



# Desarrollo Basado en Agentes

## Tema 2. Autonomía

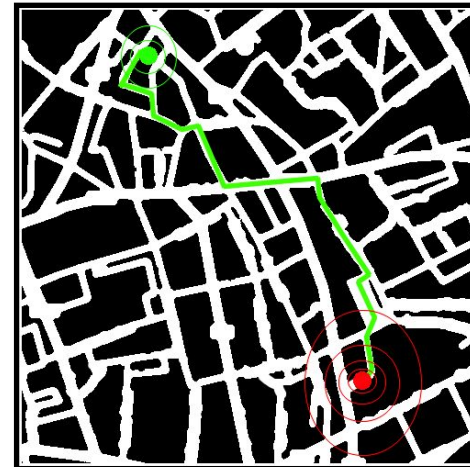
*Luis Castillo Vidal*

*Dpto Ciencias de la Computación e I.A.  
2016-2017*



# Esquema del tema

- Motivación
- Modelo básico de agentes que actúan
- Agentes deductivos
- Agentes de razonamiento práctico
- Agentes reactivos e híbridos



# Motivación

- **Focalizar el agente como individuo**
  - Los agentes deben tener capacidad de actuación autónoma sin que haya que decirles cómo actuar.
  - Poco consenso sobre lo que se considera un agente
- **Elaborar los comportamientos autónomos**
  - Modelos generales e implementables en cualquier plataforma y lenguaje de programación



# Agentes que actúan

- Entorno

- Accesible o Inaccesible
- Determinista o No Determinista
- Endógeno o Exógeno
- Discreto o continuo
- Público o privado

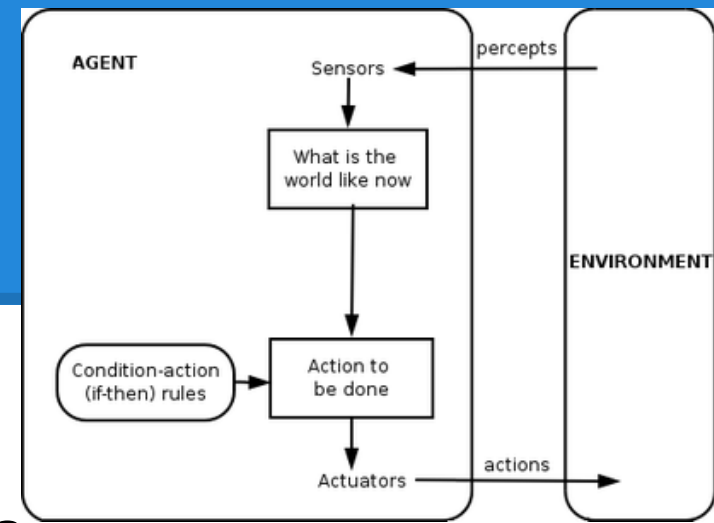
- Agente

- Reactivo
- Proactivo
- Social

- Comportamiento

- Programado
- Lógico

Percepción

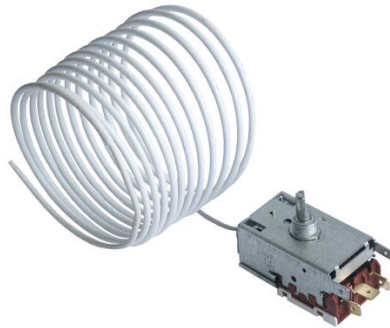


## AGENTE

Un sistema computacional **situado en un entorno** y capaz de **actuar** de forma independiente o **autónoma** para **conseguir sus propios objetivos** (sin que una persona tenga que decirle el cómo), **en representación de otra instancia** (otro agente o una persona o él mismo)

# Agentes que actúan

- Un termostato



- **Entorno** Accesible, No Determinista, Exógeno y Continuo
- **Agente** Reactivo
- **Comportamiento** Programado

# Agentes que actúan

- SIRI



- **Entorno** No Accesible, No Determinista, Exógeno, Discreto
- **Agente** Un poco proactivo
- **Comportamiento** Lógico

# Agentes que actúan

- Google car



- **Entorno** Localmente accesible, No Determinista, Exógeno, Continuo
- **Agente** Muy proactivo
- **Comportamiento** Lógico

# Agentes que actúan

- Entorno del agente
  - $E = \{e, e', e'', \dots\}$   $e$  = estado o entorno (instantánea del mundo que rodea al agente)
- Acciones que puede realizar el agente
  - $Ac = \{a, a', a'', \dots\}$   $a$  = acción
- Una ejecución o traza de un agente
  - $R = \{r, r', r'', \dots\}$  Todas las trazas posibles
  - $r \in R, r = \{ \langle e_0, a_0 \rangle, \langle e_1, a_1 \rangle, \dots, \langle e_n, \emptyset \rangle \}$

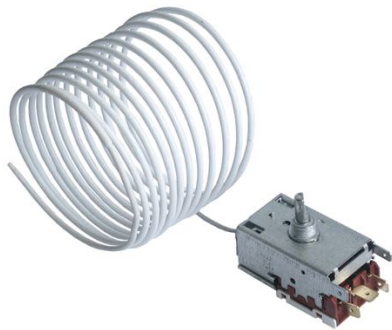


# Agentes que actúan

- Un agente autónomo
  - **Función propia** del agente → determina cómo actuar
  - Def 1.  $Ag : E \rightarrow Ac$  (agente reactivo sin memoria)  
 $Ag(ei) = aj$
- Agentes como **sistemas intencionales situados en un entorno**

