

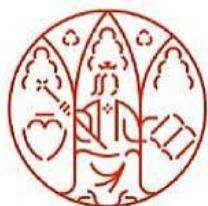
Máster en Inteligencia Artificial

Inteligencia Artificial para Sistemas Físicos

Práctica 1

Ana Gil Molina

20 enero 2025



UNIVERSIDAD
DE MURCIA

ÍNDICE

1- Ejercicio 1. Robot <i>Roombo</i>	3
1.1- Mundo	3
1.2- Arquitectura	3
1.3- Comportamientos	3
1.4- Autonomía / automatización	4
2- Ejercicio 2. Robot <i>Minotauro</i>	4
2.1- Mundo	4
2.2- Arquitectura	4
2.3- Comportamientos	5
2.4- Autonomía / automatización	5
3- Ejercicio 2. Robot <i>Minotauro++</i>	5
3.1- Mundo	5
3.2- Arquitectura	5
3.3- Comportamientos	6
3.4- Autonomía / automatización	7
3.5- Evaluación y comparación de estrategias	7

1- Ejercicio 1. Robot Roomba

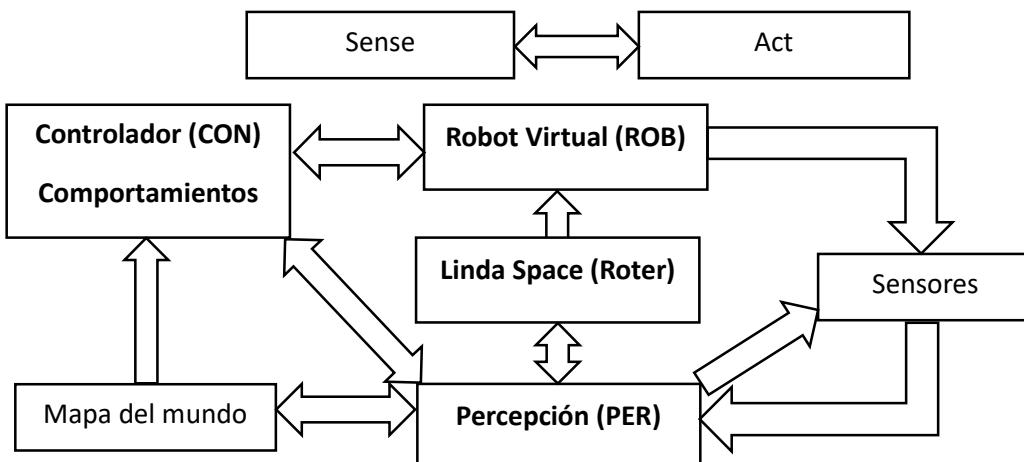
1.1- Mundo

- **Archivo:** ejercicio1.world
- **Dimensiones:** 15x15 m

Se define una zona *startRef*, utilizada como punto de referencia para inicializar el movimiento del robot. Esta zona se coloca intencionadamente sobre uno de los obstáculos, garantizando que el robot no pueda llegar a ella.

