

**RELATÓRIO DO PROJETO ELABORADO NO ÂMBITO DA PROVA  
DE APTIDÃO PROFISSIONAL**

**2023/2024**

**Curso Profissional de Técnico de Gestão e Programação de  
Sistemas Informáticos**

**RaspiPy**

**Cristiano David Paulitos Martinho**

**Professor António Travassos**

**2024**

*Dedico este projeto á minha Família,  
Namorada  
e Amigos.*

## Índice

<b>Lista de abreviaturas, siglas e acrónimos .....</b>	<b>6</b>
<b>Hardware Utilizado .....</b>	<b>7</b>
<b>Software utilizado.....</b>	<b>8</b>
Editores de códigos.....	8
Linguagens de Programação .....	8
Bibliotecas .....	8
Servidor .....	8
Base de Dados .....	8
Outros .....	8
<b>Sumário .....</b>	<b>9</b>
<b>Introdução .....</b>	<b>10</b>
<b>Objetivo da minha PAP .....</b>	<b>11</b>
<b>Robótica .....</b>	<b>12</b>
História e Evolução da Robótica .....	12
Tipos de Robôs .....	13
<b>RaspberryPi .....</b>	<b>14</b>
Principais Características do RaspberryPi .....	14
Usos Comuns do RaspberryPi .....	14

<b>Software .....</b>	<b>15</b>
Editores de código.....	15
Linguagens de programação.....	15
Bibliotecas .....	17
Servidor .....	18
Base de Dados .....	19
Outros.....	19
<b>Projeto – RaspyPi .....</b>	<b>21</b>
Programação e Desenvolvimento Web .....	21
Processamento e Reconhecimento de Voz.....	24
Identificação de Objetos .....	25
<b>Fases de Realização do Projeto .....</b>	<b>27</b>
Conceção .....	27
Implementação .....	27
Testes e Ajustes.....	28
<b>Métodos de Trabalho.....</b>	<b>30</b>
<b>Produto Final .....</b>	<b>32</b>
<b>Dificuldades Encontradas .....</b>	<b>34</b>
Problemas com Cartões SD .....	34
Configuração do Servidor.....	34
Integração da Base de Dados .....	34

Dificuldade com Manipulação Manual de Arquivos .....	35
Controlo por Teclado dos Motores do Robô .....	37
Problemas com o Cooler do RaspberryPi .....	38
Problemas com Câmera RaspberryPi .....	38
<b>Conclusão .....</b>	<b>40</b>
<b>Webgrafia .....</b>	<b>42</b>

## **Lista de abreviaturas, siglas e acrónimos**

PHP - Hypertext Preprocessor

CSS - Cascading Style Sheet

SQL - Structured Query Language

USB - Universal Serial Bus

API - Application Programming Interface

OpenCV - Open Source Computer Vision Library

YOLO – You Only Look Once

RPNs - Region Proposal Networks

GPIO - General Purpose Input/Output

NCSA - National Center for Supercomputing Applications

HTTP - Hypertext Transfer Protocol

HTTPS - Hyper Text Transfer Protocol Secure

SSL - Transport Layer Security

TCP - Transmission Control Protocol

R.U.R. - Rossum's Universal Robots

HDMI - High-Definition Multimedia Interface

IoT – Internet of Things

ARM - Acorn RISC Machine

RISC - Reduced Instruction Set Computer

DIY - Do-It-Yourself

PWM - Pulse-Width Modulation

## Hardware Utilizado

RaspberryPi Model 4-B - <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

Logitech C270 Webcam - [https://www.amazon.es/dp/B01BGBJ8Y0/ref=asc\\_df\\_B01BGBJ8Y0/?tag=ptgogshpadde=21-&linkCode=d0f&hvadid=634520192123&hvpos=&hvnetw=g&hvrnd=15248836049482578227&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1011735&hvtargid=pla-471812619030&psc=1&language=pt\\_PT&mcid=2ad533a0c39e3917b7eff-283d58e894c](https://www.amazon.es/dp/B01BGBJ8Y0/ref=asc_df_B01BGBJ8Y0/?tag=ptgogshpadde=21-&linkCode=d0f&hvadid=634520192123&hvpos=&hvnetw=g&hvrnd=15248836049482578227&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1011735&hvtargid=pla-471812619030&psc=1&language=pt_PT&mcid=2ad533a0c39e3917b7eff-283d58e894c)

2 motores ZYTD520 - <https://www.direnc.net/zytd520-6v-12v-24v-10rpm-37mm-reduktorlu-dc-motor-en>

Mini Microfone USB - [https://www.amazon.es/dp/B08BHM9VVW/ref=asc\\_df\\_B08BHM9VVW/?tag=ptgogshpadde=21&linkCode=df0&hvadid=666584406634&hvpos=&hvnetw=g&hvrnd=17348622811400214567&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1011735&hvtargid=pla-1158809861748&psc=1&language=pt\\_PT&mcid=741ee9fef733431963cd0b0af11baf2](https://www.amazon.es/dp/B08BHM9VVW/ref=asc_df_B08BHM9VVW/?tag=ptgogshpadde=21&linkCode=df0&hvadid=666584406634&hvpos=&hvnetw=g&hvrnd=17348622811400214567&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1011735&hvtargid=pla-1158809861748&psc=1&language=pt_PT&mcid=741ee9fef733431963cd0b0af11baf2)

Controlador dos motores - <https://www.botnroll.com/pt/controladores/3363-m-dulo-driver-motores-l298n.html>

Powerbank/Fonte de Alimentação - <https://www.worten.pt/produtos/powerbank-goodis-20-000-mah-micro-usb-usb-c-preto-7320994>

Fonte de Alimentação para os motores - <https://www.botnroll.com/pt/bot-n-roll-one-a-1003-bateria-12v-ni-mh-aa-800mah-wwwbotnrollcom.html>

## **Software utilizado**

### **Editores de códigos**

- Visual Studio Code
- Thonny

### **Linguagens de Programação**

- Python
- PHP
- CSS
- SQL
- JavaScript
- HTML

### **Bibliotecas**

- OpenCv
- YOLO
- RPI.GPIO
- Flask

### **Servidor**

- Apache

### **Base de Dados**

- MariaDB

### **Outros**

- WebSockets
- Node.js



## Sumário

O objetivo do projeto foi desenvolver um pequeno robô capaz de interagir com os utilizadores através de comandos de voz, controlo do robô e utilização de uma pequena câmara para fotos e até identificação de objetos, oferecendo uma experiência envolvente e educativa. Este robô tem potencial para ser utilizado não apenas para entretenimento, mas é até uma oportunidade de adaptação simplificada às novas tecnologias, perfeita, por exemplo, para ajudar idosos com a inclusão digital.

