Integrantes

- Maximiliano Miño
- Tomás Pérez
- Francisca Silva
- Matías Villegas

Descripción de proyecto

Nuestro proyecto se realizó con el objetivo de implementar, utilizando el simulador Webots, un robot móvil autónomo de cuatro ruedas, el cual posee control cinemático diferencial y sensores, Lidar y GPS, para que perciba correctamente su ubicación y elementos que estén a su alrededor.

A través de este control y uso de sensores, el robot es capaz de realizar las siguientes tareas:

- Construir un mapa local del entorno (grilla) en base a los datos obtenidos a través del sensor LIDAR
- Evitar, en tiempo real, obstáculos presentes en el entorno por la combinación de sensor LIDAR (visual) y GPS (ubicación).
- Planificar rutas con el algoritmo de planificación A*.
- Navegar de forma autónoma en el entorno, modificando su trayectoria en caso de percibir obstáculos.

Al desarrollar este robot, nos enfocamos en la percepción, planificación y control del robot de tal forma que pueda reaccionar a cambios de su entorno y tomar decisiones en tiempo real.

Arquitectura de software

1. Percepción
    a. Sensor LIDAR:
        i. Mapa parcial de la zona
        ii. Detecta obstáculos lejanos (rango de hasta 1 metro)
        iii. Construye mapa de ocupación (grilla)
    b. Sensores de distancia (sensor de ultrasonido)
        i. Detecta obstáculos cercanos frontales
        ii. Aumenta la seguridad de la evasión reactiva.
    c. GPS
        i. Obtiene posición actual (específicamente las posiciones X y Z)
2. Planificación
    a. Grilla
        i. Mapa representado como matriz
        ii. Las celdas están marcadas como libres u ocupadas
    b. Algoritmo de planificación A*
        i. Encuentra el camino más corto, o más bien el más óptimo, desde un punto inicial hacia un objetivo. Para determinar que sea óptimo hace uso de una heurística.

ii. Como la detección se realiza en tiempo real, este algoritmo se ejecuta en cada ciclo adaptándose a los cambios que ocurran en el mapa.

3. Control del robot:
    a. Navegación:
        i. Si hay obstáculo cerca, gira
        ii. Si no hay obstáculos, sigue recorrido
    b. Motores:
        i. Control cinemático diferencial (velocidad de las ruedas)
        ii. Ajustes de velocidad en función del ángulo hacia el waypoint actual

Resultados

Métricas de desempeño

Las métricas mostradas en esta sección son las métricas finales del recorrido:

- Tiempo total de navegación: 151.68 (s)
- Longitud del path: 12
- Tiempo de planificación: 0(ms)
- Porcentaje del mapa explorado:5 (%)

Análisis de algoritmos

- Precisión:
  El sistema de navegación implementado utiliza un enfoque híbrido que combina:
    - Planificación global mediante el algoritmo A* para encontrar rutas óptimas en un grid.
    - Evitación reactiva de obstáculos usando datos del LIDAR y sensores de distancia.
    - Seguimiento de ruta basado en posiciones GPS.

- Eficiencia:
    - Optimalidad garantizada (si existe solución, A* la encuentra).
    - Heurística Manhattan Distance (`|x1 - x2| + |y1 - y2|`), eficiente para movimientos en grid.
    - Se realiza una replanificación continua, permitiendo al sistema adaptarse a cambios dinámicos en el entorno.

- Robustez:
    - Detección redundante de obstáculos: el sistema combina LIDAR de 360° con sensores infrarrojos, reduciendo el riesgo de falsos negativos.

    - Umbral de seguridad conservador: ante obstáculos a menos de 0.3 m, se activa una parada de emergencia.

    - Replanificación en tiempo real: el algoritmo A* se recalcula en cada ciclo de control (cada 64 ms), manteniendo la ruta actualizada.

    - Escape reactivo básico: ante obstáculos cercanos, el sistema ejecuta giros bruscos de evasión, previniendo colisiones inminentes.

