

SCALAB

Universidad Carlos III de Madrid

Curso 2018-2019

# Robótica

# En este tema

## Robótica

### Introducción

Arquitectura clásica

Comportamiento reactivo

Búsqueda de caminos

Planificación de tareas

Otros aspectos

# Robótica



# Dos ejemplos de autonomía



Sojourner

- ▶ Duración: 30 días
- ▶ Distancia recorrida: aprox. 100 m.
- ▶ Metros/día: 3-10
- ▶ Misión: un instrumento



Opportunity

- ▶ Duración: desde 2004 hasta 2018
- ▶ Distancia recorrida (2008): 7km
- ▶ Metros/día: 3-10
- ▶ Misión: varios instrumentos (150.000 fotos)

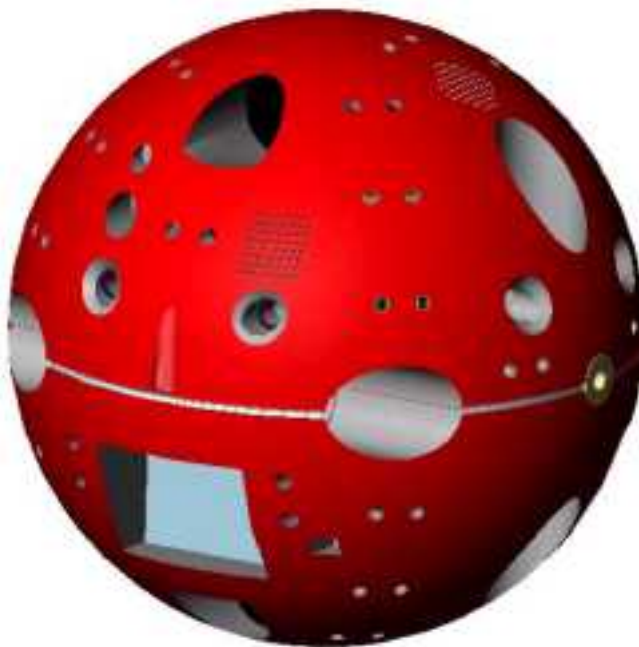
# Autonomía

Tiempo durante el cual  
? Actúa por sí mismo



<http://www.youtube.com/watch?v=4Y2Cf9nRqOw>

# PSA: personal satellite assistant



Propulsores.

Sensors de temperatura  
de sonido

Sistema de navegación.  
Sistema de inteligencia.

<http://www.youtube.com/watch?v=trtMoeNEDtc>

# Vehículos aéreos no tripulados (UAV)



<https://www.youtube.com/watch?v=BUfDhpzMTOc>



# En este tema

## Robótica

Introducción

**Arquitectura clásica**

Comportamiento reactivo

Búsqueda de caminos

Planificación de tareas

Otros aspectos

# Entradas y salidas

Importante para el examen.

## ► Sensores ⇐ Módulos de entrada al sistema.

Distintos tipos:

- Infrarojos
- Láser
- Cámaras
- Bumpers
- Sonido
- Luz, movimiento, escaleras ...

Es muy complejo controlar un robot para que sean autónomos.