O robô pode encontrar utilidade em ambientes domésticos, educacionais e em instituições de apoio a idosos, proporcionando diversão, assistência e oportunidades de aprendizagem. Além disso, o robô também poderia ser utilizado em ambientes de saúde, como clínicas e hospitais, para proporcionar entretenimento e interação com os pacientes, melhorando assim a sua experiência durante a estadia.

Com um objetivo de aprimorar as suas capacidades de comunicação e interação, poderia permitir que o robô oferecesse serviços de assistência virtual em diversas áreas, tais como, fornecimento de informações úteis, auxílio em tarefas simples do dia-a-dia e até mesmo apoio emocional.

## Introdução

A Prova de Aptidão Profissional (PAP), é um projeto prático com respetiva apresentação e defesa perante um júri, que todos os estudantes de cursos profissionais têm de realizar no final do mesmo, para a obtenção de um diploma de qualificação profissional. Deve centrar-se em temas e problemas afins aos contextos de trabalho e realiza-se sob orientação e acompanhamento de um ou mais professores. A PAP implica a planificação e desenvolvimento ou execução de um projeto prático baseado nas competências mais significativas do curso, neste caso, do curso profissional Gestão e Programação de Sistemas Informáticos.

A minha Prova de Aptidão Profissional consiste em um pequeno robô, com um RaspberryPi como o seu núcleo. O robô tem dois motores, com duas rodas ligadas a eles e os motores, por sua vez, estão conectados a um controlador de motores que está ligado ao RaspberryPi. Ligado ao RaspberryPi, também encontramos um pequeno microfone USB e uma câmera Logitech também USB.

O RaspberryPi foi usado para criar toda a lógica e programação do robô e com isso ele consegue abrir o seu próprio servidor na rede a qual está conectada que dá acesso a uma base de dados simples e a um site onde podemos não apenas controlar os motores do robô, mas também, conseguimos visualizar uma gravação em tempo real da câmera para conseguirmos controlar o robô de longe ou até mesmo se ele não estiver no nosso campo de visão. Conseguimos, ainda, tirar fotografias que nos serão disponibilizadas para download, segundos após a fotografia ser tirada, onde estas são devolvidas com identificação de objetos em tempo real, ou seja, se for tirada uma foto a uma pessoa, quando fazemos o download, veremos escrito na foto que está presente uma pessoa. É possível também fazer perguntas ao robô com as nossas próprias vozes, com o clique de um botão o microfone começará uma gravação que depois será traduzida para texto através de uma API. Por sua vez, esse texto será de seguida enviado para outra API que nos devolverá uma resposta a pergunta feita.

Por último, mas não menos importante, a cada 5 minutos é enviado para a base de dados uma lista de informações com base no desempenho do RaspberryPi naquele momento, como por exemplo, a temperatura, a percentagem de uso do CPU, entre outras informações, permitindo, assim, uma monitorização constante.

Para tudo isto se concretizar, eu tive que trabalhar bastante para conseguir aprender a trabalhar em um RaspberryPi e entender muito bem a lógica da programação que fiz para o robô funcionar, tal como o tinha idealizado.

## **Objetivo da minha PAP**

Com a escolha deste projeto para a minha PAP, tive como objetivo aprender a trabalhar com novos softwares, linguagens de programação e até com o próprio RaspberryPi. Nunca tinha trabalhado com alguns dos programas que usei para a construção e programação deste robô. Com este projeto, desafiei-me a sair da minha área de conforto e a trabalhar com algo novo para mim e, quem sabe, até encontrar um novo gosto, que neste caso seria a robótica.

Considero que foi muito importante para mim a escolha deste projeto, pois não só aprendi técnicas que levarei para o futuro, mas também aprendi que, se me esforçar a fazer algo de que gosto, consigo chegar a lugares extraordinários. Aprendi sozinho métodos de trabalho perfeitos para mim, que com certeza irei usar para trabalhos e projetos futuros, de modo a conseguir sempre os melhores resultados possíveis.

## Robótica

### História e Evolução da Robótica

A história da robótica é um testemunho do progresso tecnológico e da engenhosidade humana, com as suas raízes a remontar à antiguidade. Aqui está uma visão geral da evolução da robótica:

- **Antiguidade e Idade Média:** As primeiras tentativas de criar autómatos podem ser encontradas em relatos históricos de civilizações antigas. Por exemplo, o matemático grego Heron de Alexandria, no século I d.C., criou vários dispositivos mecânicos, incluindo um teatro automático. Na Idade Média, inventores árabes como Al-Jazari construíram autómatos para diversas funções, desde entretenimento até tarefas práticas.
- **Século XVIII e XIX:** Durante a Revolução Industrial, houve um aumento significativo no interesse por máquinas automáticas. Jacquard, um tecelão francês, desenvolveu um tear automático que utilizava cartões perfurados para controlar padrões de tecido, um precursor dos computadores modernos.
- **Século XX:** A robótica moderna começou a tomar forma no início do século XX. Em 1921, a peça “R.U.R.” de Karel Čapek introduziu a palavra “robô”. Entre os anos 1940 e 1950, os primeiros computadores eletrónicos foram desenvolvidos, fornecendo uma base para a robótica digital.
- **Robótica Industrial:** O primeiro robô industrial, Unimate, foi criado por George Devol em 1954 e instalado numa linha de produção da General Motors em 1961. Este robô revolucionou a indústria automóvel, realizando tarefas repetitivas com precisão e eficiência.
- **Robótica Moderna:** Desde os anos 1980, a robótica expandiu-se para além da indústria, entrando em áreas como medicina, exploração espacial e robôs de serviço doméstico. Avanços em inteligência artificial, sensores e materiais permitiram a criação de robôs mais versáteis e autónomos. Exemplos incluem o robô explorador de Marte, Sojourner, lançado em 1997, e robôs domésticos como o aspirador Roomba, lançado em 2002.

