

Proyecto Final: Simulación de robot autónomo.

Integrantes:

- Alex Aravena
- Samira Becerra
- Álvaro Catalán
- Leandro Chamorro

Fecha: 26 de Junio de 2025

Asignatura: Robótica y Sistemas Autónomos.

1. Diseño del Proyecto

El diseño del proyecto se fundamentó en el desarrollo de un sistema robótico móvil completamente autónomo, capaz de desenvolverse eficazmente en un entorno simulado mediante el uso del simulador Webots.

El robot cuenta con cuatro ruedas independientes, dos a cada lado. Además contamos con un conjunto de sensores para la percepción del entorno, entre los que se incluyen: un sensor ultrasónico, un sensor LIDAR y un sensor GPS. El robot está orientado para detectar obstáculos en tiempo real, crear un mapeado del entorno y planificar una ruta óptima para llegar a su destino

2. Explicación del entorno simulado en Webots.

- Es en un tablero de 20x20 metros, teniendo 80 casillas de 0,5 metros cada una.
- Escenarios:
 - a. Línea recta: Se utilizó una línea recta donde en un punto está el inicio y el lado contrario se encuentra el objetivo. Este entorno fue creado para realizar pruebas en la situación más básica posible.
 - b. Caracol: Este entorno consiste en una espiral en la que el robot deberá llegar al objetivo que se encuentra en el centro. El objetivo fue identificar la capacidad del robot de bloquear obstáculos en un entorno un poco más complejo pero lo suficientemente sencillo para probar sus capacidades.
 - c. Laberinto: Como prueba final se diseñó un laberinto en el cual el robot deberá hacer uso de todas sus capacidades para poder superar los obstáculos y llegar finalmente a su objetivo. Este es el entorno con mayor dificultad.

3. Arquitectura del software: sensores, actuadores y módulos de control.

Sensores:

- a. Lidar: Este sensor nos permite medir las distancias mediante la emisión de pulsos de luz láser y la detección de su reflejo, lo que nos ayuda a generar mapas tridimensionales del entorno y así tomar mejores decisiones.
- b. Ultrasónico: Este sensor que nos permite medir la distancia entre el sensor y un objeto usando ondas de sonido, fue utilizado en nuestro proyecto final para la detección de obstáculos y con ello evitarlos.
- c. GPS: Lo usamos para conocer la ubicación del robot en el mundo real.

Actuadores.

- Ruedas: El robot poseerá 4 ruedas, convirtiéndolo en un robot móvil.

Módulos de control.

- A*: Utilizamos el algoritmo A* como planificador de rutas.

4. Algoritmos a utilizar (evitación de obstáculos, mapeo, planificación de rutas).

Planificación de rutas:

- A* Para poder realizar una planificación de las rutas utilizamos el algoritmo A*, el cual nos ayuda a encontrar el camino más corto desde la posición actual al destino, evitando obstáculos. Este algoritmo hace uso del costo real hacia el destino y el costo estimado (heurística).

Evitación de obstáculos:

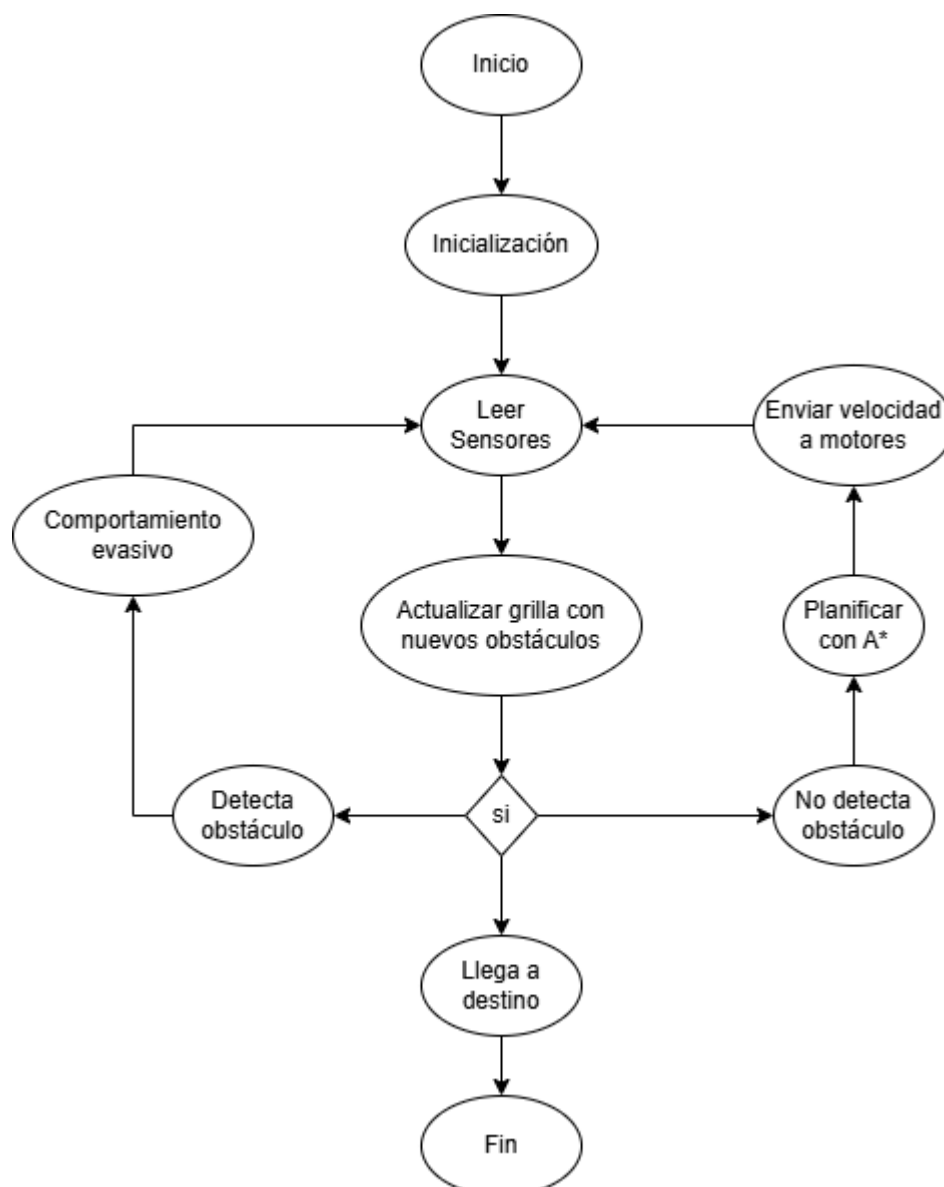
- Algoritmo Reactivo: Para poder evadir los obstáculos que los sensores detectan en el camino, el robot hace uso de un algoritmo reactivo, en el cual gira en el lugar y modifica su dirección hasta encontrar una ruta libre. Lo que permite su adaptación a entornos desconocidos y dinámicos.