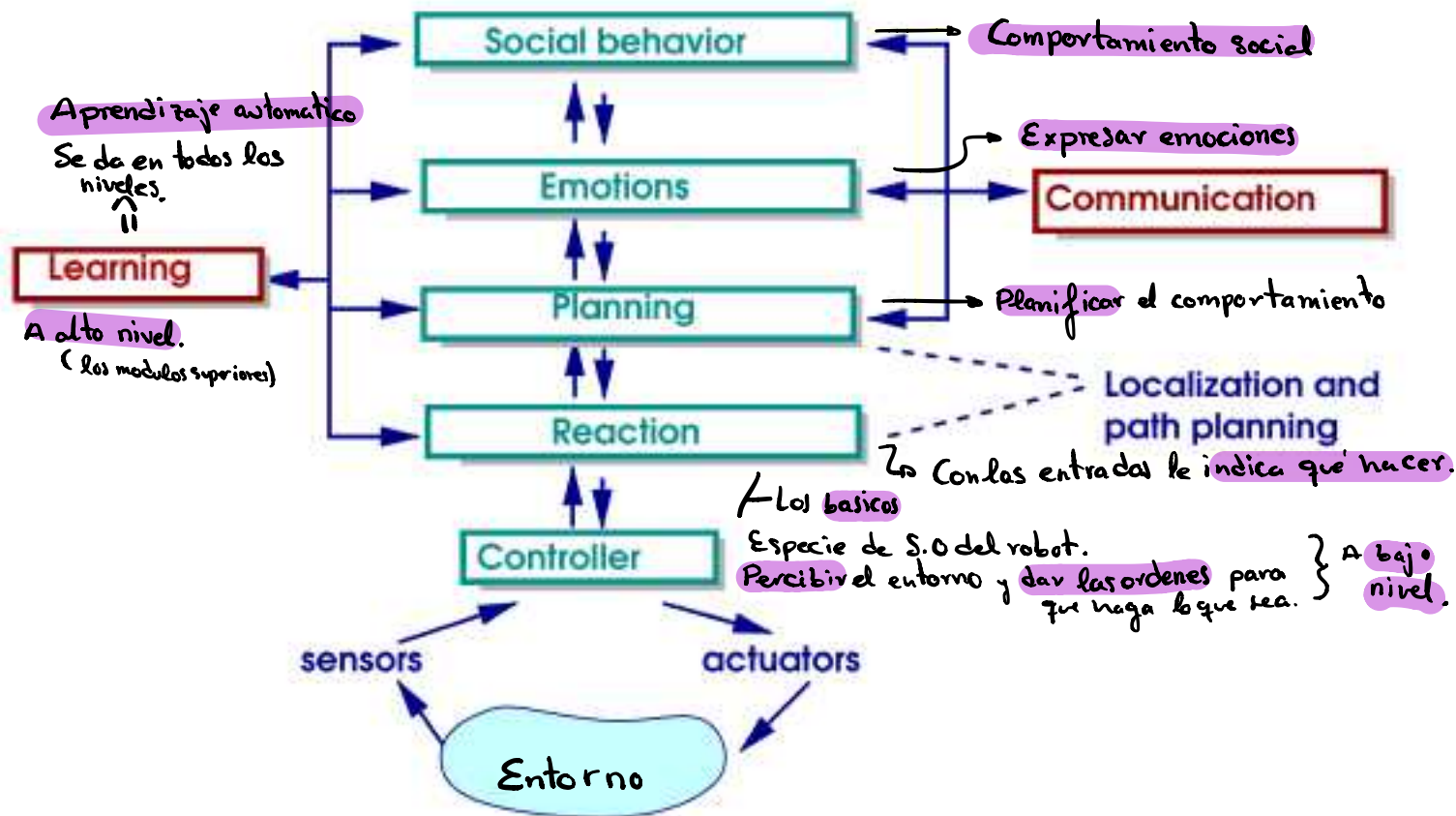
## ► Actuadores ⇐ Módulos de salida

- Velocidad de las ruedas <sup>Ángulo de giro</sup> <sub>? Un coche  
o robot con ruedas...</sub>
- Brazos <sub>→ Recoger cosas</sub>
- Sonido
- Cámara
- Software: email, etc.

# IA en control de robots

Importante para el examen.

Se lee de abajo o arriba.



