AutoCargo - Project Factory

Licenciatura em Engenharia Informática

2024/2025

Grupo Nº5:

20220523 - Abhay Kumar

20210142 - António Santos

20220672 - Mário Igreja

Docente: André Sabino

Índice

Introdução

Com base no briefing recebido, o objetivo do projeto é construir um veículo com condução manual e autónoma capaz de transportar carga de um ponto a outro, ultrapassando os obstáculos presentes no percurso.

Para alcançarmos esse propósito, iremos desenvolver um veículo manual e autónomo, equipado com um microcontrolador ESP32 e múltiplos sensores. O veículo suportara condução manual onde ele é controlado através de um Dashboard Web, ou através de condução autónoma que poderá ser ativada no Dashboard Web, no modo autónomo o veículo utilizara os seus sensores para detetar e superar os obstáculos, medir a inclinação da rampa, calcular a velocidade e monitorizar o nível de bateria.

O objetivo principal é criar um veículo capaz de transportar carga de forma manual ou autónoma de maneira eficiente e segura, tendo a capacidade de navegar pelo ambiente e levar a carga até ao fim do percurso.

Levantamento de requisitos

Requisitos Funcionais

- **Web Dashboard** - Implementação de um dashboard que permita controlar e visualizar os dados fornecidos pelo ESP32.
- **Controlo Manual** – O veículo deve ser capaz de ser controlado de maneira manual, através do Dashboard Web.
- **Controlo Autónomo** – Deve ser possível ativar o modo autónomo do veículo através do dashboard Web.
- **Transporte de carga** – O veículo deve ser capaz de transportar carga do início ao fim do percurso.

Requisitos não funcionais

- **Usabilidade** – O dashboard deve ser intuitivo e fácil de usar.

- **Segurança** - O dashboard deve apenas poder ser acessado por utilizadores autorizados.

- **Performance** – O ESP32 deve conseguir mandar os dados de maneira rápida e eficiente para o backend.

- **Fiabilidade** – O ESP32 deve detetar com exatidão os obstáculos utilizando os seus sensores.

Analise e desenho da infraestrutura computacional

Design do Sistema

Frontend – Dashboard Web onde o utilizador poderá controlar o veículo e ver a informação enviada pelo ESP32.

Backend - Irá receber pedidos do web server, comunicar com o ESP32 através de MQTT e interagir com a base de dados.

Base de dados – Armazena os dados enviados pelo ESP32 e sensores.

MQTT – Faz a comunicação entre o backend e ESP32, fazendo publish e subscribe.

ESP32 – Microcontrolador que sera programado em C++, coleta dados dos sensores e controla os motores e as rodas.