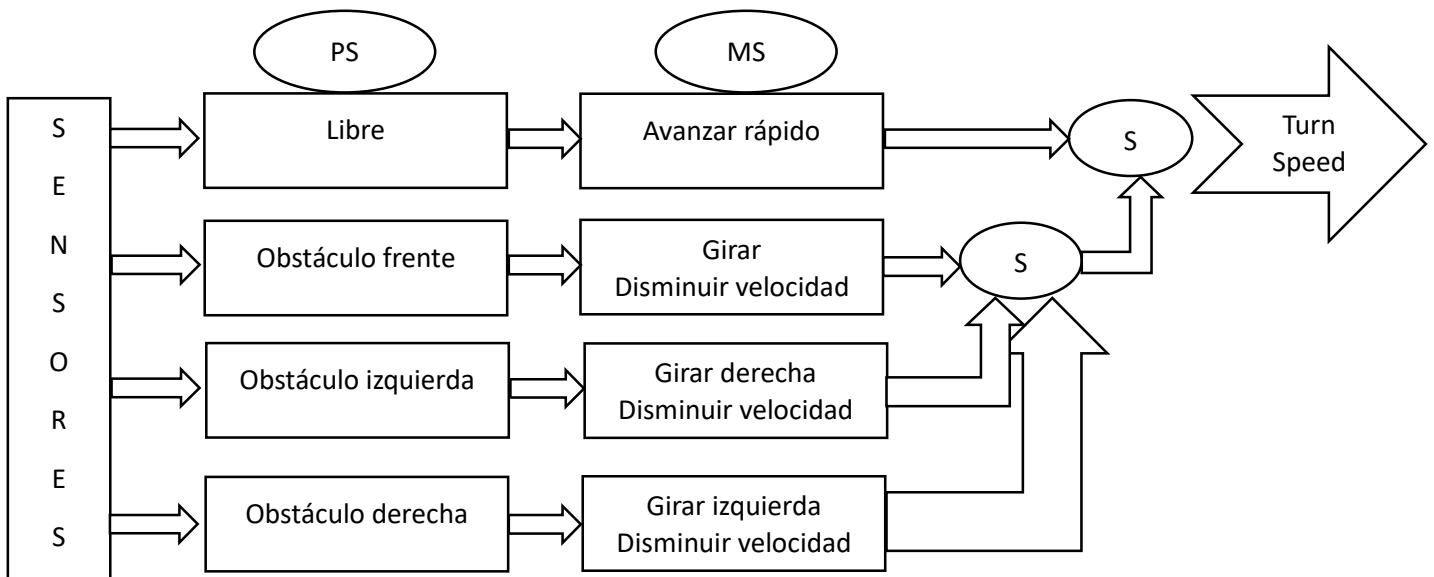
1.2- Arquitectura

- **Archivo:** ejercicio1.arch
- **Tipo de Arquitectura:** Reactiva
 - No se realiza planificación de tareas ni de trayectorias.
 - Navegación y limpieza se basan únicamente en las percepciones del entorno.