R.O.S  $\Rightarrow$  Sist. Op. de robots

# En este tema

## Robótica

Introducción

Arquitectura clásica

**Comportamiento reactivo**

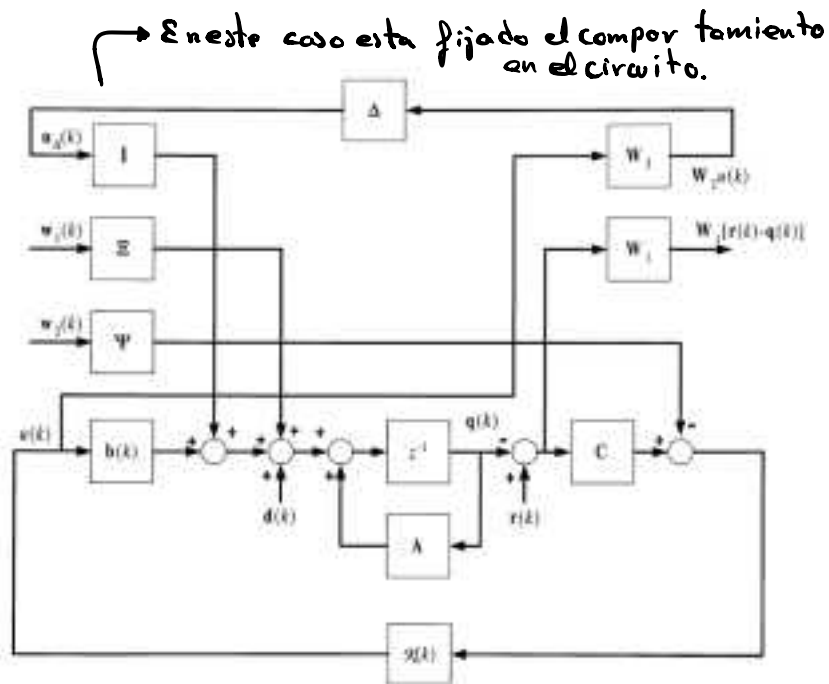
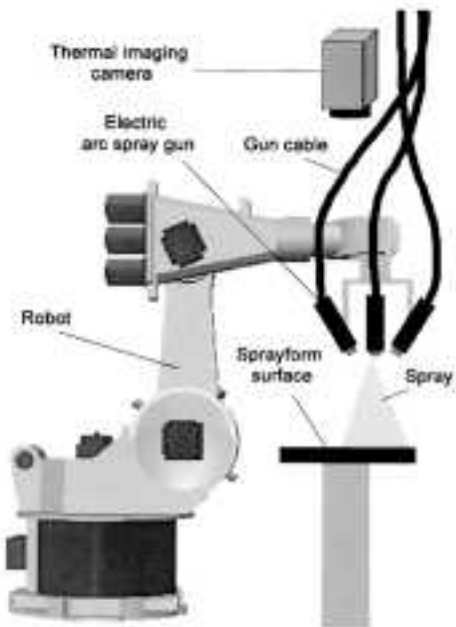
Búsqueda de caminos

Planificación de tareas

Otros aspectos

# Control clásico [Jones et. al. 2003]

Uno de los primeros ejemplos que hubo.



# Programación de comportamientos

## Mover a la derecha

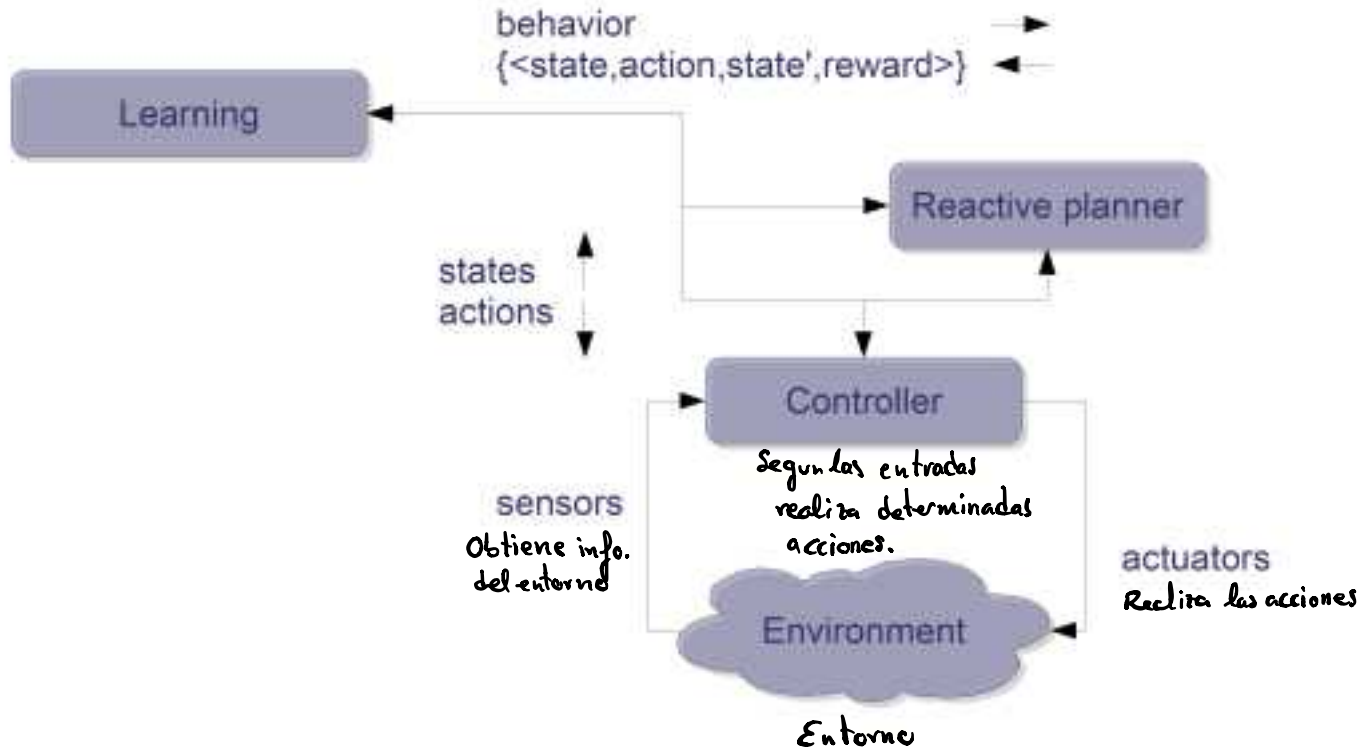
### ► Lógica clásica (reglas)

