

# LA CLASE ESTÁ A PUNTO DE COMENZAR



**Por favor:** mantenga en silencio su micrófono y  
apague su cámara

# Agentes inteligentes

## Inteligencia Artificial



**Marco Teran**

2021 - Bogotá

# Contenido

- 1 Agentes y entornos
- 2 Racionalidad
- 3 PEAS (Performance measure, Environment, Actuators, Sensors)
- 4 Tipos de entorno
- 5 Tipos de agentes
- 6 Resumen

# Agentes y entornos

# Agentes: definiciones

Rusell & Norving

“Un agente es cualquier cosa que pueda ver en su **entorno** a través de **sensores** y actuar en su entorno a través de **actuadores**”.

Nwana

“Agente: componente de software y/o hardware el cual es capaz de actuar de manera precisa con el fin de realizar tareas en **representación** de su usuario”.

# Agentes: definiciones

## Maes

“Un agente **autónomo** es un sistema computacional que habita en un entorno dinámico-complejo, **percibiendo** y **actuando** autónomamente en este **entorno**, y haciendo esto para realizar un conjunto de **objetivos** o tareas para los cuales fueron diseñados”.

## Wooldridge & Jennings

“Un agente es un **sistema computacional** que está situado en algún **ambiente**, y que es capaz de actuar **autónomamente** en dicho ambiente con el fin de cumplir sus **objetivos**”.

# El mundo de las aspiradoras



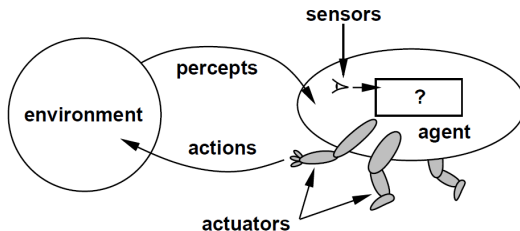
▶ ver video

# 12 different types of sensors





# Agentes y entornos



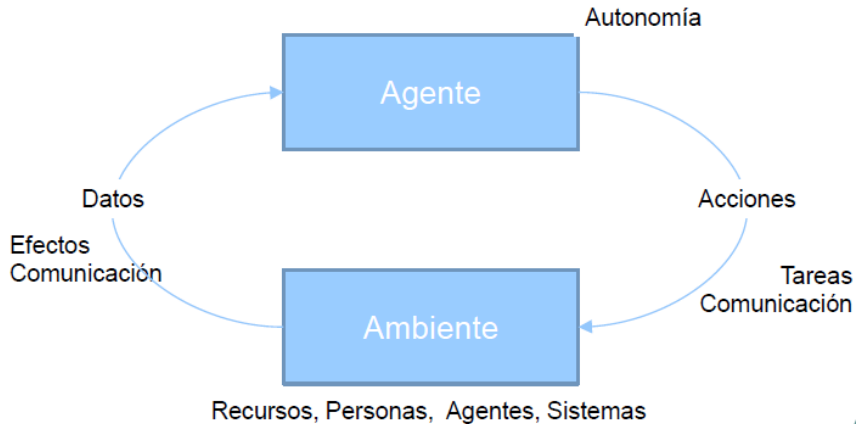
Los **agentes** pueden ser humanos, robots, softbots, termostatos, etc.

La **función del agente** asigna los historiales de percepción a las acciones:

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

El **programa del agente** se ejecuta en la **arquitectura** física para producir  $f$

# Vista abstracta de un agente



# Agentes Inteligentes y software convencional

Hay dos principales características que distinguen a los agentes inteligentes del desarrollo de software tradicional:

- 1 Inteligencia
- 2 Agencia.

# Agentes Inteligentes y software convencional

- **Inteligencia** se refiere al grado en el cual la aplicación utiliza *razonamiento*, *aprendizaje* y otras técnicas para interpretar la información o conocimiento al cual tiene acceso. Se puede decir que hay tres niveles de formas de inteligencia:
  - 1 Permite al usuario expresar sus preferencias.
  - 2 Formalizar un conjunto de reglas de razonamiento que combinadas con conocimiento a corto y largo plazo, siguiendo un proceso de inferencia puede conducir a la toma de alguna acción.
  - 3 Capacidad del agente de modificar su capacidad de razonamiento en la base nuevo conocimiento derivado de muchas fuentes, es decir, aprender.
- **Agencia** el grado en el cual el agente puede percibir su entorno y actuar en el. Define al agente, en otras palabras, para que un programa sea un agente debe poseer autonomía, habilidad social, reactividad y proactividad.