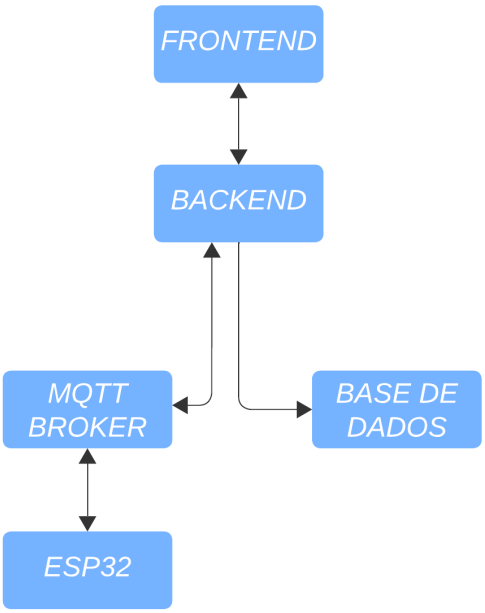

Fig. 1 Diagrama do Design do Sistema

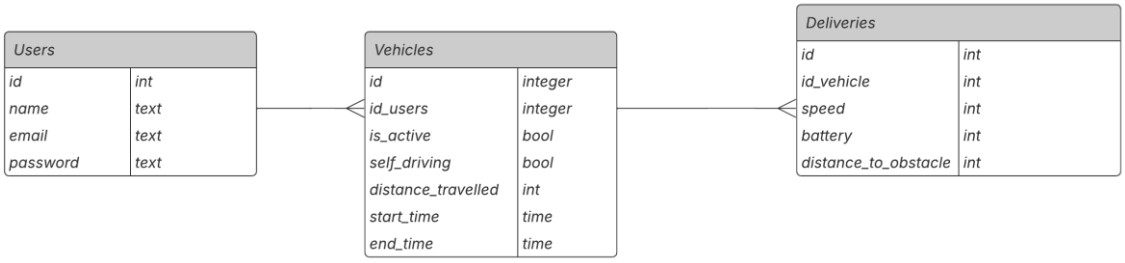

Fig. 2 Diagrama ER

Descrição da arquitetura implementada

A estrutura foi pensada na eficacia e facilidade de uso. O véiculo é controlado através de um Dashboard Web, e utiliza sensores para detetar os obstáculos, medir inclinação da rampa, calcular a velocidade e monitorizar o nível de bateria.

Temos um Esp32 responsável pela recolha dos dados dos sensores e controlar os atuadores do veículo, ele está a ser programado usando o Arduino IDE. A comunicação entre o Esp32 e o backend é realizada usando o protocolo MQTT com recurso ao broker HiveMQ.

O Backend desenvolvido em Nodejs, possui 2 camadas principais, uma REST API para que trata do registo e login, e uma camada de comunicação em tempo real

com o frontend via sockeet.io, que transmite os dados dos sensores e comandos do utlizador através do broker MQTT, no login e registo utilizamos Lex e Yacc para realizar a análise léxica e sintática das entradas do utilizador para garantir que eles seguem as regras que nós estabelecemos.

Base de Dados armazena as informações dos utilizadores e dos sensores, ela está a ser implementada utilizando Docker Compose.

Frontend feito usando Vite com React, serve para desenvolver o Dashboard Web que permite o utilizador fazer Registo, Login ou controlar o veículo seja de maneira manual ou autónoma, o dashboard também exibe dados em tempo real trasmitidos pelo Esp32.