SI  $sensor[0] < 100$  AND  $\sum_{i=1}^8 sensor[i] > 800$   
THEN  $velocidad[1] = 100, velocidad[2] = 0$

### ► Lógica borrosa

SI el sensor de la izquierda *esta cerca* AND el resto  
*está lejos*  
THEN  $velocidad[1] = alta, velocidad[2] = muybaja$

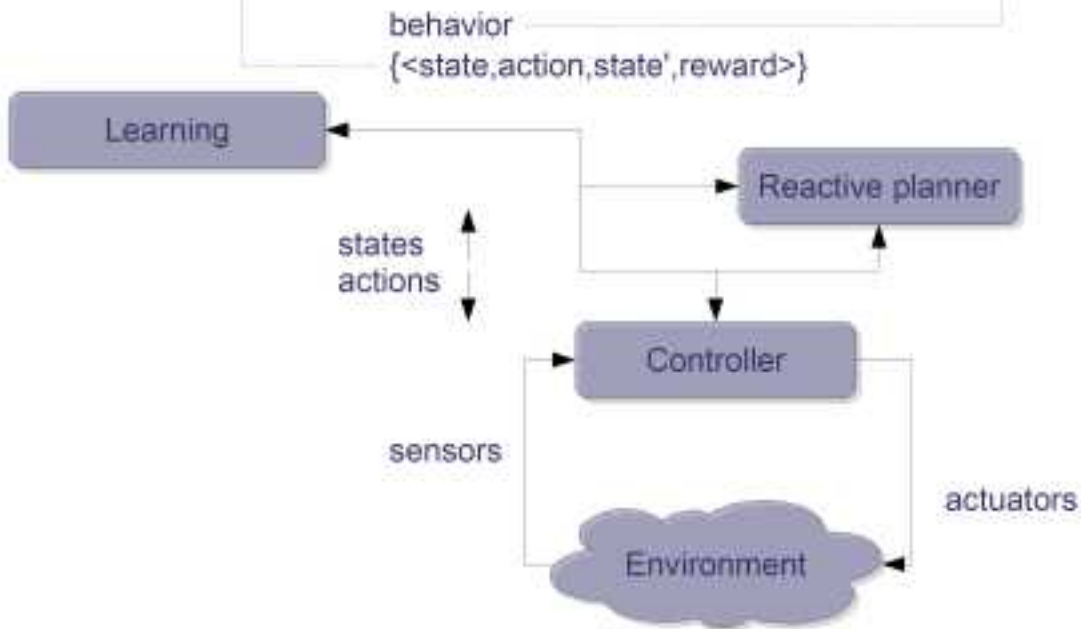
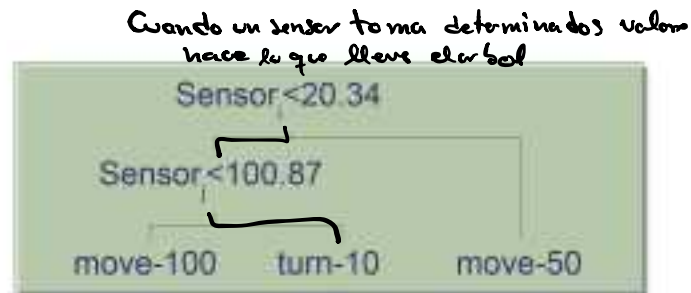
# Aprendizaje de comportamientos



