

Tema 4. Análisis de sistemas robotizados

movimiento en el espacio como la economía. Esto se debe a que los circuitos de retroalimentación utilizados son similares a los modelados en economía.

Pero veamos otro punto interesante y sobre el que trabajaremos: robots de Inteligencia Artificial (IA). La diferencia básica entre lo que puede definirse como robot de IA y un robot clásico normal es la capacidad del robot y su software para tomar decisiones, aprender y adaptarse a su entorno en base a los datos de sus sensores. Mientras la **robótica clásica** trabaja en base a comportamientos deterministas (para un conjunto de entradas, el robot produce la misma salida como una impresora 3D), la **IA aplicada al desarrollo de la robótica** permite a un robot hacer dos cosas que el robot estándar no puede: **tomar decisiones** y **aprender de la experiencia**.

En este curso no veremos la robótica en sentido amplio, sino este nuevo campo abierto por la IA. El mundo clásico de los controladores, lazos de sensores y las máquinas de estado proporcional integral derivado (PID) da paso a redes neuronales artificiales, sistemas expertos, algoritmos genéticos y planificadores de rutas de búsqueda. Conseguimos un robot que no solo reaccione a su entorno como una acción refleja, sino que tenga metas e intenciones, y que pueda aprender y adaptarse al entorno.

Entorno o environment

Nuestro primer paso sería adaptar sus capacidades al entorno y al problema que debe resolver.

El entorno es el lugar donde trabajará un robot. Se clasifican en 2 categorías: estructurados y no estructurados.

- Un **entorno estructurado**, como una línea de montaje o un banco de laboratorio, tiene todo en un espacio organizado. Se puede conocer donde está o va a estar todo objetivo del robot: sabemos qué cualidades tienen las cosas (color, tamaño,...),

Tema 4. Análisis de sistemas robotizados

dónde están colocadas en el espacio y qué forma tienen. Los robots de la línea de ensamblaje esperan que las piezas lleguen exactamente en la posición y orientación para ser agarradas y colocadas en su posición; en este caso el entorno se adapta al robot.

- Un **entorno no estructurado**, no va a tener un conjunto de objetos con las mismas cualidades, ni una colocación fija en ese espacio ni un número limitado de los mismos. Podrá tener condiciones fijas pero también (y más normalmente) variables del conjunto de objetos. Es un lugar donde el robot y el software tienen que adaptarse, no los objetos.

Control en tiempo real

Los sistemas de control automático son objetos o sistemas que, al recibir una señal de entrada, realizan alguna función de forma automática sin la intervención de las personas. En todo sistema de control podemos considerar una **señal de entrada** que actúa sobre el sistema y una **señal de salida** proporcionada por el sistema.

Los sistemas de control base permiten crear un sistema de control robusto, modular y flexible para robótica: dan **la base para controlar** el robot de manera fácil y consistente; la otra parte esencial es, el bucle o lazo, que proporciona **la base para dotar de autonomía** al robot.



Figura 21. Ejemplo de lazo abierto (túnel de lavado). Fuente: Gandhinathan y Joseph, 2019.

Tema 4. Análisis de sistemas robotizados

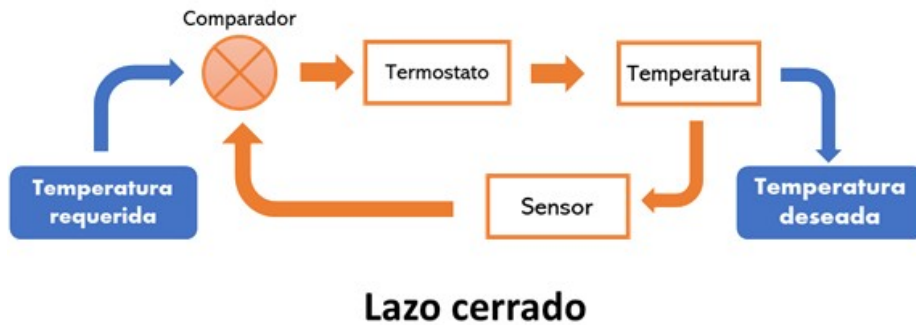


Figura 22. Ejemplo de lazo cerrado (aire acondicionado). Fuente: Gandhinathan y Joseph, 2019.

Un lazo cerrado es el **PID** que significa Proporcional, Integral, Derivado, y es un tipo de controlador que no requiere un modelo para operar. PID no es un método de IA porque no implica aprendizaje y no hay simulación de toma de decisiones. En estos casos podemos acudir a marcos o modelos para interconectar la IA con nuestro robot, en este caso veremos el lazo **OODA** (Observe-Orient-Decide-Act).

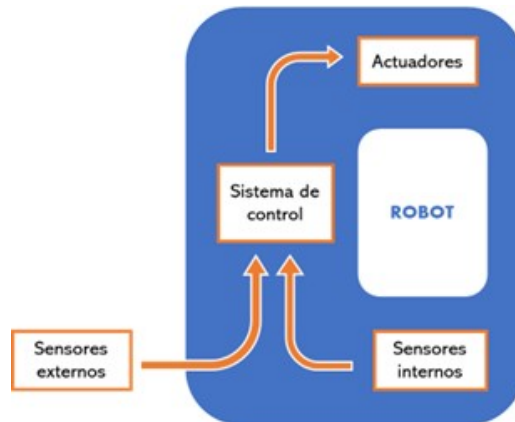


Figura 23. Esquema de las partes de un robot. Fuente: Gandhinathan y Joseph, 2019.

Como vemos, estos sistemas además tienen elementos mecánicos y sistema de control, que pueden definirse como:

- Los **sensores** son elementos que obtienen información para uso del robot. Esta información puede estar relacionada con el estado interno del robot o con sus

Tema 4. Análisis de sistemas robotizados

circunstancias externas (entorno) con lo que podemos tener diferentes tipos:

- Sensores internos.
 - Optointerruptores.
 - Codificadores.
 - Balizas y sistemas GPS.
 - Sensores externos.
- El **controlador** es circuito electrónico que pone en funcionamiento un actuador, en función del valor de entrada y del valor del sensor. Suele ser un microcontrolador o un microprocesador, cuya potencia depende del robot.
- El **actuador** es un elemento que lleva a cabo el proceso para el que se ha diseñado el sistema automático (calefacción, motor, luces, ...).

Pero si además consideramos el **tiempo real** los sistemas son más costosos y requieren un entrenamiento especial. En tiempo real no se puede interrumpir al proceso principal (control del robot) con la entrada de otros procesos de forma continua como nos sucede en nuestro ordenador o móvil (abrimos un programa y tarda tiempos diferentes según el momento). Este comportamiento no es factible en un sistema en tiempo real en el que necesitamos saber de antemano exactamente cuánto tardará un proceso en un microsegundo (básico en un piloto automático en aviación, ¿verdad?).

