

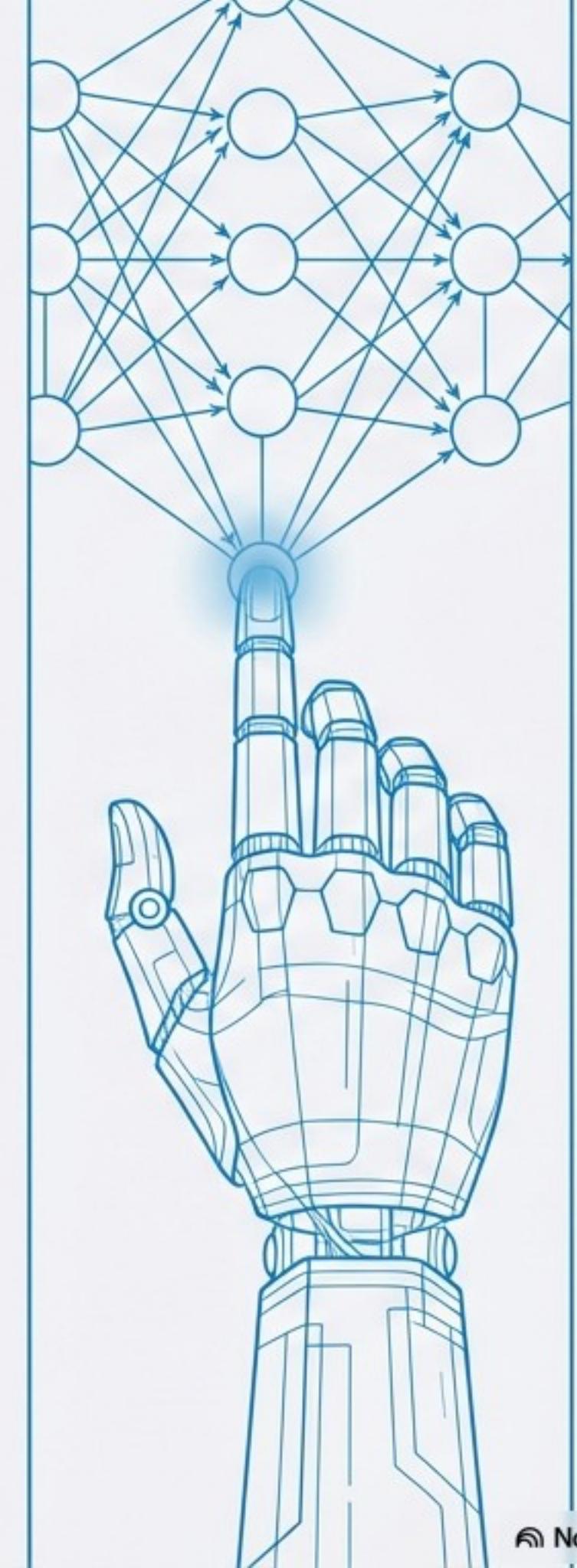
Controle Inteligente

Da Teoria à Prática com ROS 2, IA e
Robótica

Semestre 2026/1 | Curso Bacharelado em Engenharia Elétrica

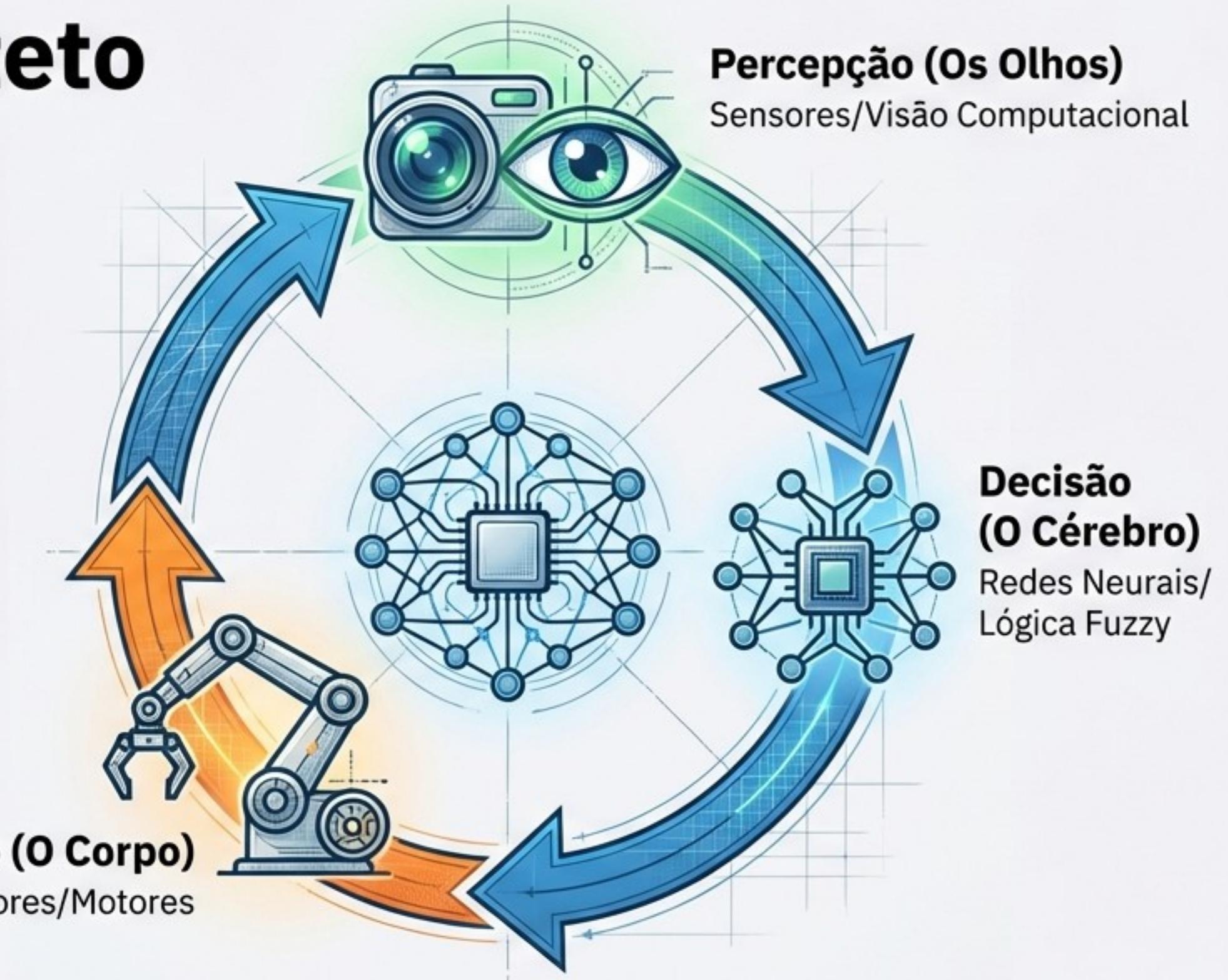
Prof. Fabio Ricardo Oliveira Bento

Onde a Engenharia Elétrica encontra a Inteligência Artificial.