# Agentes inteligentes

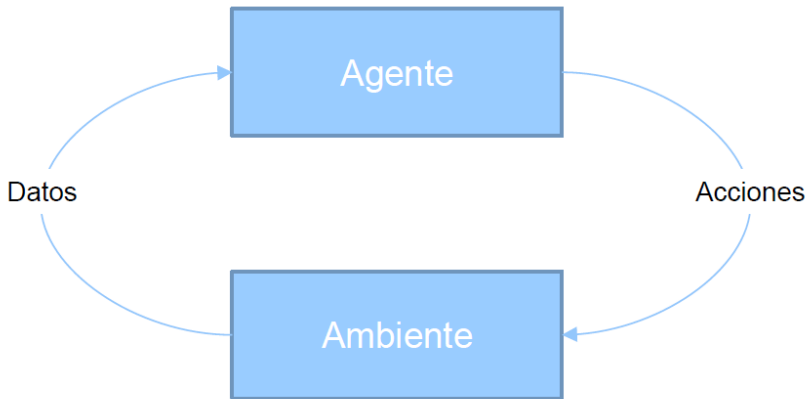
“Un **agente inteligente** es un agente capaz de ejecutar **flexiblemente acciones autónomas** con el fin de cumplir sus **objetivos**”, donde **flexibilidad** significa:

- **Reactividad:** capacidad de percibir su ambiente, y responder sin demoras a cambios que ocurren en él.
- **Pro-actividad:** capacidad de exhibir un comportamiento dirigido a objetivos, tomando la iniciativa.
- **Habilidad social:** capacidad de interactuar con otros agentes (y posiblemente humanos) a través de un lenguaje de comunicación.

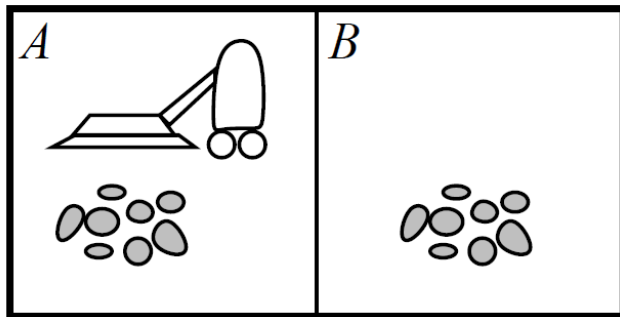
## Otras cuestiones...

- **Adaptación** Implica percibir el entorno y reconfigurar el estado mental en respuesta a este entorno.
- **Aprendizaje** Mediante distintas técnicas de aprendizaje de máquina.

# Vista abstracta de un agente



# El mundo de las aspiradoras



- **Percepciones:** ubicación del ambiente (cuadrícula) y contenido, por ejemplo, [*A*; Dirty]
- **Acciones:** *Left*, *Right*, *Suck*, *NoOp*



# El agente aspiradora

Percept sequence	Action
$[A; Clean]$	Right
$[A; Dirty]$	Suck
$[B; Clean]$	Left
$[B; Dirty]$	Suck
$[A; Clean], [A; Clean]$	Right
$[A; Clean], [A; Dirty]$	Suck
$\vdots$	$\vdots$

# El agente aspiradora

```
function REFLEX-VACUUM-AGENT([location,status]) returns an action
  if status = Dirty then return Suck
  else if location = A then return Right
  else if location = B then return Left
```

¿Qué hace la función **Right**?

¿Puede implementarse en un pequeño programa de agentes?

# Agentes y sistemas expertos

## Sistema experto

Sistema capaz de resolver problemas o dar consejos en algún dominio rico en conocimiento.

- **Agentes vs. SE** SE no interactúan directamente con el ambiente.

**Racionalidad**

# Racionalidad

La **medida de rendimiento fija** (performance measure) evalúa la **secuencia del entorno**

- ¿un *punto* por cuadrícula limpia en un tiempo  $T$ ?
- ¿un *punto* por cuadrícula limpia por paso de tiempo, *menos* uno por movimiento?
- ¿penalizar por  $> k$  cuadrículas sucias?