## Tipos de Robôs

Os robôs são classificados em várias categorias, dependendo da sua aplicação e construção:

- **Robôs Industriais:** Projetados para realizar tarefas repetitivas e precisas em ambientes de fabrico. São comuns em linhas de produção automóvel, eletrónica, e outras indústrias pesadas. Exemplos incluem braços robóticos utilizados para soldagem, pintura e montagem.
- **Robôs de Serviço:** Projetados para interagir com humanos e realizar tarefas úteis em ambientes domésticos, comerciais e de saúde. Incluem robôs como aspiradores automáticos, robôs de cozinha, e robôs de telepresença utilizados em hospitais.
- **Robôs Móveis:** Incluem robôs que podem deslocar-se autonomamente em ambientes variados. Exemplos são drones, veículos autónomos, e robôs de exploração que podem navegar por terrenos difíceis. Estes robôs utilizam uma variedade de sensores e algoritmos de navegação para se moverem de forma segura e eficiente.
- **Robôs Humanoides:** Projetados para imitar a forma e, em certa medida, o comportamento humano. São frequentemente utilizados em pesquisa, entretenimento e assistência pessoal. Exemplos incluem o ASIMO da Honda, que pode andar, correr e subir escadas, e o Pepper da SoftBank Robotics, que é capaz de interagir com pessoas e reconhecer emoções.
- **Robôs de Assistência Médica:** Utilizados em ambientes de saúde para realizar cirurgias precisas, fornecer assistência a pacientes ou ajudar na reabilitação. O Da Vinci Surgical System é um exemplo de robô cirúrgico que permite aos médicos realizar operações complexas com maior precisão.
- **Robôs de Exploração:** Desenvolvidos para explorar ambientes inacessíveis ou perigosos para os humanos, como o fundo do mar, o espaço ou áreas contaminadas. Robôs como o Curiosity Rover, que explora a superfície de Marte, são exemplos de robôs de exploração.

## RaspberryPi

O Raspberry Pi é um microcomputador de baixo custo e de tamanho reduzido, desenvolvido pela Raspberry Pi Foundation, com o objetivo de promover o ensino da informática básica em escolas e em países em desenvolvimento. Desde o seu lançamento em 2012, tornou-se extremamente popular tanto na educação quanto em projetos de hobby e profissionais devido à sua versatilidade, acessibilidade e grande comunidade de apoio.

### Principais Características do RaspberryPi

- **Processador:** Utiliza processadores ARM, que são conhecidos pela sua eficiência energética. Modelos mais recentes, como o RaspberryPi 5, vêm com um processador quad-core ARM Cortex-A76.
- **Memória:** Variantes com diferentes quantidades de RAM (1GB, 2GB, 4GB e até 8GB na versão mais recente).
- **Armazenamento:** Utiliza cartões microSD como principal meio de armazenamento, facilitando a instalação e troca de sistemas operativos.
- **Conectividade:** Equipado com portas USB, HDMI, GPIO, Ethernet, Wi-Fi e Bluetooth, permitindo uma ampla gama de aplicações e conexões.
- **Sistema Operativo:** Suporta várias distribuições Linux, com o Raspberry Pi OS (anteriormente conhecido como Raspbian) sendo a mais comum. Também é possível instalar sistemas operativos como Ubuntu, Windows 10 IoT Core e outros.

### Usos Comuns do RaspberryPi

- **Educação:** Utilizado em escolas e universidades para ensinar programação e conceitos de computação.
- **Projetos DIY:** Popular em projetos DIY, desde media centers com Kodi até estações meteorológicas e sistemas de automação residencial.
- **Prototipagem Rápida:** Utilizado por engenheiros e desenvolvedores para criar protótipos rápidos de novos produtos e conceitos.
- **Sistemas Embarcados:** Implementação em projetos de sistemas embarcados, como robótica, automação e monitoramento.

## Software

### Editores de código

- **Visual Studio Code** - O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código. Ele é customizável, permitindo que os usuários possam mudar o tema do editor, teclas de atalho e preferências. Ele é um software livre e de código aberto, apesar do download oficial estar sob uma licença proprietária. Neste projeto, o visual studio foi usado para a criação de todos os ficheiros de código HTML, CSS, PHP e JavaScript. Neles contém todo o código para o site que é usado para comandar o robô e aceder a todas as outras funcionalidades dele de forma acessível e intuitiva.
- **Thonny** - Thonny é um ambiente de desenvolvimento integrado gratuito e de código aberto para Python, projetado para iniciantes. Foi criado por Aivar Anna-maa, um programador estoniano. Ele suporta diferentes maneiras de percorrer o código, avaliação passo a passo de expressões, visualização detalhada da pilha de chamadas e um modo para explicar os conceitos de referências e heap. Thonny foi usado neste projeto para a criação de todos os ficheiros de código Python. Python foi usado neste projeto para aceder as configurações da câmera e, com a ajuda da biblioteca Flask, criar uma Flask app, que é uma pequena aplicação web junta a um pequeno servidor. Neste caso a Flask app é criada apenas para o streaming da câmera, que é disponibilizada depois no IP do RaspberryPi em uma porta.

### Linguagens de programação

- **Python:** Python é uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Atualmente, possui um modelo de desenvolvimento comunitário, aberto e gerido pela organização, sem fins lucrativos, Python Software Foundation. Python foi extremamente importante neste projeto. Sem ele, não seria possível ligar a câmera, tirar fotografias e devolvê-las com identificação de objetos em tempo real, fazer gravações de áudio, nem seria possível ligar e controlar os motores para a deslocação do robô.