Esboço do artefacto físico a construir

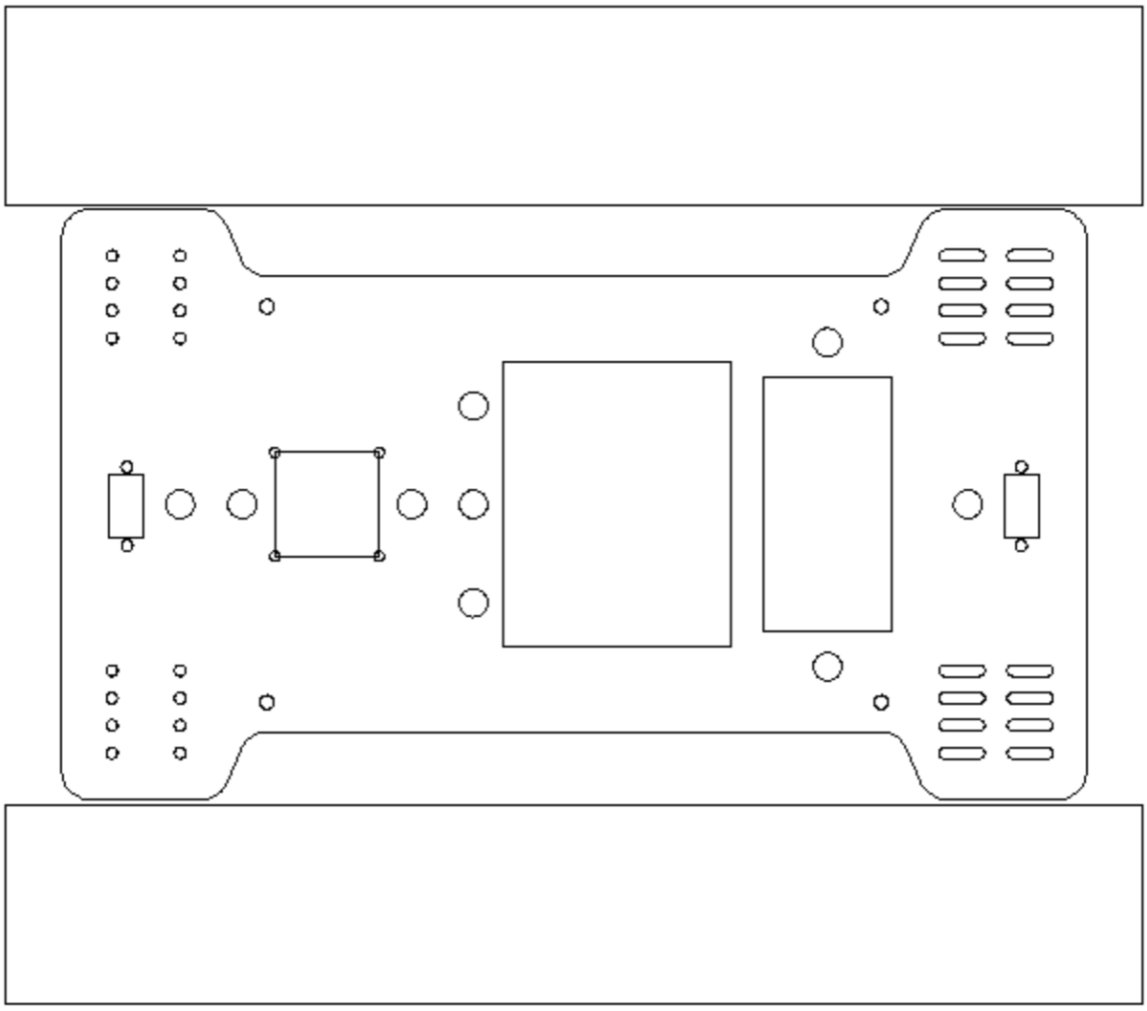

Fig. 3 Design do chassi e lagarta

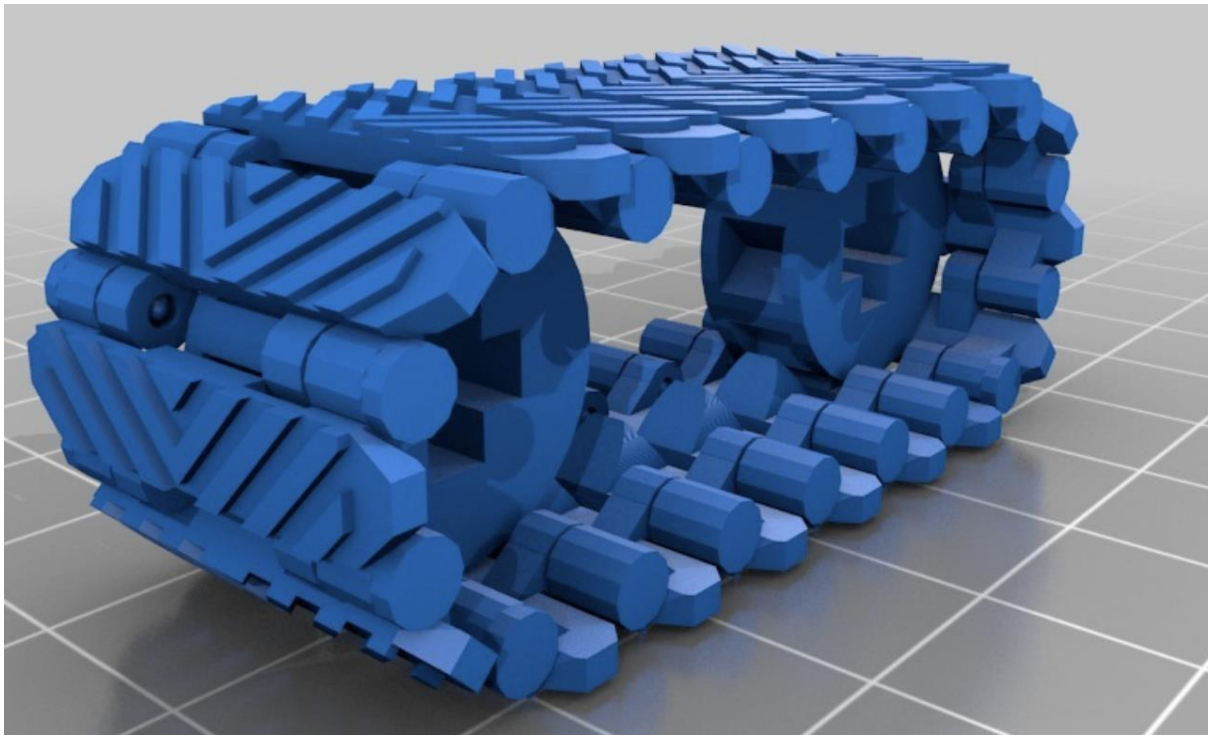

Fig. 4 Imagem da Lagarta

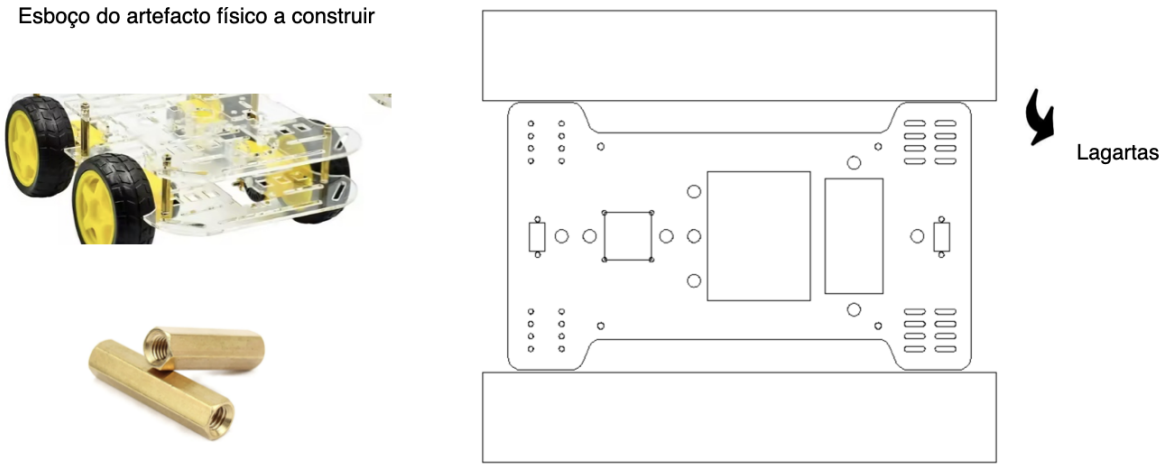

Fig. 6 Imagem do Diagrama de circuito

Para o transporte de carga, iremos implementar uma plataforma fixa ao nosso veículo, ela será acoplada ao nosso chassi, utilizando suportes metálicos com uma altura de 4 centímetros, isso garante uma distância adequada para evitar interferência com as componentes presas ao chassi.

A plataforma será feita em madeira e ela terá furos de encaixe para permitar prender o suporte de carga projetado pelos grupos de design.