# Agentes que actúan

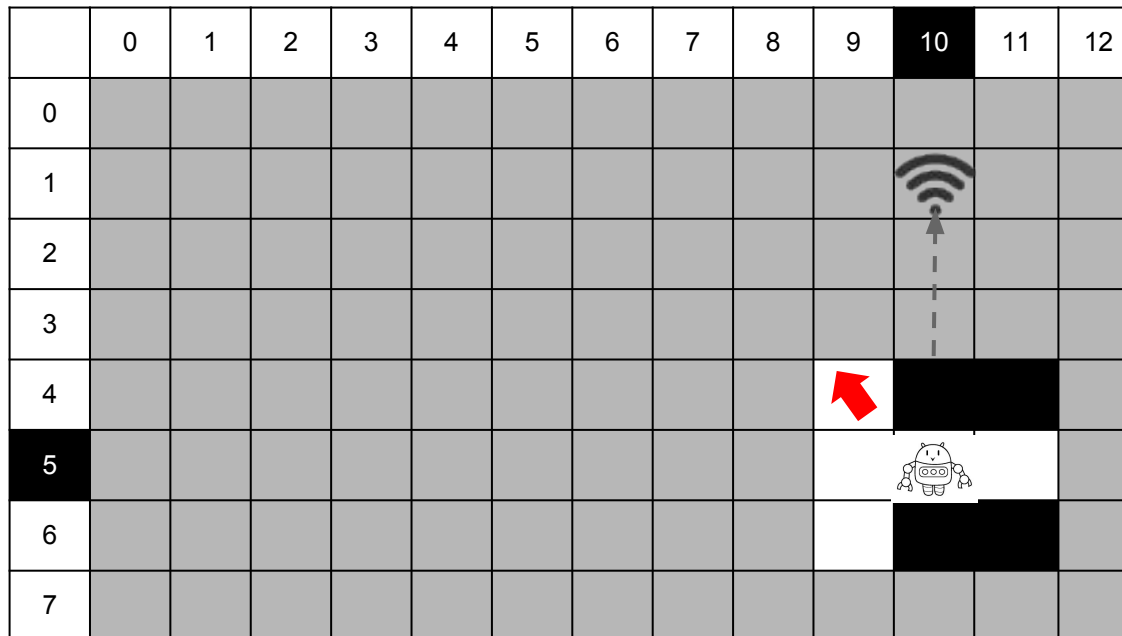
- Un agente autónomo
  - **Función propia** del agente → **determina cómo actuar**
  - Def 1.  $Ag : E \rightarrow Ac$  (agente reactivo sin memoria)



- **Entorno**
  - $e = \langle \text{temperatura} \in \mathbb{R}, \text{consigna} \in \mathbb{R} \rangle$
  - $E = \{ \langle 10, 20 \rangle, \langle 20, 20 \rangle, \langle 25, 20 \rangle, \dots \}$
- **Acciones**
  - $Ac = \{ON, OFF\}$
- **Agente**
  - $Ag(10, 20) = OFF, Ag(20, 10) = ON, Ag(10, 10) = OFF$
  - $Ag(\text{temperatura}, \text{consigna})$ 
    - Si  $\text{temperatura} > \text{consigna} \rightarrow ON$
    - Si  $\text{temperatura} \leq \text{consigna} \rightarrow OFF$
- **Ejecución**
  - $r = \{ \langle \langle 12, 10 \rangle, ON \rangle, \langle \langle 11, 10 \rangle, ON \rangle, \langle \langle 10, 10 \rangle, OFF \rangle, \langle \langle 10, 10 \rangle, OFF \rangle, \langle \langle 11, 10 \rangle, ON \rangle, \langle \langle 10, 10 \rangle, null \rangle \}$

# Agentes que actúan

- Un agente autónomo
  - **Función propia** del agente → determina cómo actuar
  - Def 1.  $Ag : E \rightarrow Ac$  (agente reactivo sin memoria)



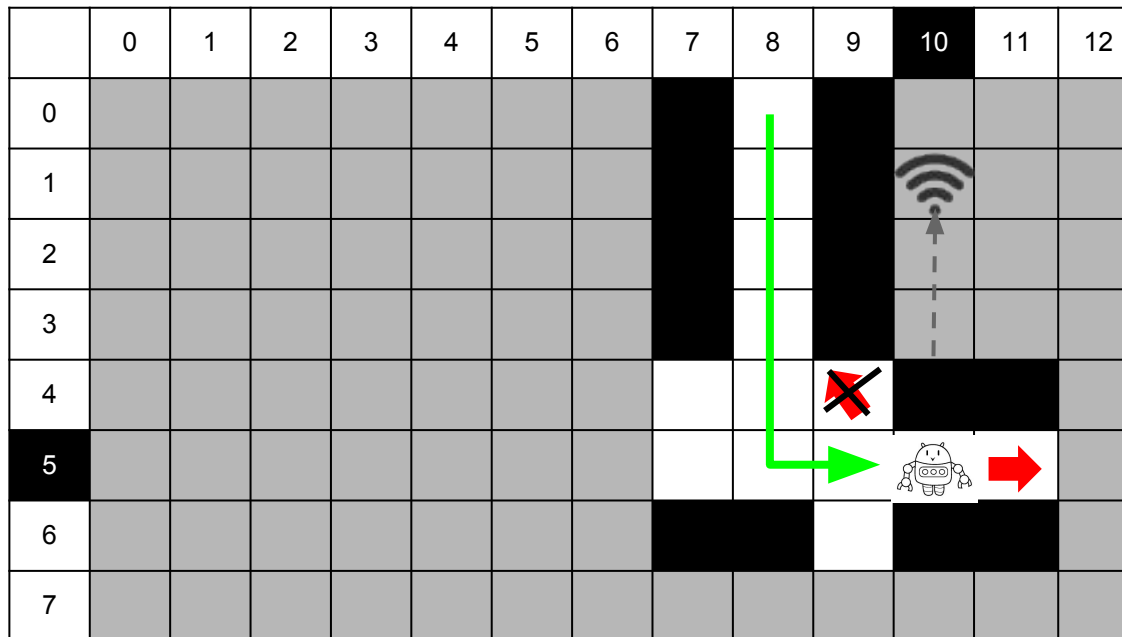
# Agentes que actúan

- Un agente autónomo

- Def 2.  $Ag : R \rightarrow Ac$

(agente reactivo con memoria)

$$Ag(ri) = aj$$



# Agentes que actúan

- Un agente autónomo

- Def 2.  $Ag : R \rightarrow Ac$

$$Ag(ri) = a_j$$

(agente reactivo con memoria)

