, M. Rovatsos y G. Wei (eds.), *Agents and Computational Autonomy - Potential, Risks, and Solutions*. (p. 83-94) Lecture Notes in Computer Science 2969. Springer.
- Weber, A. y Varela, F. J. (2002) Life after Kant: Natural purposes and the autopoietic foundations of biological individuality. *Phenomenology and the Cognitive Sciences*, 1, pp. 97-125.
- Wheeler, M. (1997) Cognitions Coming Home: the Reunion of Life and Mind. In P. Husbands y I. Harvey (eds.), *Proc. of the 4th Euro. Conf. on Artificial Life*. (pp. 10-19) Cambridge, MA: The MIT Press.
- White., J. E. (1994) *Telescript technology: The foundation for the electronic marketplace*. White paper, General Magic, Inc., 2465 Latham Street, Mountain View, CA 94040.
- Wooldridge, M. (2001) *Introduction to Multiagent Systems*. San Francisco, CA: John Wiley and Sons, Ltd.
- Wooldridge, M. y Jennings, N. R. (1995) Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), pp. 115-152.