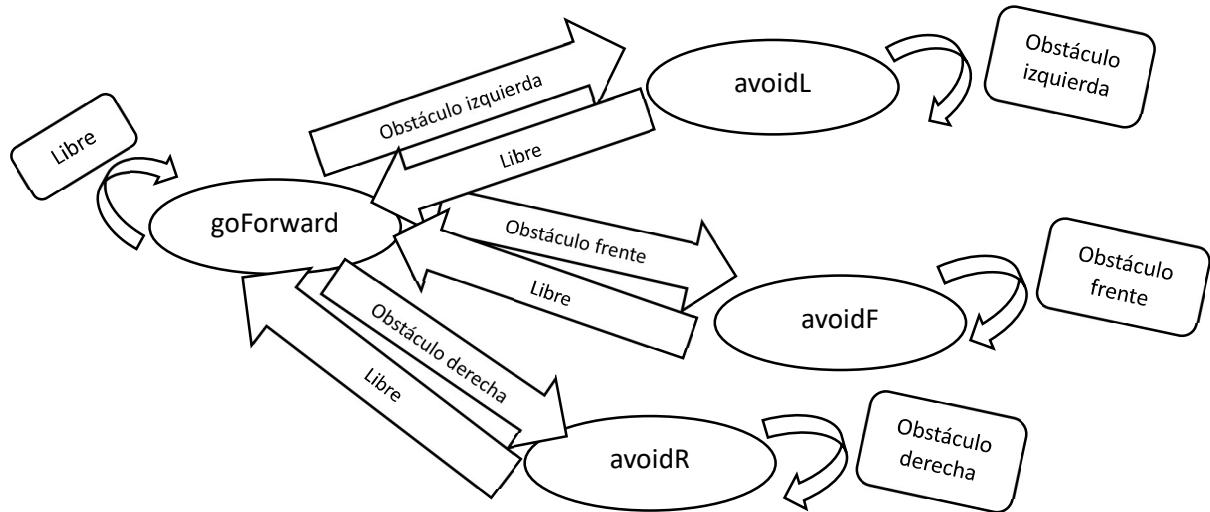


1.3- Comportamientos

- **Archivo:** ejercicio1.b

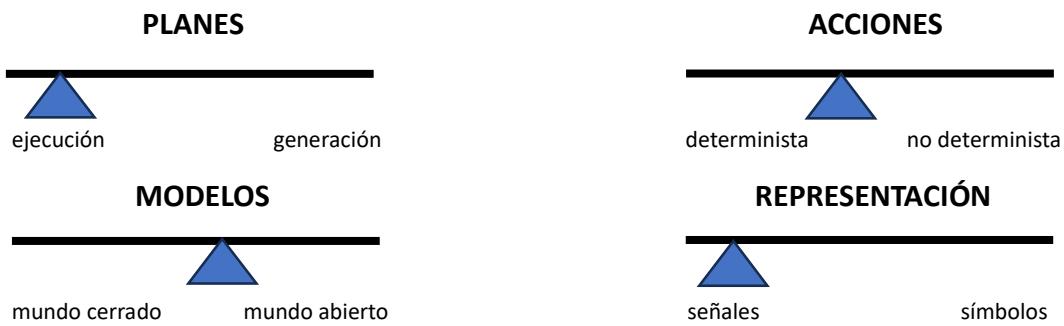


Inteligencia Artificial para Sistemas Físicos



Como los comportamientos de evitación (*avoidL*, *avoidR* y *avoidF*) tienen la misma prioridad, cuando varios de ellos se activan al mismo tiempo puede haber conflictos.