# Racionalidad

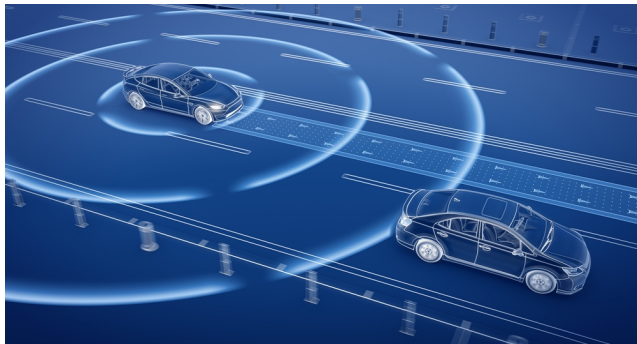
Un **agente racional** elige la acción que maximiza el valor **esperado** de la medida de rendimiento **dada la secuencia de percepciones hasta la fecha**

- Racional  $\neq$  omnisciente
  - las percepciones pueden no proporcionar toda la información relevante
- Racional  $\neq$  clarividente
  - los resultados de la acción pueden no ser los esperados
- Racional  $\neq$  exitoso

Racional  $\rightarrow$  exploración, aprendizaje, autonomía

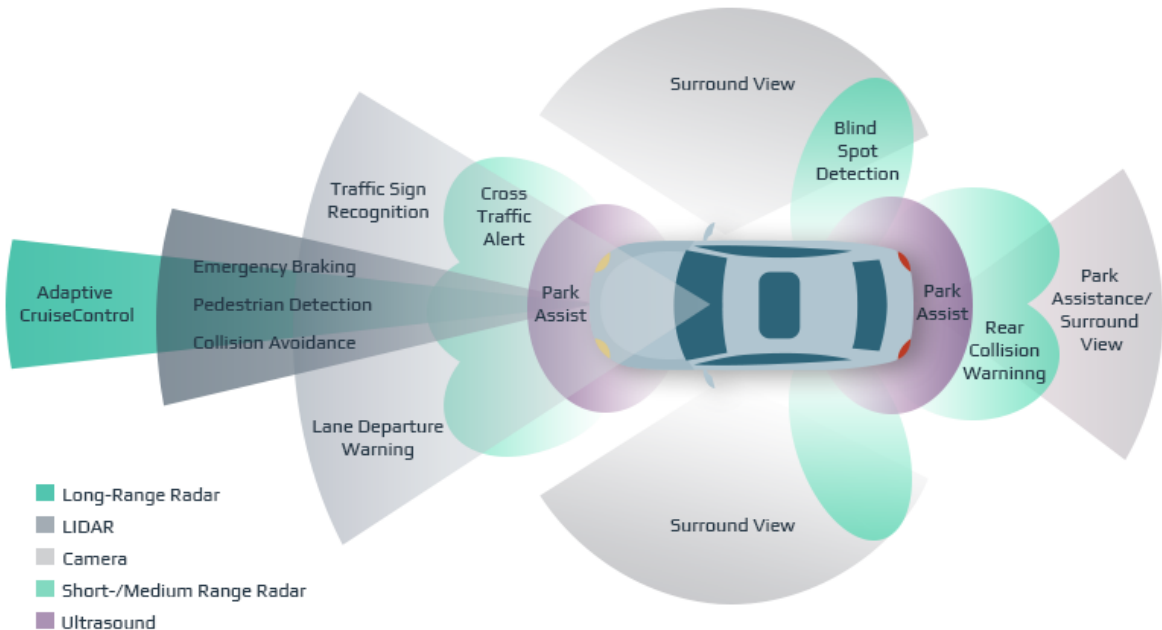
**PEAS (Performance measure,  
Environment, Actuators, Sensors)**