A Jornada do Arquiteto de Inteligência

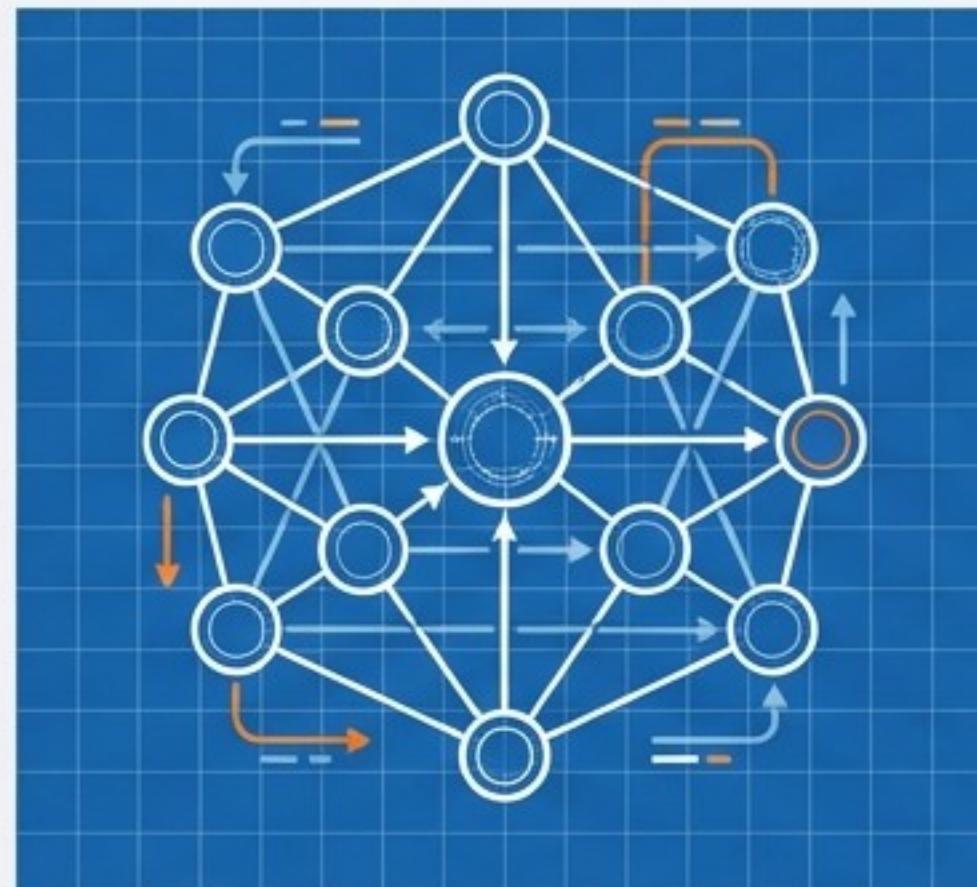
Não estudaremos apenas algoritmos; construiremos agentes. Este curso segue o fluxo de desenvolvimento de um robô autônomo moderno, transformando teoria abstrata em sistemas funcionais.



Objetivo: Agente Autônomo para Navegação (Inteligência como software modular via middleware).

O Stack Tecnológico Industrial

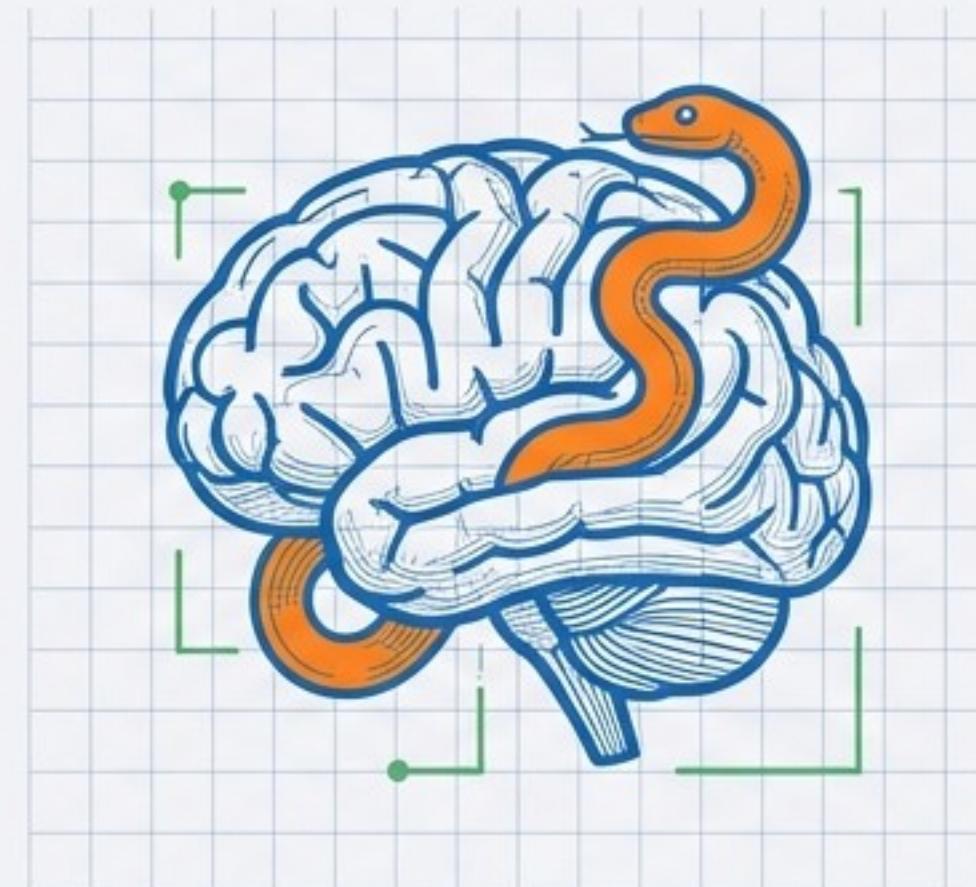
Ferramentas de ponta alinhadas ao mercado (Ubuntu 24.04).