1.4- Autonomía / automatización



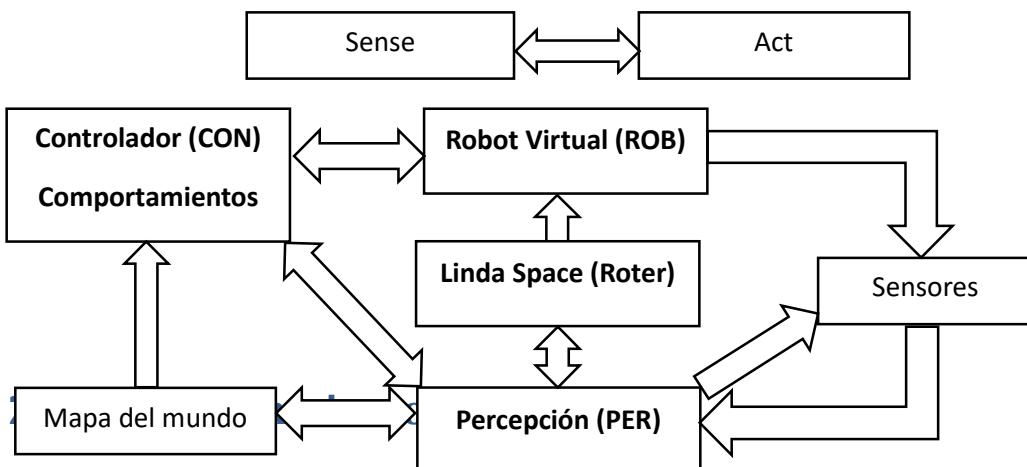
2- Ejercicio 2. Robot Minotauro

2.1- Mundo

- **Archivo:** *ejercicio2.world*
- **Dimensiones:** 20x20 m

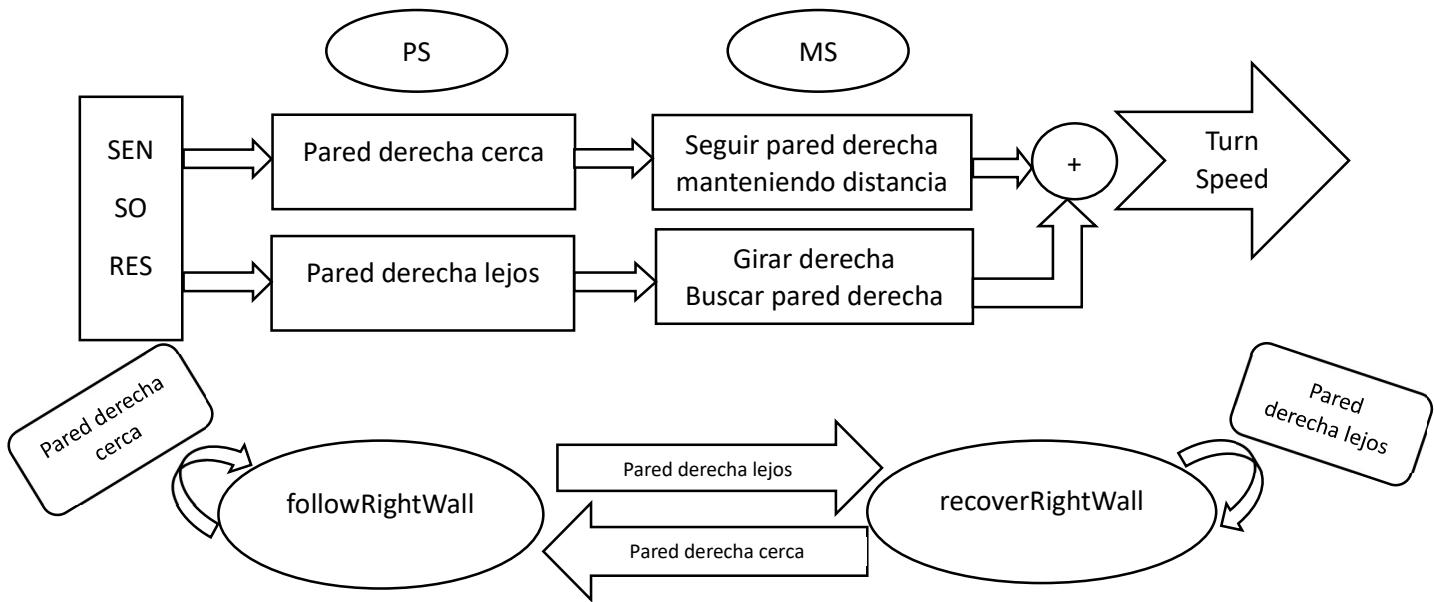
2.2- Arquitectura

- **Archivo:** *ejercicio2.arch*
- **Tipo de Arquitectura:** Reactiva



2.3- Comportamientos

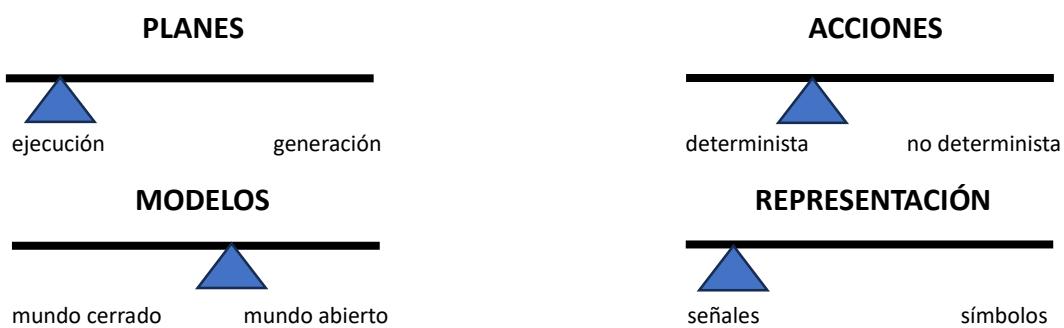
- **Archivos:** ejercicio2derecha.b, ejercicio2izquierda.b



Idem para seguir la pared izquierda. El seguimiento funciona igual en ambos casos.