Si el sistema de control es **duro** en tiempo real, el hardware está imponiendo restricciones de tiempo y asegurándose de que los recursos de la computadora estén disponibles cuando se necesiten.

En el caso de **control suave**, el software realiza un seguimiento de la ejecución de la tarea, generalmente en forma de **frames**, que son intervalos de tiempo establecidos,

Tema 4. Análisis de sistemas robotizados

como los fotogramas de una película. Cada *frame* es un intervalo de fijo, por ejemplo 1/15 de segundo, y el programa divide sus tareas en partes que puede ejecutar en esos *frames*. Se trata de un tipo de sistema que brinda una base estable para que el robot funcione de manera constante y sin problemas.

El sistema **tiempo real suave** es más apropiado para robots no tan críticos como los que hacen recolección o limpieza, por ejemplo, mientras que un sistema de **tiempo real duro** es crítico para los robots que realizan tareas de seguridad.

Todos los controles que se agregan con este tipo de control corregirán las fluctuaciones en la sincronización del robot que son inevitables en tiempo real. Lo que nos permite hacer es realizar el ciclo de control de manera consistente todo el tiempo. Veremos cómo nos ayuda la programación de robots en las siguientes páginas.

Lazo OODA

Presentaremos un modelo para la toma de decisiones que proviene de la Fuerza Aérea de los Estados Unidos, llamado lazo OODA (Observar-Orientar-Decidir-Actuar) para entender qué aporta este proceso automatizado.

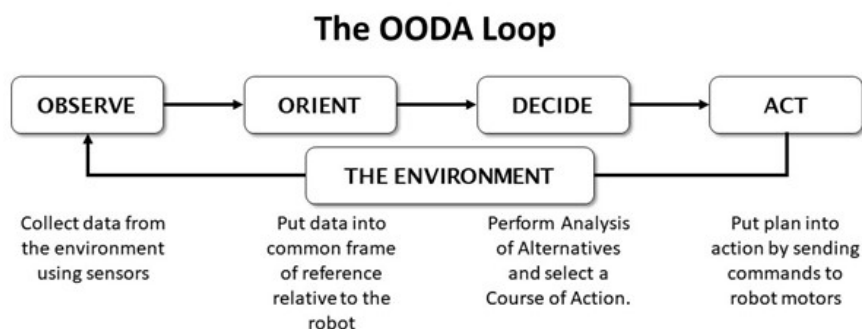


Figura 24. Definición de lazo OODA. Fuente: Govers, s.f.

El lazo OODA fue inventado por el coronel John Boyd, un hombre también llamado El padre del F-16. Las ideas del coronel Boyd todavía se citan ampliamente en la

Tema 4. Análisis de sistemas robotizados

actualidad, y su lazo OODA se utiliza para describir la inteligencia artificial de robots, la planificación militar o las estrategias de marketing con la misma utilidad. La OODA proporciona un modelo de cómo podría funcionar una máquina pensante que interactúa con su entorno.

En qué nos ayuda: permitiendo que haya robots que no funcionen simplemente ejecutando comandos o siguiendo instrucciones paso a paso, sino **estableciendo objetivos y luego trabajando para lograrlos**. Así, el robot tiene libertad para establecer su propio recorrido o determinar cómo llegar a su objetivo.

Por ejemplo, si queremos que recoja un determinado material en la superficie lunar, el robot decidirá cuál, cómo acercarse de la mejor manera y cómo recoger el juguete. Esto no nos valdría para que el robot trabaje como una mano teleoperada, ya que este enfoque trabaja con la programación de instrucciones individuales, como avanzar, mover a la derecha, extender el brazo, abrir la mano de forma independiente y sin darle a los robots información alguna de por qué hacer esos movimientos.

Más adelante veremos cómo hacer uso de lazos de control para el diseño de robots.

Tema 4. Análisis de sistemas robotizados

4.4. Programación de robots y aplicaciones

El nivel de programación de un robot difiere en función de su utilidad. En este campo el desarrollo de la Inteligencia Artificial en las últimas décadas ha marcado un antes y un después en la evolución de la robótica.

Es innegable que la unión de la Inteligencia Artificial y la robótica ha mejorado notablemente su capacidad. Gracias a los algoritmos con redes neuronales entrenados por medio del Machine Learning (Aprendizaje Automático) y el Deep Learning (Aprendizaje profundo), ahora son capaces de recabar y analizar una gran cantidad de información. Gracias al procesamiento de datos masivo por medio del Big Data, los robots son capaces de dar una respuesta revolucionando por completo la robótica.

Los componentes de software que rodean a un robot son **genéricos** como es el sistema operativo (desde Linux a otros propietarios) y lenguajes de programación de propósito general (Python, Java, C y otros), así como un **software específico** de programación de robots. Este software debe permitir la construcción de un robot con métodos y funciones que le permitan acceder a la información del sensor, controlar todos los componentes del robot y conectarse al robot.

Framework Robot Operating System (ROS)

Una de las herramientas más utilizadas es el framework **Robot Operating System (ROS)** que es un conjunto de librerías y herramientas a través de las cuales es posible reproducir diversos comportamientos robóticos y crear robots. Sus principales características que lo hacen muy atractivo son:

- ▶ El código desarrollado se puede adaptar a cualquier otro robot pues todos sus paquetes funcionan en una capa de abstracción de hardware.
- ▶ Permite trabajar en varias máquinas simultáneamente.

Tema 4. Análisis de sistemas robotizados

- Es de código abierto y de uso gratuito, además de integrable con muchas librerías de robótica de terceros.

Por ejemplo, podemos reutilizar en un robot que busca rocas en Marte o aguacates en una plantación, podemos usar una computadora para obtener la información de la cámara de la roca/aguacate y procesarla (es la roca buscada o está madura la fruta), otra máquina para lanzar el movimiento que comanda el robot, y finalmente el robot tomará la roca/aguacate. Al seguir este flujo de trabajo, las computadoras no realizarán demasiadas tareas computacionales y la ejecución resulta ser más fluida. En la figura podemos ver un ejemplo de aplicación del lazo OODA para este robot.

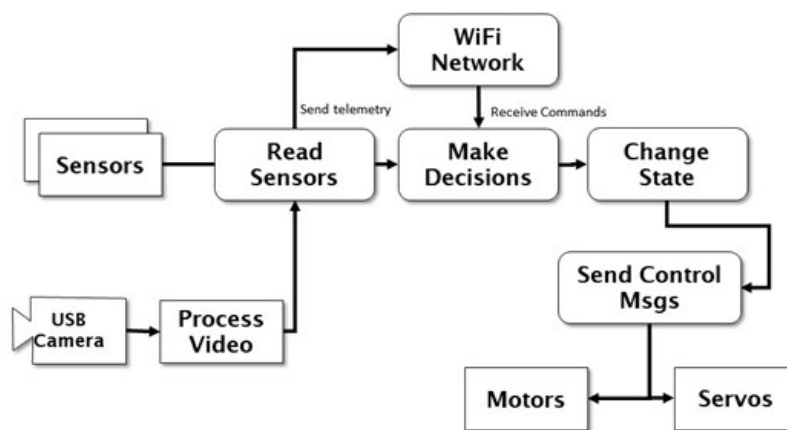


Figura 25. Un ejemplo de instanciación del lazo OODA en el SW y HW de un robot. Fuente: Govers, s.f.