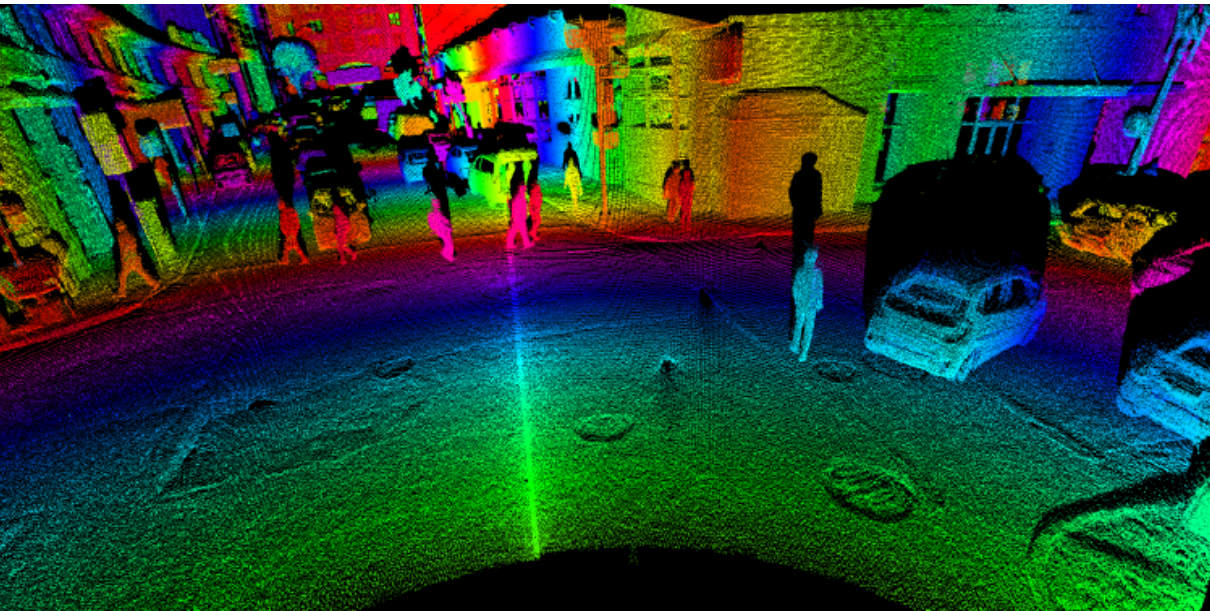
# Carros autónomos



▶ ver video







# PEAS (Performance measure, Environment, Actuators, Sensors)

Para diseñar un agente racional, debemos especificar el **entorno de la tarea** (task environment)

Consideremos, por ejemplo, la tarea de diseñar un taxi autonomo:

- ¿Medida de rendimiento?
- ¿Entorno?
- ¿Actuadores?
- ¿Sensores?

# PEAS (Performance measure, Environment, Actuators, Sensors)

Para diseñar un agente racional, debemos especificar el **entorno de la tarea** (task environment)

Consideremos, por ejemplo, la tarea de diseñar un taxi autonomo:

- **¿Medida de rendimiento?** seguridad, destino, prots, legalidad, comodidad, ...
- **¿Entorno?** calles/carreteras de EE.UU., trazado, peatones, clima, ...
- **¿Actuadores?** dirección, acelerador, freno, bocina, altavoz/pantalla, ...
- **¿Sensores?** vídeo, acelerómetros, indicadores, sensores del motor, teclado, GPS, ...

# Agente de compras por Internet

- ¿Medida de rendimiento?
- ¿Entorno?
- ¿Actuadores?
- ¿Sensores?

# Agente de compras por Internet

- **¿Medida de rendimiento?** precio, calidad, adecuación, eficiencia
- **¿Entorno?** sitios WWW actuales y futuros, proveedores, expedidores
- **¿Actuadores?** mostrar al usuario, seguir la URL, rellenar el formulario
- **¿Sensores?** Páginas HTML (texto, gráficos, scripts)

# Tipos de entorno

# Tipos de entorno

## El tipo de entorno determina en gran medida el diseño del agente

El mundo real es (por supuesto) parcialmente observable, estocástico, secuencial, dinámico, continuo y multiagente

- Observable
- Determinista
- Episódico
- Estático
- Discreto
- Agente único



# Tipos de agentess

# Tipos de agentess

Cuatro tipos básicos en orden de generalidad creciente:

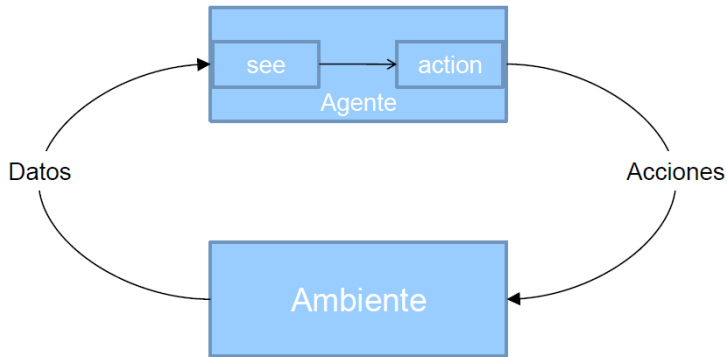
- Agentes puramente reactivos (*simple reflex agents*)
- Agentes con estado (*reflex agents with state*)
- Agentes basados en objetivos (*goal-based agents*)
- Agentes basados en la utilidad (*utility-based agents*)

Todos ellos pueden convertirse en **agentes de aprendizaje** (*learning agents*)

# Agentes puramente reactivos(*simple reflex agents*)

Toma de decisiones enteramente basada en el presente.

