

Ingeniero en Software y Tecnologías Emergentes

Inteligencia Artificial

Tipos de Agentes Inteligentes

ISyTE

Dr. Héctor Zatarain Aceves

hector.zatarain@uabc.edu.mx

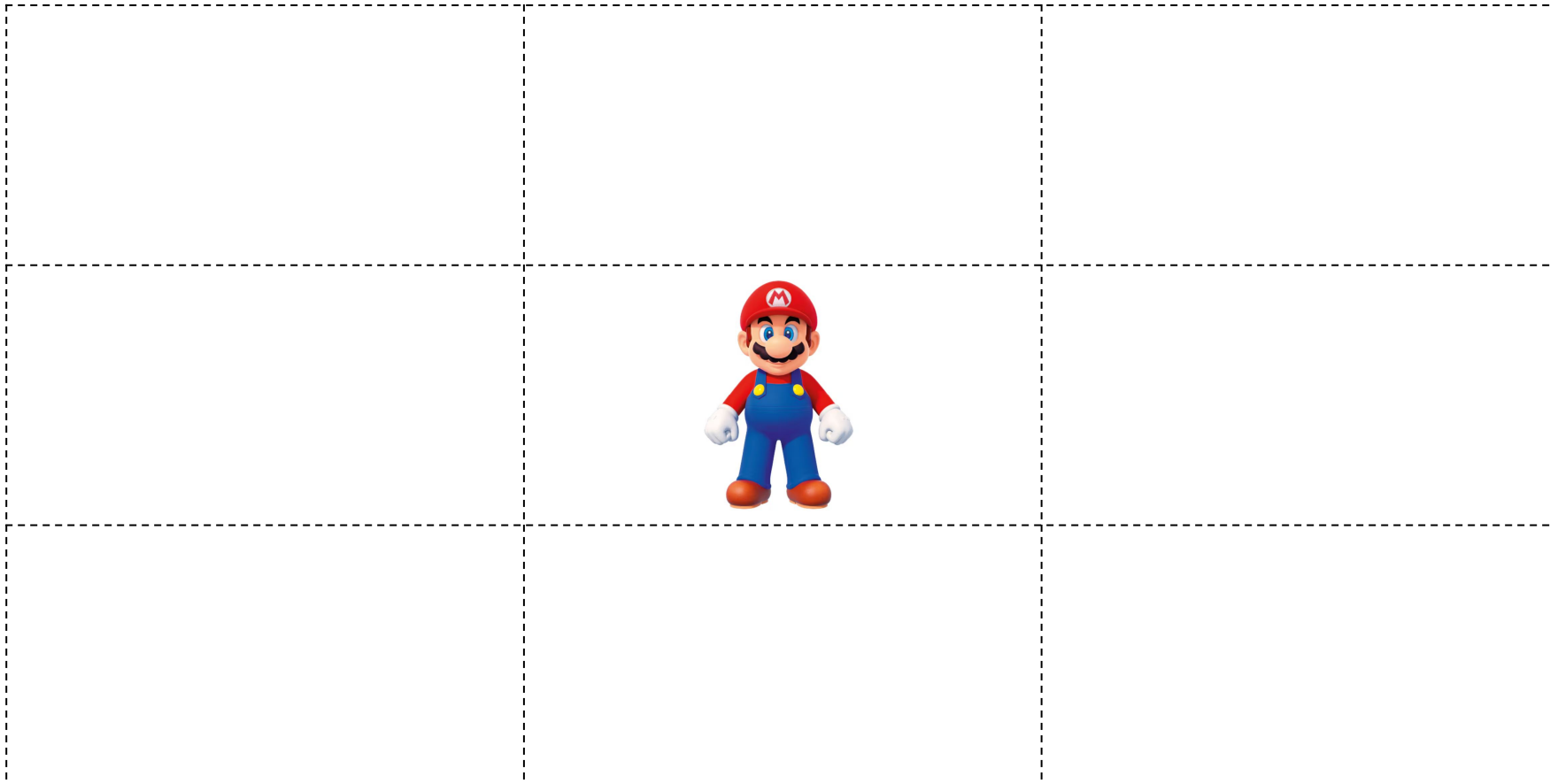
Actividad 5 - Bot saves princess

Actividad 5 - Bot saves princess



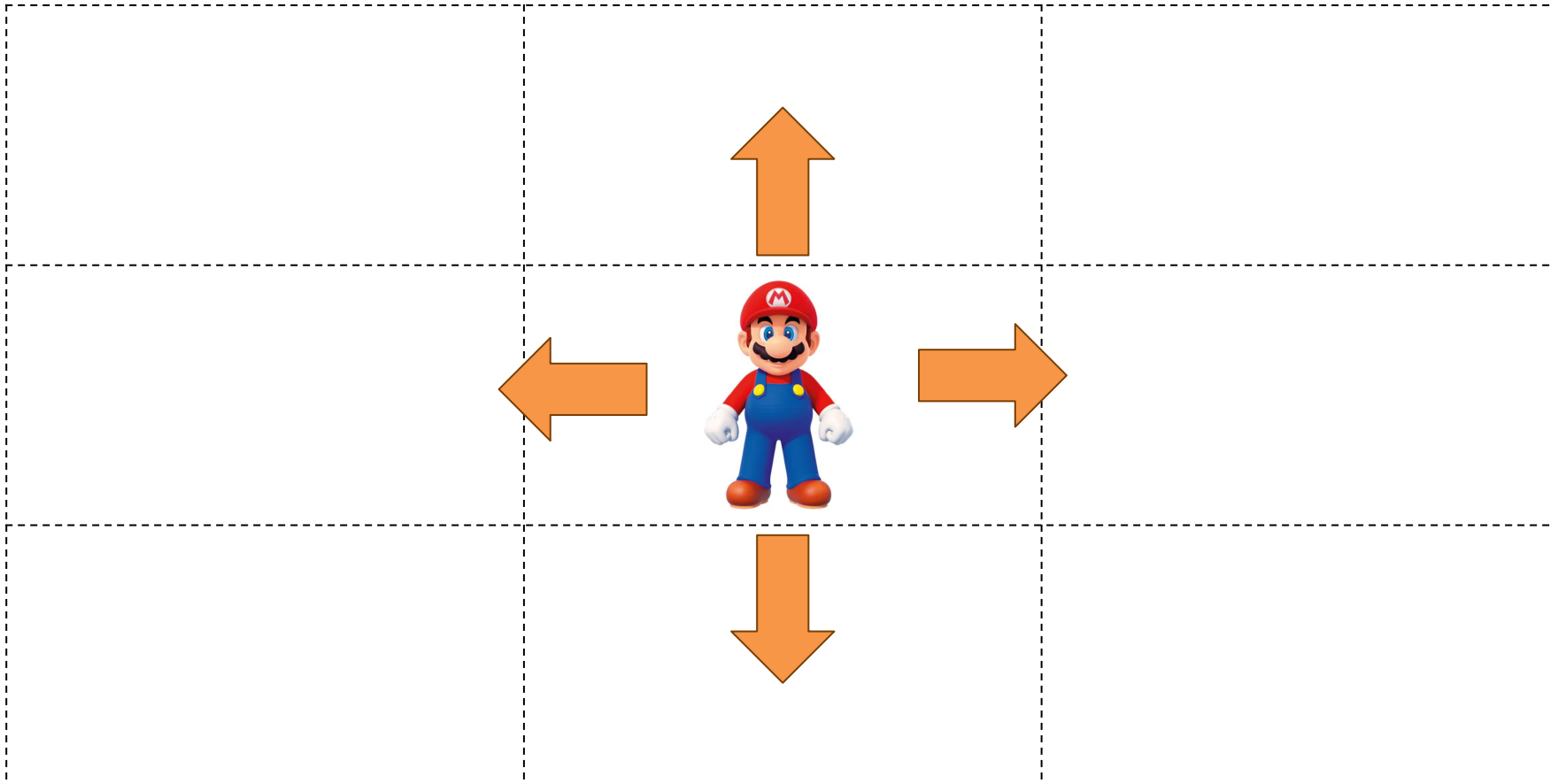
Actividad 5 - Bot saves princess

- Supongamos que **Mario** es un agente en una cuadrícula.