Mapeo

- Para realizar un mapeo del entorno hicimos uso de un sensor LIDAR.

5. Diagramas de flujo y pseudocódigo de la solución.

Diagrama de flujo



Pseudocódigo

INICIAR robot

CONFIGURAR motores, sensores ultrasónicos, LIDAR y GPS

CREAR grilla vacía de 80x80

DEFINIR punto objetivo (x_goal, y_goal)

MIENTRAS el robot esté en ejecución:

LEER valores de sensores ultrasónicos

LEER posición del robot con GPS

LEER datos del LIDAR

CONVERTIR posición actual y objetivo a coordenadas de grilla

PARA cada lectura del LIDAR:

SI distancia < 1.0:

CALCULAR posición del obstáculo

MARCAR celda en la grilla como ocupada

SI se detecta obstáculo cercano:

FIJAR velocidades para girar

SINO:

LLAMAR a plan_path() con el mapa y obtener ruta

SI la ruta tiene al menos 2 puntos:

OBTENER siguiente waypoint

CALCULAR ángulo y distancia al waypoint

SI está cerca del waypoint:

AVANZAR al siguiente

SINO SI el ángulo es mayor a un umbral:

AJUSTAR velocidades para girar

SINO:

FIJAR velocidades para avanzar recto

ENVIAR velocidades a los motores

ESPERAR siguiente ciclo

TERMINAR ejecución

6. Resultados obtenidos (métricas de desempeño del robot).

- a. Línea recta.
 - i. Tiempo total de navegación: 1 minuto y 24 segundos.
 - ii. Longitud del path(celdas): 75 celdas.
- b. Caracol.
 - i. Tiempo total de navegación: 16 minutos y 30 segundos.
 - ii. Longitud del path(celdas): 372 celdas.
- c. Laberinto.
 - i. Tiempo total de navegación: 2 minutos y 51 segundos.
 - ii. Longitud del path(celdas): 95 celdas.

7. Análisis de los algoritmos utilizados (precisión, eficiencia).

Algoritmo A*: Este algoritmo demostró ser preciso y consistente a la hora de generar rutas de manera óptima desde la posición de inicio hasta el destino al escoger una ruta razonable con el entorno que se conoce.

- Precisión: El algoritmo demuestra encontrar rutas óptimas en los escenarios presentados pese a no conocerlos previamente.
- Eficiencia: Su toma de decisiones a la hora de elegir un nuevo objetivo es rápida.

Algoritmo Reactivo: El algoritmo utilizado para evitar obstáculos resultó muy efectivo ya que impedía que el robot chocara con los objetos, siempre mantenía un margen de distancia para evitar colisiones. En caso de encontrarse con un obstáculo de frente el robot giraba para poder evitarlo.

- Precisión: El algoritmo resultaba muy efectivo al enfrentarse a obstáculos nuevos.
- Eficiencia: Era extremadamente rápido, ya que su procesamiento no era complejo.

8. Reflexión sobre mejoras y optimización del sistema.

En cuanto a las mejoras y optimización del sistema, una de las áreas más relevantes a reforzar es la toma de decisiones tanto en la reacción reactiva como en la planificación de rutas. Actualmente, el sistema utiliza un enfoque relativamente básico para la evasión de obstáculos, el cual puede generar respuestas poco eficientes o tardías ante situaciones dinámicas del entorno.

Por otro lado, en el aspecto de planificación, el algoritmo A* puede mejorarse si es que no solo se toma en cuenta el costo del camino, sino también factores como la probabilidad de colisión o de caminos bloqueados en el futuro.

9. Lecciones aprendidas y posibles extensiones del proyecto.

A través de este proyecto aprendimos cómo realizar una simulación de un robot autónomo haciendo uso de diversos sensores para probar la eficiencia de los mismos en diferentes entornos. Sumado a esto aprendimos a implementar una planificación de ruta haciendo uso del algoritmo de A^* para que el robot sea capaz de generar una ruta eficiente hacia el objetivo.

Otro apartado destacable es la implementación de diferentes circuitos los cuales nos permitieron ver que el robot actúa de diferentes formas según la dificultad de los obstáculos que se le presenten.

Para futuros proyectos sería interesante probar los mismos entornos pero haciendo uso de diferentes sensores para ver cuál se adecua mejor a cada tipo de obstáculo y saber mejor qué sensor es más efectivo para llegar al objetivo en cada situación. Además se podría hacer uso de distintos entornos con formas simétricas o con obstáculos más aleatorios para comprobar la efectividad tanto del algoritmo implementado como de otras implementaciones.