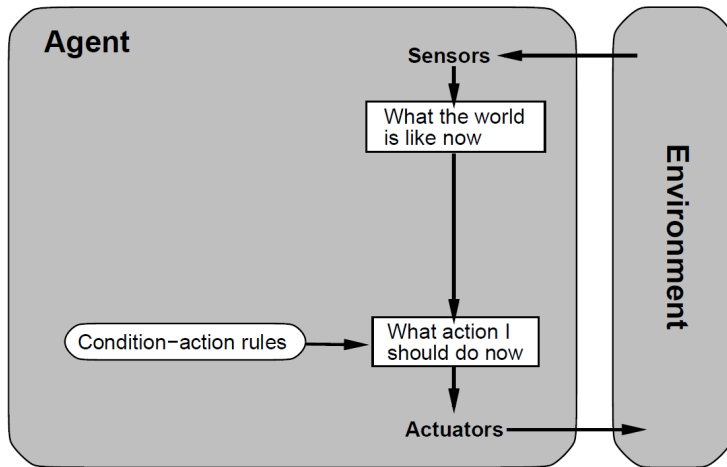
$$action : S \rightarrow A$$



$$see : S \rightarrow P$$

$$action : P^* \rightarrow A$$

## Agentes puramente reactivos (*simple reflex agents*)



# Percepción y acción

Ventajas de la división.

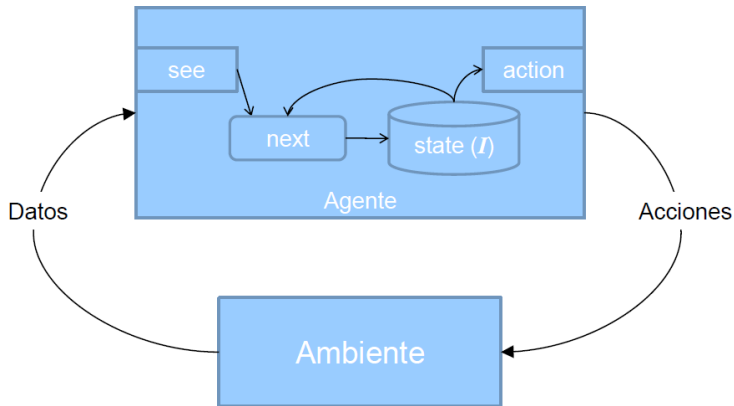
Sea

- $X$  : “La temperatura de la habitación es Ok”.
- $Y$  : “Ventana cerrada”.

$$S = \{\{\neg x, \neg y\}, \{\neg x, y\}, \{x, \neg y\}, \{x, y\}\}$$

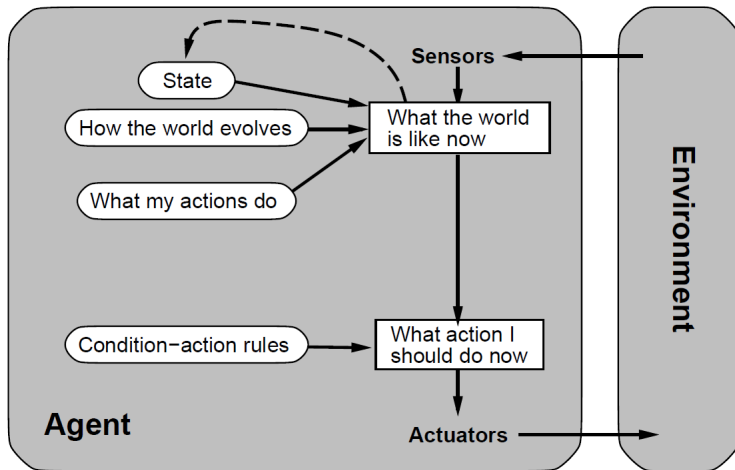
$$see(s) = \begin{cases} p_1 & si\ s = s_1 \\ p_2 & si\ s = s_2 \\ p_3 & si\ s = s_3\ o\ s_4 \end{cases} \quad action(p) = \begin{cases} cerrarvent & p_1 \\ calefacción\ on & p_2 \\ calefacción\ off & p_3 \end{cases}$$