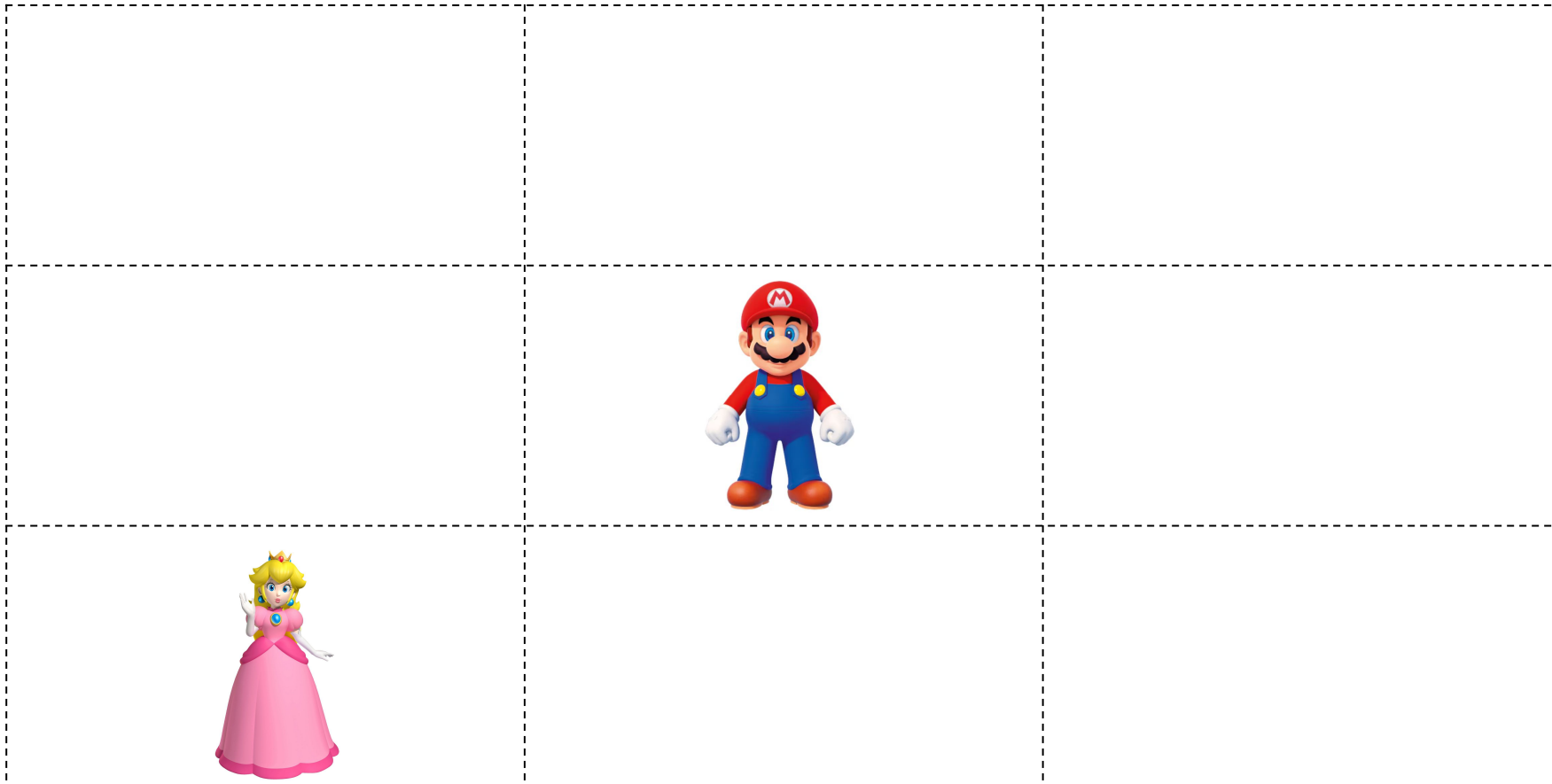
Actividad 5 - Bot saves princess

- Supongamos que **Mario** es un agente en una cuadrícula.
- La **acciones del agente Mario**: LEFT, RIGHT, UP o DOWN



Actividad 5 - Bot saves princess

- Supongamos que **Mario** es un agente en una cuadrícula.
- La **acciones del agente Mario**: LEFT, RIGHT, UP o DOWN
- La **finalidad** del agente **Mario** es encontrar a la **princesa Peach**.



Actividad 5 - Bot saves princess

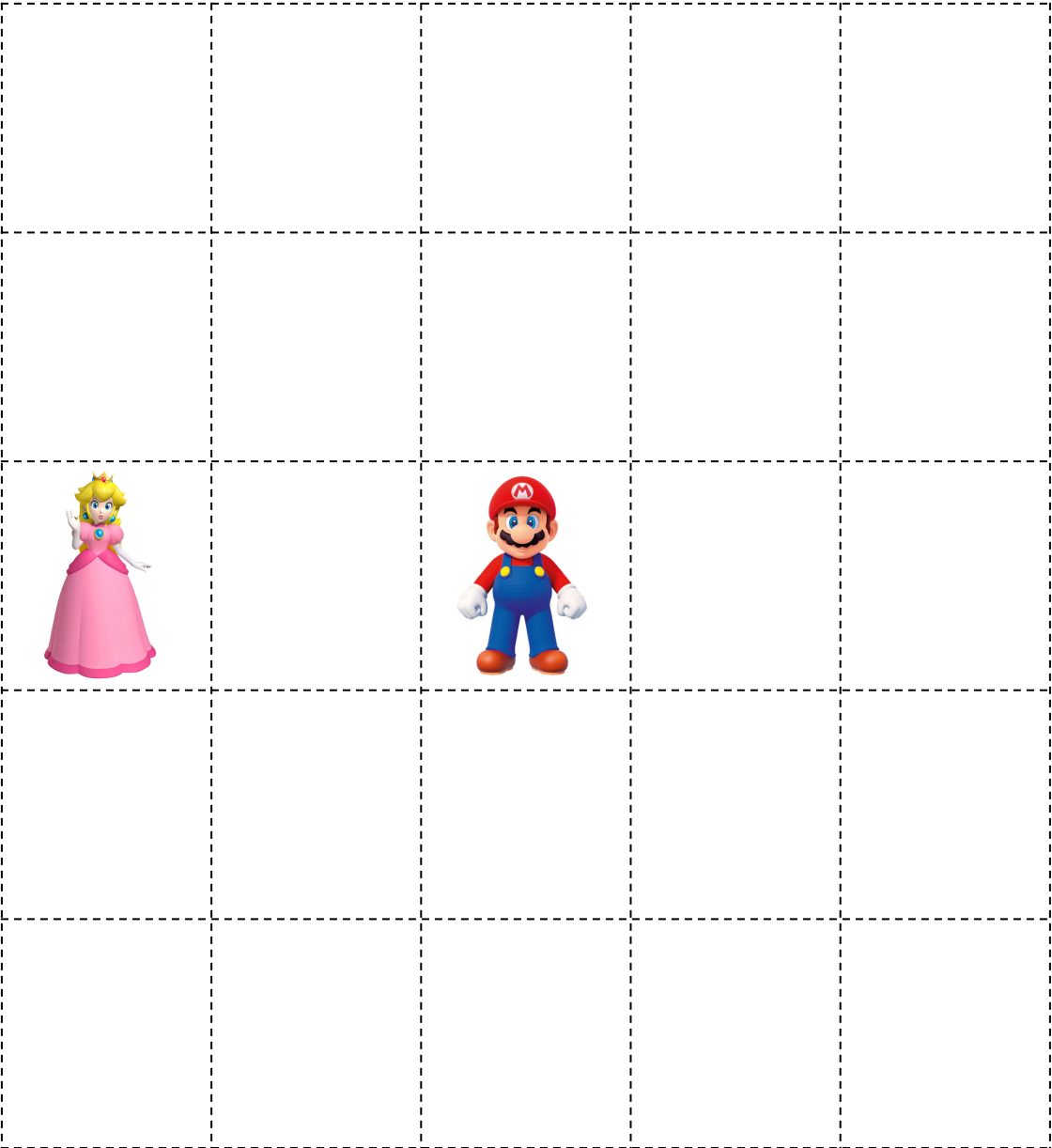
La **función del agente** Mario.

