Pipeline + Software

Sistema operacional da Jetson: Ubuntu 20 + ROS2 + sdk zed

1. Percepção (ZED2 + LiDAR)

Responsável por interpretar o ambiente. A câmera ZED2 + LiDAR

- **Saídas nos tópicos:**
  - /perception/occupancy → mapa de ocupação (occupancy grid)
  - /perception/navigable_mask → máscara gerada com as áreas que o robô pode percorrer
  - /perception/obstacles → obstáculos móveis detectados no caminho

Deteçoes:

- **Áreas livres e obstáculos estáticos:**
  Usamos a imagem de profundidade da ZED (/zed_node/depth) junto com uma rede de segmentação semântica, que identifica classes treinadas como rua, muro, calçada, etc.
- **Obstáculos móveis:**
  São detectados com a nuvem de pontos da ZED (/zed_node/point_cloud) e com os dados do LiDAR do tópico (/lidar/scan). Juntos, eles ajudam a reconstruir o ambiente em 3D.

---

2. Mapping (SLAM)

Gera o mapa global da arena. O carro roda pela arena e constrói esse mapa com os sensores.

- **Sensores usados:**
  - LiDAR 2D
  - ZED2 (profundidade + pose)
  - IMU e odometria da própria ZED2
- **Tópicos importantes:**
  - /zed_node/pose → posição estimada do robô
  - /lidar/scan e /zed_node/point_cloud → leitura do ambiente
  - /tf → transformações entre frames (odom, map, base_link...)
- **Saídas:**
  - my_map.pgm e my_map.yaml → usados depois para visualização no RViz e navegação

---

3. Planejamento (Planning)

Pode rodar apenas com o mapa salvo, ou com o mapa e a percepção em tempo real.

- **Global Planner:**
  - Usa o mapa feito pelo SLAM para traçar o melhor caminho até o destino (goal).
  - Gera um /planning/nav_msgs/Path com os pontos que o carrinho deve seguir.
- **Local Planner:**
  - Combina esse caminho global com a occupancy grid da percepção (/perception/occupancy), ajustando o trajeto caso haja algum obstáculo móvel
  - Gera comandos de movimento publicados em /planning/cmd_vel (mensagem do tipo Twist).

---

4. Controle (Control)

Transforma os comandos de velocidade em sinais para os motores.

- **Entradas:**
  - /planning/cmd_vel → velocidades calculadas pelo planner
  - /zed_node/odom → feedback da odometria.
- **Saídas:**
  - /servo_pwm → controla a direção
  - /motor_pwm → controla a velocidade

Esses sinais vão para o controlador de motor (VESC ou Arduino)(a definir).