



# TECNOLOGÍA Y ARQUITECTURA ROBÓTICA

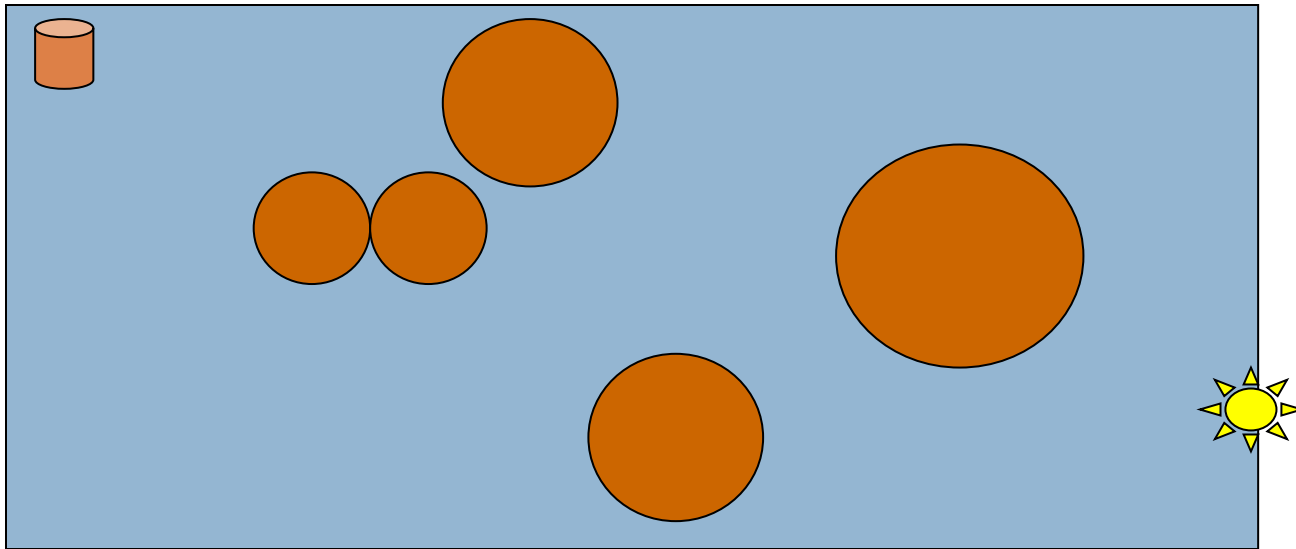
## Sesión 8

- **Arquitecturas deliberativas y reactivas**



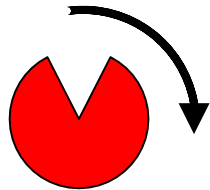
# Arquitecturas de robots

- Organización de la generación de acciones a partir de las percepciones del robot
- Estrategias que permiten el **control** del robot
- Ejemplo: ir al destino evitando los obstáculos

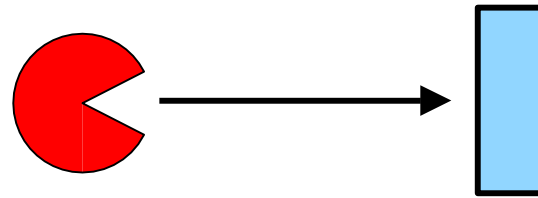


# Comportamientos

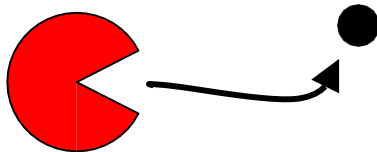
- Los comportamientos son las primitivas básicas de los sistemas de control de robots.
- Deben estar bien definidos, ser autocontenidos y ser independientes