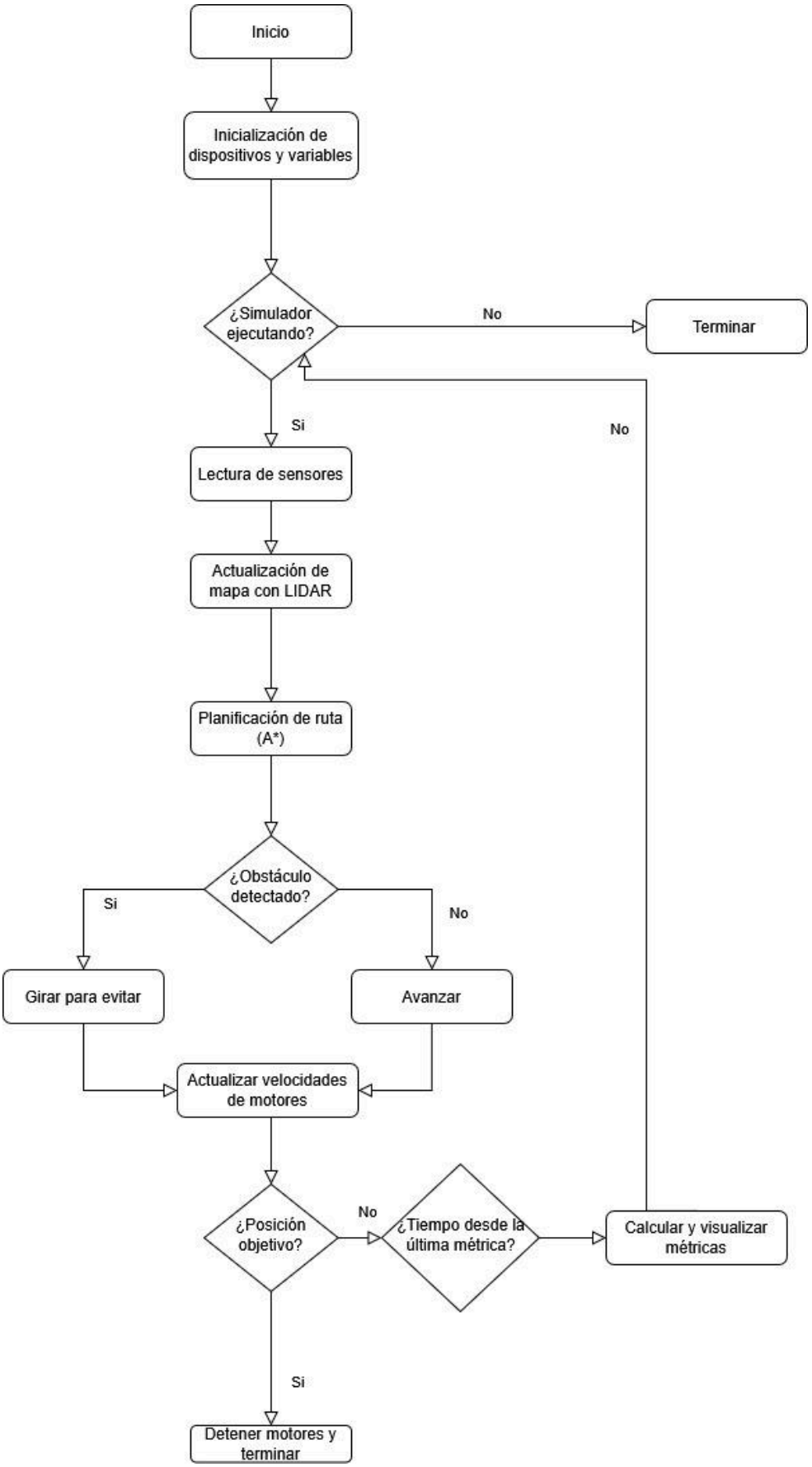
Reflexión de mejoras

- Mayor precisión en la percepción de obstáculos
- Integrar un sensor RGB para la detección de colores y ajustar el algoritmo de planificación para que el robot realice cambios en su trayectoria, dependiendo del color que detecte.
- Lograr que la toma de decisiones en cuanto a posibles caminos que pueda seguir sea más eficiente