Secuencia de percepciones	Acción
Mario está a la derecha de Peach	RIGHT
Mario está a la izquierda de Peach	LEFT
Mario está arriba de Peach	DOWN
Mario está abajo de Peach	UP

Actividad 5 - Bot saves princess

La **función del agente** Mario.

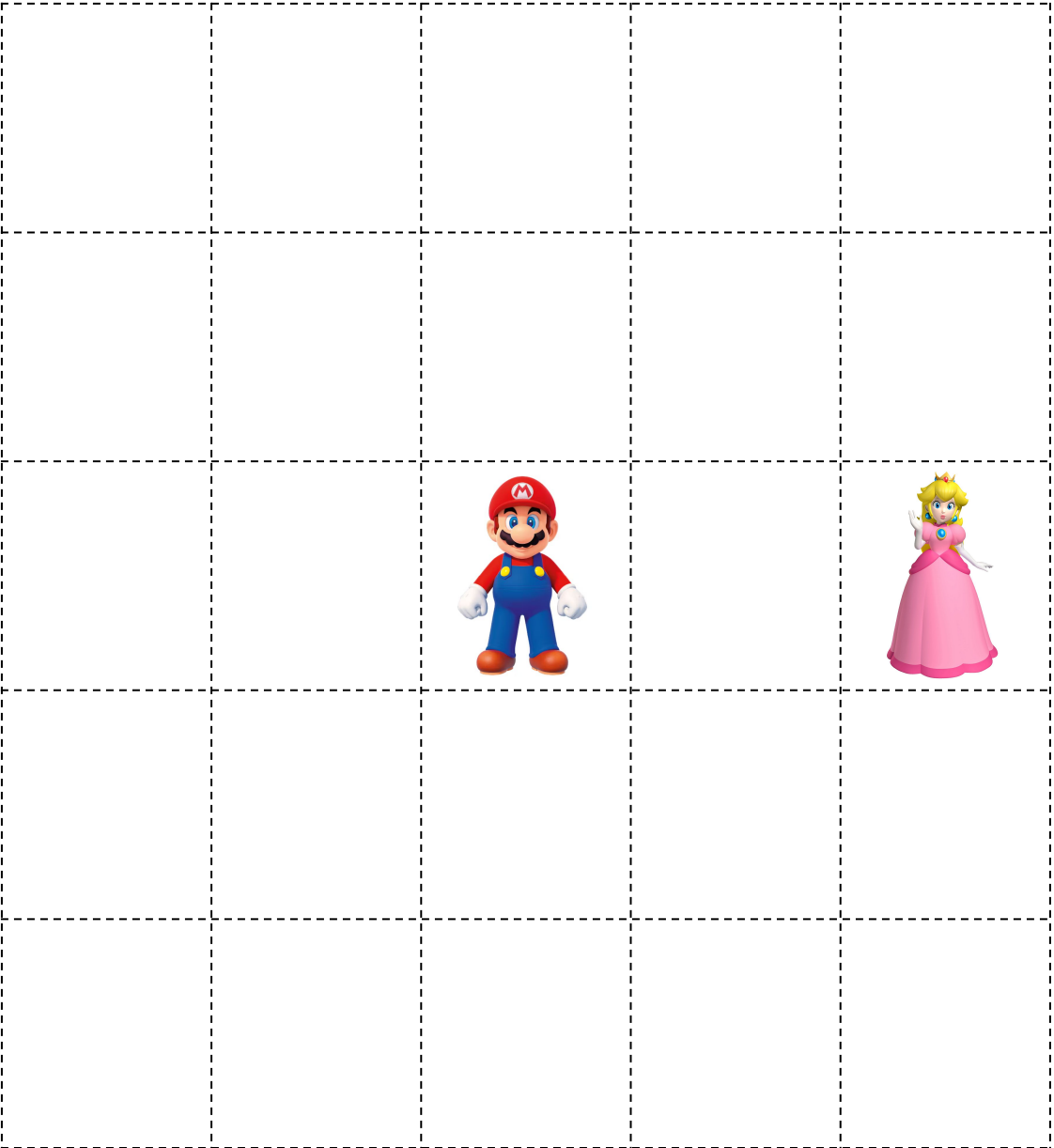
Secuencia de percepciones	Acción
Mario está a la derecha de Peach	RIGHT
Mario está a la izquierda de Peach	LEFT
Mario está arriba de Peach	DOWN
Mario está abajo de Peach	UP



Actividad 5 - Bot saves princess

La **función del agente** Mario.