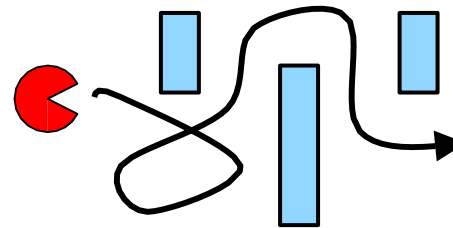
Girar 90° a la derecha



Ir hacia adelante hasta encontrar obstáculo



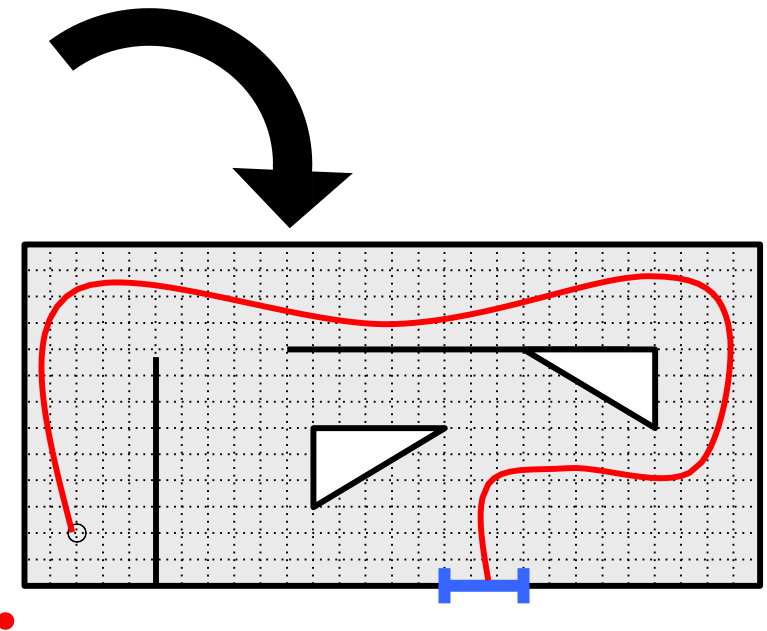
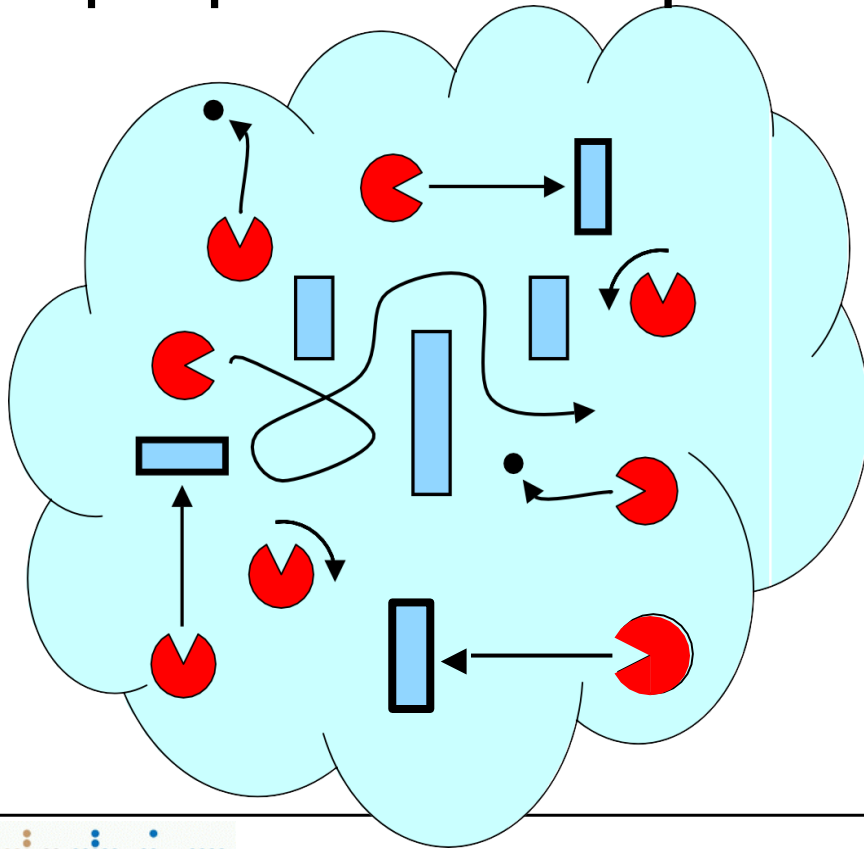
Coger la pelota



Explorar el terreno

# Comportamientos

- El objetivo fundamental es crear comportamientos que permitan cumplir la misión deseada.