# Aprendizaje a partir de ejemplos

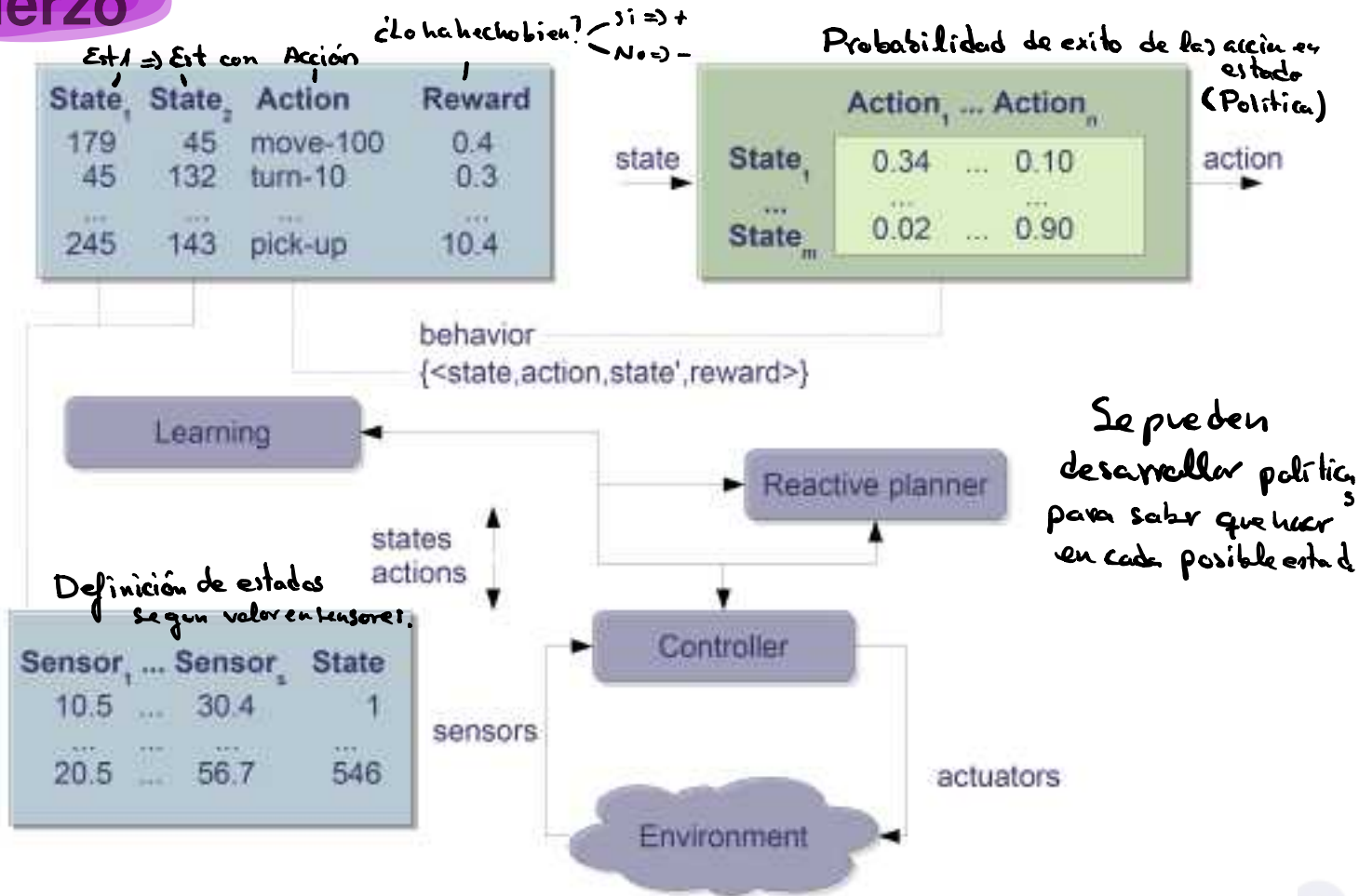
*Cuando tiene estos valores  $\Rightarrow$  Hacer esta acción*