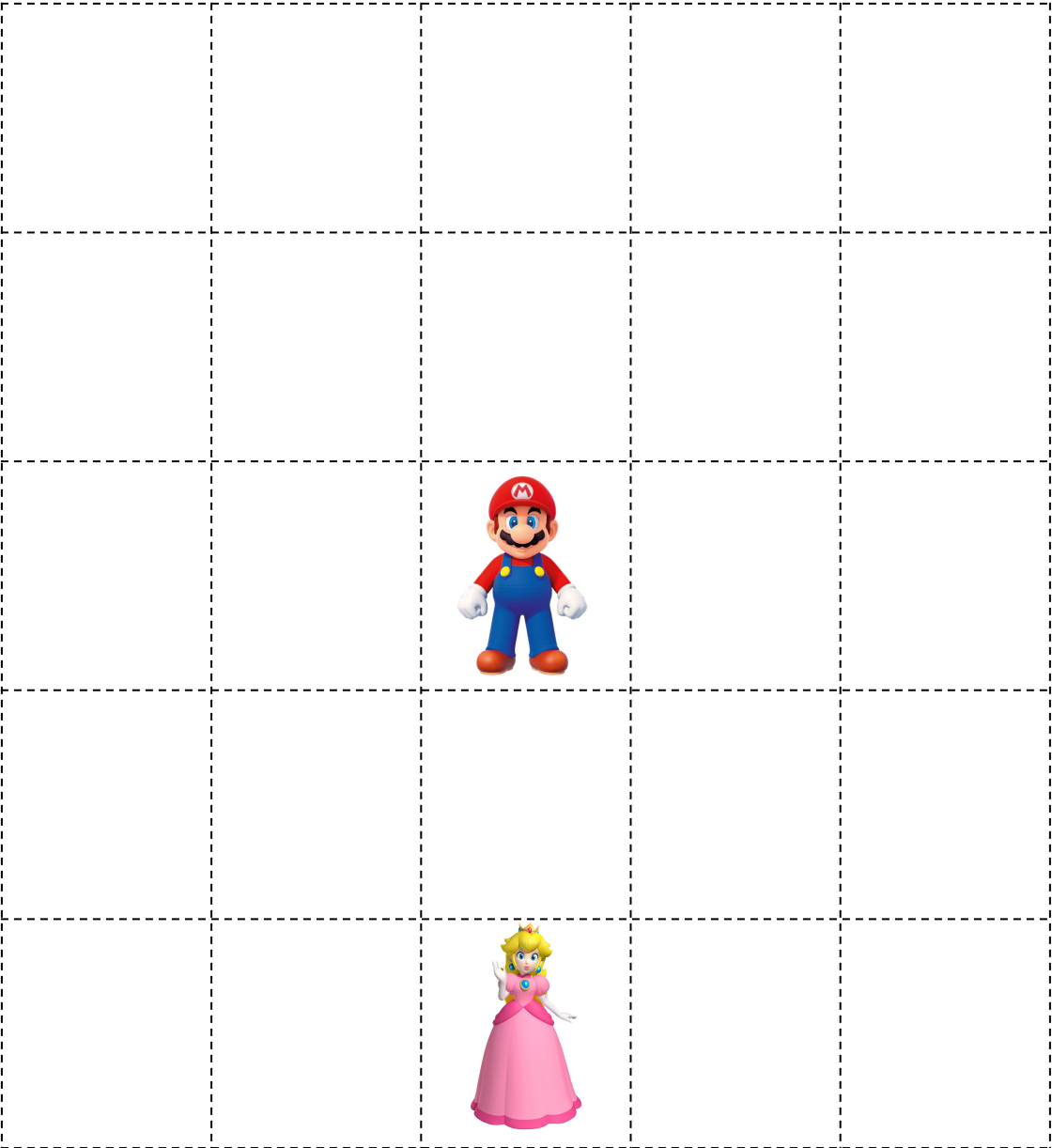
Secuencia de percepciones	Acción
Mario está a la derecha de Peach	RIGHT
Mario está a la izquierda de Peach	LEFT
Mario está arriba de Peach	DOWN
Mario está abajo de Peach	UP



Actividad 5 - Bot saves princess

La **función del agente** Mario.

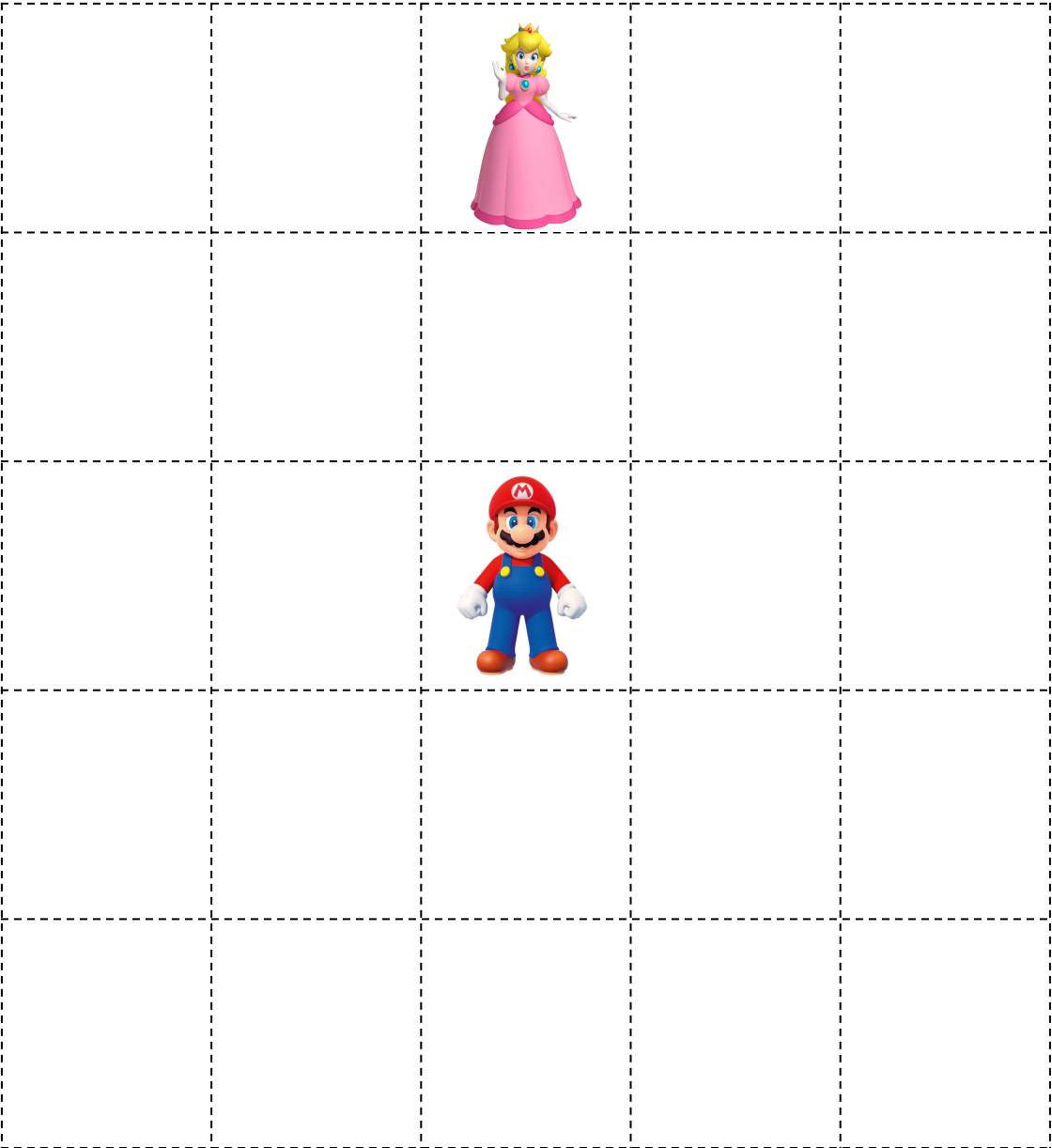
Secuencia de percepciones	Acción
Mario está a la derecha de Peach	RIGHT
Mario está a la izquierda de Peach	LEFT
Mario está arriba de Peach	DOWN
Mario está abajo de Peach	UP



Actividad 5 - Bot saves princess

La **función del agente** Mario.

Secuencia de percepciones	Acción
Mario está a la derecha de Peach	RIGHT
Mario está a la izquierda de Peach	LEFT
Mario está arriba de Peach	DOWN
Mario está abajo de Peach	UP



Actividad 5 - Bot saves princess

Entren a la siguiente liga:

<https://www.hackerrank.com/challenges/saveprincess>

Seguir las instrucciones y codificar en su lenguaje de programación favorito el **programa del agente** para que se mueva en su medio y cumpla su objetivo.