ROS es la herramienta de robótica más utilizada, tanto por investigadores como por empresas. Se está convirtiendo en un estándar para las tareas de robótica. Además, ROS está en constante evolución para resolver nuevos problemas y se está adaptando a diferentes tecnologías. Todos estos hechos lo convierten en un buen tema para estudiar y practicar.

Tema 4. Análisis de sistemas robotizados

Conceptos de ROS

Al igual que cualquier otro software, se trata de aprender un lenguaje y cómo realizar ciertas tareas con él. Las ideas clave detrás de las funciones de ROS que ayudarán al alumno a comprender sus procesos internos son los siguientes:

- ▶ **Nodo o Node** : Un nodo ROS es un proceso encargado de realizar tareas y cálculos. Se pueden combinar entre sí utilizando *topics* u otras herramientas más complejas. Un robot puede tener muchos nodos para realizar sus cálculos. Por ejemplo, un robot móvil autónomo puede tener un nodo cada uno para la interfaz de hardware, lectura de escaneo láser y localización y mapeo. Para crearlos podemos utilizar librerías ROS.
- ▶ **Servidor de parámetros**: el servidor de parámetros es algo bastante útil en ROS. Un nodo puede almacenar una variable en el servidor de parámetros y también establecer su privacidad. Si el parámetro tiene un alcance global, todos los demás nodos pueden acceder a él. El parámetro ROS se ejecuta junto con el maestro ROS.
- ▶ **Tema o Topic**: Un *Topic* es un canal de información entre nodos que funcionan de forma unidireccional. Es un flujo de trabajo unidireccional porque los nodos pueden suscribirse a *Topic*, pero un *Topic* no sabría qué nodos están suscritos.
- ▶ **Master**: Es un servicio que proporciona un nombre y registro a los nodos restantes. Su función principal es habilitar nodos individuales para que puedan ubicarse entre sí y establecer una comunicación entre pares.
- ▶ **Servicios**: Es otro tipo de método de comunicación, similar a los *Topic*. Los *Topic* usan la interacción publicar o suscribirse, pero en los servicios, se usa un método de solicitud o respuesta. Un nodo actuará como proveedor de servicios, que tiene una rutina de servicio en ejecución, y un nodo cliente solicita un servicio del servidor. El servidor ejecutará la rutina de servicio y enviará el resultado al cliente. El nodo cliente debe esperar hasta que el servidor responda con los resultados.
- ▶ **Paqueteo Package**: los paquetes son el núcleo de la organización ROS. Dentro de

Tema 4. Análisis de sistemas robotizados

estos paquetes, puede encontrar nodos, librerías, conjuntos de datos o componentes útiles para construir una aplicación robótica.

- ▶ **Mensajes:** los nodos ROS pueden comunicarse entre sí de muchas formas. En todos los métodos, los nodos envían y reciben datos en forma de mensajes ROS. El mensaje ROS es una estructura de datos que utilizan los nodos ROS para intercambiar datos.
- ▶ **Pila o Stack:** una pila o stack ROS es un conjunto de nodos que, en conjunto, proporcionan alguna funcionalidad. Puede ser útil para dividir tareas entre nodos cuando la funcionalidad a desarrollar es demasiado compleja.
- ▶ **Bags:** Las bolsas son una utilidad útil en ROS para la grabación y reproducción de *Topics* ROS. Mientras trabajamos con robots, puede haber algunas situaciones en las que necesitemos trabajar sin hardware real.

Así, el concepto de gráfico ROS (nivel computacional ROS) constituye nodos, temas (topics), mensajes, maestro, servidor de parámetros, servicios y bolsas:

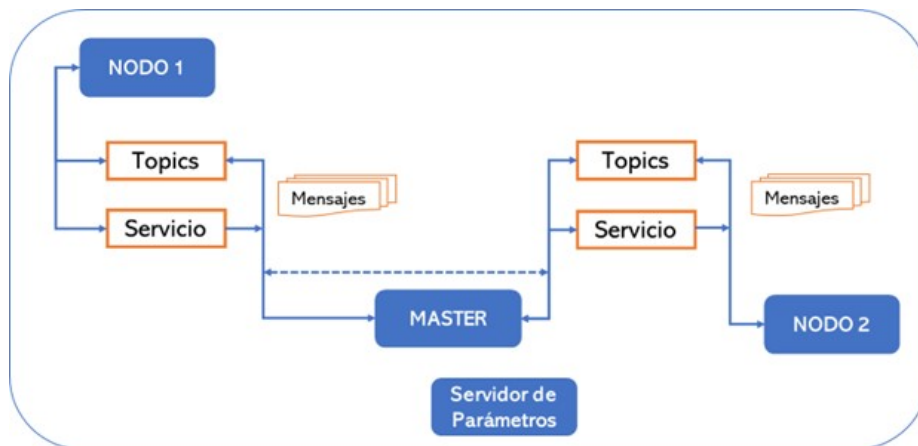


Figura 26. Diagrama conceptual del gráfico computacional ROS. Fuente: Gandhinathan y Joseph, 2019.

Tema 4. Análisis de sistemas robotizados

Su utilización en un ejemplo real podemos verla en la siguiente figura:

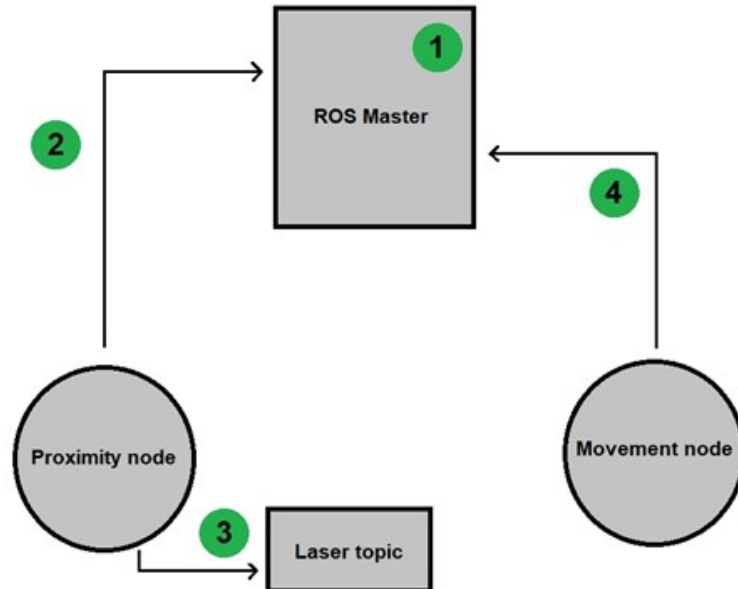


Figura 27. Esquema de un ejemplo real de sistema ROS. Fuente: Morena Alberola et al., 2019.