## Agentes con estado (*reflex agents with state*)

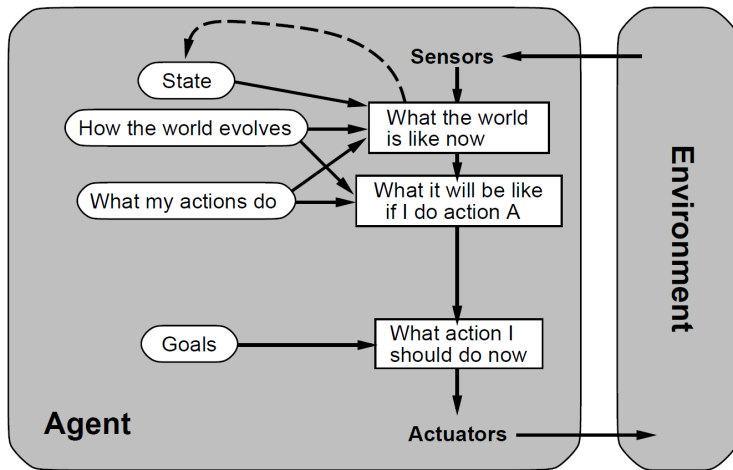


$$see : S \rightarrow P \quad action : I \rightarrow A \quad next : I \times P \rightarrow I$$

## Agentes con estado (*reflex agents with state*)

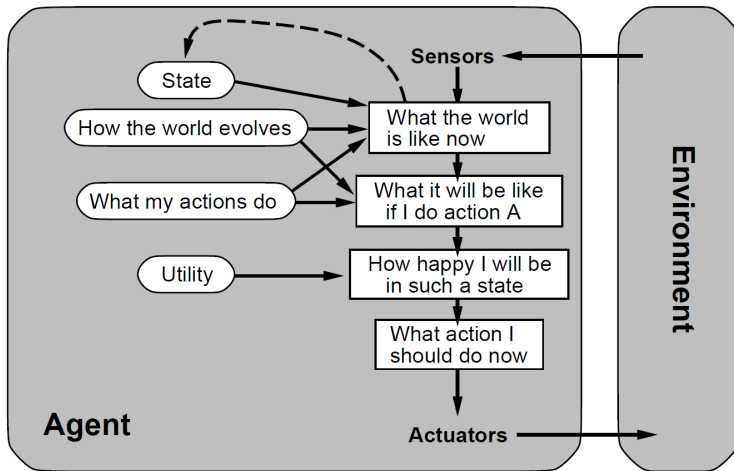


# Agentes basados en objetivos (*goal-based agents*)

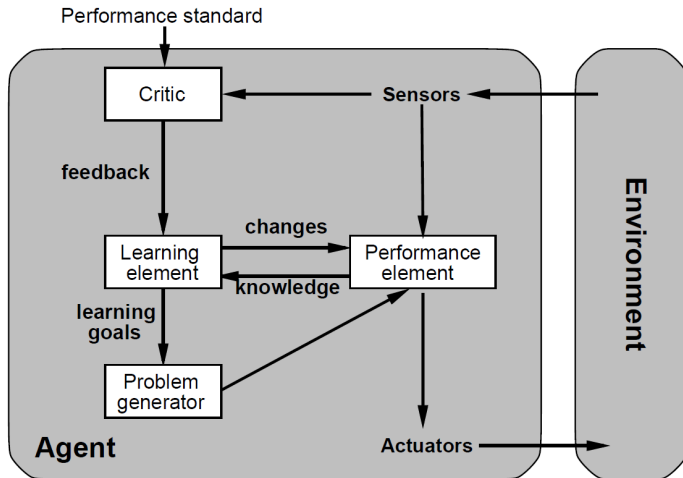




# Agentes basados en la utilidad (*utility-based agents*)



# Agentes de aprendizaje (*learning agents*)



# Resumen

# Resumen

- Los **agentes** interactúan con el **entorno** a través de **actuadores** y **sensores**
- La **función del agente** describe lo que hace el agente en todas las circunstancias
- La **medida de rendimiento** evalúa la secuencia del entorno
- Un agente **perfectamente racional** maximiza el rendimiento esperado
- Los **programas de agente** implementan (algunas) funciones de agente
- Las descripciones de los PEAS definen los entornos de las tareas
- Los entornos se clasifican en varias dimensiones:
  - ¿observables? ¿deterministas? ¿episódicos? ¿estáticos? ¿discretos? ¿un solo agente?
- Existen varias arquitecturas básicas de agentes:
  - reflejo, reflejo con estado, basada en objetivos, basada en la utilidad

# Muchas gracias por su atención

*¿Preguntas?*



**Contacto:** Marco Teran  
**webpage:** [marcoteran.github.io/](http://marcoteran.github.io/)