



**Faculdade de Design,  
Tecnologia e Comunicação**  
Universidade Europeia

Project Factory 2025

Projeto: FTP Robotics

**Elementos do Grupo:**

Carla Ribeiro - 20220411

Julia Disconzi - 20220007

Tiago Costa – 20190887

**Repositório GitHub:** <https://github.com/Carlarib2/Project-Factory-PBL>

# *Índice*

Introdução	3
Projetos Relacionados	4
Levantamento de requisitos Hardware e Software	5
Análise e desenho da infraestrutura computacional	6
Esboço do artefacto físico	7
Diagrama de circuitos	8
Resumo diagrama	9
Lista de materiais necessários	10
Distribuição de tarefas	11 / 12
Plano de trabalho	13

# Introdução

Imagine um pequeno veículo autônomo navegando por um percurso cheio de desafios, rampas inclinadas, terrenos irregulares e obstáculos inesperados. Agora, acrescente a isso a necessidade de transportar diferentes tipos de cargas, como esferas, cubos e cilindros, sem deixar nada cair pelo caminho. Esse é o desafio que estamos assumindo neste projeto, unindo tecnologia, criatividade e trabalho em equipe para encontrar a melhor solução.

Este projeto faz parte das unidades curriculares de **Computação Física e IoT e Project Factory**, onde o objetivo é desenvolver um sistema inteligente capaz de controlar e monitorar o veículo em tempo real. Além disso, estamos colaborando com os estudantes de **Design de Produção**, que ficarão responsáveis pela criação da estrutura física que carregará as cargas.

Nosso veículo não será apenas um protótipo, mas um sistema completo que combina sensores, motores, algoritmos de controle e um painel interativo para acompanhar sua operação. Para enfrentar os obstáculos do percurso, ele precisará de estratégias de navegação eficientes, garantindo que consiga desviar de paredes móveis, manter estabilidade em superfícies irregulares e até lidar com rampas que podem fazê-lo saltar.

O desenvolvimento será feito em três fases principais:

**Exploração e Planejamento** – Onde vamos estudar as melhores tecnologias, entender os desafios e definir o design do veículo.

**Construção e Testes** – Vamos colocar a mão na massa, montar o hardware, programar os controles e realizar os primeiros experimentos.

**Otimização e Ajustes Finais** – Depois de testar, é hora de melhorar o desempenho, corrigir falhas e deixar o sistema pronto para os desafios finais.

Ao longo do projeto, vamos trabalhar com metodologias ágeis, garantindo flexibilidade para adaptações e melhorias conforme testamos nossas ideias. No final, esperamos ter um veículo autônomo funcional, pronto para enfrentar qualquer obstáculo e transportar cargas com eficiência.

# *Projetos relacionados*

Serve Robotics : robôs autônomos que entrega comida em calçadas, onde utiliza sensores LiDAR e GPS



---

Amazon Scout: Também robôs autônomos de entregas, que utilizam câmeras para a navegação autônoma.



---

Shifu Ramen: Restaurante Asiático usa robot para auxílio na entrega dos pedidos dos clientes, usando uma câmera para identificar a mesa do pedido que leva.



# *Levantamentos e requisitos*

## **Hardware :**

### Placa Controladora

ESP32 : Processador principal do sistema, responsável pelo controle do veículo e comunicação sem fio.

### Sensores

MPU-6050 (Acelerômetro + Giroscópio) : Mede a orientação, rotação e rotação do veículo.

HC-SR04 (Sensor Ultrassônico) : Mede a distância para evitar obstáculos.

Sensores Infravermelhos : Detectam linhas e objetos próximos ao veículo.

### Motores e Controle

4 Motores DC : Responsáveis pela locomoção do veículo.

Driver de Motor TB6612 : Controla os motores DC, permitindo ajuste de velocidade e direção.

### Fonte de Alimentação

Bateria Li-Po : Fornece energia para todo o sistema.

Regulador de Tensão : Converte 7,4V da bateria para 5V e 3,3V.

### Componentes Adicionais

Jumpers e Conectores : Para interligação entre os componentes.

Placa Perfurada : Para montagem dos circuitos.