O Sistema Nervoso

ROS 2 (Jazzy/Humble)

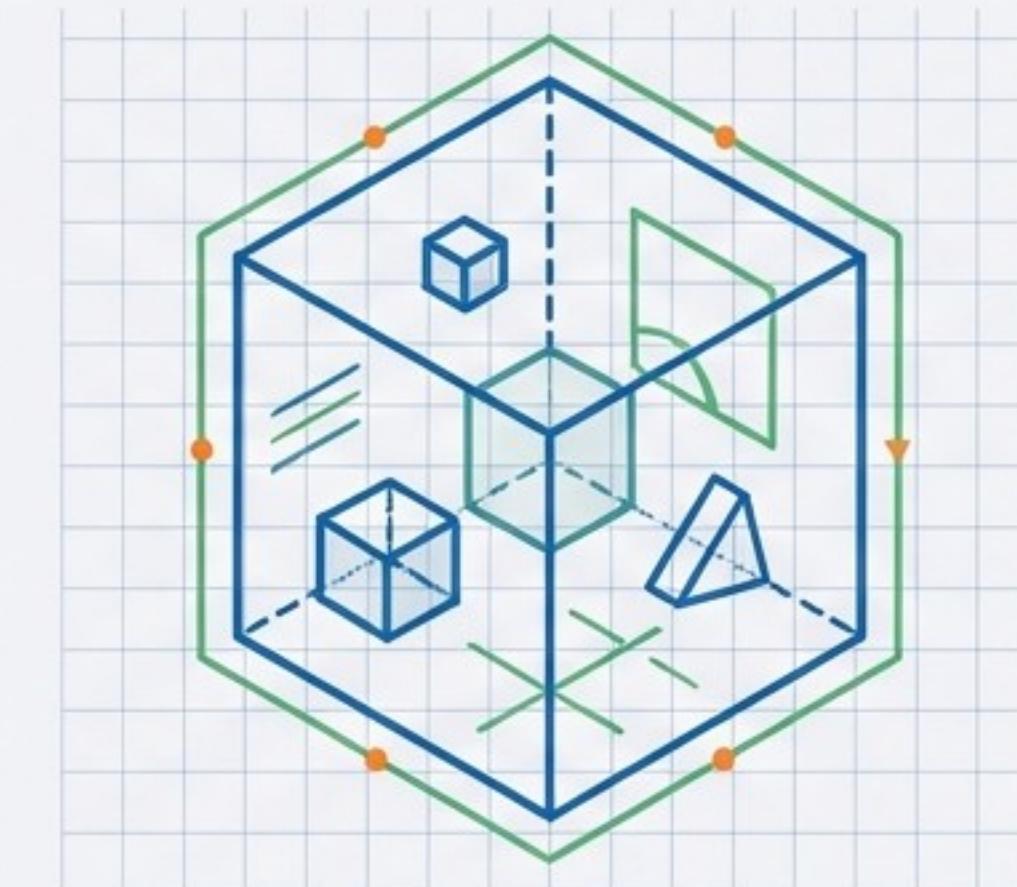
O padrão industrial de middleware. Conecta sensores, atuadores e algoritmos.



O Cérebro

Python 3 + PyTorch

Biblioteca líder para Deep Learning e construção de redes neurais complexas.



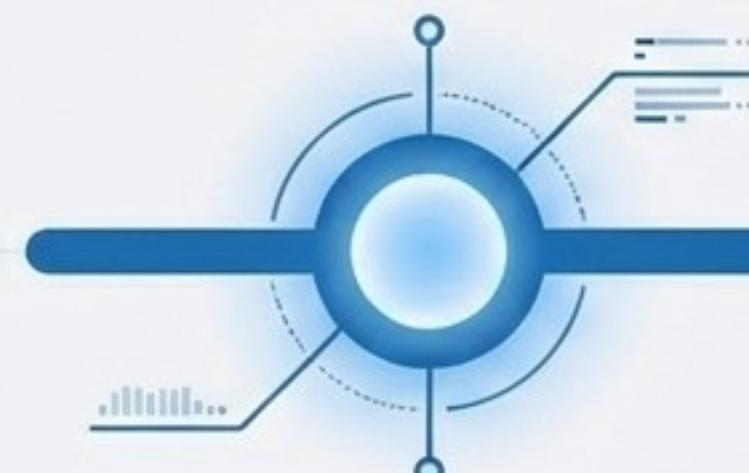
O Mundo

Gazebo Sim

Simulação física realista para validação segura (Sim-to-Real).

Roadmap do Semestre 2026/1

Fase 1: Infraestrutura (Aulas 1-4)

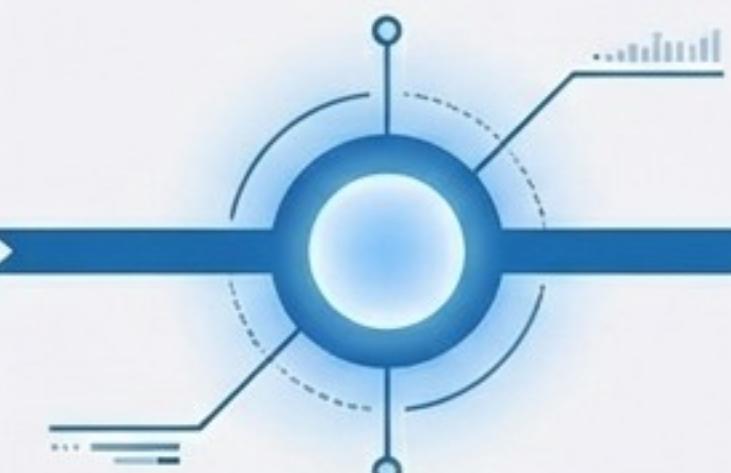


O Sistema Nervoso

ROS 2, Nós, Tópicos



Fase 2: Percepção (Aulas 5-10)

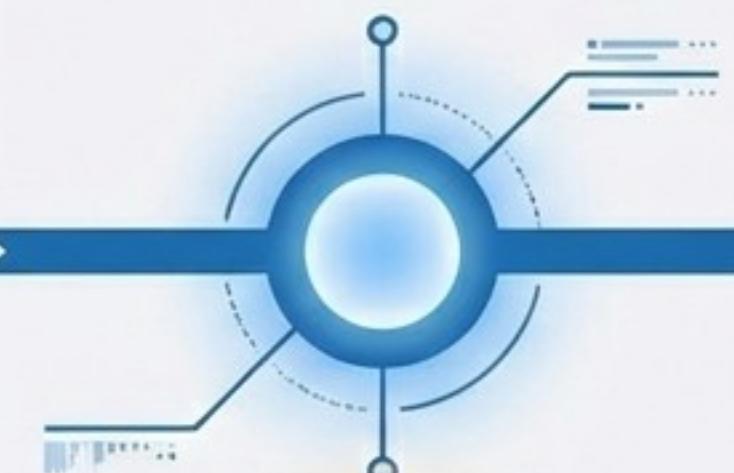


Os Olhos

Deep Learning,
CNNs, PyTorch

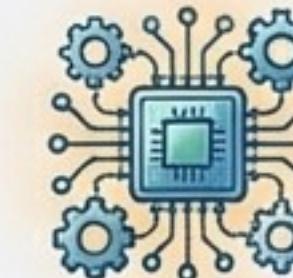


Fase 3: Otimização (Aulas 11-15)