Sensor <sub>1</sub>	...	Sensor <sub>n</sub>	Class
23.45	...	34.23	move-100
10.11	...	33.23	turn-10
...	...	...	...
30.10	...	32.54	pick-object

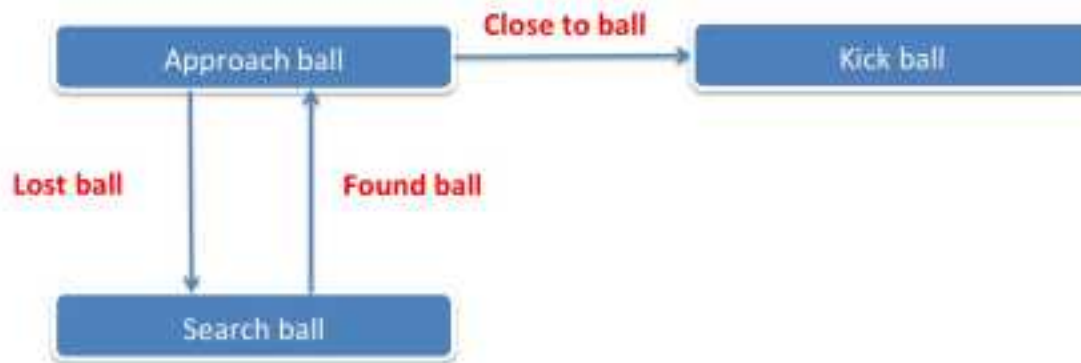




# Aprendizaje a partir de ejemplos. Aprendizaje por refuerzo



# Combinación de comportamientos simples

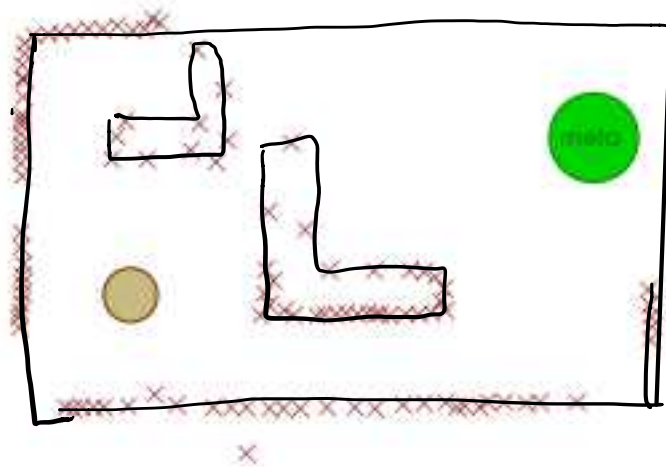


# Descripción jerárquica



# Mapas

- Generación de mapas: preprogramados, aprendidos



Segun los datos de los sensores podemos localizar el robot para saber que acci6n realizar, para ello se genera un mapa con las medidas tomadas

- Localización: GPS, balizas, Markov, Montecarlo, Filtros de Kalman  

$$p(\text{State}=(x,y) \mid \text{Sensors}=(s[0], \dots, s[n]))$$
- Búsqueda de caminos (Path finding)