Este ejemplo presenta a un robot que cambia de dirección al detectar un obstáculo cercano. Los pasos serían los siguientes:

1. Se habilita el maestro ROS. Se traduce en que el sistema ROS se ha iniciado y puede ejecutar cualquier nodo.
2. El nodo de proximidad se inicia y extrae información del sensor láser. Avisa al maestro para que publique esta información obtenida. Si no hay ningún problema y la información tiene el tipo esperado, el maestro permitirá que el nodo publique a través de un *Topic*.
3. Una vez que el maestro permite que el nodo publique, la información se pasa a un *Topic* y se publica. En este caso, el nodo de proximidad publica la información en el tema láser.
4. El nodo de movimiento le pide al maestro que se suscriba a un *Topic* de láser. Una vez suscrito, obtendrá la información publicada y trabajará con ella para decidir la

Tema 4. Análisis de sistemas robotizados

siguiente acción que debe realizar el robot.

En resumen, ambos nodos pueden compartir información utilizando el servicio maestro, que notifica a ambos nodos sobre la existencia del otro.

Tema 4. Análisis de sistemas robotizados

4.5. Referencias bibliográficas

Archéologie. (s.f.). *Man Walks On The Lune*.
<https://archeologie.culture.gouv.fr/archeo-sous-marine/en/mediatheque/man-walks-lune>

Blog de Tecnología. (2017, mayo 2). *Programar con Scratch*.
<https://josepanadero.wordpress.com/2017/05/02/programar-con-scratch/>

Carfra. (2024, abril 15). *Robótica: una década de transformaciones*.
<https://carfra.com.ar/robotica-una-decada-de-transformaciones/>

Digital Trends. (2020, enero 9). *Loro, el pájaro que ayuda a las personas con discapacidad*. <https://es.digitaltrends.com/tendencias/loro-dispositivo-discapacitados-ces-2020/>

Gandhinathan, R. y Joseph, L. (2019). *ROS Robotics Projects: Build and control robots powered by the Robot Operating System, machine learning, and virtual realit*. Packt Publishing.

Govers, F. X. (s.f.). *Inteligencia artificial para robótica*.

Hanson Robotics. (s.f.). *Sophia*. <https://www.hansonrobotics.com/sophia/>

HM Sanchinarro. (s.f.). *Cirugía robótica*.
<https://www.hmsanchinarro.com/especialidades/programas-medicos/robot-da-vinci>

Interempresas. (s.f.). *Brazo robótico: de 6 ejes con un radio de funcionamiento de 1.300 mm*. <https://www.interempresas.net/Robotica/FeriaVirtual/Producto-Brazo-robotico-UR-10-78220.html>

iRobot. (s.f.). *Aspira, friega y se vacía solo, así de sencillo*. <https://www.irobot.es/>

Tema 4. Análisis de sistemas robotizados

La Razón. (2020, noviembre 20). *Revolución dron: del taxi aéreo a la agricultura 3.0*.

<https://www.larazon.es/medio-ambiente/20201120/k6p2j6evtrdszdw6hu5klchaiu.html>

Mobius Bionics. (s.f.). *LUKE Arm Details*. <https://mobiusbionics.com/luke-arm/>

Morena Alberola, A., Molina Gallego, G. y Garay Maestre, U. (2019). *Artificial Vision and Language Processing for Robotics*. Packt Publishing.

Omron. (s.f.). *Robots Móviles Autónomos (AMR)*.

<https://industrial.omron.es/es/products/autonomous-mobile-robot>

Revista de Robots. (2021, marzo 19). *EL ROBOT KIWIBOT TE LLEVA LA COMIDA A CASA*. <https://revistaderobots.com/robots-y-robotica/el-robot-kiwibot-te-lleva-la-comida-a-casa/>

Revista de Robots. (2023, julio 1). *Robótica. Qué es la robótica y para qué sirve*.

<https://revistaderobots.com/robots-y-robotica/que-es-la-robotica/?cn-reloaded=1>

Revista de Robots. (2024, mayo 2). *Robots industriales. Qué es un robot industrial, tipos y ejemplos de robots*.

<https://revistaderobots.com/robots-y-robotica/robots-industriales-y-robotica-industrial/>

Oreilly. (s.f.). *The washing machine*. [https://www.oreilly.com/library/view/learn-robotics-programming/9781789340747/816cc655-56a4-4fb6-91e6-](https://www.oreilly.com/library/view/learn-robotics-programming/9781789340747/816cc655-56a4-4fb6-91e6-318ea37d8ed3.xhtml)

[318ea37d8ed3.xhtml](https://www.oreilly.com/library/view/learn-robotics-programming/9781789340747/816cc655-56a4-4fb6-91e6-318ea37d8ed3.xhtml)

Jet Propulsion Laboratory. (s.f.). *PIA16239: High-Resolution Self-Portrait by Curiosity*

Rover Arm Camera. <https://photojournal.jpl.nasa.gov/catalog/PIA16239>

5.1. Introducción y objetivos

En este tema se explica a alto nivel el **concepto y la estructura general de los agentes inteligentes y los sistemas expertos**. Estos conceptos forman parte de los fundamentos básicos de la inteligencia artificial y, de una forma u otra, deben considerarse a la hora de diseñar una solución basada en esta disciplina.

De forma detallada, los objetivos que persigue esta unidad son:

- ▶ Ser capaz de definir las peculiaridades de un agente inteligente.
- ▶ Identificar las aplicaciones de los agentes inteligentes.
- ▶ Conocer los fundamentos de diseño y estructura de los agentes inteligentes.
- ▶ Conocer los conceptos clave de los sistemas expertos.
- ▶ Identificar qué proyectos son más adecuados para los sistemas expertos.

5.2. Introducción a los agentes inteligentes

En temas anteriores se ha discutido el significado de inteligencia y comportamiento racional. Todos estos contenidos previos deben tenerse en mente a la hora abordar el presente tema.

Un agente es un ente que recibe información del entorno y actúa en consecuencia. Se espera de él que actúe de forma inteligente.

Un ejemplo de actuación inteligente es la que está orientada a la **maximización de un objetivo** previamente establecido usando toda la información disponible en ese momento. Por ejemplo, si el objetivo es la supervivencia, una acción inteligente sería escapar u ocultarnos de un depredador.

De forma abstracta, un agente **es una función** que convierte las percepciones en acciones.