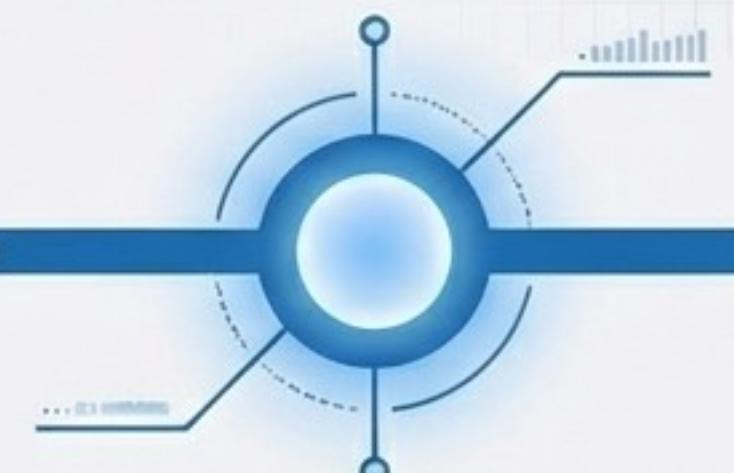


Os Reflexos

Lógica Fuzzy,
Algoritmos Genéticos



Fase 4: Autonomia (Aulas 16-23)



A Cognição

Deep RL, Sim2Real



Unidade I: O Sistema Nervoso (ROS 2)

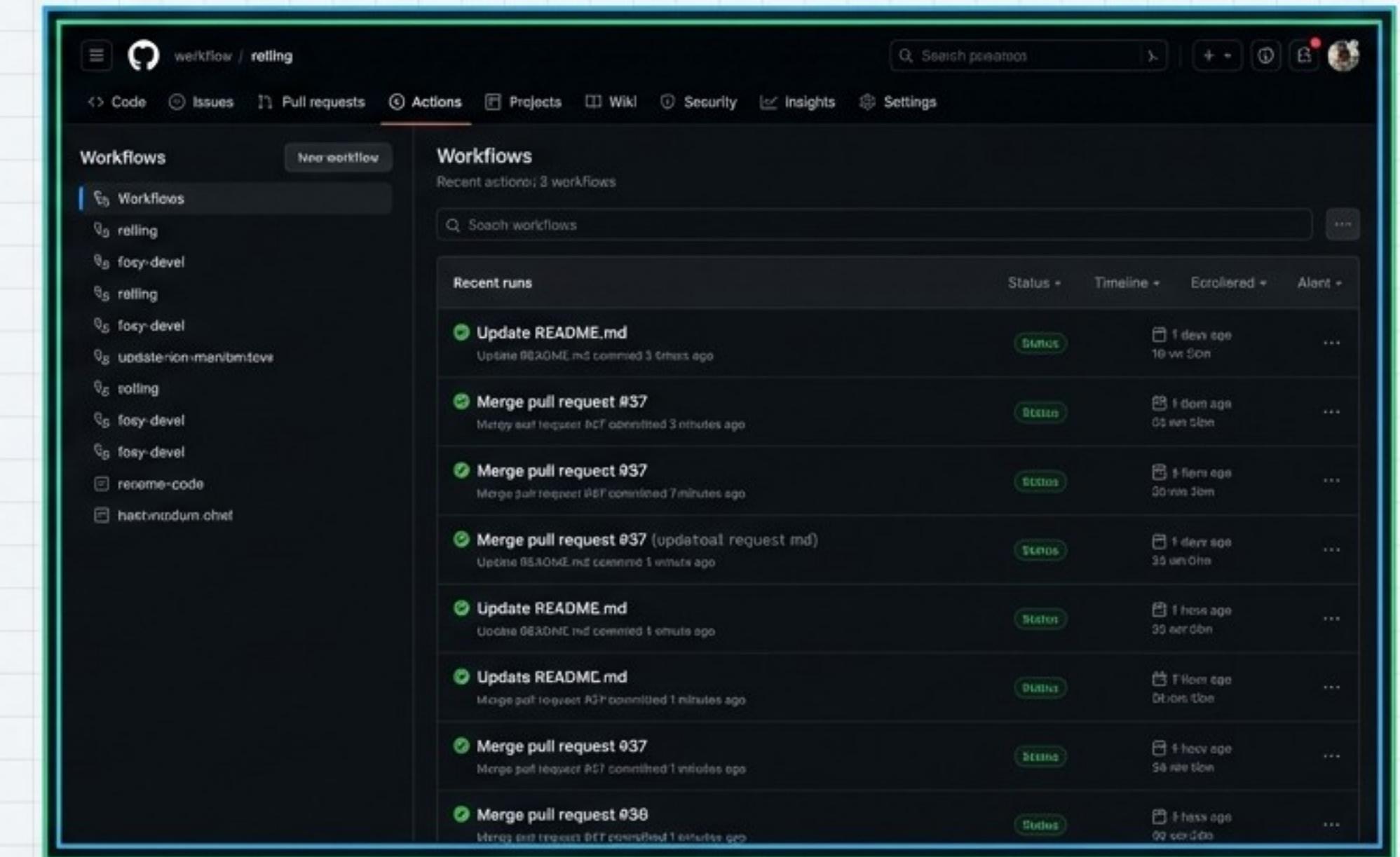
Antes de haver inteligência, deve haver conectividade.

Conceitos Fundamentais:

- **Nodes (Nós)**: Processos individuais (neurônios) executando tarefas.
- **Topics & Services**: Canais de dados (sinapses) para fluxo de informação.
- **Diferencial**: Transição para ROS 2 (DDS Middleware) para tempo real.

Hands-on:

- Configuração do Workspace Linux e criação do pacote 'Talker/Listener'.



Unidade II: Visão e Percepção

Transformando pixels em tensores.