# En este tema

## Robótica

Introducción

Arquitectura clásica

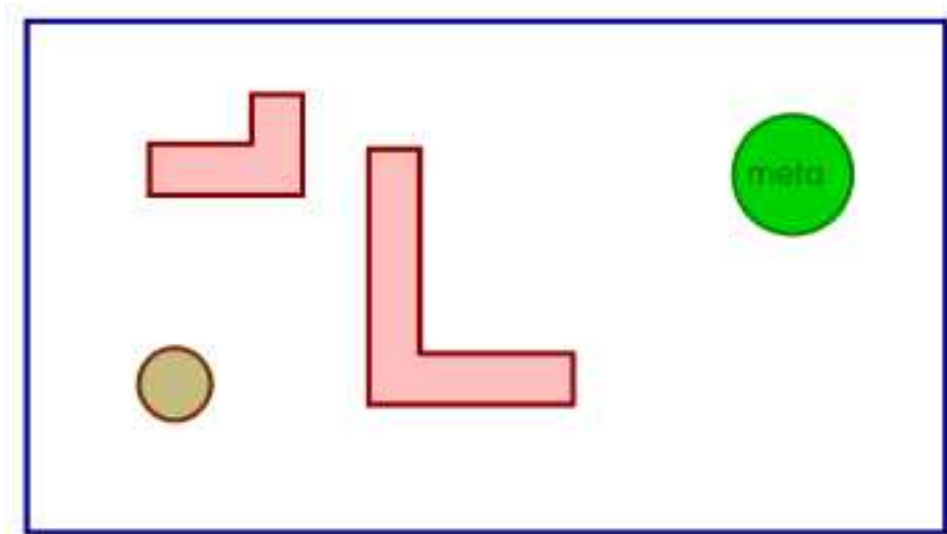
Comportamiento reactivo

**Búsqueda de caminos**

Planificación de tareas

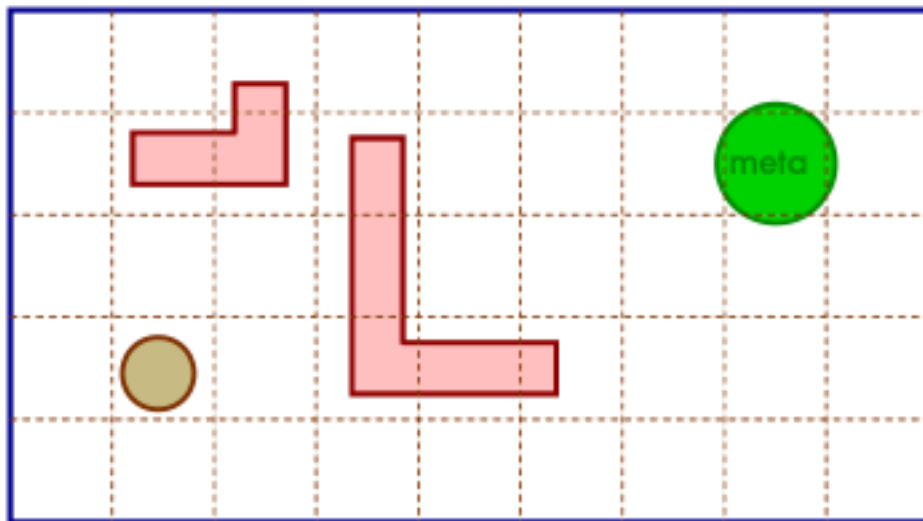
Otros aspectos

# Encontrar el mejor camino

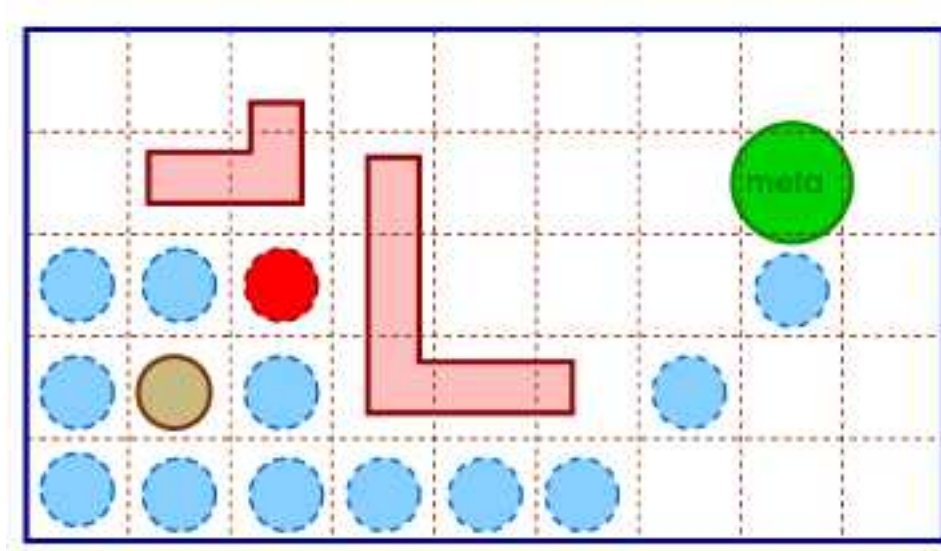


# Encontrar el mejor camino

Discretizamos el problema, lo pasamos a celdas donde hay una meta y nuestra posición. Podemos aplicar algoritmos de búsqueda.



# Búsqueda A\*





# En este tema

## Robótica

Introducción

Arquitectura clásica

Comportamiento reactivo

Búsqueda de caminos

**Planificación de tareas**

Otros aspectos

Se definen los estados y acciones, y se hace planificación y replanificación hasta que hace lo que esperamos.

► Estado

```
at(robot1, room1),  
opened_door(door1, room1, room2),  
closed_door(door2, room1, room3, ...
```

► Acciones

```
abrir, cerrar, mover, ...
```

► Metas

```
at(letters, room3)
```

► Plan

```
mover(robot1, room1, room2, door1), ...
```

► Métricas: tiempo, coste, ...

# Ejemplo en robot humanoide



<https://www.youtube.com/watch?v=tovwPsDm-6c>

[https://www.youtube.com/watch?v=g2NZ\\_EasJv0&v1=de](https://www.youtube.com/watch?v=g2NZ_EasJv0&v1=de)

# En este tema

## Robótica

Introducción

Arquitectura clásica

Comportamiento reactivo

Búsqueda de caminos

Planificación de tareas

Otros aspectos

# Emociones y comportamiento social



<http://www.youtube.com/watch?v=3GkI374ZkM4>