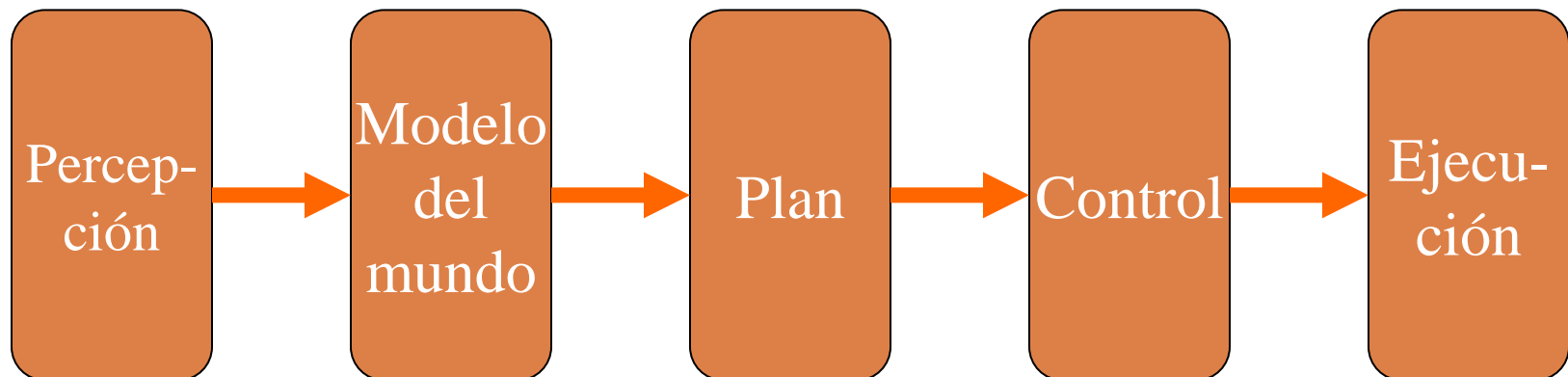
# Escala temporal en una arquitectura

- ¿A qué velocidad reacciona una arquitectura de control?  
¿Tiene en cuenta acciones futuras?
- Arquitecturas deliberativas
  - ▣ Tienen en cuenta el futuro (plan) antes de ejecutar las acciones  $\Rightarrow$  escala temporal a largo plazo
- Arquitecturas reactivas
  - ▣ No consideran acciones futuras, simplemente reaccionan  $\Rightarrow$  escala temporal a corto plazo
- Arquitecturas híbridas
  - ▣ Tienen en cuenta el futuro (capa deliberativa) pero también son capaces de reaccionar con rapidez (capa reactiva)

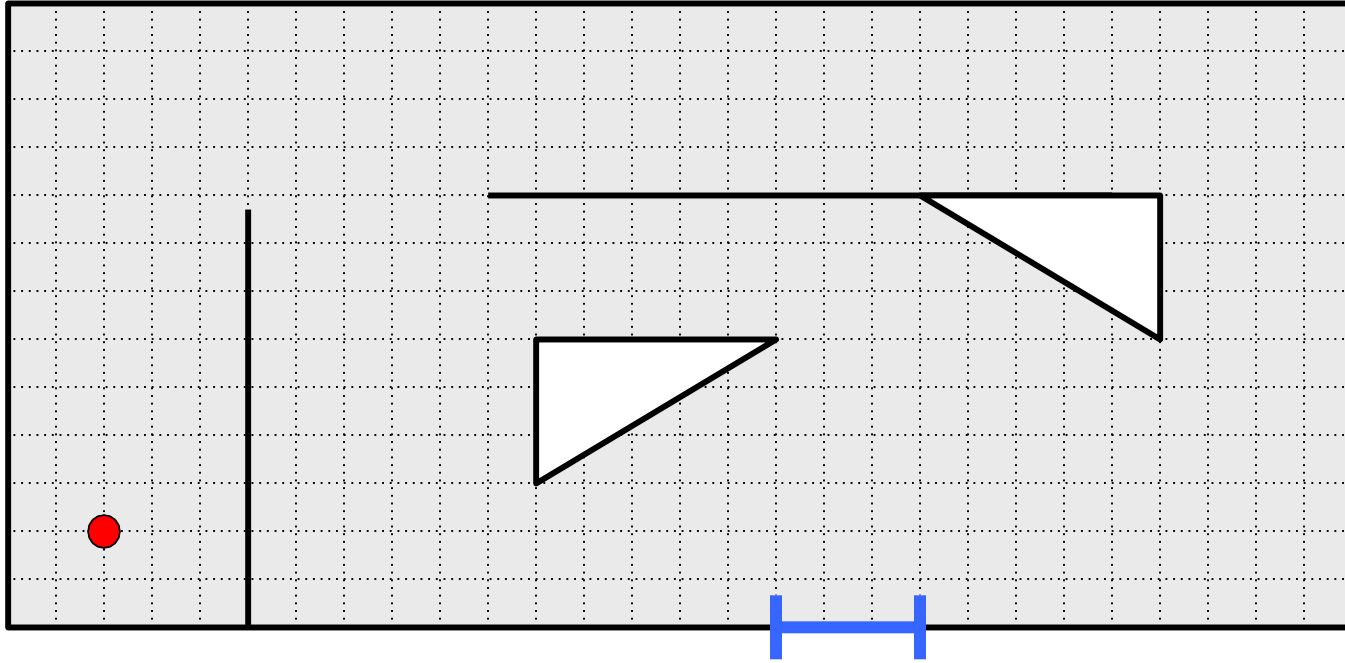


# Arquitecturas deliberativas

- Basada en el paradigma de *sentir-planificar-actuar* (SPA):
  - ▣ Se tiene un modelo completo (mapa) del entorno
  - ▣ Se construye un plan de acción basado en el mapa para realizar la tarea
  - ▣ Se ejecuta el plan