- **HTML:** O HTML é o componente base da web. Isso quer dizer que ele permite a construção de websites e a inserção de novos conteúdos, como imagens e vídeos, por meio dos hipertextos. Os hipertextos são a junção de vários elementos (palavras, vídeos e conteúdos), que, quando conectados, estabelecem uma rede de dados que permite a comunicação, o armazenamento e o compartilhamento de informações. O HTML contribuiu apenas para o desenvolvimento do website neste projeto.
- **PHP:** PHP é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web. Figura entre as primeiras linguagens passíveis de inserção em documentos HTML, dispensando em muitos casos o uso de arquivos externos para eventuais processamentos de dados. O código é interpretado no lado do servidor pelo módulo PHP, que também gera a página web a ser visualizada no lado do cliente. O PHP, juntamente com o HTML, contribuiu para o desenvolvimento do website do projeto, mas com um pouco mais de impacto, o PHP foi utilizado para aceder a base de dados e nela inserir as informações sobre o comportamento das várias componentes do RaspberryPi. Também foi utilizado para incluir os ficheiros Python no site, permitindo, por exemplo, controlar os motores do robô apenas com um clique do teclado.
- **CSS:** CSS é usado para adicionar estilo a uma página HTML. O CSS separa o conteúdo da representação visual do site. Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também é possível criar tabelas, adicionar e ajustar imagens, usar variações de layouts e assim por diante. Sem o CSS, o site do projeto seria dificilmente interessante e até feio, pois seria apenas um ecrã branco com texto corrido em preto e botões sem forma.
- **SQL:** SQL é uma linguagem de domínio específico desenvolvida para gerir dados relacionais em um sistema de gerenciamento de bases de dados, ou para processamento de fluxo de dados em um sistema de gerenciamento de fluxo de dados. SQL serviu para fazer consultas à base de dados, como a inserção das informações na mesma ou a listagem das informações no site.



- **JavaScript:** JavaScript (frequentemente abreviado como JS) é uma linguagem de programação interpretada estruturada, de script em alto nível. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web. JavaScript permite páginas da Web interativas e, portanto, é uma parte essencial dos aplicativos da web. O JavaScript foi essencial para o projeto. No site, ele foi utilizado para a criação de animações, mas foi mais usado no backend do projeto, através da manipulação manual de ficheiros, para aceder as informações em tempo real da percentagem de uso do CPU, temperatura e a percentagem do uso de memória. Foi também muito útil para debug do sistema sempre que algum erro desconhecido aparecia.

## Bibliotecas

- **OpenCV:** OpenCV é uma biblioteca de software de código aberto de visão computacional. OpenCV foi construído para fornecer uma infraestrutura comum para aplicações de visão computacional. A biblioteca tem mais de 2.500 algoritmos otimizados, que inclui um conjunto abrangente de algoritmos clássicos e de última geração de visão computacional. Esses algoritmos podem ser usados para detetar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, rastrear movimentos de câmeras, rastrear objetos em movimento, extrair modelos 3D de objetos, produzir nuvens de pontos 3D a partir de câmeras estéreo, unir imagens para produzir alta resolução. imagem de uma cena inteira, encontrar imagens semelhantes em uma base de dados de imagens, remover olhos vermelhos de imagens tiradas com flash, seguir os movimentos dos olhos, reconhecer cenários e estabelecer marcadores para sobrepô-los com realidade aumentada, etc. OpenCV tem mais de 47 mil usuários comunidade e número estimado de downloads superior a 18 milhões. Apesar do OpenCV ser uma biblioteca realmente extraordinária, eu não consegui encontrar nela o que realmente estava à procura. Mesmo assim, isso não me impediu de a usar. A biblioteca OpenCV foi utilizada no projeto para capturar vídeo de uma câmera USB, definir o tamanho do frame de vídeo, ler frames da câmera, codificar frames como imagens JPEG e gerar uma sequência de frames como um fluxo de bytes. O fluxo de bytes resultante é então enviado sobre HTTP como uma resposta para a rota /video\_feed, permitindo que feed de câmera seja visto em um navegador web. Em seguida, o programa que construi no robô, utilizando JavaScript, pega o que é enviado por HTTP e mostra no web site.

- **YOLO:** YOLO é um sistema de deteção de objetos em tempo real que deteta objetos em RPNs ou pós-processamento. É um algoritmo popular baseado em aprendizado de máquina usado para tarefas de deteção de objetos. YOLO funciona dividindo a imagem de entrada em uma grade de células. Cada célula é responsável por detetar objetos dentro dos seus próprios limites. O algoritmo prevê caixas delimitadoras e probabilidades de classe diretamente de imagens completas. Essa abordagem torna o YOLO rápido e eficiente, permitindo que ele processe imagens em tempo real. Como não encontrei o que procurava com a biblioteca OpenCV, decidi procurar outros programas que funcionassem como eu queria. Encontrei o YOLO e, após algum estudo, descobri que podia juntar o melhor de dois mundos, ao combinar partes da biblioteca OpenCV com o programa YOLO. Então, fiz isso. Enquanto o feed da câmera é exibido no site, também é possível tirar fotos com apenas um clique de um botão na tela. Quando isso acontece, OpenCV e YOLO trabalham juntos para identificar todos os objetos na imagem da melhor forma possível. O YOLO divide a imagem de entrada em uma grade de células, e o OpenCV faz o seu melhor para identificar os objetos dentro de cada célula. Descobri que, ao trabalhar assim, é possível uma melhor identificação. No entanto, o Raspberry Pi não é o computador mais poderoso existente, então tive que usar versões antigas, tanto da biblioteca OpenCV quanto do programa YOLO. Com isso, a identificação de objetos não é tão boa comparada à que seria se estivesse trabalhando com um computador mais potente em vez de um Raspberry Pi.
- **RPI.GPIO:** RPi.GPIO é uma biblioteca que fornece um módulo Python para controlar os pins GPIO de um RaspberryPi. Ele permite o controlo dos pinos de entrada e saída do RaspberryPi, permitindo a criação de projetos de automação, robótica e outros que requerem interação com o mundo físico. O RPi.GPIO cria uma interface entre o Python e os pins GPIO do RaspberryPi. Ele fornece uma série de funções que permitem que configurar os pinos como entrada ou saída, ler os valores dos pinos, escrever valores nos pinos e muito mais. Neste projeto, a biblioteca RPi.GPIO permite o controlo dos motores para o robô conseguir-se mover.