# Entornos multi-robot *(varios robots que se deben coordinar)*

- ▶ Algunos dominios son multi-robot
- ▶ Necesidad de coordinación, negociación, cooperación y comunicación
- ▶ Coordinación
  - ▶ Centralizada: si robot1 cercano a robot2, mover robot2
  - ▶ Distribuida: si muy cerca de otro robot en la derecha mover a la izquierda
- ▶ Comunicación. Necesidad de un lenguaje como KQML
  - ▶ `inform X Y`
  - ▶ `request X Y`
- ▶ Ejemplos
  - `http://www.youtube.com/watch?v=pJVS-9sMiVY`
  - `https://www.youtube.com/watch?v=4ErEBkj\_3PY`
  - `http://www.youtube.com/watch?v=qBUFX41e1mc`

# Ejemplo de robots sociales. Teatro



<https://www.youtube.com/watch?v=cJu1VweV3BQ>

# Aplicaciones



Industry



Medicine



Transport



Surveillance



Exploration, science



Entertainment



# Más robots

► **P3(Honda)**

[http://www.youtube.com/watch?v=7T\\_O9BkkFik](http://www.youtube.com/watch?v=7T_O9BkkFik)

► **Perro**

<http://www.youtube.com/watch?v=cHJJQ0zNNOM>

► **Spirit y Opportunity**

[http://www.youtube.com/watch?v=Rljneh\\_N9WI](http://www.youtube.com/watch?v=Rljneh_N9WI)

► **Boss**

<http://www.youtube.com/watch?v=1ULl63ERek0>

► **Nao**

<http://www.youtube.com/watch?v=2STNYNF4lk>

► **Quadrópteros**

[http://www.youtube.com/watch?v=\\_sUeGC-8dyk](http://www.youtube.com/watch?v=_sUeGC-8dyk)

► **Jugadores de fútbol**

<https://www.youtube.com/watch?v=6BchV1Pk7yc>