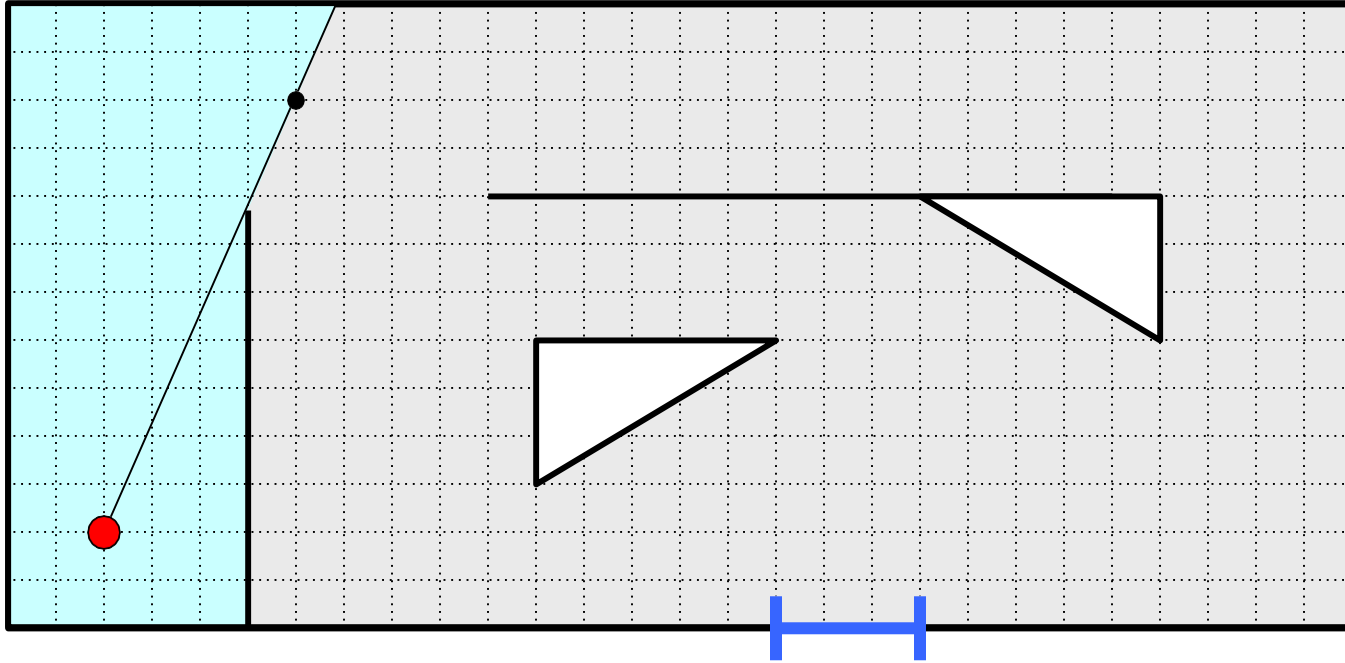


# Ejemplo de planificación en arquitectura deliberativa



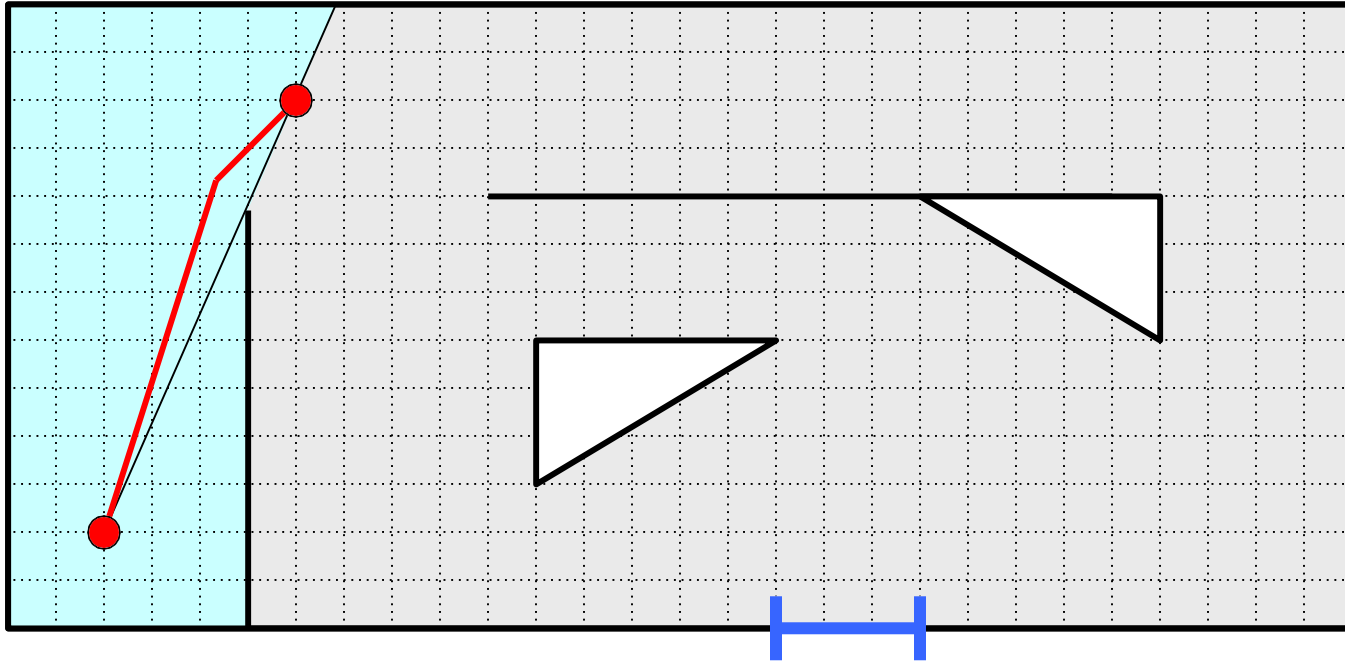