Divisor de Tensão (Resistores 1kΩ e 2kΩ) : Para ajustar o nível de tensão no pino ECHO do HC-SR04.

## **Software:**

### Ambiente de Desenvolvimento

Arduino IDE: Utilizado para programar e testar o código do ESP32.

EasyEDA : Ferramenta para criar o diagrama do circuito elétrico.

MQTT (Message Queuing Telemetry Transport) : Protocolo de comunicação para troca de dados entre o veículo e um servidor na nuvem.

### Bibliotecas Necessárias

Wire.h (I2C): Comunicação com o MPU-6050.

Adafruit\_MPU6050: Facilita a leitura dos dados do acelerômetro e giroscópio.

NewPing.h: Para medição de distância com o HC-SR04.

WiFi.h: Para conexão com redes Wi-Fi.

PubSubClient.h: Para comunicação MQTT.

### Funcionalidades do Software

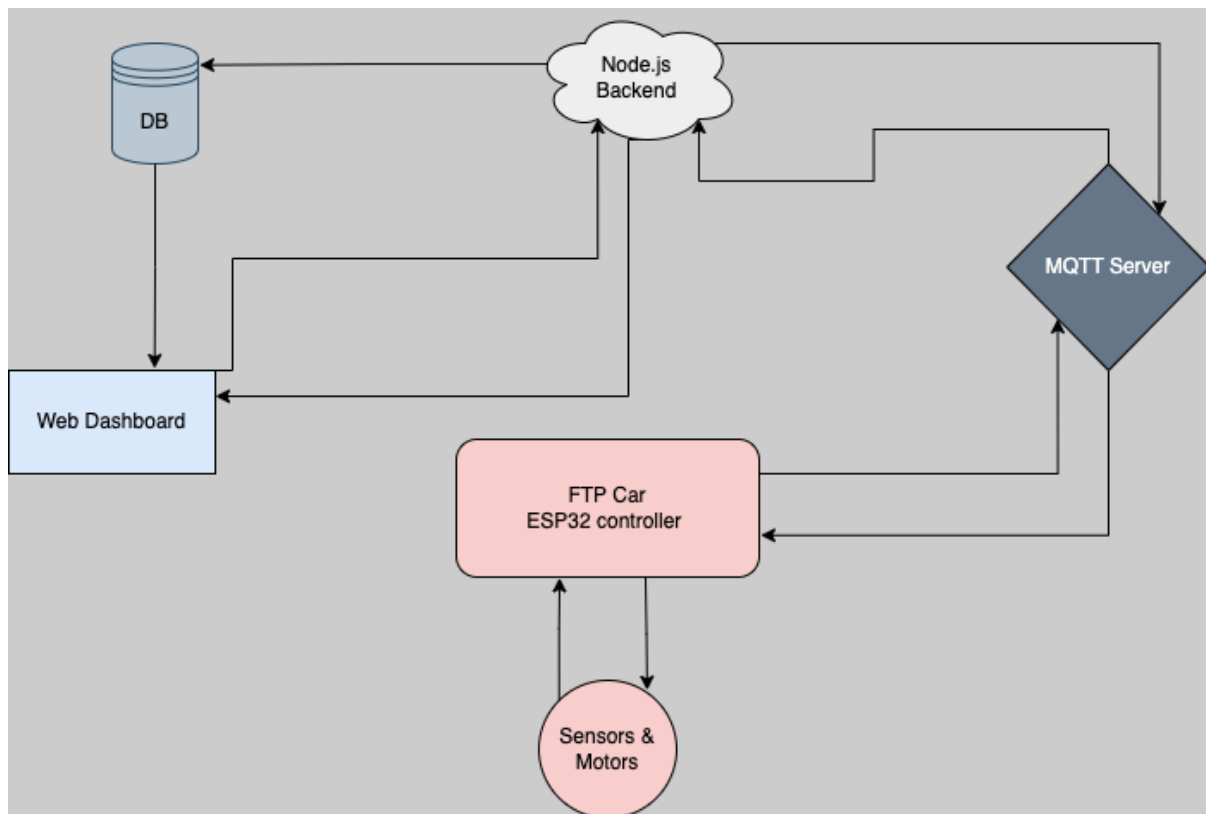
Leitura de Sensores: Captura dados dos sensores ultrassônicos, infravermelhos e do MPU-6050.