## Servidor

- **Apache:** O Servidor HTTP Apache ou Servidor Apache ou HTTP Daemon Apache ou somente Apache, é o servidor web livre criado em 1995 por um grupo de desenvolvedores da NCSA, tendo como base o servidor web NCSA HTTPd criado por Rob McCool. Para garantir segurança nas transações HTTP, o servidor dispõe de um módulo chamado `mod_ssl`, o qual adiciona a capacidade do servidor atender requisições utilizando o protocolo HTTPS. Este protocolo utiliza uma camada SSL para criptografar todos os dados transferidos entre o cliente e o servidor, provendo maior grau de segurança, confidencialidade e confiabilidade dos dados. Os arquivos de configuração, por padrão, em ambientes Unix-like (ambiente do RaspberryPi), residem no diretório `/etc/apache`. O servidor é configurado por um arquivo mestre nomeado `httpd.conf` e opcionalmente pode haver configurações para cada diretório utilizando arquivos com o nome `.htaccess`, onde é possível utilizar autenticação de utilizadores pelo próprio protocolo HTTP utilizando uma combinação de arquivo `.htaccess` com um arquivo `.htpasswd`, que guardará os utilizadores e respetivas senhas.

## Base de Dados

- **MariaDB:** MariaDB é um sistema de gerenciamento de bases de dados que surgiu como bifurcação do MySQL, criado pelo próprio fundador do projeto após a sua aquisição pela Oracle. A segurança é muito importante no mundo de hoje e é um foco especial para os desenvolvedores do MariaDB. O projeto mantém os seus próprios patches de segurança situados no core de MySQL. Para cada versão do MariaDB os desenvolvedores também incluem todos os patches de segurança do MySQL e as suas melhorias, caso necessário. Quando problemas de segurança críticos são descobertos, os desenvolvedores imediatamente preparam e distribuem novas versões do MariaDB para obter as correções o mais rápido possível. Com estas informações eu decidi que MariaDB seria o que eu iria usar para gerir a minha base de dados.

## Outros

- **WebSockets:** WebSocket é uma tecnologia que permite a comunicação bidirecional por canais full-duplex sobre um único soquete TCP. WebSockets estão presentes no projeto para que vários utilizadores possam aceder ao site e ter atualizações em tempo real ao mesmo tempo.

- **Node.js:** Node.js é um software de código aberto, multiplataforma, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador web. A principal característica do Node.js é sua arquitetura assíncrona e orientada por eventos. O runtime do Node.js é single-thread ou seja trabalha com apenas uma única thread (chamada de Event Loop) é responsável por executar o código JavaScript, sem a necessidade de criar novas threads, o que torna o código mais simples de manter, com isto, o Node.js está a ser utilizado no projeto para ajudar o JavaScript a fazer o seu trabalho. Quando o JavaScript tem que executar algum código ou ficheiro, o Node.js funciona como um recetor da informação que é dada ao executar o código ou ficheiro. Isso funciona muito bem para poupar recursos do RaspberryPi, assim como funciona muito bem para debug do código, facilitando muito a procura dos erros.

## Projeto – RaspyPi

### Programação e Desenvolvimento Web

A interface web do projeto foi desenvolvida utilizando HTML, CSS, PHP e JavaScript. Esta interface permite aos utilizadores controlar o robô, visualizar a transmissão de vídeo em tempo real, capturar fotografias, utilizar o chat com reconhecimento de voz e resposta por texto e por último aceder as informações de percentagem de uso do CPU, temperatura do RaspberryPi e percentagem de uso da memória.

### Componentes

- **HTML:** Estrutura da página web.
- **CSS:** Estilo e layout da página web.
- **JavaScript:** Interatividade e funcionalidades dinâmicas.
- **PHP:** Lógica do servidor e integração com o backend.
- **Apache:** Servidor web utilizado para hospedar o site.

### Funcionalidades da Interface Web

- **Controlo do Robô:** Interface para controlar o movimento do robô através do teclado.
- **Transmissão de Vídeo em Tempo Real:** Visualização do vídeo capturado pela câmara USB em tempo real.
- **Captura de Fotografias:** Botão para capturar fotografias e processá-las com YOLOv5.
- **Chat com o Robô:** Sistema de chat que permite gravação de áudio, transcrição com WhisperV3, e resposta de texto com ChatGPT-4.
- **Monitorização do Sistema:** Página de informações sobre a utilização da CPU, memória e temperatura do RaspberryPi.

### Integração com Base de Dados

Para armazenar e recuperar informações sobre o desempenho do Raspberry Pi e outros dados, foi utilizada uma base de dados. A integração é feita utilizando PHP para interagir com a base de dados a partir da interface web.

## Componentes

- **MySQL:** Sistema de gerenciamento de base de dados utilizado para armazenar informações.
- **PHP:** Linguagem utilizada para executar consultas à base de dados e manipular dados.

## Fluxo de Trabalho

- **Coleta de Dados:** Scripts Python coletam dados sobre a utilização da CPU, memória e temperatura.
- **Armazenamento de Dados:** Os dados são enviados para a base de dados.
- **Recuperação de Dados:** A interface web permite ao utilizador consultar a base de dados para visualizar informações específicas.

### Exemplo de Código para Inserção de Dados com PHP

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "database";

// Criar conexão
$conn = new mysqli($servername, $username, $password, $dbname);
// Verificar conexão
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Inserir dados na tabela
$sql = "INSERT INTO performance_data (cpu_usage, memory_usage, temperature)
VALUES ('".$_POST["cpu"]."', '".$_POST["memory"]."', '".$_POST["temperature"]."')";
if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

## **Processamento e Reconhecimento de Voz**

### **Reconhecimento de Voz**

- O reconhecimento de voz é uma tecnologia que converte fala em texto. Neste projeto, o reconhecimento de voz é implementado utilizando a API WhisperV3 do OpenAI. Este serviço é capaz de transcrever áudio em texto com alta precisão.