# Ejemplo de planificación en arquitectura deliberativa



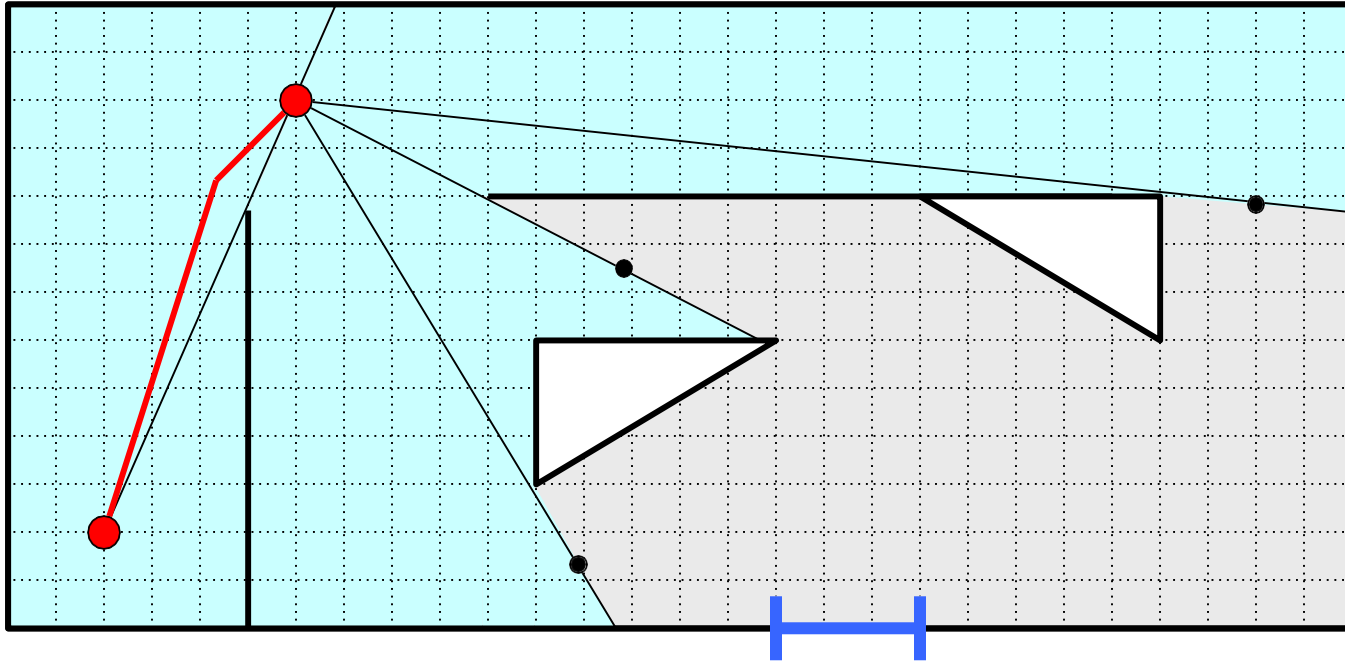
El robot utiliza sensores para crear un mapa local del mundo e identificar áreas sin explorar