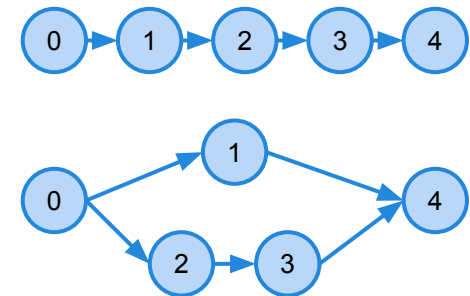
- Def 3.  $Ag : R \rightarrow \{Ac\}^n$

$$Ag(ri) = \{a_0, a_1, a_2, \dots, a_n\}$$

(agente deliberativo con mem.)

$$Ag(ri) = \{a_0, a_1, a_2, a_3, a_4\}$$

$$Ag(ri) = \{a_0, [a_1, \{a_2, a_3\}], a_4\}$$



# Agentes que actúan

Proyecto  
InfoSiadex

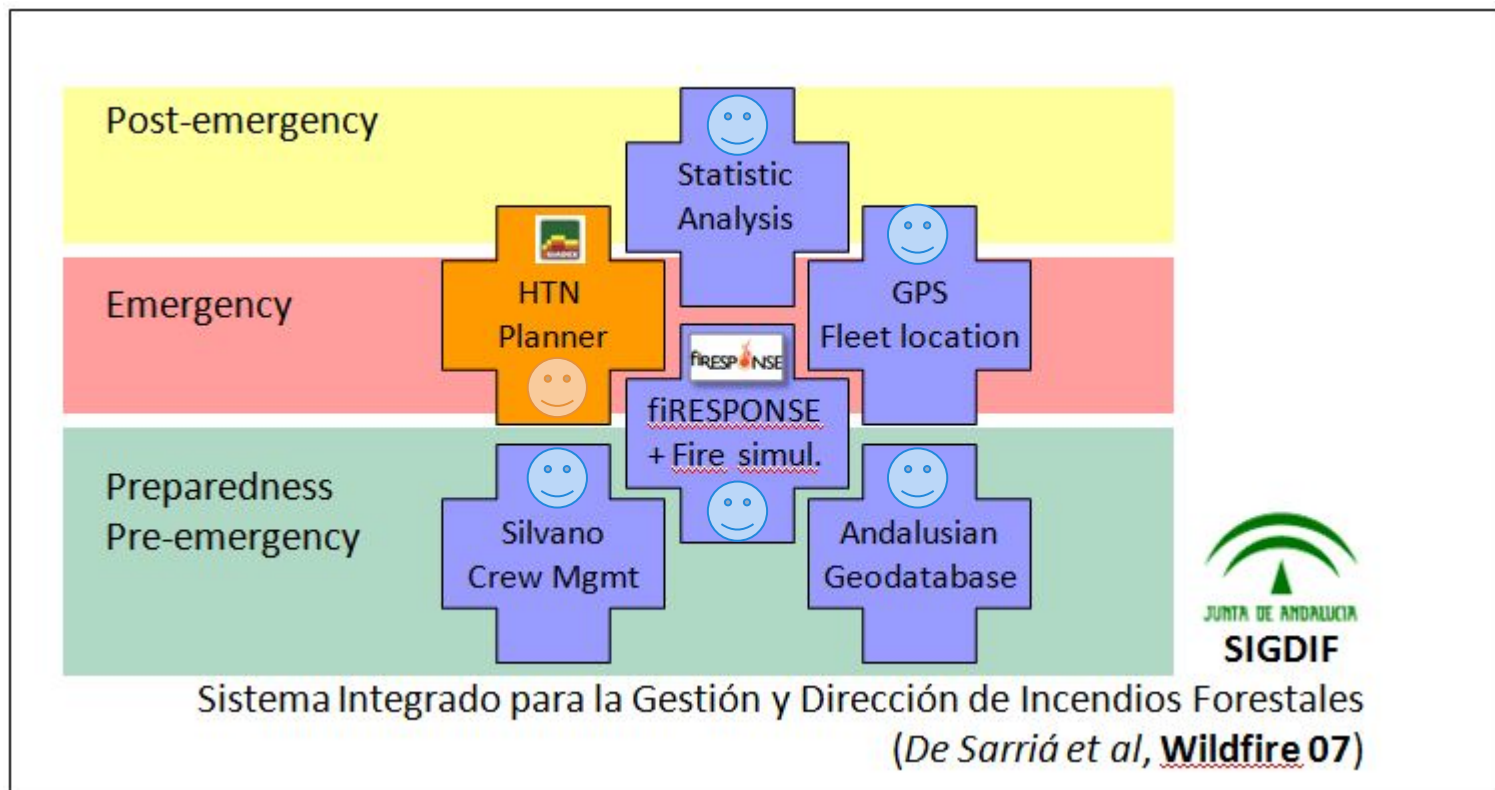
 IAActive  
Intelligent Technologies

[+Info](#)

- Un agente autónomo

- Def 3.  $Ag : R \rightarrow \{Ac\}^n$

(agente deliberativo con mem.)