Tipos de programas de agentes

Tipos de programas de agentes

Existen cuatro tipos básicos de **programas de agente**:

- 1.- Agentes reactivos simples
- 2.- Agentes reactivos basados en modelos
- 3.- Agentes basados en objetivos
- 4.- Agentes basados en utilidad

(1) Agentes reactivos simples

El tipo de agente más sencillo es el agente reactivo simple.

Estos agentes seleccionan las **acciones** sobre la base de las **percepciones actuales**, ignorando el resto de las **percepciones históricas**.

(1) Agentes reactivos simples

El agente **reactivo simple** utiliza reglas de **condición-acción**.

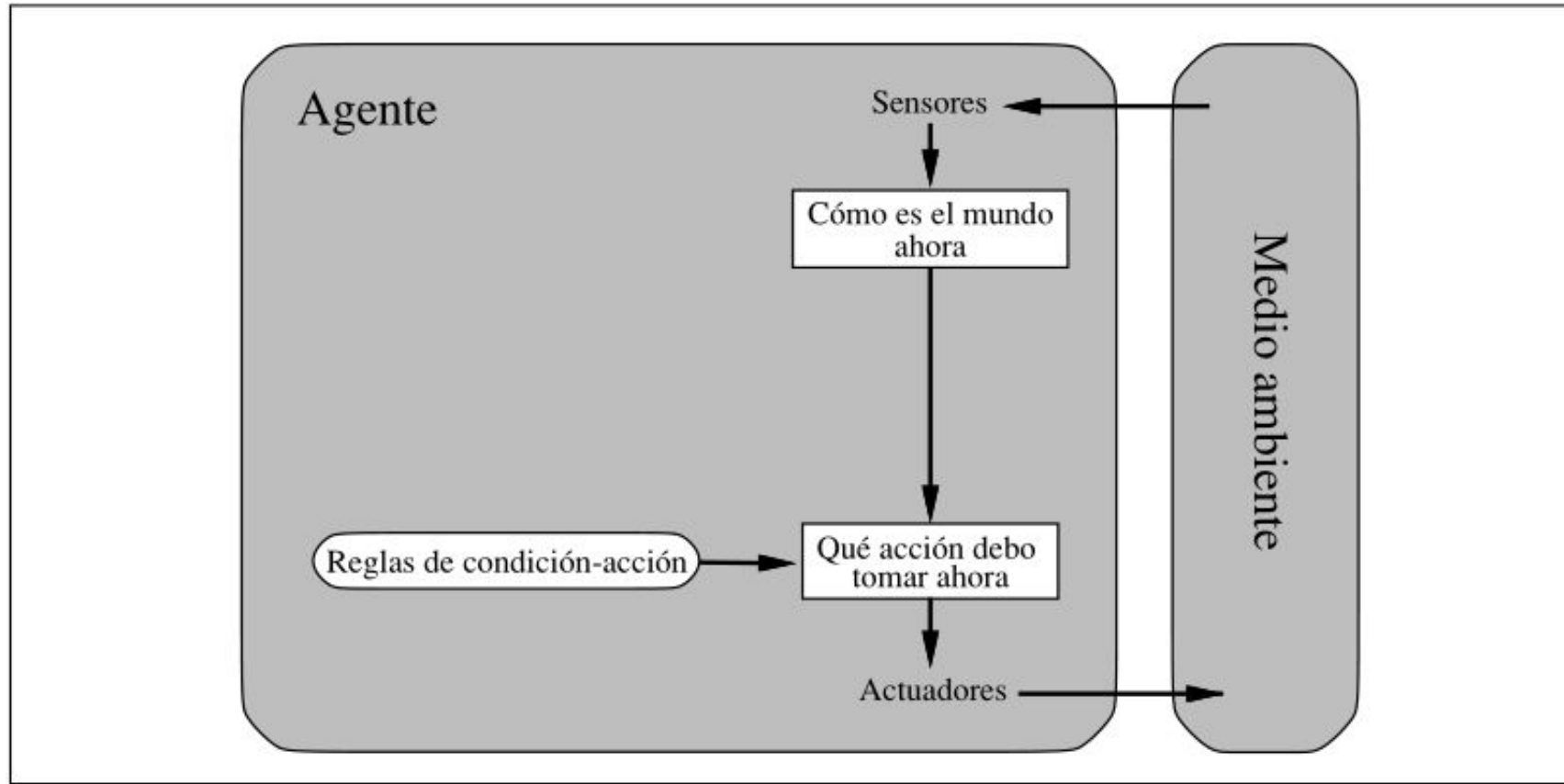


Figura 1. Esquema de un agente reactivo simple.

(1) Agentes reactivos simples

El agente **reactivo simple** utiliza reglas de **condición-acción**.

```
función AGENTE-REACTIVO-SIMPLE(percepción) devuelve una acción  
  estático: reglas, un conjunto de reglas condición-acción  
  
  estado ← INTERPRETAR-ENTRADA(percepción)  
  regla ← REGLA-COINCIDENCIA(estado, reglas)  
  acción ← REGLA-ACCIÓN[regla]  
  devolver acción
```

Figura 2. Ejemplo del programa de un agente reactivo simple.

(1) Agentes reactivos simples

Ejemplo.

El agente aspiradora es un **agente reactivo simple** porque toma sus **decisiones** sólo con base en la **localización actual** y si ésta está sucia.

función AGENTE-ASPIRADORA-REACTIVO([*localización*, *estado*]) **devuelve** una acción

si *estado* = *Sucio* **entonces devolver** *Aspirar*

de otra forma, si *localización* = *A* **entonces devolver** *Derecha*

de otra forma, si *localización* = *B* **entonces devolver** *Izquierda*

Figura 3. Programa del agente aspiradora en el entorno definido por las dos cuadrículas.

(2) Agentes reactivos basados en modelos

En entornos parcialmente observables un agente debe mantener algún tipo de **estado interno** que dependa de la historia percibida y que de ese modo refleje por lo menos algunos de los aspectos no observables del estado actual.

La actualización de la información de estado interno según pasa el tiempo requiere codificar **dos tipos de conocimiento** en el programa del agente.

(2) Agentes reactivos basados en modelos

Primero, el agente necesita información acerca de cómo evoluciona el mundo **independientemente** del agente.

Segundo, el agente necesita información sobre cómo **afectan** al mundo las acciones del agente.

Este conocimiento acerca de «cómo funciona el mundo», tanto si está implementado con un circuito booleano simple o con teorías científicas completas, se denomina **modelo del mundo**. Un agente que utilice este modelo es un **agente basado en modelos**.