Processamento e Decisão: Identificação de obstáculos e ajuste de rota.

Controle dos Motores: Ajuste da velocidade e direção dos motores via driver TB6612.

Comunicação via MQTT: Envio de dados do veículo para um servidor remoto.

# Análise e design da infraestrutura computacional



O nosso sistema consiste num **carro telecomandado com um ESP32** que recolhe dados dos sensores e controla os motores. Para comunicar com o resto do sistema, usamos um **servidor MQTT**, que funciona como intermediário na troca de mensagens entre o carro e o backend.

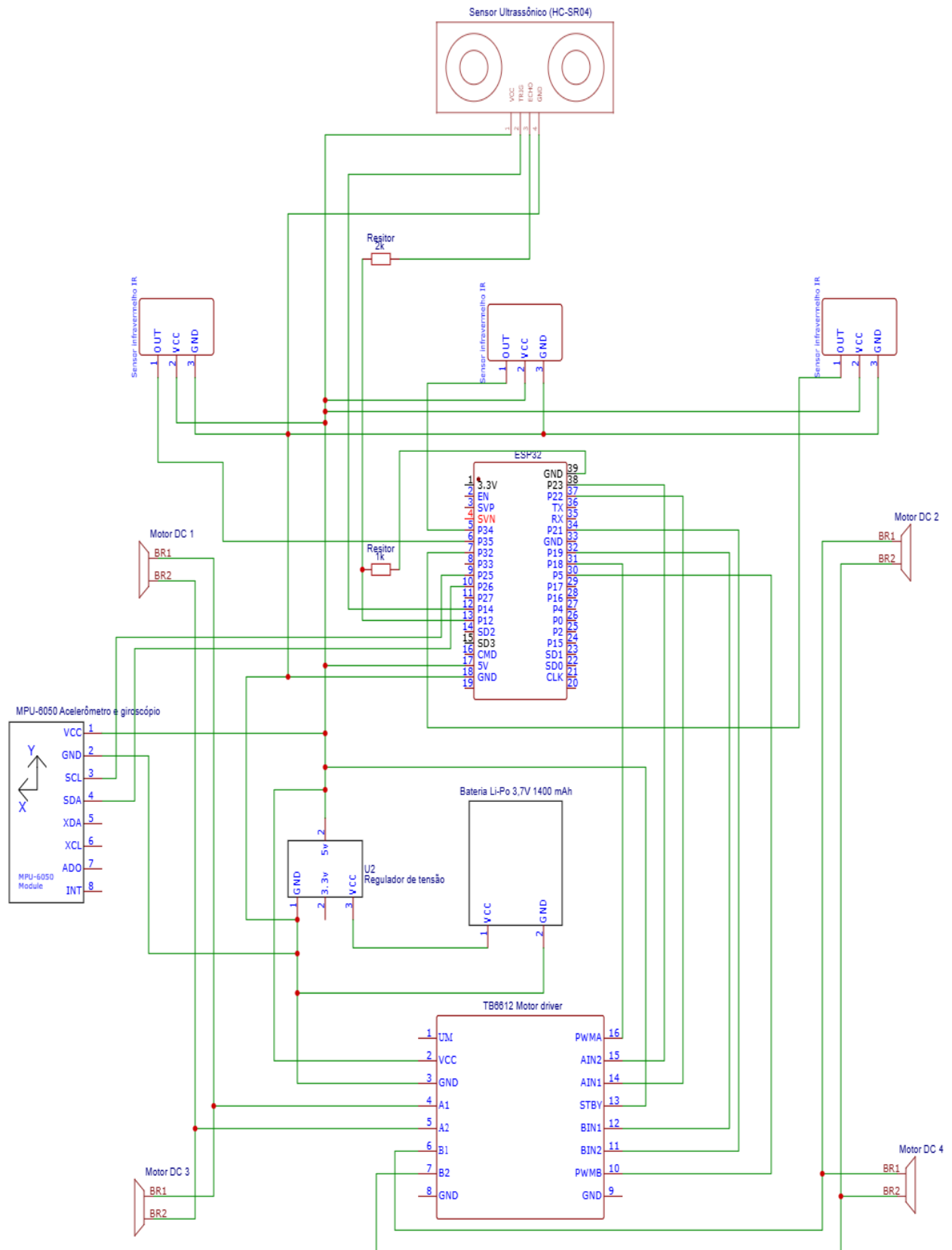
O **ESP32** envia informações dos sensores (como distância e orientação) para o **servidor MQTT**, que depois reencaminha esses dados para o **backend em Node.js**. Este backend processa os dados, guarda-os numa **base de dados** e pode também enviar comandos de volta para o ESP32, como ajustar a velocidade ou desviar de obstáculos.