### **Componentes**

- Microfone: Captura o áudio do utilizador.
- API WhisperV3: Processa o áudio e retorna a transcrição em texto.

### **Fluxo de Trabalho**

- Captura do Áudio: O utilizador grava um áudio de 5 segundos através da interface web.
- Envio para WhisperV3: O áudio é enviado para a API WhisperV3.
- Transcrição: A API converte o áudio em texto e retorna o resultado.

### **Conversão de Texto em Voz e Resposta**

Após a transcrição do áudio, o texto é enviado para a API do ChatGPT-4 para gerar uma resposta. Esta resposta é então exibida na interface web.

### **Componentes**

- API ChatGPT-4: Recebe o texto transcrito e retorna uma resposta.

### **Fluxo de Trabalho**

- Envio do Texto Transcrito: O texto transcrito é enviado para a API do ChatGPT-4.
- Geração de Resposta: A API gera uma resposta baseada no texto fornecido.
- Exibição da Resposta: A resposta é exibida na interface web.



## Identificação de Objetos

### Processamento de Imagens e Vídeos

Visão computacional é um campo da inteligência artificial que permite aos computadores extrair informações úteis de imagens e vídeos. Neste projeto, isso inclui a captura de vídeo em tempo real e a identificação de objetos em fotos.

#### Componentes

- **Câmera USB:** Captura imagens e vídeos.
- **OpenCV:** Biblioteca de visão computacional utilizada para processamento de imagens e transmissão de vídeo.
- **YOLOv5:** Modelo de machine learning utilizado para identificação de objetos.

#### Fluxo de Trabalho

- **Captura de Imagens e Vídeo:** A câmera USB captura imagens e vídeo em tempo real.
- **Processamento em Tempo Real:** OpenCV é utilizado para processar e transmitir o vídeo para a interface web.
- **Captura de Fotos:** Fotos podem ser capturadas a partir do vídeo em tempo real.

### Reconhecimento de Objetos com YOLOv5

YOLO é um modelo de deep learning utilizado para detetar e classificar objetos em imagens. É conhecido por sua alta precisão e velocidade.

#### Componentes

- **Modelo YOLOv5:** Carregado e utilizado para identificar objetos nas fotos capturadas.
- **Python e OpenCV:** Utilizados para pré-processamento das imagens e integração com o modelo YOLOv5.

### Fluxo de Trabalho

- **Captura de Fotos:** Uma fotografia é capturada com a câmara.
- **Processamento com YOLOv5:** A foto é processada pelo modelo YOLOv5 para detetar objetos.
- **Identificação de Objetos:** Os objetos identificados são marcados na foto.
- **Download da Fotografia:** A fotografia com as identificações é disponibilizada para download.

## **Fases de Realização do Projeto**

### **Conceção**

A fase de conceção do projeto focou-se na definição clara dos objetivos e funcionalidades essenciais do robô, bem como na seleção das tecnologias e componentes adequados para atingir esses objetivos. As principais atividades nesta fase incluíram:

#### **1. Definição dos Objetivos:**

- Estabelecer as funcionalidades principais do robô como o controle por teclado, transmissão de vídeo em tempo real, captura de fotografias e identificação dos objetos presentes nelas, chat com reconhecimento de voz e monitorização do desempenho do sistema.

#### **2. Pesquisa de Tecnologias:**

- Identificar as tecnologias mais adequadas para cada funcionalidade, como o uso do OpenCV para processamento de imagens e vídeo, YOLOv5 para identificação de objetos, APIs WhisperV3 e ChatGPT-4 do OpenAI para reconhecimento de voz e geração de respostas em texto, HTML, CSS, PHP, e JavaScript para o desenvolvimento da interface web, e Apache para poder hospedar o servidor.

### **Implementação**

A fase de implementação iniciou o desenvolvimento prático do robô e da interface web. Esta fase foi subdividida em várias atividades específicas:

#### **1. Configuração do RaspberryPi:**

- Instalação do sistema operativo e configuração inicial do RaspberryPi.
- Conexão e configuração da câmara USB e do controlador de motores para permitir o movimento do robô.

## **2. Desenvolvimento do Software:**

- Programação dos scripts em Python para controle dos motores utilizando os pinos GPIO e para a captura e processamento de vídeo com OpenCV.
- Implementação do modelo YOLOv5 para identificação de objetos nas fotos capturadas.
- Desenvolvimento do backend em PHP para interagir com a base de dados e o servidor Apache.
- Criação da interface web utilizando HTML, CSS e JavaScript para controlar o robô, visualizar o vídeo em tempo real, capturar fotos, e interagir com o chat.

## **3. Integração de APIs:**

- Integração com a API WhisperV3 para reconhecimento de voz e a API Chat-GPT-4 para gerar respostas em texto.
- Implementação de scripts para coleta de dados de desempenho do RaspberryPi e armazenamento na base de dados.

## **Testes e Ajustes**

A fase de testes e ajustes foi fundamental para garantir que todas as funcionalidades do robô e da interface web funcionassem corretamente e de forma integrada. As principais atividades nesta fase incluíram:

### **1. Testes Funcionais:**

- Verificação de todas as funcionalidades individuais, como controle do robô, transmissão de vídeo, captura de fotos, identificação de objetos e chat com reconhecimento de voz.

### **2. Testes de Integração:**

- Testes para assegurar que todos os componentes do sistema funcionassem de forma integrada, verificando a comunicação entre os scripts Python, a interface web, e as APIs externas.

### **3. Ajustes e Otimizações:**

- Correção de bugs identificados durante os testes.
- Otimização do desempenho do sistema, especialmente na transmissão de vídeo em tempo real e no processamento de imagens com YOLOv5.

### **4. Testes de Usabilidade:**

- Avaliação da interface web por utilizadores finais para garantir que a experiência de utilização fosse intuitiva e satisfatória.

## **Métodos de Trabalho**

Durante o desenvolvimento deste projeto, o foco esteve sempre na flexibilidade e na adaptação às necessidades específicas do robô. O desenvolvimento foi realizado de forma direta, utilizando o ambiente de desenvolvimento Visual Studio Code para escrever e editar o código-fonte.