En caso de que el robot pierda la pared de referencia, se activa el comportamiento *recoverRightWall* (o *recoverLeftWall*, según corresponda), diseñado para buscar y recuperar dicha pared. Si las paredes están muy lejos, el robot podría quedar atrapado girando continuamente a la derecha (o izquierda, según corresponda).

2.4- Autonomía / automatización



3- Ejercicio 2. Robot Minotauro++

3.1- Mundo

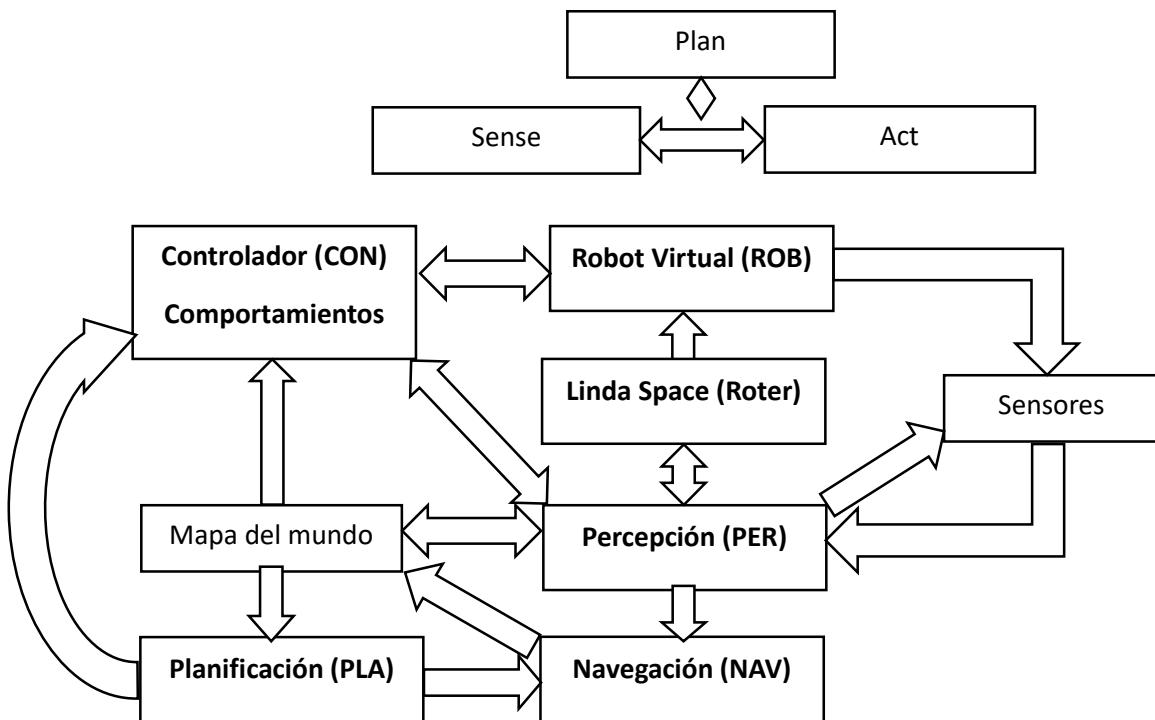
- **Archivo:** ejercicio3.world
- **Dimensiones:** 20x20 m