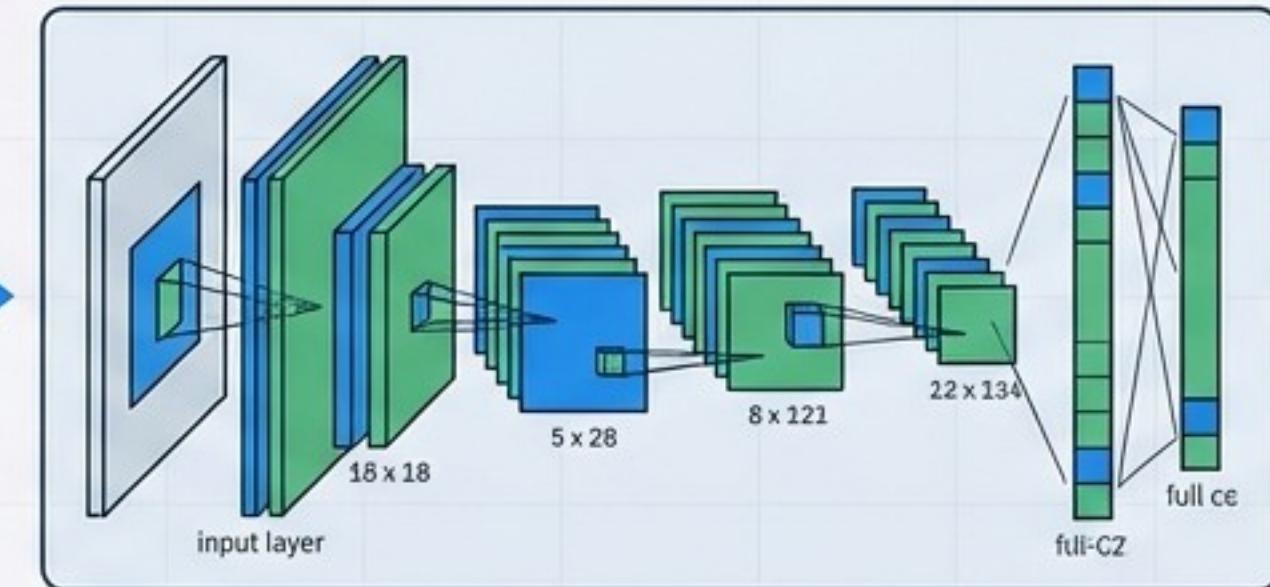
Raw Data



sensor_ms-
gs/Image



PyTorch
Tensor



Decisão:
Parede vs.
Caminho

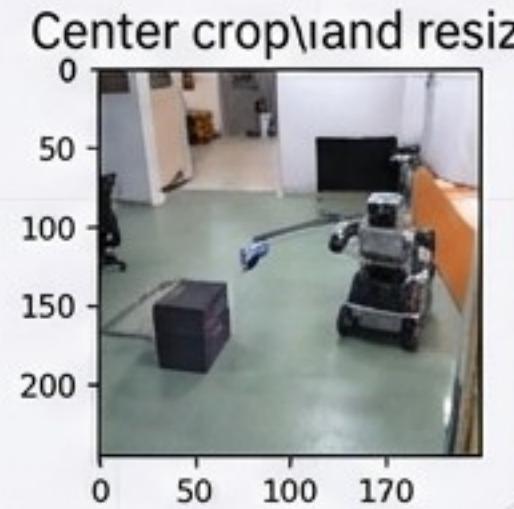
```
>>> ax.set_title('Adjust brightness', size=15)
>>> ax = fig.add_subplot(2, 5, 9)
>>> img_adj_brightness = transforms.functional.adjust_brightness(
...     img, brightness_factor=1.3
... )
>>> ax.set_title('Center crop\nand resize', size=15)
>>> ax = fig.add_subplot(2, 5, 10)
>>> img_center_crop = transforms.functional.center_crop(
...     img, [0.7*218, 0.7*178]
```



Original



Brightness-adjustent



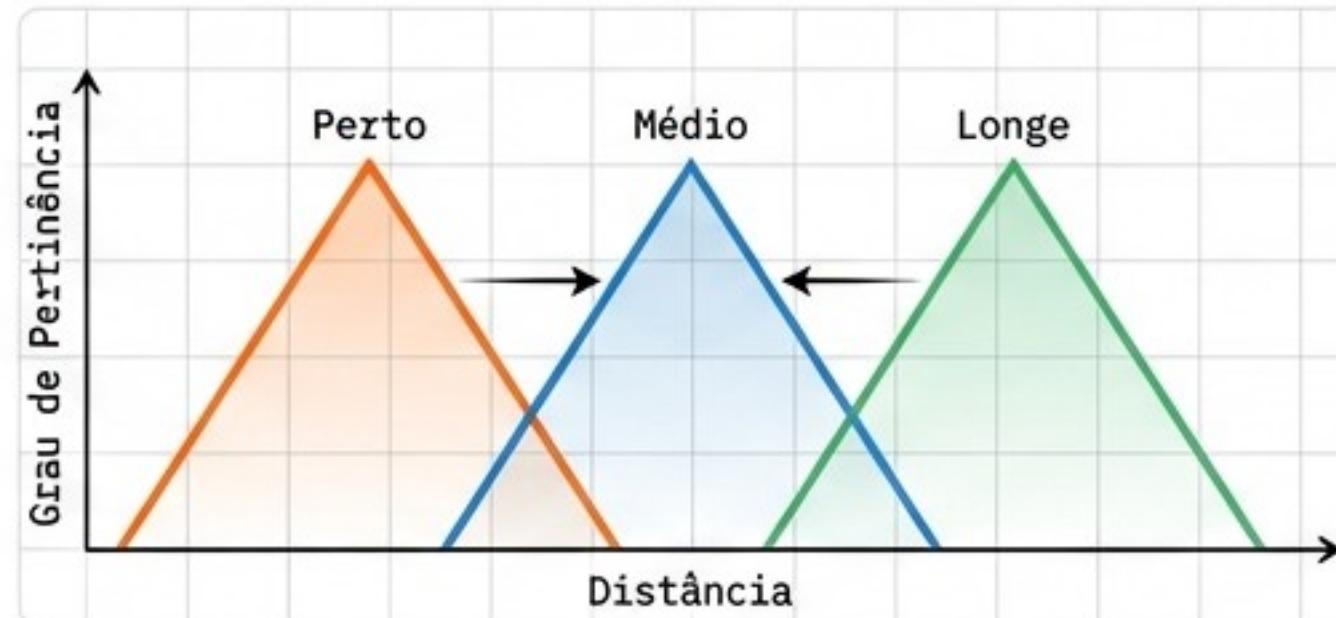
Center crop\and resize

Objetivo Prático: Nós de inferência que identifica obstáculos e publica comandos de parada.