$f: P^* \rightarrow A$

P^* : indica un conjunto de percepciones finito.

A : indica la acción que llevamos a cabo en función de dichas percepciones.

Los agentes son muy útiles para modelar entornos complejos donde las acciones que lleven a cabo unos agentes puedan afectar a movimientos posteriores de otros entes. Se recibe información de un entorno y las propias acciones que los agentes llevan a cabo insertan nueva información en dicho entorno.

Tema 5. Sistemas Expertos

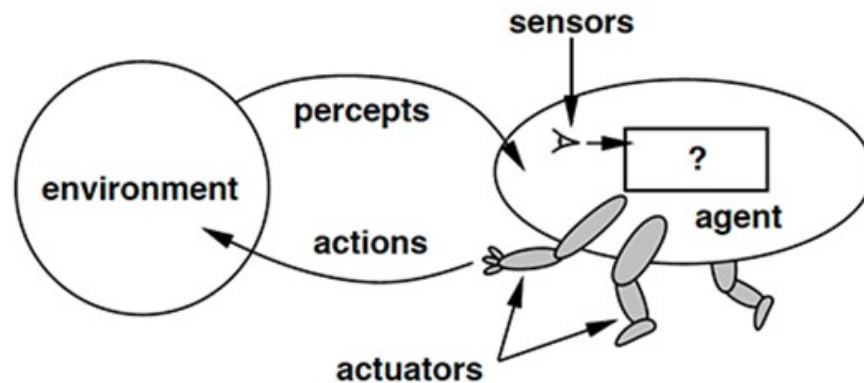


Figura 1. Un agente y su entorno. Fuente: Russel y Norving, 2004.

Ilustraremos estos conceptos mediante un sencillo ejemplo práctico. Supongamos que hemos creado un robot aspirador. Hemos hecho un gran trabajo y, por tanto, nuestro robot puede considerarse un buen ejemplo de agente inteligente. ¿Cómo actuaría este robot aspirador? Resumiendo, la función asociada al agente (que estaría ejecutándose de forma continua mientras el robot estuviese encendido) podría ser como la que sigue (Russell y Norving, 2004):

```
function Reflex-Vacuum-Agent([location,status]) returns an action  
  
if status = Dirty then return Suck  
  
else if location = Right then return Right  
  
else if location = Left then return Left
```

Explicado de forma sencilla, el robot dispone de un sensor que, dentro de un radio de acción, le indica que hay algún tipo de residuo o suciedad por recoger. Si así fuese, el mismo sensor le proporciona la localización concreta donde actuar. Si está a su derecha, el robot toma la decisión de moverse a la derecha. Una vez posicionado en el punto concreto, activa el aspirador. Tras ejecutar estas acciones, el entorno de partida ha cambiado, el residuo o resto de basura ha desaparecido y, por tanto, la información asociada a dicho entorno no es la misma que al inicio.

Tema 5. Sistemas Expertos

Como podemos esperar, sacar al mercado un robot aspirador implica crear un producto más completo del aquí reflejado. No obstante, la dinámica del procedimiento es común a todos los agentes inteligentes, todos tienen en cuenta la información que proporciona el entorno para, en función de una configuración previa, llevar a cabo una serie de acciones que permite maximizar el objetivo inicial planteado.

5.3. Comportamiento y entorno de los agentes inteligentes

Para diseñar un agente inteligente **debemos especificar de antemano el entorno en el que se va a ejecutar la tarea**. Esto implica definir cómo se va a evaluar si la tarea se está realizando correctamente, qué información se recoge del entorno, cuáles son las posibles acciones y qué tipo de sensores vamos a emplear. Todo ello constituye lo que se llama el **entorno de trabajo**, para cuya denominación se utiliza el acrónimo REAS (Rendimiento, Entorno, Actuadores/Acciones, Sensores).

Por ejemplo, supongamos que queremos diseñar ahora un coche autónomo. Podríamos trabajar con la siguiente descripción de elementos:

- ▶ **Indicadores de rendimiento:** número de kilómetros recorridos, número de infracciones de tráfico cometidas, número de accidentes, valoración media (o mediana) de la comodidad del viaje por parte de los ocupantes del vehículo, número de situaciones de riesgo detectadas, coste del sistema, etc.
- ▶ **Entorno:** calles de la ciudad, señales, otros vehículos, peatones...
- ▶ **Acciones posibles:** arrancar, parar, frenar, acelerar, girar, cambiar marcha...
- ▶ **Sensores:** velocímetros, frenos, GPS, cámaras de vídeo, etc.

A la hora de diseñar el agente se debe tener en especial consideración las **características del entorno**. Por ejemplo, no es lo mismo un entorno totalmente observable y una lógica de secuencias simple como puede ser la planteada para aspirar una habitación, que un entorno más estocástico e incluso con fenómenos no directamente observables como pueden ser los mercados financieros. En este último caso y en el ejemplo concreto del diseño de un agente inteligente para llevar a cabo operaciones bursátiles automatizadas, es probable que no se disponga del detalle de

Tema 5. Sistemas Expertos

todas las acciones llevadas a cabo por todos los agentes. Podríamos llegar a percibir las consecuencias de alguna acción concreta sin tener totalmente claro a quién asociar la acción raíz.

Propiedades de los entornos de trabajo

Los agentes inteligentes se emplean en entornos muy variados y diversos. Aun así, es posible identificar un número de dimensiones concretas en las que categorizar estos entornos. Estas dimensiones influyen de manera crítica en el diseño del agente. Las dimensiones que podemos considerar son las siguientes:

Totalmente observable vs. parcialmente observable

Diremos que el entorno de trabajo es totalmente observable si los sensores del agente le proporcionan información sobre todos los aspectos relevantes para la toma de decisiones. En otro caso, el entorno de trabajo es parcialmente observable. Un entorno puede ser parcialmente observable debido al ruido y a la existencia de sensores poco exactos o porque los sensores no reciben información de parte del sistema. Por ejemplo, en un sistema de inversión bursátil automatizado hay que tomar decisiones sin tener visión de qué están plateando o ejecutando otros actores en ese mismo momento.

Determinista vs. estocástico

Un entorno de trabajo es determinista cuando el siguiente estado del entorno está totalmente determinado por el estado actual y la acción ejecutada por el agente. En otro caso, hablamos de un entorno estocástico. Un entorno estocástico añade incertidumbre a la toma de decisiones. Por ejemplo, y en el caso de uso de un coche autónomo, el entorno es claramente estocástico, ya que en cualquier momento puede aparecer una avería, pinchazo o peatón inoportuno que altere el rumbo de los acontecimientos.