# Agentes que actúan

Proyecto  
InfoSiadex

 **IAActive**  
Intelligent Technologies

[+Info](#)

- Un agente autónomo

- Def 3.  $Ag : R \rightarrow \{Ac\}^n$

(agente deliberativo con mem.)



- **Entorno**

- Estado de recursos e incendios en Andalucía (ontología de 130+ clases, 2000+ instancias)
  - Eventos **exógenos** (día/noche, operadores, clima)
  - **No determinista**

- **Acciones**

- Cualquier actividad de extinción de incendios, gestión de recursos o administrativa (60+ acciones abstractas, 30+ acciones atómicas, parametrizadas)

- **Agente**

- Un plan de ataque al incendio, completo y temporizado, para las próximas 6 horas (en menos de 10 seg)
  - $Ag(r)$  no es sencilla de definir :-)

- **Ejecución reactiva**

- **Incertidumbre:** Si algo no se ejecuta bien, se rediseña un plan nuevo (Silent-Mode, USPTO)

# Agentes que actúan

Proyecto  
InfoSiadex

 IAActive  
Intelligent Technologies

[+Info](#)

- Un agente autónomo
  - Def 3.  $Ag : R \rightarrow \{Ac\}^n$  (agente deliberativo con mem.)





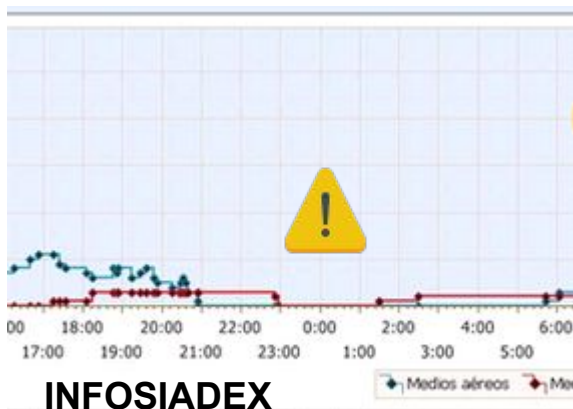
# Agentes que actúan

- El efecto de un agente en su entorno, la **función de transición**
  - $T : R \times Ac \rightarrow E$       Agente/Entorno determinista  
 $T(ri, aj) = ek$
  - $T : R \times Ac \rightarrow \mathbb{P}(E)$       Agente/Entorno no determinista  
 $T(ri, aj) = \{ek1, ek2, ek3\}$   
  
 $T(<moneda\_cruz>, Lanzar\_Moneda) = \{<moneda\_cara>, <moneda\_cruz>\}$
  - Necesario para predecir el efecto de una acción

# Agentes que actúan

- El efecto de un agente en su entorno, la **función de transición**

- $Ag(r) = \emptyset \Rightarrow$
- $\forall a \in Ac, T(r,a) = \emptyset \Rightarrow r$  es GAMEOVER
  - Debido a
    - No hay acciones permitidas
    - La ejecución de la acción no produciría efectos
  - Excepción: eventos exógenos



# Agentes que actúan

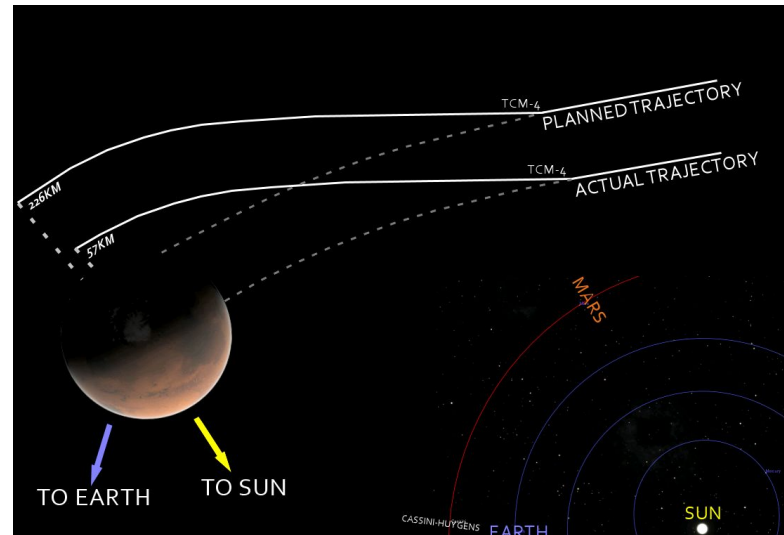
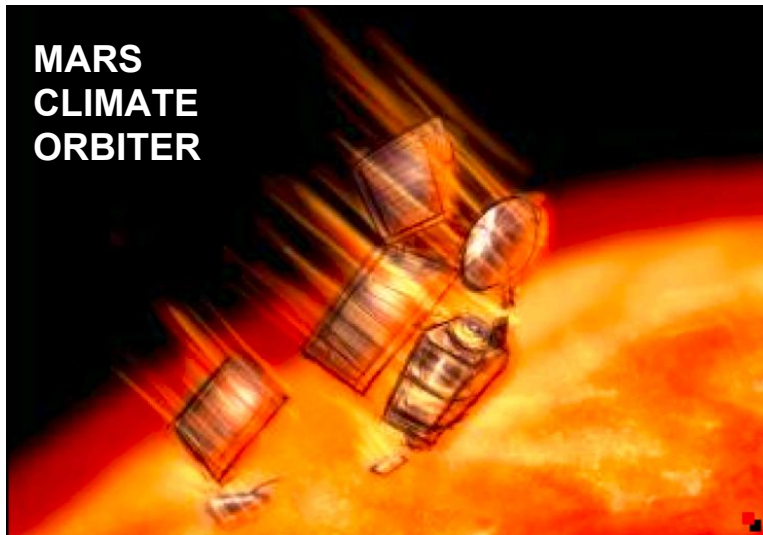