# Ejemplo de planificación en arquitectura deliberativa



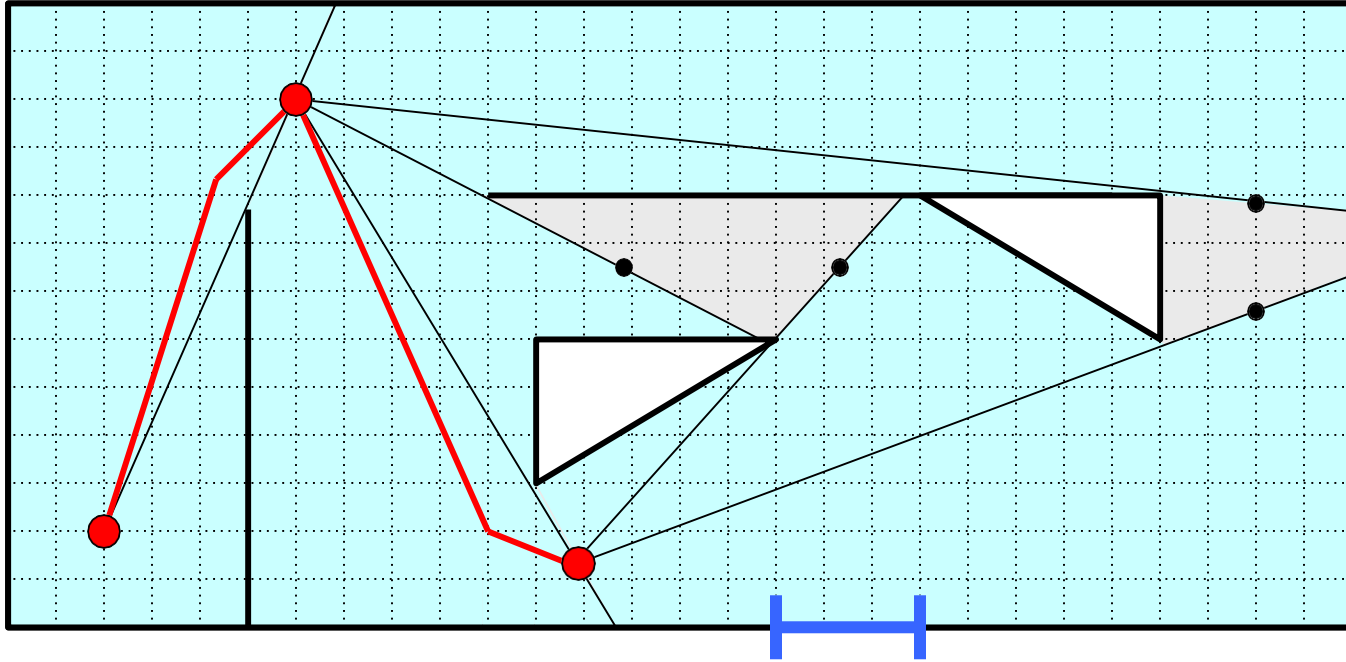
El robot se mueve al punto medio del del área sin explorar

# Ejemplo de planificación en arquitectura deliberativa



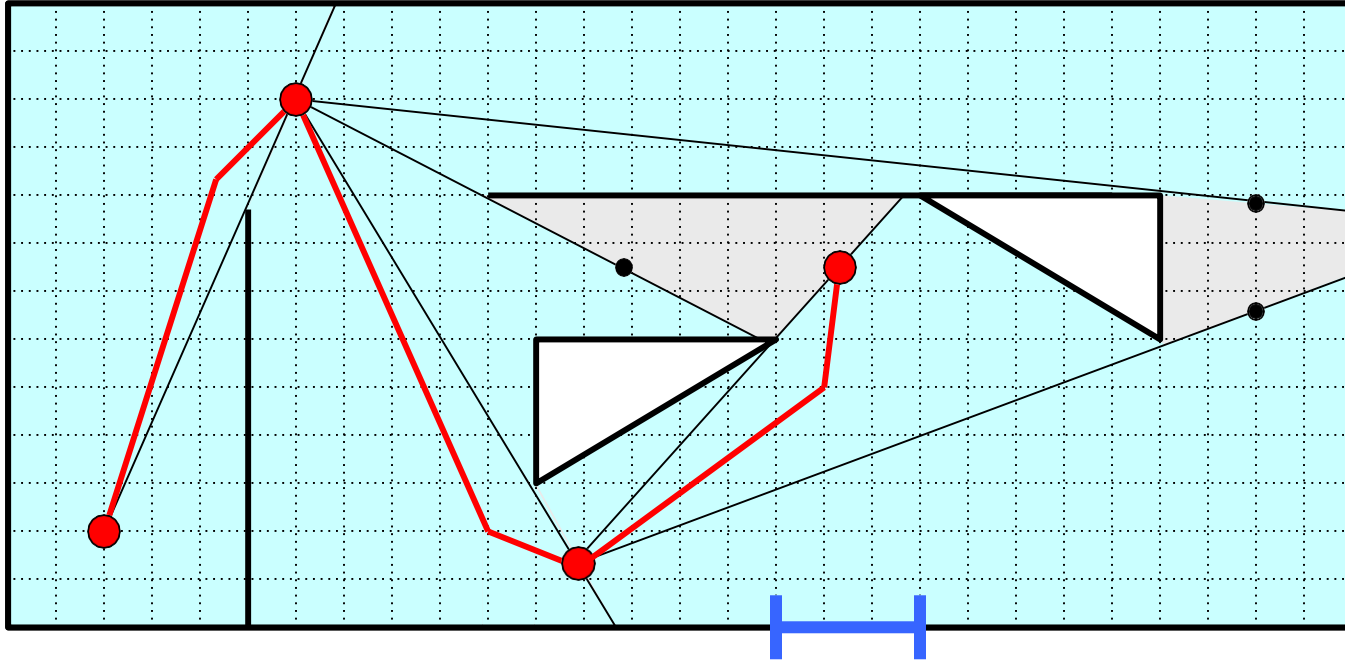
El robot realiza un segundo scan de los sensores y coteja los nuevos datos con el mapa global.