(2) Agentes reactivos basados en modelos

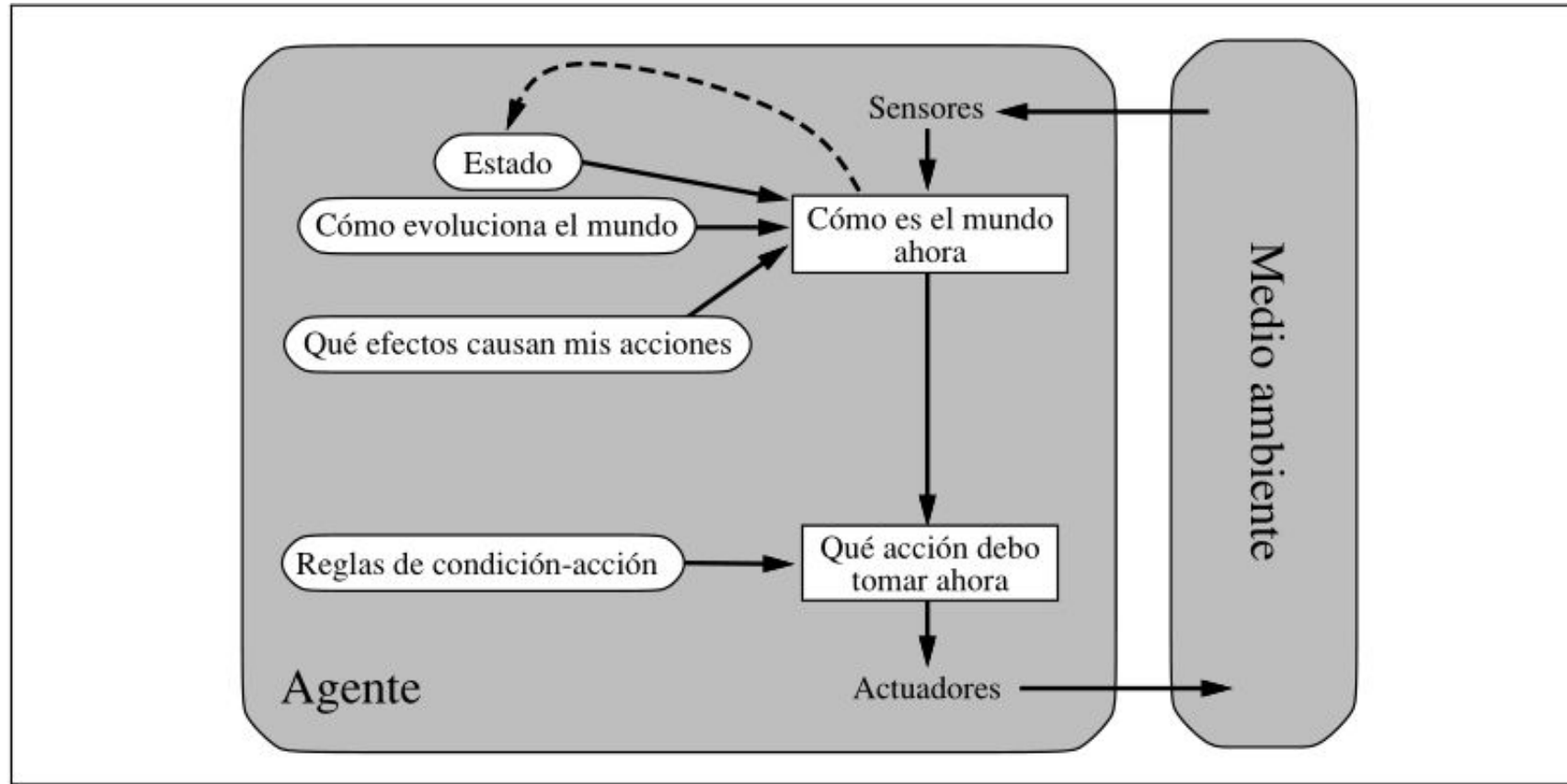


Figura 4. Esquema de un agente reactivo basado en modelos.

(3) Agentes basados en objetivos

El conocimiento sobre el estado actual del mundo **no es siempre suficiente** para decidir qué hacer.

Además de la descripción del estado actual, el agente necesita algún tipo de información sobre su **meta** que describa las situaciones que son deseables.

El programa del agente se puede combinar con información sobre los resultados de las acciones posibles (información utilizada para actualizar el estado interno) para **seleccionar las acciones** que permitan alcanzar un **objetivo**.

(3) Agentes basados en objetivos

En algunas ocasiones, la **selección de acciones** basadas en objetivos es **directa**, cuando alcanzar los objetivos es el resultado inmediato de una **acción individual**.

En otras ocasiones, puede ser más complicado, cuando el agente tiene que considerar **secuencias complejas** para encontrar el camino que le permita alcanzar el objetivo.

La **búsqueda** y la **planificación** son los subcampos de la IA centrados en encontrar **secuencias de acciones** que permitan a los agentes alcanzar sus **metas**.

(3) Agentes basados en objetivos

La **toma de decisiones** de este tipo de agentes es fundamentalmente diferente de las reglas de **condición-acción** descritas anteriormente.

- ¿Qué pasa si yo ejecuto una acción, me hará esta acción feliz?

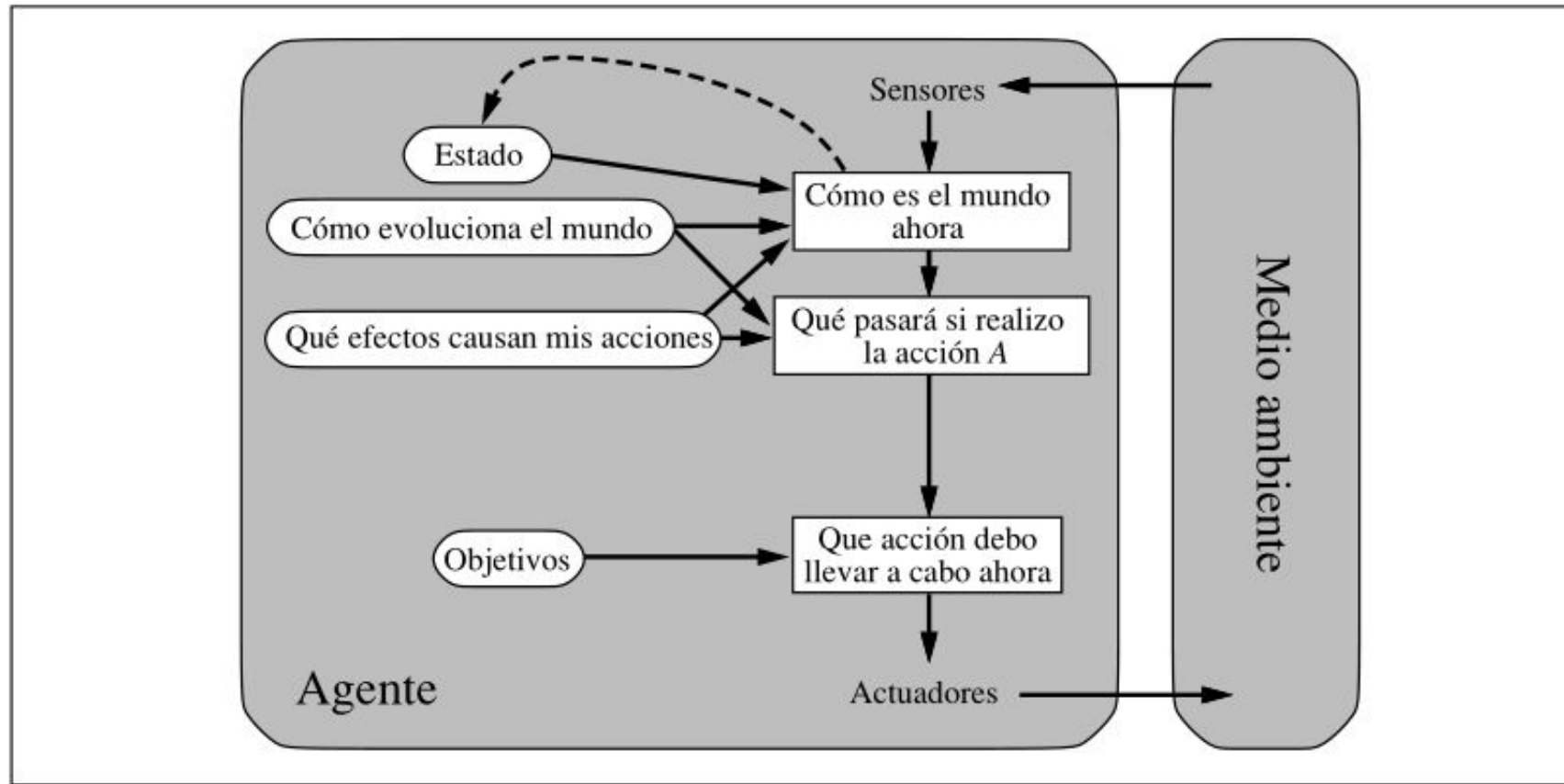


Figura 5. Esquema de un agente basado en objetivos.