# Agentes que actúan

- Estado interno del agente
  - KB = una base de conocimiento que describe el estado interno del agente
- Modelamos la percepción como una imagen del entorno, no el entorno mismo
  - Problema de la transducción
  - Completa/Incompleta                      Precisa/Imprecisa
  - Cierta/Incierta                              Puntual/Retrasada

# Agentes que actúan

- Estado interno del agente
  - KB = una base de conocimiento que describe el estado interno del agente
- Modelamos la percepción como una imagen del entorno, no el entorno mismo



# Agentes que actúan

- **Función de observación**

- $\text{Obs}: E \rightarrow \text{Per} \subseteq \text{KB}$
- Información de sensores y mensajes recibidos

- **Función propia del agente**

- $R^{\text{KB}} = \{r, r', r'', \dots\}$  Todas las ejecuciones posibles
- $r \in R^{\text{KB}}, r = \{ \langle \text{kb}0, a0 \rangle, \langle \text{kb}1, a1 \rangle, \dots, \langle \text{kb}n, \emptyset \rangle \}$
- Def 4.  $\text{Ag} : R^{\text{KB}} \rightarrow \text{Ac}$  (reactivo)
- Def 5.  $\text{Ag} : R^{\text{KB}} \rightarrow \{\text{Ac}\}^n$  (deliberativo)

- **Función de transición**

- $T : R^{\text{KB}} \times \text{Ac} \rightarrow E$  (determinista)
- $T : R^{\text{KB}} \times \text{Ac} \rightarrow \mathbb{P}(E)$  (no determinista)

# Agentes que actúan


- **¿Es lo mismo?**

- $Ag : R^{KB} \rightarrow Ac \quad r \in R^{KB} = \{ \langle kb_i, a_i \rangle \}$
- $Ag : R \rightarrow Ac \quad r \in R = \{ \langle e_i, a_i \rangle \}$

- Interpretación de las percepciones del agente
- Percepción incompleta del mundo (estado)
- Estados cognitivos internos/privados del agente

# Agentes que actúan

## Un primer algoritmo de **agentes básicos**

1. Inicializar KB // Conocimiento de base
2. Inicializar traza  $r = \emptyset$
3. while (true)
  - a. **Percibir** entorno  $E \rightarrow \text{Per}$
  - b.  $\text{KB} = \text{KB} \cup \text{Per}$  // Añadir percepción a KB
  - c.  $\text{tr} = r \cup \langle \text{KB}, \emptyset \rangle$  // Traza provisional
  - d. Sea  $P = \{a \in A_c / T(\text{tr}, a) \neq \emptyset\}$  // Acciones posibles
  -  e. **Seleccionar**  $a' \in P$  // Clave  $\rightarrow$  **Problema de búsqueda**
  - f. **Ejecutar**  $a$
  - g.  $r = r \cup \{\langle \text{KB}, a \rangle\}$  // Actualizar la traza




# Agentes que actúan

- No todas las acciones son igual de buenas
- **Función de utilidad**
  - $U : E \rightarrow \mathbb{R}$
  - $U : \mathbb{R}^{KB} \times E \rightarrow \mathbb{R}$ 
    - Min? Max? Sum? Prod? Avg?  $\rightarrow$  A determinar
    - No siempre está tan claro qué optimizar
      - Optimización multicriterio ponderada


# Agentes que actúan

## Un algoritmo para agentes optimizadores

1. Inicializar KB // Conocimiento de base
2. Inicializar traza  $r = \emptyset$
3. while (true)
  - a. **Percibir** entorno  $E \rightarrow \text{Per}$
  - b.  $\text{KB} = \text{KB} \cup \text{Per}$  // Añadir percepción a KB
  - c.  $\text{tr} = r \cup \langle \text{KB}, \emptyset \rangle$  // Traza provisional
  - d. Sea  $P = \{a \in A_c / T(\text{tr}, a) \neq \emptyset\}$  // Acciones posibles
  -  e. **Seleccionar**  $a' \in P$ 
    - i. que maximiza  $U(\text{tr}, T(\text{tr}, a))$  // Hillclimbing
  - f. **Ejecutar**  $a$  //  $\text{Ag}(r) = \arg \max_{a/T(r,a) \neq \emptyset} (U(r, T(r, a)))$
  - g.  $r = r \cup \{\langle \text{KB}, a \rangle\}$  // Actualizar la traza

# Agentes que actúan

## Un algoritmo para agentes deliberativos

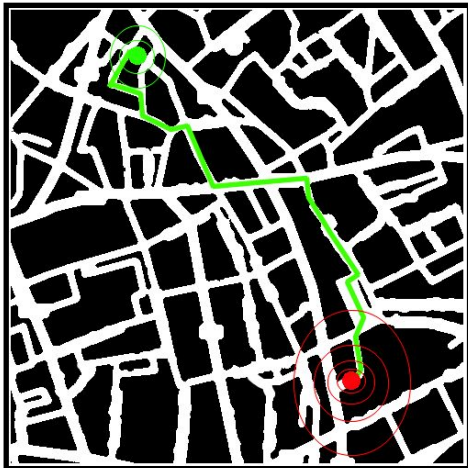
1. Inicializar KB // Conocimiento de base
2. Inicializar traza  $r = \emptyset$
3. Inicializar plan  $\Pi = \emptyset$
4. while (true)
  - a. **Percibir** entorno  $E \rightarrow \text{Per}$  // Sólo se percibe al inicio de la planificación
  - b.  $\text{KB} = \text{KB} \cup \text{Per}$  // Añadir percepción a KB
  - c. while (length( $\Pi$ )  $\leq$  TOPE) // Bucle para construir el plan
    -  i.  $\text{tr} = r \cup \langle \text{KB}, \emptyset \rangle$  // Traza provisional
    - ii. Sea  $P = \{a \in A_c / T(\text{tr}, a) \neq \emptyset\}$  // Acciones posibles
    - iii. **Seleccionar**  $a' \in P$  // Problema de búsqueda global
    - iv. **Añadir** a al plan  $\Pi = \Pi \cup a$
    - v.  $r = r \cup \{\langle \text{KB}, a \rangle\}$
    - vi.  $\text{KB} = \text{KB} \cup T(\text{tr}, a)$  // Simular en KB la ejecución de la acción
  - d. **Ejecutar**  $\Pi$  // Si TOPE==1  $\rightarrow$  Reactivo