Diagramas de circuitos

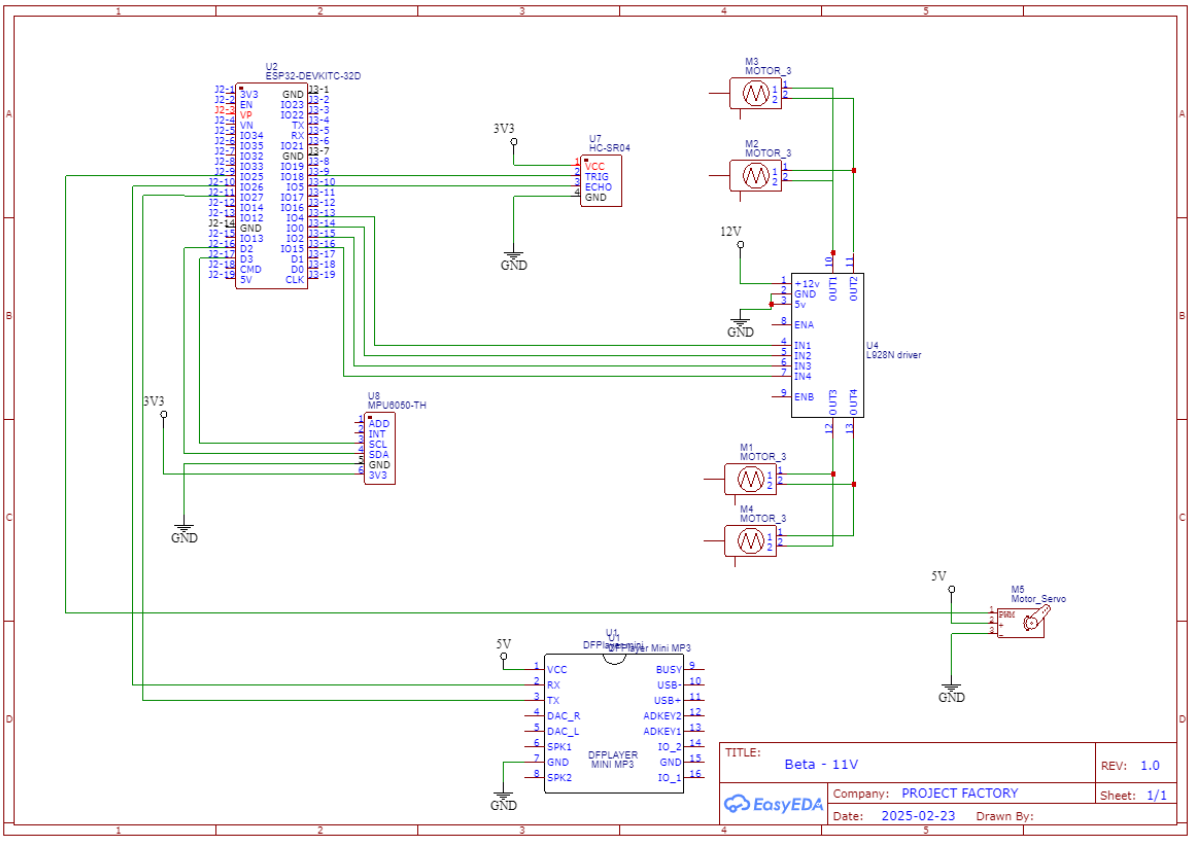

Fig. 6 Imagem do Diagrama de circuito

Lista de material necessário

- MP3 Audio player
- ESP32 microcontrolador
- Breadboard
- Pilhas
- Suporte para pilhas
- Driver Motor L298N
- Motor elétrico
- Lagartas
- Sensor acelerômetro e giroscópio MPU-6500

- Sensor ultrassônico HC-SR04
- Jumpers
- Colunas
- Sensor de Voltagem

Enquadramento com outras UCs

Compiladores - Implementação de lex e yacc no Login e Register para os dados estejam no formato desejado antes de serem enviados para o backend.

Ética e Deontologia Profissional – Faremos a análise ética do processo de construção do veículo, considerando princípios de responsabilidade ambiental, transparência e segurança. Avaliaremos a utilidade do sistema para a sociedade, o impacto ambiental e o uso consciente da tecnologia, a proteção de dados dos usuários e os possíveis riscos associados à automação.

Descrição de atividades realizadas

Ao longo dos últimos 2 meses, concentrámo-nos no desenvolvimento da estrutura computacional e física. Nós decidimos dividir as tarefas para o desenvolvimento ser mais rápido e organizado.

Inicialmente decidimos no focar na criação do Dashboard Web, na comunicação entre o Esp32 e o backend através do protocolo MQTT utilizando o broker HiveMQ.

Após termos a comunicação finalizada, demos início à implementação do controlo manual e autónomo em paralelo, na condução manual o utilizador envia comandos pelo dashboard para o Esp32 através do backend e broker MQTT, enquanto na condução autónoma assim que o código é enviado para o Esp32 o veículo começa a usar o sensor ultrassónico para detetar se há obstáculos a serem desviados.

Com a condução manual e autónoma em andamento, realizamos reuniões com os grupos de design, onde em conjunto definimos o formato e dimensões do veículo, com o design feito nós pegamos contraplacado para a montagem do chassi do veículo, com o chassi feito iniciamos a construção das lagartas do veículo, onde foram feitas usando impressão 3D.