3.2- Arquitectura

- **Archivo:** ejercicio3.arch
- **Tipo de Arquitectura:** Híbrida

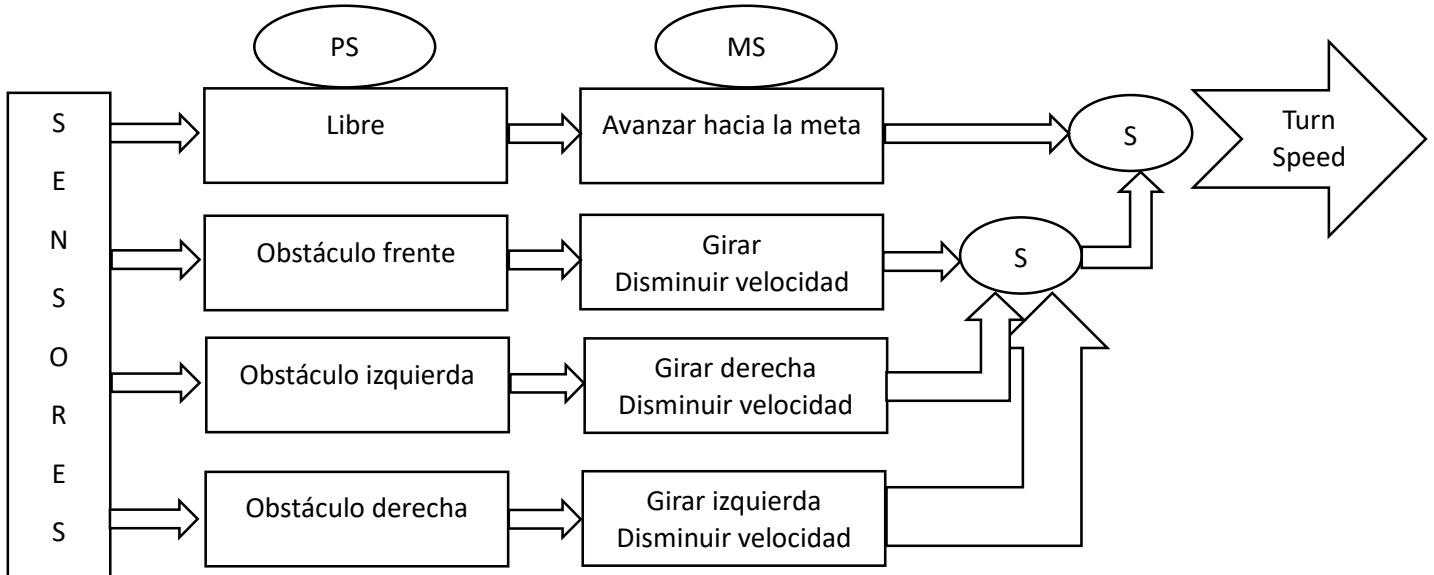
Inteligencia Artificial para Sistemas Físicos

- Deliberativa para planificar la mejor trayectoria y hacer un modelo del entorno.
- Reactiva para evitar los obstáculos y seguir el punto de *look-ahead*.