(4) Agentes basados en utilidad

Las metas por sí solas **no son realmente suficientes** para generar comportamiento de gran **calidad** en la mayoría de los entornos.

Las metas sólo proporcionan una cruda distinción **binaria** entre los estados de “*felicidad*” y “*tristeza*”.

Mientras que una **medida de eficiencia** más general debería permitir una comparación entre estados del mundo diferentes de acuerdo al nivel **exacto de felicidad** que el agente alcance cuando se llegue a un estado u otro.

(4) Agentes basados en utilidad

La terminología tradicional utilizada en estos casos para indicar que se prefiere un estado del mundo a otro es que un estado tiene mayor **utilidad** que otro para el agente.

La **utilidad** es la medida de satisfacción del agente.

Una **función de utilidad** mapea un estado (o una secuencia de estados) en un número real, que representa un nivel de felicidad.

(4) Agentes basados en utilidad

Una **función de utilidad** permite tomar decisiones racionales en **dos tipos** de casos en los que las metas son inadecuadas.

Primero, cuando hay **objetivos en conflicto** de tal forma que sólo se puede alcanzar algunos de ellos (por ejemplo, velocidad y seguridad). La función de utilidad ayuda a **determinar el equilibrio** adecuado.

Segundo, cuando hay **varios objetivos** por los que se pueda guiar el agente, y ninguno de ellos se pueda alcanzar con certeza, la utilidad proporciona un mecanismo para ponderar la probabilidad de éxito en función de la importancia de los objetivos.

(4) Agentes basados en utilidad

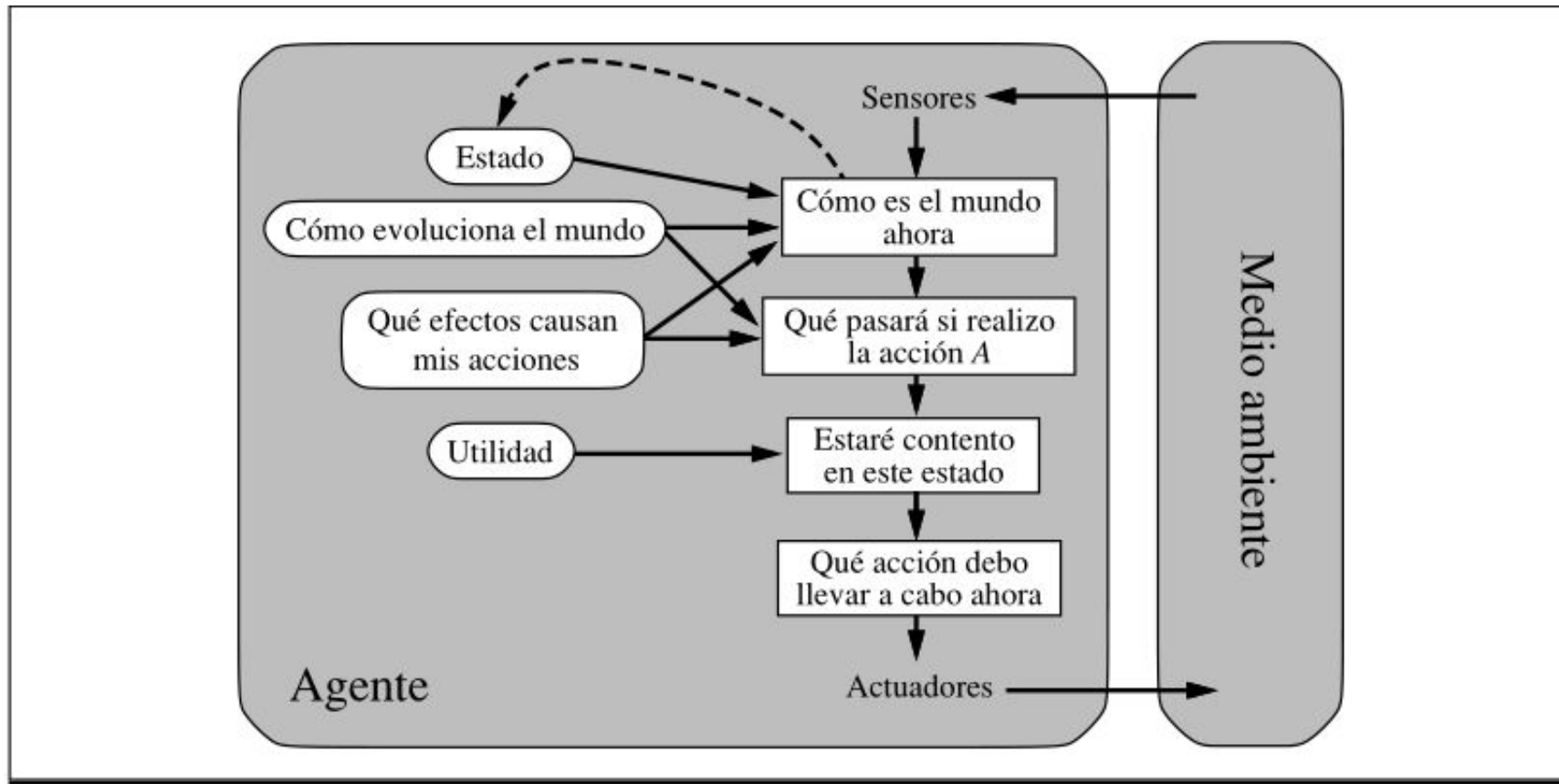


Figura 6. Esquema de un agente basado en utilidad y basado en modelos. Utiliza un **modelo del mundo**, junto con una **función de utilidad** que calcula sus **preferencias** entre los estados del mundo. Después selecciona la **acción** que le lleve a alcanzar la **mayor utilidad esperada**, que se calcula haciendo la **media** de todos los **estados resultantes** posibles, ponderado con la **probabilidad** del resultado.

(4) Agentes basados en utilidad

Ejemplo.

Suponga que en un concurso se le plantea una elección. Aceptar directamente un premio de un millón de dólares o se juega el premio a un volado de moneda, de forma que si sale cara, lo pierde todo, y si sale cruz, gana tres millones de dólares.

¿Qué haría un agente racional?

(4) Agentes basados en utilidad

Probablemente, como lo haría mucha más gente, usted desestimaría jugársela a cara o cruz y querría decir esto que usted es **irracional**.

Suponiendo que usted confía en que la moneda no está trucada, el valor utilidad monetario esperado de la apuesta planteada es:

$$\text{Valor de utilidad esperado} = \frac{1}{2}(\$0) + \frac{1}{2}(\$3,000,000) = \$1,500,000$$

Agentes que aprenden

Agentes que aprenden

Turing (1950), en su temprano y famoso artículo, consideró la idea de programar sus **máquinas inteligentes** a mano.

Estimó cuánto tiempo podía llevar y concluyó que “sería deseable utilizar algún método más rápido”.