Distribuição de tarefas

Para a distribuição nós utilizamos o Jira, e a divisão foi feita da seguinte maneira:

Abhay	António	Mário
Esboço do veículo a construir	Desenvolvimento do MQTT	Esboço do veículo a construir
Construção do Dashboard Web	Desenvolvimento da condução manual	Construção do diagrama de circuitos.
Slides da apresentação.	Modelação da Base de Dados	Desenvolvimento da condução autónoma

Bibliografia

Manjunathan, A., Kumar, A. S., Udhayanan, S., Selvi, C. T., & Stonier, A. A. (2021). Design of Autonomous Vehicle Control using IoT. *IOP Conference Series Materials Science and Engineering*, *1055*(1), 012008. https://doi.org/10.1088/1757-899x/1055/1/012008

Obstacle avoiding robot. (n.d.). Arduino Project Hub. https://projecthub.arduino.cc/aakash11/obstacle-avoiding-robot-8018ae

Rakesh-I. (n.d.). *GitHub - rakesh-i/ESP32-Autonomous-car: Autonomous car using ESP32.* GitHub. https://github.com/rakesh-i/Autonomous-car/

Obstacle Detection and Avoidance Robot with Multi-Control System Using IoT. (2023, November 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/10369630

Hackster.io. (2023, November 29). *IoT Robot Car using MQTT*. https://www.hackster.io/virgilvox/iot-robot-car-using-mqtt-2c8c55

Obstacle avoiding robot using ESP32. (n.d.). Obstacle Avoiding Robot Using ESP32. https://www.robotique.tech/robotics/obstacle-avoiding-robot-using-esp32/

Bernstein, C., Brush, K., & Gillis, A. S. (2021, January 27). *MQTT (MQ Telemetry Transport)*. Search IoT. https://www.techtarget.com/iotagenda/definition/MQTT-MQ-Telemetry-Transport

Team, H. (2024, February 14). Introducing the MQTT Protocol – MQTT Essentials: Part 1. *HiveMQ*. https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/

Manandhar, G. (2021, December 25). Geshan's Blog. *Geshan's Blog*. https://geshan.com.np/blog/2021/12/docker-postgres/

ESP32 - Ultrasonic Sensor. (n.d.). ESP32 Tutorial. https://esp32io.com/tutorials/esp32-ultrasonic-sensor

ArduinoGetStarted.com. (2025, March 29). *How to create Class and Object on Arduino IDE | Arduino FAQs*. Arduino Getting Started. https://arduinogetstarted.com/faq/how-to-create-class-and-object-on-arduino-ide

ESP32IO.com. (2025). *ESP32 - MP3 Player | ESP32 Tutorial*. ESP32 Tutorial. https://esp32io.com/tutorials/esp32-mp3-player

Shehab Magdy. (2024, August 16). *Connecting ESP32 to HiveMQ for Real-time IoT Data Streaming with MQTT*. Medium; Engineering IoT. https://medium.com/engineering-iot/connecting-esp32-to-hivemq-for-real-time-iot-data-streaming-with-mqtt-8813f48cb1a4

ESP32 MQTT Publish Subscribe with Arduino IDE | Random Nerd Tutorials. (2018, June 13). Randomnerdtutorials. https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/

Lombardo, P. G. (2023, May 16). *The Ultimate Guide on How to Use MQTT with Node.js*. Hivemq.com; HiveMQ. https://www.hivemq.com/blog/ultimate-guide-on-how-to-use-mqtt-with-node-js/

Thingiverse.com. (2025). *Treads by enrohtkcalb*. Thingiverse. https://www.thingiverse.com/thing:803058

Line Following Robot. (n.d.). Projecthub.arduino.cc. https://projecthub.arduino.cc/lightthedreams/line-following-robot-34b1d3

Using lex with yacc. (2025). Sco.com. http://osr600doc.sco.com/en/SDK_tools/_Using_lex_with_yacc.html

LEX-YACC. (2025). Github.io. https://silcnitc.github.io/ywl.html

Example program for the lex and yacc programs. (n.d.). Www.ibm.com. https://www.ibm.com/docs/en/aix/7.1?topic=information-example-program-lex-yacc-programs