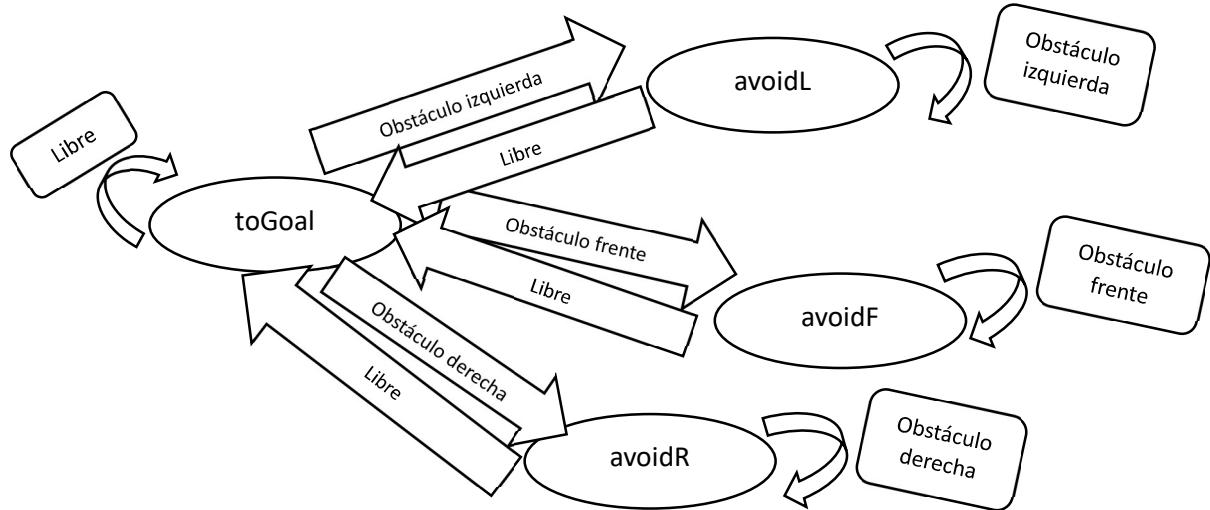
3.3- Comportamientos

- Archivo: *ejercicio3.b*

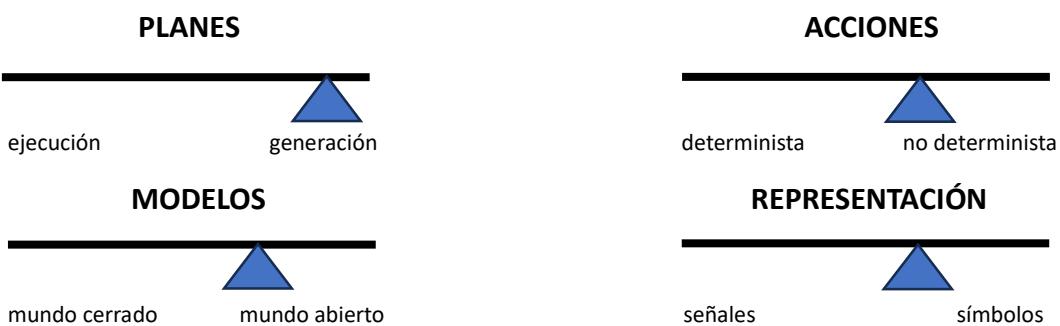


Se calcula la mejor trayectoria desde el principio. Sin embargo, al no conocerse el mapa *a priori*, la trayectoria se va actualizando a medida que se descubre y modela el entorno, teniendo en cuenta los obstáculos encontrados en el camino. Un posible problema es que el robot siga un camino aún no explorado, creyendo que es el más corto, para finalmente descubrir que no tiene salida, teniendo que retroceder y recalcular una nueva trayectoria.

Inteligencia Artificial para Sistemas Físicos



3.4- Autonomía / automatización



3.5- Evaluación y comparación de estrategias

- Al establecer un rango máximo del sonar de 1.5, el robot detecta menos celdas, pero identifica los obstáculos con mayor precisión al acercarse a ellos, en comparación con un rango mayor, por ejemplo, de 3. De este modo, un rango bajo permite encontrar el mejor camino (en nuestro mapa particular) en menos tiempo, ya que, gracias a la mayor precisión en la detección de obstáculos cercanos, se mejora la orientación del robot y se evita que repita innecesariamente el mismo camino, cosa que sí ocurre cuando se toma un rango mayor.
- Estableciendo una distancia de *look-ahead* de 15, el robot detecta mejor los obstáculos en comparación con distancias de 10 o de 20. En estos últimos casos, es más propenso a dejar celdas vacías cuando en realidad hay un obstáculo, lo que lo obliga a revisarlas varias veces. Con distancias menores, se enfoca demasiado cerca de su posición actual. Con distancias mayores, el robot planifica demasiado lejos, no detectando correctamente detalles cercanos.
- Tomando un tamaño de 0.025 para las celdas, tiende a dejar algunas celdas con obstáculos sin detectar, por ser demasiado pequeñas y haber tantas, teniendo que revisarlas varias veces. Con un tamaño de 0.225, las celdas son demasiado grandes y se detectan obstáculos donde realmente no los hay, haciendo que el robot regrese por el camino que ya ha recorrido, intentando buscar otro camino donde no lo hay. Finalmente, tomamos un tamaño intermedio de 0.075.
- Debido a la forma del laberinto definido, al utilizar la estrategia reactiva de seguir la pared derecha, el robot logra completarlo rápidamente. Sin embargo, al seguir la pared izquierda, debido al cruce de caminos, se desvía por caminos más largos, lo que resulta menos eficiente en comparación con el enfoque híbrido.