# Agentes que actúan

En conclusión, y de forma simplificada, por cada agente

- Definir la **percepción** del entorno  $\text{Obs}: E \rightarrow \text{Per} \subseteq \text{KB}$
- Definir la autonomía de **decisión**
  - Definir la **función propia**  $\text{Ag} : R^{\text{KB}} \rightarrow \text{Ac}$
  - Definir la **función de utilidad**,  $U : R^{\text{KB}} \times E \rightarrow \mathbb{R}$
  - Definir la **función de transición**  $R^{\text{KB}} \times \text{Ac} \rightarrow E$
- Definir la **ejecución** de las acciones
- El algoritmo de control viene a ser el mismo
  - Percibir  $\rightarrow$  Decidir/Razonar  $\rightarrow$  Actuar  $\rightarrow$  Percibir  $\rightarrow$  ...

# Caso de uso: Gugel Car



El problema que se deberá resolver es el diseño e implementación de un **vehículo autónomo** que se desplaza por las **calles** de una ciudad. El vehículo que deberá llegar hasta un **punto de destino** situado en una de las calles, a partir de una señal que le indica al vehículo a la distancia a la que se encuentra el destino teniendo en cuenta que el vehículo podrá encontrar **obstáculos** en el camino que deberá bordear para no chocar con ellos.

# Caso de uso: Gugel Car : Sensores

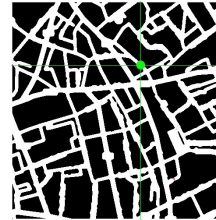


  
Sensor de  
BATERÍA



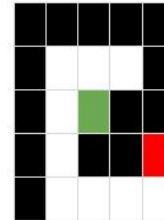
75%

  
Sensor de  
POSICIÓN



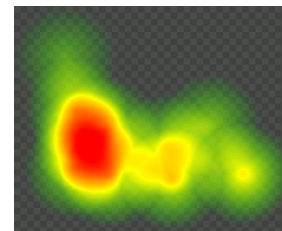
(x=100 , y=25)

  
Sensor de  
OBSTÁCULOS

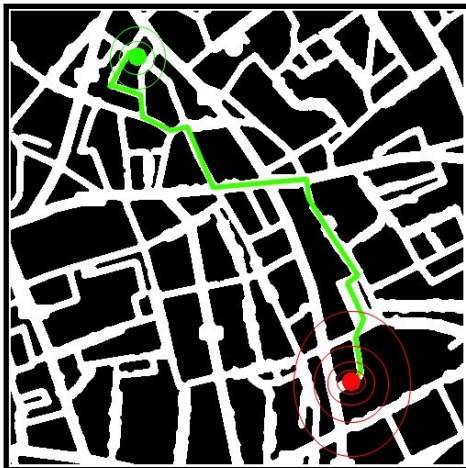


1	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	1	2
1	0	0	0	0

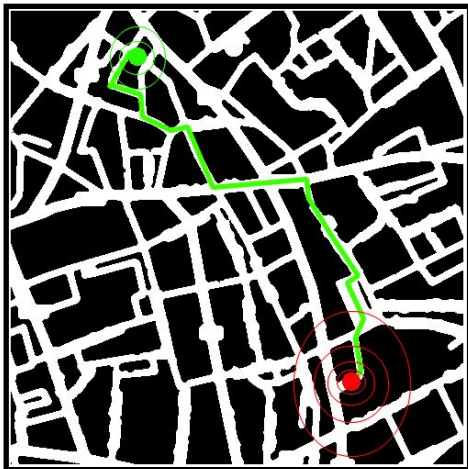
  
Sensor de  
DISTANCIA




67.94	68.62	69.31	70.00	70.71
68.68	69.35	70.03	70.72	71.42
69.42	70.09	70.76	71.44	72.13
70.17	70.83	71.50	72.18	72.86
70.93	71.58	72.24	72.91	73.59