Secuencial vs. episódico

En un entorno secuencial las acciones realizadas en el corto plazo pueden afectar a largo plazo. Un ejemplo claro es el motor inteligente de un juego de ajedrez. Un mal movimiento puede dar al traste con una partida, aunque el jaque mate no llegue de manera inmediata. Por el contrario, en un entorno de trabajo episódico, las acciones a realizar en el siguiente episodio no dependen de las acciones que se realizaron en episodios previos. Por ejemplo, un robot ubicado en una cadena de montaje puede tener el objetivo de detectar piezas defectuosas y sacarlas del ciclo de fabricación. En este caso no se tiene en cuenta las piezas descartadas con anterioridad ni tampoco lo que se haga con esas piezas en el futuro. Los entornos episódicos son más simples puesto que no requieren el estudio de ciertas dependencias entre las acciones que se llevan a cabo.

Estático vs. dinámico

En un entorno de trabajo dinámico, el entorno puede cambiar mientras el agente está analizando la situación para decidir qué acción tomar. En otro caso, el entorno es estático. En el caso del coche autónomo, el entorno es dinámico puesto que todos los vehículos y peatones se están moviendo mientras el robot decide qué es lo próximo que debe hacer.

Discreto vs. continuo

Un entorno de trabajo discreto tiene un número finito de estados distintos, aunque este número sea muy elevado. Esta situación se da en juegos como el ajedrez, que poseen un elevadísimo número de combinaciones, pero aun así finito. Por el contrario, el recurrido caso del coche autónomo sería un ejemplo de entorno de trabajo continuo, puesto que la velocidad y ubicación de los coches y peatones tienen un rango de valores continuos a los que hay que unir parámetros varios como ángulos de giro, inclinación de la carretera, etc.

Tema 5. Sistemas Expertos

Agente individual vs. multiagente

Un entorno de trabajo donde un único agente trata de resolver un determinado problema sin interactuar con otros entes inteligentes sería un ejemplo de entorno de trabajo con agente individual. Por el contrario, entornos de trabajo como el ajedrez, donde dos agentes compiten entre sí o como el caso de uso del coche autónomo, donde varios agentes inteligentes tienen que convivir, serían ejemplos de entorno de trabajo multiagente.

Encontraremos el caso más complejo ante un entorno de trabajo que sea parcialmente observable, estocástico, secuencial, dinámico, continuo y multiagente. Realmente, este caso extremo suele ser uno de los más frecuentes.

La tabla siguiente muestra la clasificación de algunos ejemplos de uso según la tipología del entorno de trabajo (Russell y Norving, 2004). Todos los casos se refieren a un agente inteligente realizando la labor referenciada, ya puede ser jugar a un determinado juego, conducir un taxi (robotizado) o enseñar inglés.

Tema 5. Sistemas Expertos

Entorno de trabajo	Observable	Determinista/ estocástico	Secuencial/ episódico	Estático/ dinámico	Discreto/ continuo	Individual/ multiagente
Crucigrama	Totalmente	Determinista	Secuencial	Estático	Discreto	Individual
Ajedrez	Totalmente	Determinista	Secuencial	Semi	Discreto	Multi
Póker	Parcialmente	Determinista	Secuencial	Estático	Discreto	Multi
Backgammon	Totalmente	Estocástico	Secuencial	Estático	Discreto	Multi
Taxi autónomo	Parcialmente	Estocástico	Secuencial	Dinámico	Continuo	Multi
Diagnóstico médico	Parcialmente	Estocástico	Secuencial	Dinámico	Continuo	Individual
Análisis de imagen	Totalmente	Determinista	Episódico	Semi	Continuo	Individual
Robot clasificador	Parcialmente	Estocástico	Episódico	Dinámico	Continuo	Individual
Tutor interactivo de inglés	Parcialmente	Estocástico	Secuencial	Dinámico	Discreto	Multi

Tabla 1. Ejemplos de entorno de trabajo y sus características. Fuente: basado en Russell y Norving, 2004, p. 50.

5.4. Estructura de los agentes inteligentes

Un agente se compone de **arquitectura más software**. Por arquitectura entendemos el conjunto de sensores físicos, unidades de procesamiento y todo tipo de dispositivo que permite al agente comunicarse con el entorno y desencadenar acciones. El *software* recoge las instrucciones que guían el comportamiento del agente.

Podemos generalizar los distintos tipos de agentes inteligentes en cuatro grandes categorías de complejidad creciente. Esto significa que agentes categorizados en capas de mayor complejidad podrían ser capaces de realizar tipologías de tareas asociadas a agentes más simples, pero no al revés.

En primer lugar, encontramos a los **agentes reactivos simples**. Estos agentes reaccionan ante un estímulo o información de entorno externo y actúan en función de un conjunto de reglas o instrucciones previamente estipuladas. Toda la información previa o histórica es descartada. Por ejemplo, supongamos que los sensores de un robot aspirador informan de que justo en la ubicación actual hay unas migas de pan. El robot pasaría a succionar la suciedad sin plantearse nada más.

Tema 5. Sistemas Expertos

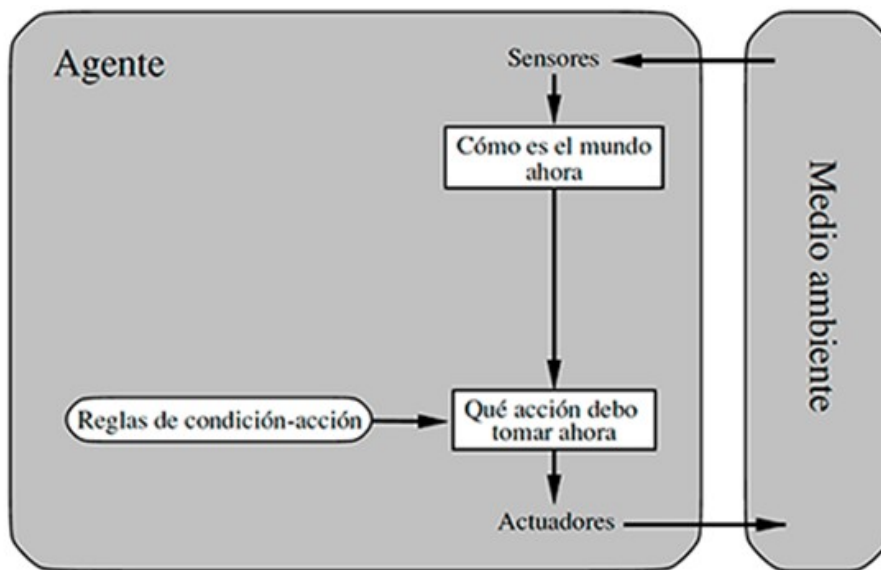


Figura 2. Agente reactivo simple. Fuente: Russel y Norving, 2004, p. 54.

Estos agentes se encuentran bastante limitados, puesto que solo se puede tomar la decisión correcta si el entorno es totalmente observable.