### **1. Ambiente de Desenvolvimento:**

- Utilização do Visual Studio Code como principal ambiente de desenvolvimento, proporcionando uma experiência de codificação eficiente e produtiva.
- A manipulação manual dos arquivos foi uma prática comum, permitindo um controlo preciso sobre o sistema e uma compreensão detalhada da estrutura e funcionamento dos componentes. Isto incluiu a edição manual de configurações do servidor web, ajustes na base de dados e a implementação de funcionalidades específicas do RaspberryPi.

### **2. Desenvolvimento de Software:**

- Programação em Python para controlar os motores e processar o vídeo, utilizando bibliotecas como RPi.GPIO e OpenCV.
- Desenvolvimento de scripts para comunicação com as APIs WhisperV3 e ChatGPT-4, implementando processamento de voz para texto e geração de respostas em texto.
- Uso do YOLOv5 para identificação de objetos, treino e ajuste do modelo para melhorar a precisão nas fotos capturadas pelo robô.

### **3. Desenvolvimento Web:**

- Criação da interface web com HTML, CSS e JavaScript, garantindo uma experiência de utilizador intuitiva.
- Implementação do backend em PHP para gerenciar a lógica do servidor e integrar com a base de dados.
- Uso do Flask para facilitar a transmissão de vídeo em tempo real, integrando com OpenCV para capturar e enviar frames da câmara USB do RaspberryPi para o site.

#### **4. Integração de Componentes:**

- Testes contínuos de integração para garantir que todos os componentes funcionassem juntos de forma suave.
- Ajustes constantes baseados no feedback dos testes, garantindo a estabilidade e a eficiência do sistema.

#### **5. Monitorização e Ajustes:**

- Implementação de scripts para monitorização do desempenho do RaspberryPi, coletando dados sobre uso da CPU, memória e temperatura.
- Armazenamento desses dados em uma base de dados, permitindo análise histórica e ajuste do desempenho do sistema conforme necessário.

## Produto Final

Este projeto demonstra a integração de várias tecnologias para criar um robô avançado controlado por teclado, com capacidades de visão computacional, reconhecimento de voz, e uma interface web interativa. Utilizando um RaspberryPi como base, foram implementadas funcionalidades robustas que permitem ao robô capturar e transmitir vídeo em tempo real, realizar captura de fotos com identificação de objetos e interagir com os utilizadores via chat com reconhecimento de voz.

A interface web, desenvolvida com HTML, CSS, PHP e JavaScript, oferece uma plataforma amigável e intuitiva para os utilizadores controlarem o robô e acederem a diversas funcionalidades. A transmissão de vídeo em tempo real, facilitada por OpenCV e Flask, permite que os utilizadores vejam o que o robô vê, proporcionando uma experiência interativa e imersiva. A funcionalidade de captura de fotos, aliada ao modelo YOLOv5, permite a identificação precisa de objetos, tornando o robô útil para várias aplicações práticas, desde vigilância até análise de ambientes.

A integração de reconhecimento de voz e resposta em texto através das APIs WhisperV3 e ChatGPT-4 do OpenAI acrescenta uma camada de interatividade que aproxima ainda mais a interação homem-máquina, permitindo conversas naturais e eficientes com o robô. Esta capacidade de compreender e responder a comandos de voz torna o robô acessível e fácil de usar, mesmo para utilizadores com pouca experiência técnica. Além disso, a funcionalidade de monitorização do desempenho do RaspberryPi, com coleta de dados sobre utilização da CPU, memória e temperatura, permite um acompanhamento detalhado do funcionamento do sistema. A integração com uma base de dados para armazenamento e recuperação desses dados, permite que os utilizadores possam consultar informações históricas e realizar análises de desempenho de forma eficaz.

O uso do servidor Apache para hospedar o site garante que a aplicação web seja robusta e capaz de lidar com múltiplos utilizadores simultaneamente. A combinação de tecnologias front-end e back-end proporciona uma experiência de utilizador ágil e responsiva, essencial para a interação em tempo real com o robô.



Em resumo, este projeto exemplifica a aplicação prática de técnicas modernas de inteligência artificial, visão computacional, e desenvolvimento web para criar um sistema integrado e funcional. O robô não só demonstra capacidades técnicas avançadas, como também apresenta um potencial significativo para várias aplicações no mundo real, desde a automação doméstica até a segurança e monitoramento ambiental.

## **Dificuldades Encontradas**

### **Problemas com Cartões SD**

No início do projeto, enfrentei uma série de problemas significativos com cartões SD. Ao todo, tive de adquirir cinco cartões SD diferentes, pois o Raspberry Pi corrompia-os rapidamente. Esta situação resultou numa perda frequente de dados e progresso, uma vez que, sempre que um cartão SD se estragava, eu tinha de recomeçar o projeto do zero. Isso incluía a configuração do sistema operativo, a instalação e configuração do servidor web Apache e da base de dados, bem como a reinstalação de todas as dependências necessárias. Este processo era moroso e frustrante, e atrasou consideravelmente o desenvolvimento do projeto. Deste modo, optei por utilizar técnicas de backup regulares e investiguei soluções para minimizar a corrupção dos cartões SD, tais como a utilização de cartões de maior qualidade e a implementação de práticas de escrita de dados mais eficientes.

### **Configuração do Servidor**

Configurar o servidor web no RaspberryPi apresentou-se um desafio muito significativo. Além das questões de segurança e desempenho, também surgiram obstáculos relacionados à compatibilidade de software e à otimização dos recursos limitados do dispositivo. A necessidade de garantir a estabilidade e a acessibilidade do servidor, exigiu uma abordagem mais cuidadosa na configuração e na seleção das melhores práticas de segurança.

### **Integração da Base de Dados**

A integração do MariaDB com o servidor web também representou uma etapa complexa do projeto. Foi necessário dedicar tempo significativo à configuração da base de dados, para garantir um bom funcionamento e a integridade dos dados armazenados. Além disso, foram necessárias considerações adicionais para otimizar o desempenho da base de dados e garantir a eficiência das consultas, especialmente num ambiente de recursos limitados como o RaspberryPi.