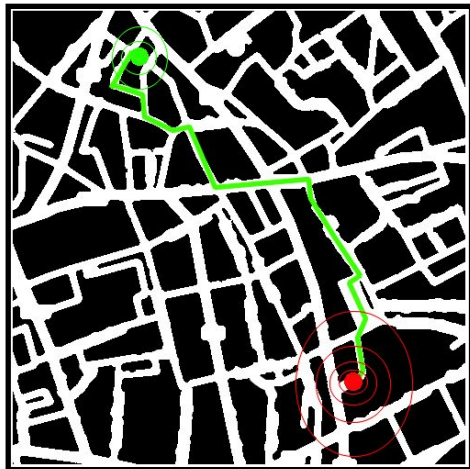


# Caso de uso: Gugel Car : Acciones

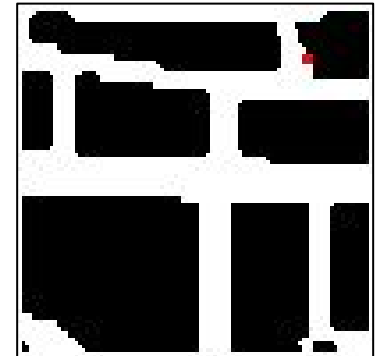


NW	N	NE
W		E
SW	S	SE

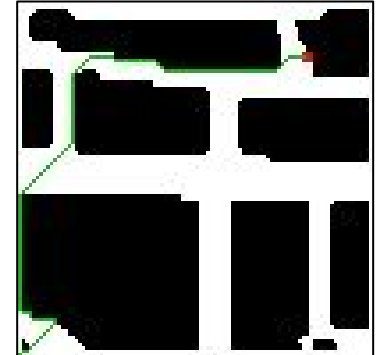
# Caso de uso: Gugel Car : Trazas



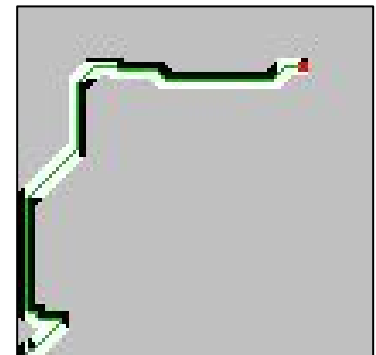
MUNDO Y  
OBJETIVO



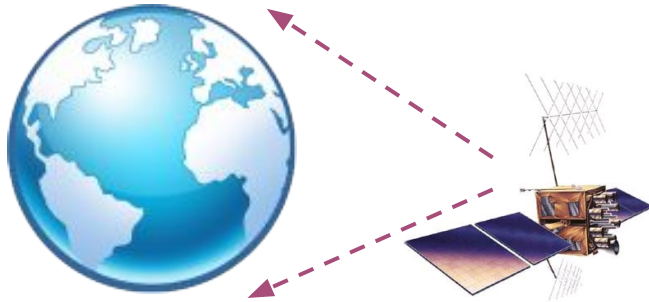
RECORRIDO



TRAZA DE  
PERCEPCIÓN







{“connected”:“YES”, “ready”:“YES”, “gps”:{“x”:10,“y”:5}, “goal”:“No”, “gonio”:{“alpha”:0, “dist”:4.0}, “battery”:100, “radar”:[0,0,0,0,0,0,0,1,1]}

### JSON String

- Connected = “YES”
- Ready = “YES”
- GPS
  - X = 10
  - Y = 5
- Goal = “No”
- Goniometro
  - Alpha = 0°
  - Dist = 4.0
- Battery = 100
- Radar = [0,0,0,0,0,0,0,1,1]



	0	1	2	3	4	5	6	7	8	9	10	11	12
0													
1													
2													
3													
4										0	0		
5										0	(0)	0	
6										0	1	1	
7													

	0	1	2	3	4	5	6	7	8	9	10	11	12
0													
1													
2													
3													
4													
5													
6													
7													

# Agentes que actúan

**Agente::Bot**

$r = \emptyset$

KB = {"connected":"YES", "ready":"YES", "gps":{"x":10,"y":5}, "goal":"No",  
"gonio":{"alpha":0, "dist":4.0}, "battery":100, "radar":[0,0,0,0,0,0,0,1,1]}

$tr = r \cup KB$

$Ac=\{N, E, S, W\}$

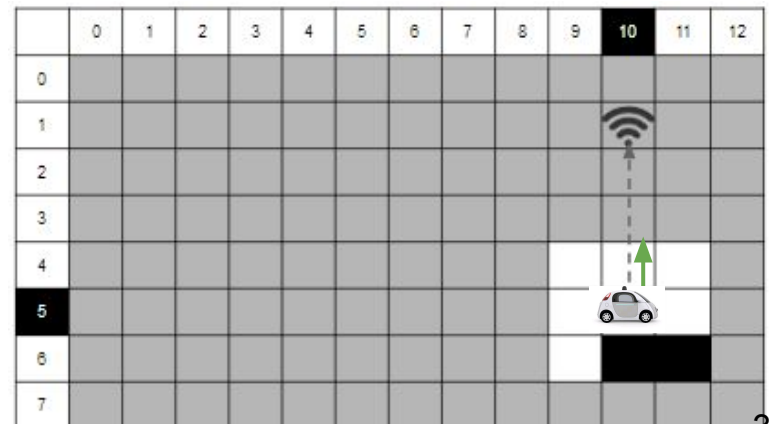
$T(tr, N) = \{ \dots \text{"gps":{"x":10,"y":4}}, \dots \text{"gonio":{"alpha":0, "dist":3.0}} \dots \}$ ,  $U(tr, T(tr,N)) = -3.0$

$T(tr, E) = \{ \dots \text{"gps":{"x":11,"y":5}}, \dots \text{"gonio":{"alpha":0, "dist":5.0}} \dots \}$ ,  $U(tr, T(tr,E)) = -5.0$

$T(tr, S) = \emptyset$

$T(tr, W) = \{ \dots \text{"gps":{"x":9,"y":5}}, \dots \text{"gonio":{"alpha":0, "dist":5.0}} \dots \}$ ,  $U(tr, T(tr,W)) = -5.0$

**$Ag(tr) = \arg \max(-3.0, -5.0, -5.0) = N$**



# Agentes que actuan

**Agente::Drone**

$r = \emptyset$

KB = {"connected": "YES", "ready": "YES", "gps": {"x": 10, "y": 5}, "goal": "No",  
"gonio": {"alpha": 0, "dist": 4.0}, "battery": 100, "radar": [0, 0, 0, 0, 0, 0, 0, 1, 1]}