Una evolución más completa de estos agentes son los **agentes reactivos basados en modelos**. Estos agentes incorporan el histórico de información a la toma de decisiones. La historia previa se refleja en un determinado estado interno del agente, por lo que ya no solo se considera la información proveniente del entorno, sino que además debo tener en cuenta el estado en el que se encuentra el agente. Por ejemplo, imaginemos un videojuego de acción donde en un momento dado están disparando a nuestro personaje. El nivel de salud del personaje podría bajar o, por el contrario, podría mantenerse estático si nos encontramos en un estado «protegido» porque previamente hemos capturado algún tipo de objeto que ha creado un escudo protector. El estado del juego sería capaz de representar este amplio abanico de situaciones y entender cómo funciona su mundo particular. Y para entender su mundo necesita representar dicho conocimiento de manera formal, ya sea mediante un sistema de reglas, transacciones de estados, circuitos *booleanos* o ecuaciones matemáticas. Esa representación es lo que conocemos como modelo. Un agente que

Tema 5. Sistemas Expertos

utilice este tipo de modelos es un agente basado en modelos.

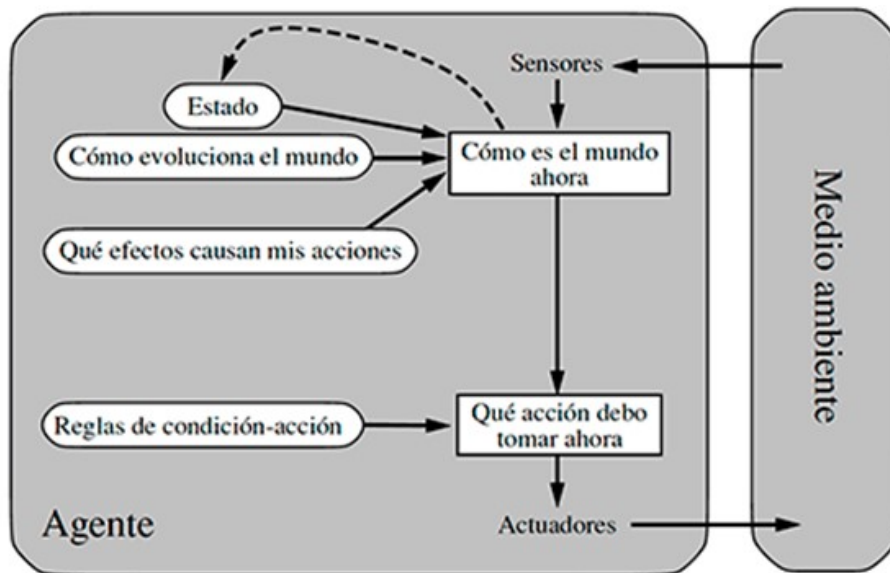


Figura 3. Agente reactivo basado en modelos. Fuente: Russel y Norving, 2004, p. 56.

Los **agentes inteligentes como los anteriores hacen que sea posible determinar de antemano un conjunto de reglas de condición-acción**. Por ejemplo, si los sensores del vehículo autónomo detectan que el coche de delante frena, se debe iniciar la frenada. Si los sensores del robot aspirador detectan suciedad a la derecha, el robot debe desplazarse a la derecha.

En muchas ocasiones no se dispone del conocimiento necesario para poder construir este conjunto de reglas, por lo que se establece un objetivo a conseguir. Un objetivo o meta podría ser, por ejemplo, llegar a un destino concreto. Hablaremos en este caso de **agentes basados en objetivos**.

Tema 5. Sistemas Expertos

Agentes basados en objetivos y función de utilidad

La Figura 4 muestra la arquitectura general de un agente basado en objetivos.

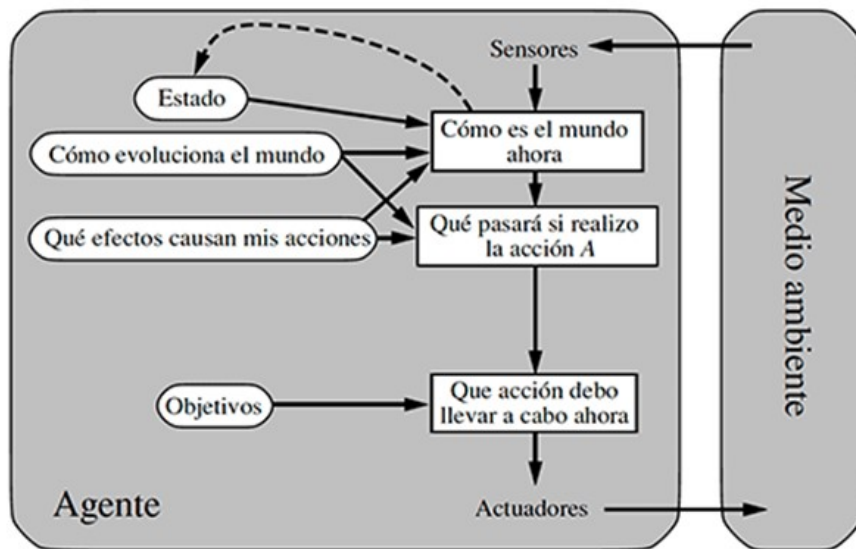


Figura 4. Agente basado en objetivos. Fuente: Russel y Norving, 2004, p. 57.

Este tipo de agentes son más flexibles, puesto que el objetivo que les guía obliga a representar explícitamente el conocimiento que soporta su decisión. Pero también son más complejos (dependiendo del objetivo planteado). Por ejemplo, el objetivo puede llegar con el coche a una determinada ubicación. Este tipo de tareas obliga a realizar una búsqueda de las distintas soluciones posibles y planificar un itinerario.

Los **agentes basados en objetivos** no permiten satisfacer nuestras necesidades en un gran número de ocasiones. Podemos pedir al navegador del GPS de nuestro vehículo que nos lleve a una ciudad concreta, pero nos sentiremos tremendamente defraudados y enfadados si nos damos cuenta de que nos está haciendo dar un rodeo innecesario o nos lleva por caminos poco transitables. Nos gustaría considerar no solo la meta a conseguir sino también en qué condiciones se debe alcanzar dicha meta para que consideremos plenamente satisfactoria la solución planteada por el algoritmo. Factores como llegar en el mínimo tiempo posible, recoger la mínima

Tema 5. Sistemas Expertos

distancia posible, evitar peajes, etc., nos aporta valor sobre la utilidad de la solución.

Los **agentes basados en utilidad** incorporan una función de utilidad al modelo de tal forma que es posible medir la satisfacción de los usuarios con la solución. Esta función de utilidad suele contemplarse como una función matemática a maximizar o minimizar. Por ejemplo, y como se ha comentado antes, se puede minimizar la distancia recorrida, minimizar el gasto en combustible, minimizar el tiempo invertido. Debemos notar que, matemáticamente hablando, un problema de maximización es similar a un problema de minimización, puesto que el $\max\{f(x)\}$ es igual al $\min\{f(x)\}$.