- Pseudo código:
  INICIO
-     Inicializar Webots y dispositivos:
-         - Motores (4 ruedas)
-         - Sensores de distancia (2)
-         - LIDAR
-         - GPS
-
-     Inicializar variables:
-         - Grid de mapa GRID_SIZE x GRID_SIZE
-         - Array para almacenar path
-         - Tiempos de inicio y métricas
-
-     MIENTRAS el simulador está ejecutando HACER
-         // Detección de obstáculos
-         Leer sensores de distancia
-         SI distancia < umbral ENTONCES
-             ds_detect_near ← VERDADERO
-         FIN SI
-
-         // Obtener posición actual del GPS
-         robot_x, robot_y ← Leer GPS
-
-         // Actualizar mapa con datos del LIDAR
-         Limpiar grid
-         PARA cada punto en el escaneo LIDAR HACER
-             SI distancia es válida Y distancia < rango_máximo
  ENTONCES
-                 Calcular posición obstáculo (obs_x, obs_y)
-                 Convertir a coordenadas de grid (cell_x,
  cell_y)
-                 SI está dentro del grid ENTONCES
-                     Marcar celda como obstáculo (1)
-                     SI distancia muy cercana ENTONCES
-                         lidar_detect_near ← VERDADERO
-                     FIN SI
-                 FIN SI
-             FIN SI
-         FIN PARA
-
-         // Marcar posición actual en el grid
-         robot_cell_x, robot_cell_y ← Convertir (robot_x,
  robot_y) a celdas de grid
-         Marcar celda actual como explorada (2)
-
-         // Planificación de ruta con A*
-         start ← (robot_cell_x, robot_cell_y)
-         goal ← (GRID_SIZE-2, GRID_SIZE-2)

```
-          path_length ← plan_path(grid, start, goal, path,
   MAX_PATH_LEN)
-
-          // Control de movimiento
-          SI lidar_detect_near O ds_detect_near ENTONCES
-              // Girar para evitar obstáculo
-              left_speed ← SPEED
-              right_speed ← -SPEED
-          SINO SI path_length <= 1 ENTONCES
-              // Girar para explorar
-              left_speed ← SPEED
-              right_speed ← -SPEED
-          SINO
-              // Avanzar
-              left_speed ← SPEED
-              right_speed ← SPEED
-          FIN SI
-
-          Aplicar velocidades a motores
-
-          // Reporte de métricas cada 5 segundos
-          SI tiempo_actual - last_metrics_time >= 5 ENTONCES
-              Calcular porcentaje de mapa explorado
-              Mostrar métricas:
-                  - Tiempo de navegación
-                  - Longitud del path
-                  - Tiempo de planificación
-                  - Porcentaje explorado
-                  - Posición GPS
-                  - Valores de sensores
-                  - Mapa actual (ASCII)
-              Actualizar last_metrics_time
-          FIN SI
-
-          // Condición de parada
-          SI robot_cell_x == stop_x Y robot_cell_y == stop_y
   ENTONCES
-              Detener motores
-              Terminar programa
-          FIN SI
-      FIN MIENTRAS
-
-      Limpiar Webots
-  FIN
```

https://drive.google.com/file/d/1okHyaiwYb5XyZ3asdKFAeZYeT31lfQq3/view?usp=sharing


Inicio
Inicialización de dispositivos y variables
¿Simulador ejecutando?
No
Terminar
Si
No
Lectura de sensores
Actualización de mapa con LIDAR
Planificación de ruta (A*)
¿Obstáculo detectado?
Si
No
Girar para evitar
Avanzar
Actualizar velocidades de motores
¿Posición objetivo?
No
¿Tiempo desde la última métrica?
Calcular y visualizar métricas
Si
Detener motores y terminar