Unidade III: Adaptação e Otimização

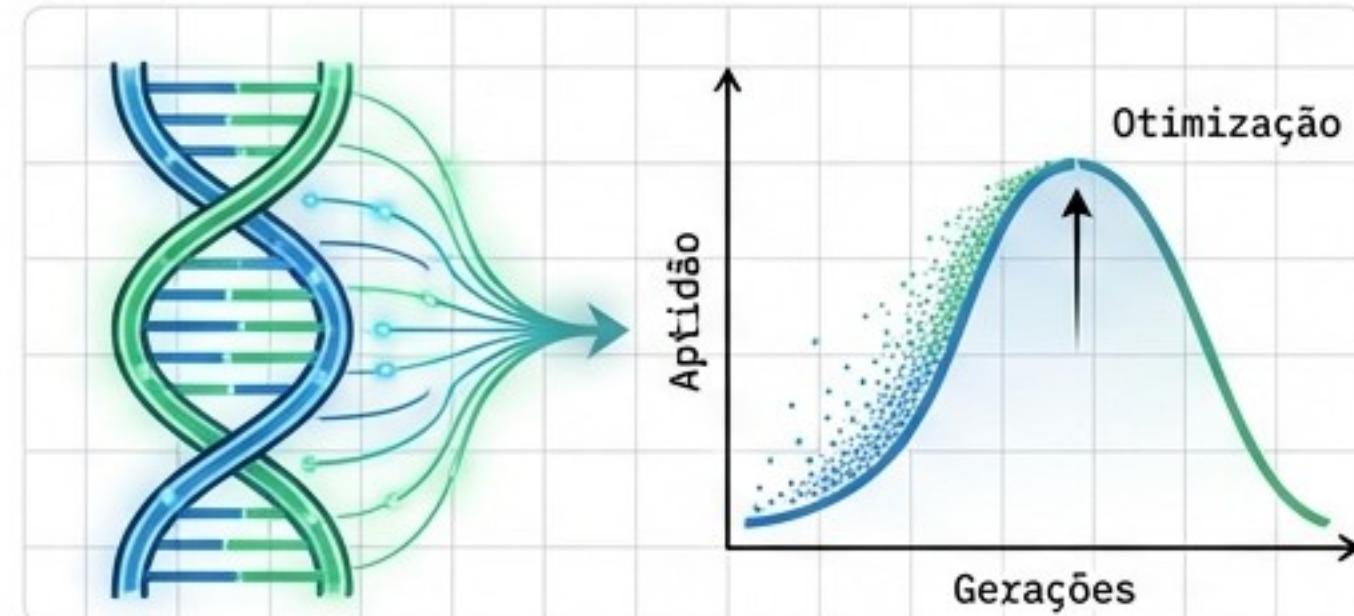
Hibridização de técnicas clássicas e modernas.

Lógica Fuzzy (Reflexos)



Fuzzy Navigator: Regras linguísticas para evitar colisões. Tratamento de incerteza (ex: Se "Perto" então "Freie").

Algoritmos Genéticos (Evolução)



Neuroevolução: Otimização automática de hiperparâmetros de controle e arquiteturas neurais.

Unidade IV: O Cérebro (Reinforcement Learning)

Aprendizado por tentativa e erro.



Conceitos: Agente, Ambiente, Reward, MDPs.

Algoritmos: DQN (Deep Q-Networks) e PPO (Proximal Policy Optimization).

Desafio: Mapless Navigation (Navegação sem mapa prévio).

Dados de Treinamento (Exemplo)

```
>>> torch.manual_seed(1)
>>> print(sample(model, starting_str='The island'))
Trainaining data 1 [2275 deep=0, 0.59, 0.0a24, 1.70, 0.333317]
Trainaining data 1 [2275 deep=0, 0.49, 0.0a23, 1.70, 0.136998]
Trainaining data 1 [2175 deep=0, 0.49, 0.0a24, 1.80, 0.327917])
```

Metodologia: O Laboratório R&D

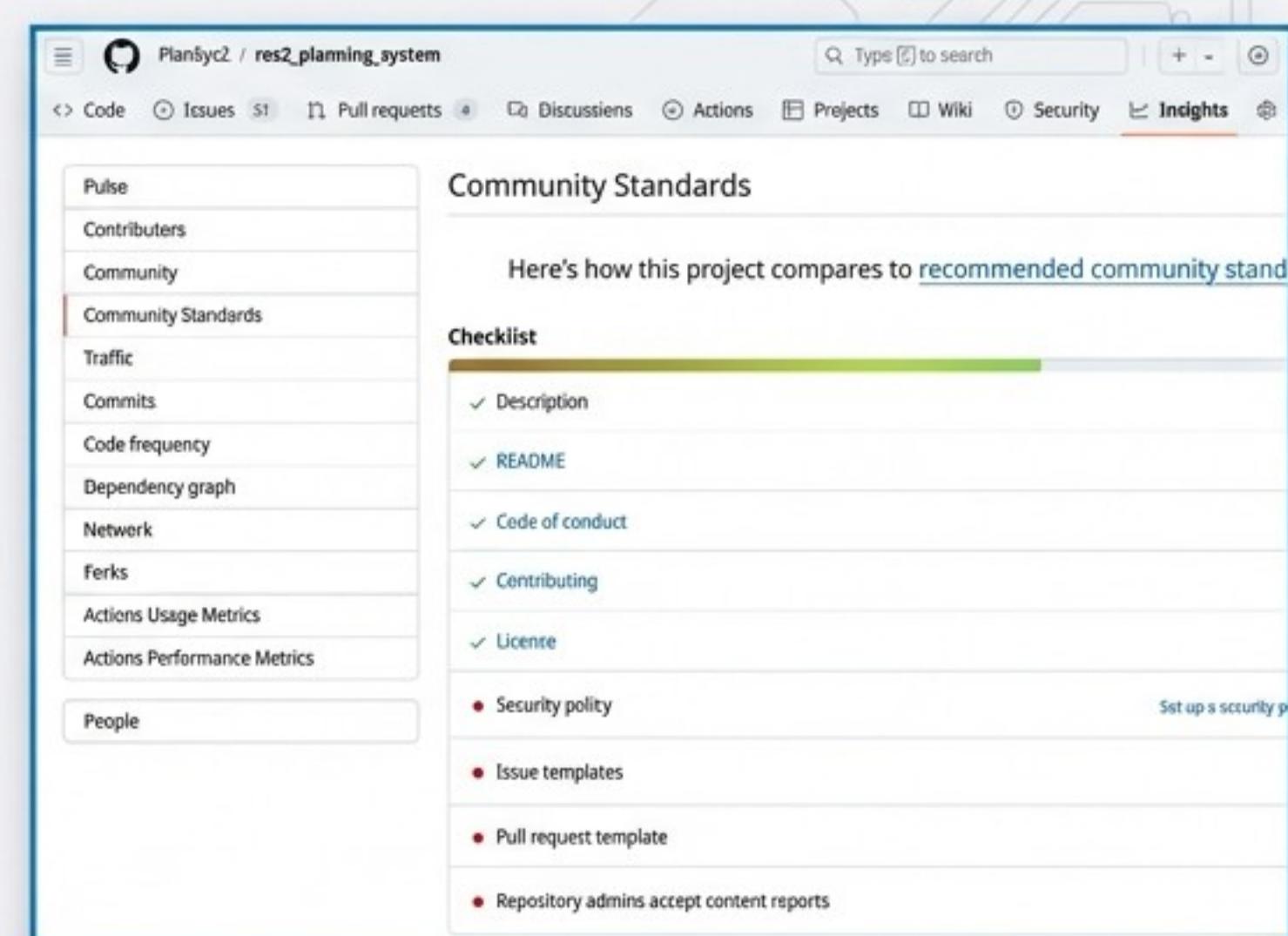
Ambiente de desenvolvimento profissional (Sim2Real).