$tr = r \cup KB$

$Ac = \{N, E, S, W\}$

$T(tr, N) = \{ \dots \text{"gps": \{"x": 10, "y": 4\}, \dots \text{"gonio": \{\dots, "dist": 3.0\} \dots} \}, U(tr, T(tr, N)) = -3.0$

$T(tr, E) = \{ \dots \text{"gps": \{"x": 11, "y": 5\}, \dots \text{"gonio": \{\dots, "dist": 5.0\} \dots} \}, U(tr, T(tr, E)) = -5.0$

$T(tr, S) = \emptyset$

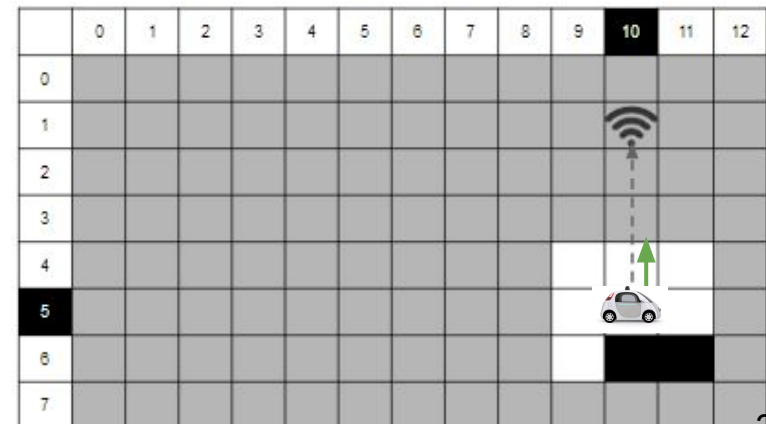
$T(tr, W) = \{ \dots \text{"gps": \{"x": 9, "y": 5\}, \dots \text{"gonio": \{\dots, "dist": 5.0\} \dots} \}, U(tr, T(tr, W)) = -5.0$

$Ag(tr) = \arg \max(-3.0, -5.0, -5.0) = N$

**Beliefs**

**Desire**  
 $U(r, T(r, a)) == 0$

**Intention**



# Agentes que actuan

**Agente::Drone**

$r = \emptyset$

KB = {"connected":"YES", "ready":"YES", "gps":{"x":10,"y":5}, "goal":"No",  
"gonio":{"alpha":0, "dist":4.0}, "battery":100, "radar":[0,0,0,0,0,0,0,1,1]}

$tr = r \cup KB$

$Ac = \{N, E, S, W\}$

**Capabilities**

$T(tr, N) = \{ \dots \text{"gps"}:\{\text{"x"}:10, \text{"y"}:4\}, \dots \text{"gonio"}:\{\dots, \text{"dist"}:3.0\} \dots \}, U(tr, T(tr, N)) = -3.0$

$T(tr, E) = \{ \dots \text{"gps"}:\{\text{"x"}:11, \text{"y"}:5\}, \dots \text{"gonio"}:\{\dots, \text{"dist"}:5.0\} \dots \}, U(tr, T(tr, E)) = -5.0$

$T(tr, S) = \emptyset$

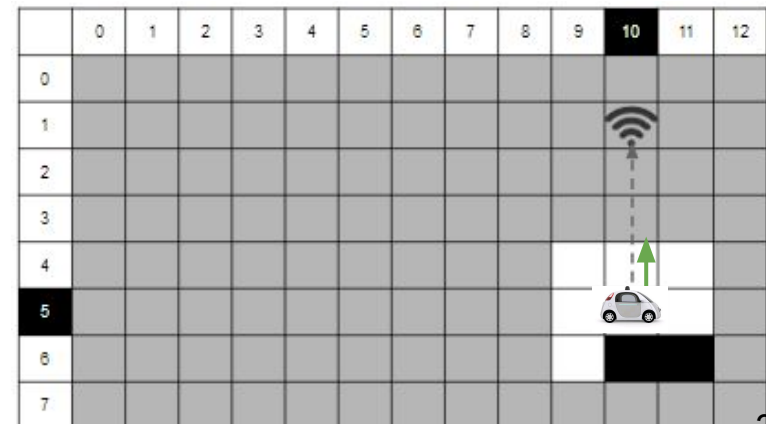
$T(tr, W) = \{ \dots \text{"gps"}:\{\text{"x"}:9, \text{"y"}:5\}, \dots \text{"gonio"}:\{\dots, \text{"dist"}:5.0\} \dots \}, U(tr, T(tr, W)) = -5.0$

$Ag(tr) = \arg \max(-3.0, -5.0, -5.0) = N$

**Obligation or Commitment**

**Beliefs or Mental state**

**Decision**



# Agentes que actúan

- Programado v.s. lógico

```
int x=10, y=5;  
if (accion.equals("NORTE"))  
    y--;
```

Lógica en el código

```
posicion_x(10, s1) ∧ posicion_y(5, s1)  
  
//Axioma Mover Norte, Cálculo Situacional  
posicion_y(Y, S) → posicion_y(Y-1, S+1)  
  
// Modus Ponens  
{5/Y, s1/S}  $\frac{P \rightarrow Q, P}{\therefore Q}$   
  
posicion_y(4, s2)
```

Lógica en la KB

- ¿Ventajas e inconvenientes?

# Agentes que actúan

- KB Como una teoría lógica → **Estado mental o cognitivo** del agente
  - Hechos conocidos (creencias)
  - KB con incertidumbre
    - Probabilidad
    - Posibilidad
  - Mecanismos de inferencia
    - Deducción
    - Abducción
    - Representación de acciones
      - Precondiciones y efectos