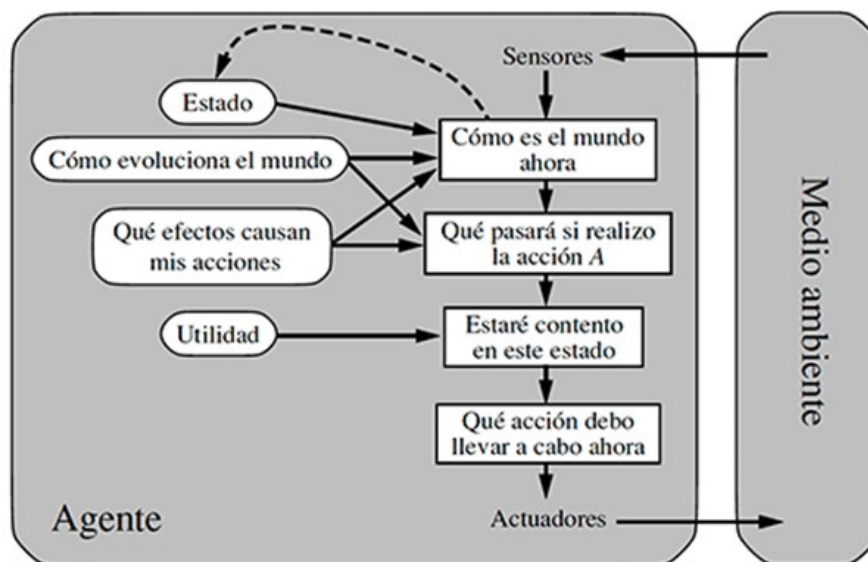


Figura 5. Agente basado en función de utilidad. Fuente: Russel y Norving, 2004, p. 59.

Se considera que un agente que posea una función de utilidad explícita puede tomar decisiones racionales con la ayuda de un algoritmo de propósito general que no dependa de la función específica de utilidad a maximizar o minimizar.

Con un **algoritmo de propósito general**, el tiempo de ejecución depende solamente del volumen o tamaño de los datos de entrada. Por el contrario, en un **algoritmo de propósito específico** el tiempo de ejecución depende de las propiedades asociadas

Tema 5. Sistemas Expertos

a los parámetros de entrada y no tiene por qué estar directamente relacionado con el tamaño de los datos de entrada.

Agentes que aprenden

Los agentes explicados hasta ahora reciben «su conocimiento» del entorno a través de sus programadores. Son estos quienes introducen las reglas correspondientes, los objetivos a conseguir o la función de utilidad. Un enfoque diferente es enseñar a las máquinas a aprender por sí mismas las características de la tarea a abordar. De esta forma, la propia máquina podría ser capaz de descubrir nuevos enfoques y mecanismos eficientes para llevar a cabo la tarea asignada. Este tipo de agentes da pie a una nueva categoría, los **agentes que aprenden**.

Un agente que aprende se puede dividir en **cuatro** componentes conceptuales tal y como muestra la Figura 6 (Russell y Norving, 2004).

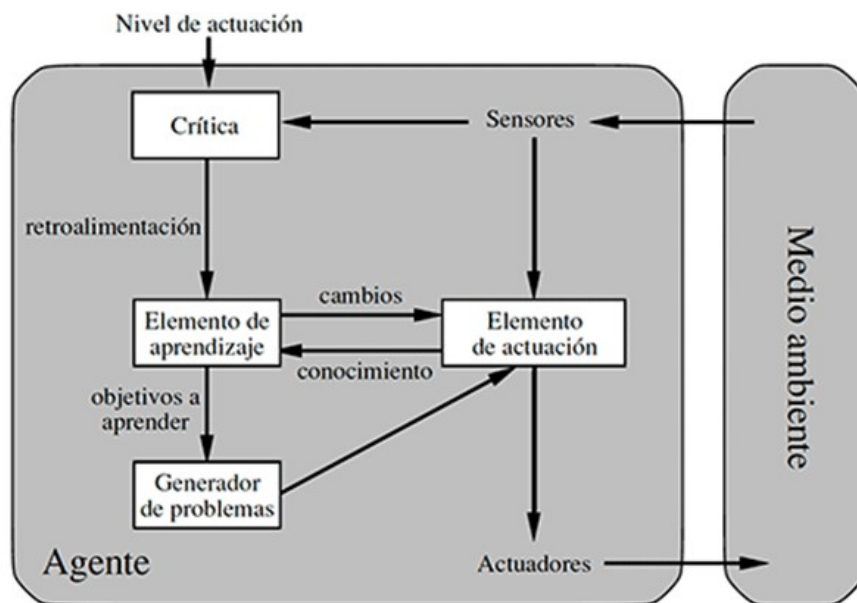


Figura 6. Estructura de un agente que aprende. Fuente: Russel y Norving, 2004, p. 60.

La actuación del agente genera críticas, entendidas estas como evaluaciones sobre la actividad del agente inteligente. Los propios sensores del agente se convierten en

Tema 5. Sistemas Expertos

receptores de las críticas. Debemos entender estas críticas como una retroalimentación muy valiosa para evolucionar y mejorar el funcionamiento del agente.

La retroalimentación recibida alimenta el elemento de aprendizaje, que es el motor encargado de desarrollar mejoras. Los cambios realizados son compartidos con el elemento de actuación. Este último componente es el responsable de seleccionar las acciones a ejecutar considerando los estímulos recibidos a través de los sensores.

Por último, el **generador de problemas** tiene la misión de sugerir acciones exploratorias (similares a un experimento) que, aunque puedan ser consideradas como no óptimas a corto plazo, son útiles para llegar a soluciones mejores que las actuales a largo plazo.

La arquitectura reflejada en la última imagen debe entenderse como una arquitectura de alto nivel donde solo se refleja la composición e interacción entre los elementos principales del esquema. De hecho, como **elemento de actuación** debe entenderse una representación simplificada de lo que anteriormente se había considerado como el agente completo ya que recibe estímulos y determina las acciones a realizar. Una posible implementación de este tipo de agentes es usando **aprendizaje por refuerzo**.

Para acabar de entender el último tipo de agente explicado podemos acudir al manual *Inteligencia Artificial: un enfoque moderno* (Russell y Norving, 2004). Los autores nos plantean el caso de un taxi automatizado, sin conductor. Podemos suponer que el usuario entra en el taxi e introduce la ubicación de destino. Aquí, el elemento de actuación agruparía todos los conocimientos y procedimientos que tiene el taxi para seleccionar las acciones a ejecutar en base a los requerimientos del conductor. La parte crítica, gracias a la observación del entorno, proporciona al elemento de aprendizaje la información necesaria para aprender y evolucionar su comportamiento. Un ejemplo puede ser un cambio de carril que ha provocado que un