Ambiente:

- Ubuntu 24.04 / Docker
- Controle de Versão (Git)

Workflow:

1. Teoria Matemática
2. Simulação (Gazebo)
3. Integração (ROS 2 Nodes)



Ênfase na qualidade de código, documentação e práticas profissionais de software.

Desafio Final: O Robô Navegador Inteligente

O Cenário: Um labirinto simulado no Gazebo com obstáculos estáticos e dinâmicos.

A Tarefa: Navegar de A para B.

A Regra: Apenas sensores a bordo (LiDAR/Câmera). Sem mapa global.

Labs 1-6: Setup & Visão



Labs 7-10: Treinamento RL & Competição

Conexão com a Pesquisa (Grupo GAIA)

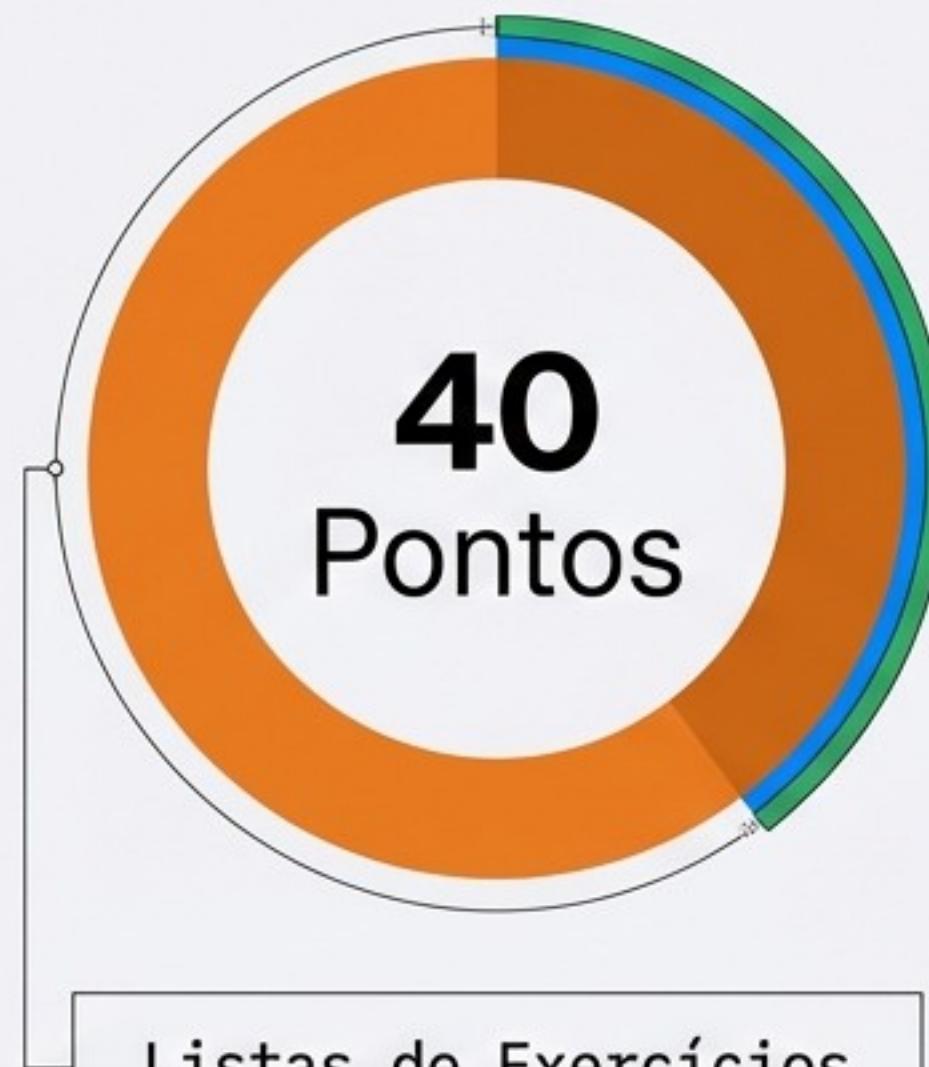
Oportunidades de Iniciação Científica.



- **Projeto:** Agente Autônomo para Navegação em Ambientes Virtuais Complexos.
- **Tópicos Avançados:** Reward Shaping, Dueling Networks, PPO.
- **Objetivo:** Publicação acadêmica e desenvolvimento científico.