# Ejemplo de planificación en arquitectura deliberativa



El robot sigue explorando

# Ejemplo de planificación en arquitectura deliberativa



El robot debe reconocer cuándo está frente a áreas que ya ha explorado.

# Arquitecturas deliberativas:

## Conclusiones

### □ Ventajas:

- El tener un modelo del entorno permite optimizar las acciones para obtener el “mejor” plan

### □ Limitaciones:

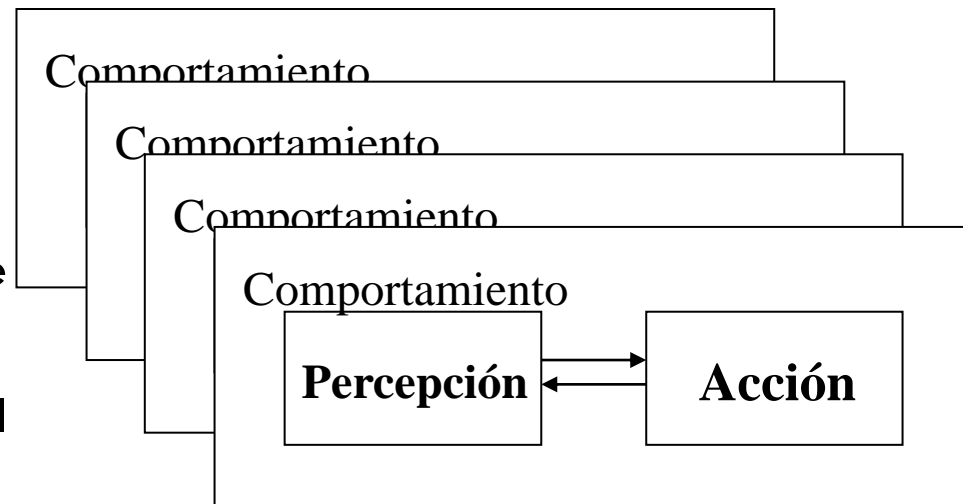
- Necesidad de un modelo preciso del entorno
- Altos requerimientos de cómputo y memoria
- Dificultad de operar en un mundo dinámico o desconocido
- Reacción “lenta” a situaciones imprevistas

# Arquitecturas Reactivas

- En el enfoque reactivo hay una *conexión directa* de percepción a acción sin necesidad de un modelo del mundo
- Normalmente se considera una serie de **comportamientos** que realizan diferentes tareas en forma “paralela”
- Los sistemas puramente reactivos no usan ninguna representación interna del entorno y no tienen en cuenta acciones futuras
- Emplean muy poca (o ninguna) información de estados

# Reglas

- Las arquitecturas reactivas consisten en una colección de reglas reactivas, que a partir de situaciones específicas generan acciones específicas.
- Similar a los reflejos
  - ▣ Al no intervenir el “cerebro”, los reflejos actúan muy rápido
- Las reglas se ejecutan concurrentemente y en paralelo
- Las posibles situaciones se extraen directamente de la información de los sensores
- Las acciones son las respuestas del sistema (comportamientos)





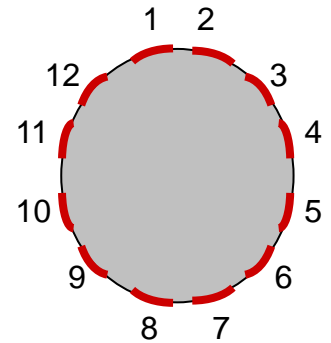
# Mapas incompletos

- En general, no se usan mapas del entorno completos en los sistemas reactivos
- Las situaciones más importantes hacen que se activen las reacciones más apropiadas
- Se usan respuestas por defecto para cubrir todos los casos posibles
- **Ejemplo:** controlador reactivo para navegación segura
  - ▣ Si el sensor de choque izquierdo activo → girar a la derecha
  - ▣ Si el sensor de choque derecho activo → girar a la izquierda
  - ▣ Si ambos activos → retroceder y girar a la izquierda
  - ▣ En otro caso, seguir la trayectoria



# Ejemplo – Navegación segura

- Un robot con 12 sensores s3nar, alrededor del robot
- Dividimos los rangos s3nares en dos zonas
  - ▣ Zona de peligro: cosas demasiado cercanas
  - ▣ Zona segura: distancia razonable a los objetos