El método que propone es construir **máquinas** que **aprendan** y después enseñarlas.

Agentes que aprenden

El **aprendizaje** tiene otras ventajas, como se ha explicado anteriormente: permite que el agente opere en medios inicialmente desconocidos y que sea más competente que si sólo utilizase un conocimiento inicial.

Un **agente que aprende** se puede dividir en **cuatro** componentes conceptuales:

- **Elemento de aprendizaje.**
- **Elemento de actuación.**
- **Crítica.**
- **Generador de problemas.**

Agentes que aprenden

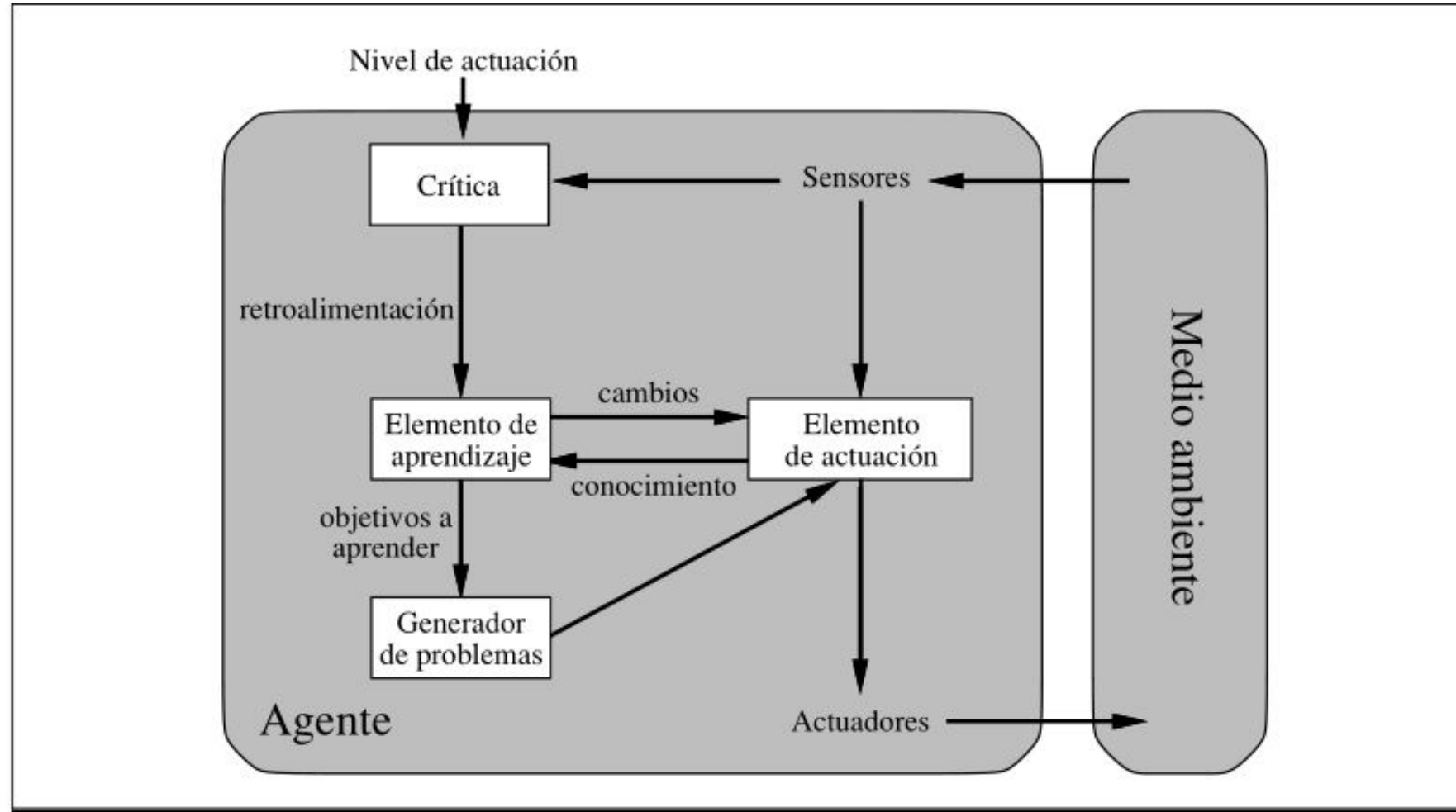


Figura 7. Esquema de un agente que aprende. Cualquiera de los cuatro tipos básicos de programas de agentes pueden incluir el aprendizaje.

Agentes que aprenden

El elemento de **actuación** es lo que anteriormente se había considerado como el agente completo: recibe estímulos y determina las acciones a realizar.

El elemento de **aprendizaje** se realimenta con las críticas sobre la actuación del agente y determina cómo se debe modificar el elemento de actuación para proporcionar mejores resultados en el futuro.

Agentes que aprenden

La distinción más importante entre el elemento de **aprendizaje** y el elemento de **actuación** es que el primero está responsabilizado de hacer mejoras y el segundo se responsabiliza de la selección de acciones externas.

La **crítica** indica al elemento de aprendizaje qué tal lo está haciendo el agente con respecto a un nivel de actuación fijo.

La crítica es necesaria porque las percepciones por sí mismas no prevén una indicación del **éxito del agente**.

Agentes que aprenden

El último componente del agente con capacidad de aprendizaje es el **generador de problemas**.

Es responsable de sugerir acciones que lo guiarán hacia experiencias nuevas e informativas, es decir, recomienda **acciones exploratorias** al agente.

Agentes que aprenden

Si el agente está dispuesto a explorar un poco, y llevar a cabo algunas acciones que no sean totalmente **óptimas a corto** plazo, puede descubrir acciones mejores a **largo plazo**.

Nota:

El elemento de **aprendizaje** puede hacer cambios en cualquiera de los componentes de “conocimiento” que se muestran en los cuatro tipos básicos de **programas de agente**.

2.2. Diferencias entre el Enfoque Orientado a Objetos y Orientado a Agentes

**Enfoque Orientado a
Objetos (OO)**

VS

**Enfoque Orientado a
Agentes (OA)**

2.2. Diferencias entre el Enfoque Orientado a Objetos y Orientado a Agentes

Aspecto	Enfoque Orientado a Objetos (OO)	Enfoque Orientado a Agentes (AOA)
Unidad básica	Objeto (instancia de una clase).	Agente (entidad autónoma).
Control	Control centralizado o distribuido, pero coordinado por el programador.	Autonomía: cada agente decide sus acciones en función de su percepción y objetivos.
Comunicación	Intercambio de mensajes prediseñados (métodos invocados).	Comunicación flexible, basada en lenguajes de agentes (ej. KQML, FIPA ACL).
Estados internos	Limitados, definidos por atributos y métodos.	Complejos: incluyen creencias, deseos e intenciones (modelo BDI).
Entorno	Pasivo; los objetos no perciben su entorno, actúan cuando se les llama.	Dinámico: el agente percibe, razona y actúa proactivamente.

Referencias bibliográficas

Russell, S. J. and Norvig, P. (2016). Artificial Intelligence: A Modern Approach (3rd ed). Pearson Education Limited.

Poole, D.L. and Mackworth, A.K. (2017). Artificial Intelligence: Foundations of Computational Agents (2nd ed). New York, NY: Cambridge University Press.



ISyTE

Dr. Héctor Zatarain Aceves

Email: hector.zatarain@uabc.edu.mx

Teléfono: (646) 152 8244 Ext. 64350