A interação com os utilizadores acontece através de uma **dashboard web**, onde podemos visualizar os dados do carro em tempo real. Esta dashboard comunica com o backend para obter informações e apresentar o estado do veículo de forma acessível.

## Esboço do artefacto físico



# Diagrama de circuitos





# *Resumo diagrama*

- ✓ ESP32: Microcontrolador principal, processa sensores e controla motores.
- ✓ Sensor HC-SR04: Mede a distância até obstáculos usando ultrassom.
- ✓ Sensor IR (Infravermelho): Detecta linhas e obstáculos próximos.
- ✓ MPU6050: Sensor de movimento (Acelerômetro + Giroscópio) para equilíbrio e navegação.
- ✓ Motores DC: Movem as rodas do robô.
- ✓ TB6612: Driver de motores que recebe comandos do ESP32 e controla os motores DC.
- ✓ Regulador de Tensão: Reduz a tensão da bateria para níveis seguros (5V/3.3V).
- ✓ Resistores: Usados para limitar corrente e proteger circuitos.
- ✓ Bateria : Fonte de energia do robô (7.4V LiPo).

# *Lista de materiais*

- ✓ Chassi
- ✓ 4 motores DC com redutor
- ✓ 4 rodas
- ✓ Driver de motor TB6612
- ✓ ESP32
- ✓ MPU-6050
- ✓ 1 sensor HC-SR04
- ✓ 3 Sensores IR
- ✓ Bateria Li-Po 7.4V 5000mAh
- ✓ Regulador de Tensão DC-DC Buck Converter (XL4015) de 5A
- ✓ Jumpers macho-macho, macho-fêmea, fêmea-fêmea
- ✓ Breadboard

# *Distribuição de tarefas*

Milestone 1 – Semana 5 (Entrega: 6 de março de 2025)

Objetivo: Estruturar o projeto, definir requisitos e apresentar a proposta.

Tarefas:

- Pesquisa de projetos semelhantes : Carla
- Levantamento de requisitos (hardware e software): Júlia
- Desenho da infraestrutura computacional: Tiago
- Esboço do artefacto físico (protótipo) : Carla
- Diagramas de circuitos : Júlia
- Lista preliminar de materiais necessários : Todos
- Plano de trabalho e distribuição de tarefas : Tiago
- Preparação dos slides da apresentação: Carla
- Relatório de trabalho: Todos
- Código inicial do ESP32 e backend : Todos

Milestone 2 – Semana 10 (Entrega: 10 de abril de 2025)

Objetivo: Desenvolver e testar a infraestrutura computacional.

Tarefas:

- Montagem do circuito físico : Carla
- Programação do ESP32 (sensores, motores, comunicação MQTT) : Tiago
- Desenvolvimento do backend (MQTT Server, API, base de dados) : Júlia
- Criação da dashboard web : Carla
- Testes individuais dos módulos (sensores, motores, comunicação) : Todos
- Relatório sobre a arquitetura implementada e funcionalidades : Júlia
- Gravação e edição do vídeo de demonstração : Tiago
- Preparação dos slides para apresentação : Carla

Milestone 3 – Semana 14 (Entrega: 29 de maio de 2025)

Objetivo: Finalizar o sistema e preparar a demonstração final.

Tarefas:

- Integração final do sistema (ESP32 + Backend + Dashboard) : Todos
- Testes finais e ajustes : Todos
- Documentação completa do projeto : Júlia
- Gravação e edição do vídeo final : Tiago
- Preparação dos slides para a apresentação final : Carla

# Plano de trabalho