**Si**  $\text{m3nimo}(\text{sonares } 1, 2, 3, 12) < \text{zona de peligro}$  **Y** en movimiento  $\rightarrow$  stop

**Si**  $\text{m3nimo}(\text{sonares } 1, 2, 3, 12) < \text{zona de peligro}$  **Y** parado  $\rightarrow$  retroceder

**Si no**

moverse hacia adelante

# Ejemplo – Navegación segura

- En entornos dinámicos, añadimos otra capa

**Si** sonar 11 o 12 < zona de peligro **Y**

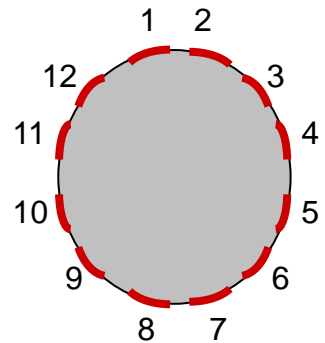
sonar 1 o 2 < zona de peligro

**entonces** girar a la derecha

**Si** sonar 3 o 4 < zona de peligro

**entonces** girar a la izquierda

- El robot se aleja de los obstáculos antes de tenerlos demasiado cerca.
- Combinando ambos controladores podemos obtener un comportamiento para que el robot se mueva sin colisión
- Debemos tener en cuenta que hay condiciones mutuamente exclusivas



# Selección de acciones

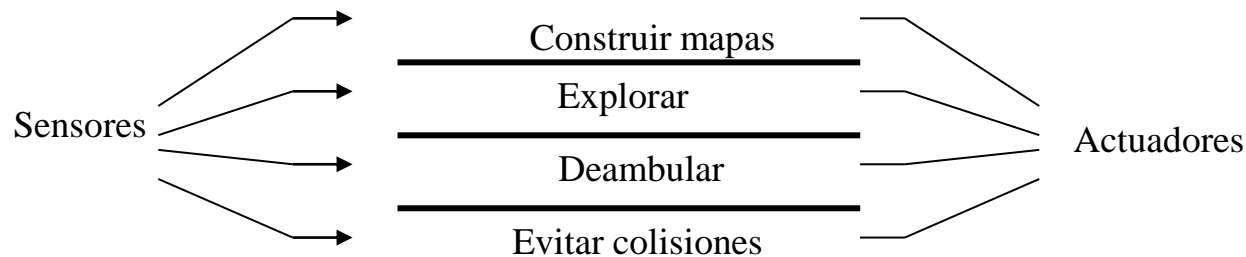
- En la mayoría de los casos las normas que rigen los comportamientos son activadas por más de una regla al mismo tiempo
  - ▣ Se envían dos o más comandos a los actuadores
- Decidir qué acción realizar implica una selección
- Se necesita alguna estrategia para escoger o combinar comportamientos que entren en conflicto
- **Arbitraje:** decidir de entre múltiples acciones o comportamientos, cuál realizar
- **Fusión:** combinar múltiples acciones para generar una sola orden

# Diseño de sistemas reactivos

- Descomponer el sistema de control en un conjunto de capas de comportamientos reactivos
- Cada comportamiento transforma la información de los sensores en órdenes para los motores
- Los comportamientos de bajo nivel (como evitar obstáculos) se pueden ejecutar en tiempo real al no necesitar mucha capacidad de cómputo
- Los comportamientos de alto nivel se invocan sólo cuando es necesario

# Diseño de sistemas reactivos

- Se trata de intentar unir las múltiples reglas para producir comportamientos efectivos y que permitan cumplir los objetivos propuestos



- Los dos modelos más conocidos de arquitecturas reactivas son
  - ▣ Arquitectura “subsumida” (*Subsumption architecture*)
  - ▣ Campos potenciales

# Arquitecturas “subsumidas”

- Subsumir significa “tener prioridad sobre”.
- Comportamientos organizados jerárquicamente por niveles que se ejecutan concurrente y asíncronamente
  - ▣ Cada nivel tiene asignada una prioridad
  - ▣ Las respuestas de los niveles superiores subsumen (predominan sobre) las respuestas de los niveles inferiores
  - ▣ Los niveles altos deberían estar a cargo de comportamientos orientados a garantizar la supervivencia
  - ▣ Los niveles inferiores se encargan de las tareas más básicas
  - ▣ Los sistemas se construyen de abajo hacia arriba

# Beneficios de la arquitectura *subsumida*

- Los sistemas se diseñan de manera incremental
  - ▣ Se evitan problemas por la complejidad de la tarea
  - ▣ Ayuda al diseño y al proceso de depuración
- Robustez
  - ▣ Si un nivel superior falla, los inferiores siguen funcionando
- Modularidad
  - ▣ Cada competencia se incluye en niveles distintos, por lo que el sistema es relativamente sencillo de diseñar y mantener
  - ▣ Las reglas y niveles se pueden reutilizar en diferentes robots y para diferentes tareas.





# Problemas

- No hay representación interna del mundo
  - ▣ Razonamientos complejos?
- No hay planificación estratégica
- ¿Cómo introducir aprendizaje?
- ¿Cómo crear objetivos y/o modificar los ya propuestos?
- Número de niveles máximo posible