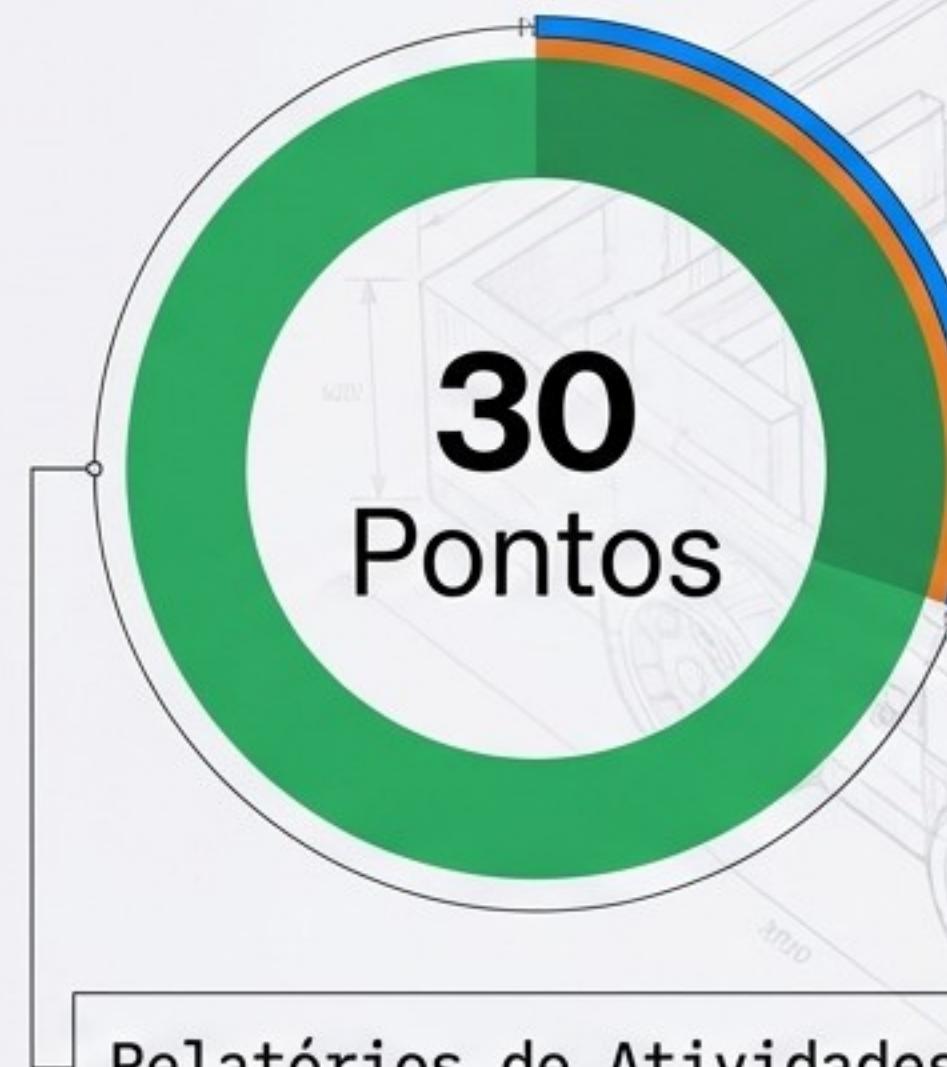
Sistema de Avaliação

Foco no progresso prático consistente.



Listas de Exercícios

Consolidação teórica semanal.



Relatórios de Atividades

Documentação de laboratório
(Markdown/Jupyter).

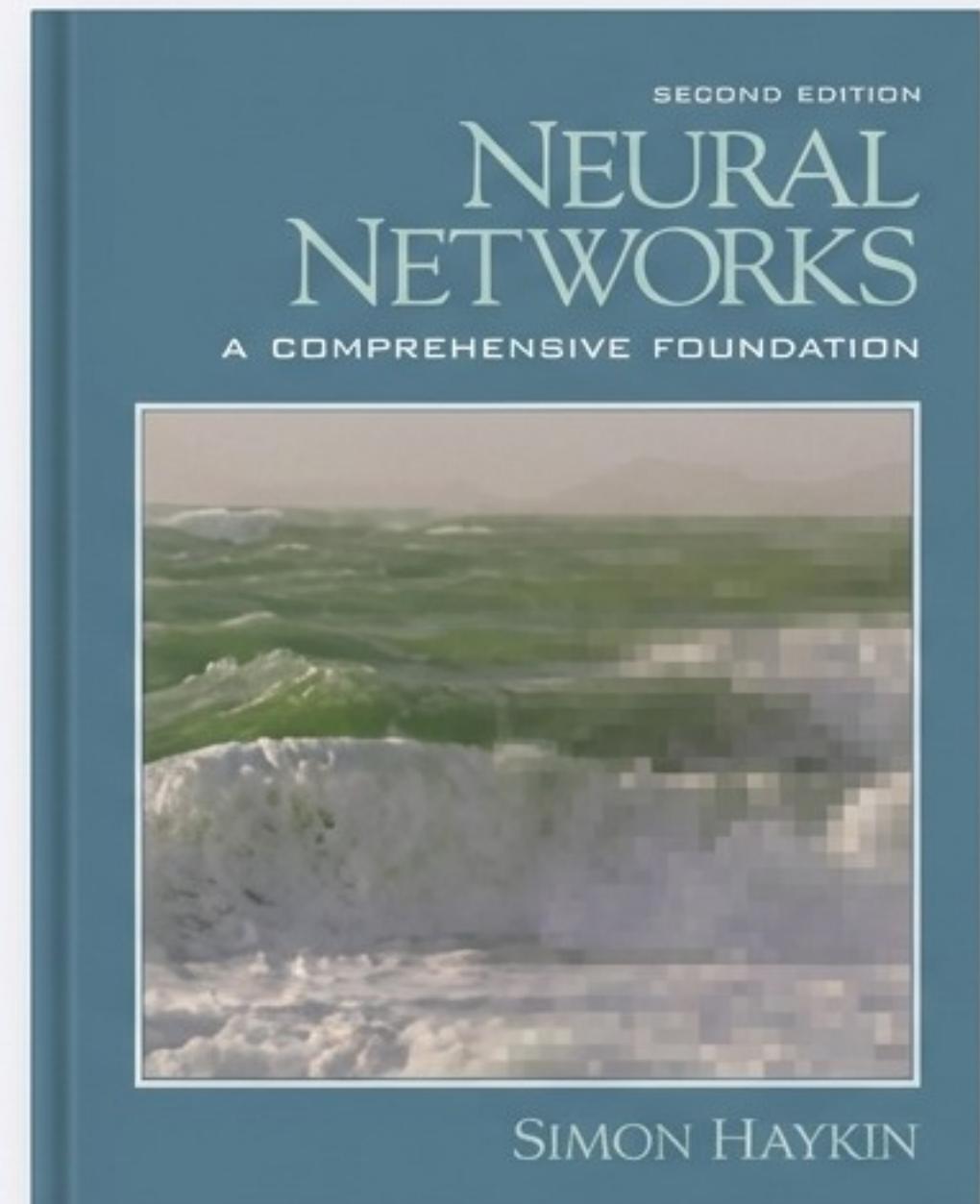


Projeto Integrador

Apresentação e performance
no desafio do labirinto.

Bibliografia Essencial & Estado da Arte

- 1. Mastering ROS 2 for Robotics Programming** (Renard, 2025) - *Integração ROS 2 + AI.*
- 2. Deep Reinforcement Learning Hands-On** (Lapan, 2024) - *Algoritmos PPO/DQN.*
- 3. Hands-On Machine Learning** (Géron, 2025) - *Fundamentos de ML/CNNs.*
- 4. Neural Networks: A Comprehensive Foundation** (Haykin) - *Teoria de base.*



Pré-requisitos & Preparação

- Conhecimento de Controle Automático.
- Familiaridade com Python (Obrigatório).
- Conforto com linha de comando Linux.

Setup do Workspace

Instalar Ubuntu 24.04 ou configurar WSL (Windows Subsystem for Linux) antes da primeira aula prática.