## **Dificuldade com Manipulação Manual de Arquivos**

Durante o desenvolvimento do projeto, enfrentei várias dificuldades relacionadas com a manipulação manual de ficheiros no RaspberryPi para obter informações sobre CPU, memória e temperatura. Estes problemas surgiram por vários motivos:

### **1. Falta de Documentação e Familiaridade:**

- Inicialmente, tive dificuldades em encontrar e interpretar a documentação adequada sobre como aceder e ler os arquivos de sistema do RaspberryPi que contêm informações sobre a utilização da CPU, memória e temperatura.

### **2. Complexidade dos Arquivos de Sistema:**

- Os arquivos de sistema que armazenam estes dados, não são sempre intuitivos ou bem documentados, o que complicou a compreensão de que ficheiros deveriam ser lidos e como extrair as informações necessárias de forma correta.

### **3. Erros de Sintaxe e Leitura:**

- Cometi vários erros ao tentar escrever o código para ler e interpretar os dados dos arquivos de sistema. Isto incluía erros de sintaxe em Python, bem como dificuldades em processar corretamente os dados lidos.

### **4. Soluções e Aprendizagem:**

- Após várias tentativas e pesquisas, consegui identificar os arquivos corretos no diretório `/proc` e `/sys` que continham as informações necessárias:
  - Para a utilização da CPU, utilizei o arquivo `/proc/stat`.
  - Para a memória, acedi ao arquivo `/proc/meminfo`.
  - Para a temperatura, utilizei o arquivo `/sys/class/thermal/thermal_zone0/temp`.

- Desenvolvi scripts em Python para ler estes arquivos, processar as informações e armazená-las na base de dados. Por exemplo, para obter a temperatura da CPU, o script lê o valor em milicelcius e converte para graus Celsius.

Apesar das dificuldades iniciais, estas experiências resultaram numa aprendizagem significativa sobre a estrutura do sistema de arquivos do Linux no Raspberry Pi e melhoraram as minhas habilidades em manipulação e processamento de dados através de scripts Python.

## **Controlo por Teclado dos Motores do Robô**

Inicialmente, a configuração dos pinos GPIO do Raspberry Pi para controlar os motores, exigiu um entendimento profundo da documentação e experimentação prática para encontrar a configuração correta. Identificar os pinos adequados e configurá-los para saída PWM foi um processo crucial para garantir o controlo preciso dos motores.

Uma vez configurados os pinos GPIO, o desafio seguinte foi desenvolver a lógica para mapear os comandos do teclado para os movimentos dos motores. Converter os pressionamentos de teclas em ações específicas dos motores, como avançar, recuar e girar, exigiu uma abordagem cuidadosa para garantir que o robô respondesse de forma previsível e precisa aos comandos do utilizador.

A calibração dos motores também se mostrou uma etapa crucial. Ajustar a velocidade e a direção dos motores para corresponder aos comandos do teclado exigiu testes repetidos e ajustes finos para alcançar o comportamento desejado. Garantir um movimento suave e preciso do robô foi essencial para a utilização e eficácia do sistema.

Além disso, a latência no controlo dos motores apresentou desafios adicionais. A garantia de uma resposta rápida e precisa aos comandos do teclado foi fundamental para proporcionar uma experiência de controlo intuitiva e responsiva. Isso exigiu otimizações no código e no hardware para minimizar qualquer atraso perceptível entre a entrada do teclado e a resposta dos motores.

Durante todo o processo, a depuração e os testes foram fundamentais para identificar e corrigir erros no controlo dos motores. Foram realizados testes extensivos para verificar o funcionamento correto de cada comando do teclado e depurar problemas, como motores que não respondiam ou se moviam de forma inconsistente.

Superar esses desafios exigiu dedicação, pesquisa e experimentação. A colaboração com outros membros da equipa e a consulta à comunidade online foram recursos valiosos para resolver problemas e aprimorar o controlo por teclado dos motores do robô.

## **Problemas com o Cooler do RaspberryPi**

Durante o desenvolvimento do projeto, enfrentei um desafio inesperado quando o cooler do RaspberryPi falhou em funcionar corretamente. O sobreaquecimento do RaspberryPi poderia comprometer a estabilidade e o desempenho do sistema, o que exigia uma solução rápida e eficaz.

Após uma análise inicial, constatei que o cooler estava com defeito e precisava ser substituído urgentemente. Infelizmente, não havia um cooler compatível disponível no momento. Diante dessa situação, tomei a decisão de procurar uma solução alternativa para garantir a continuidade do projeto.

Foi então que surgiu a ideia de desmontar uma Hard Drive antiga e remover o cooler dela. Apesar de não ser uma solução convencional, acreditei que poderia ser o suficiente para resolver este problema, uma vez que era o único cooler de tamanho adequado que consegui encontrar.

Após desmontar a unidade de disco e extrair o cooler, realizei os ajustes necessários para instalá-lo no RaspberryPi. Felizmente, a adaptação foi bem-sucedida e o novo cooler conseguiu manter a temperatura do RaspberryPi dentro dos limites operacionais aceitáveis.

Essa experiência ressaltou a importância do planeamento de contingência e da capacidade de improvisação durante projetos de engenharia. Mesmo diante de contratempos inesperados, fui capaz de encontrar uma solução criativa e eficaz para garantir a continuidade do projeto.

A resolução desse problema demonstrou não apenas a minha habilidade técnica, mas também a minha determinação em superar obstáculos e alcançar os meus objetivos.

## **Problemas com Câmera RaspberryPi**

Durante a fase inicial do projeto, deparei-me com um obstáculo relacionado com a câmara. Inicialmente, planeava utilizar uma Picamera para capturar vídeo em tempo real e realizar reconhecimento de objetos. No entanto, enfrentei dificuldades técnicas que impediram o funcionamento correto da Picamera, mesmo após várias tentativas de solução de problemas.

Diante dessa situação, tomei a decisão de procurar uma alternativa viável para garantir a funcionalidade de captura de vídeo do robô. Após uma pesquisa cuidadosa, optei por adquirir uma câmara USB Logitech como substituta. Embora essa decisão tenha acarretado em custos adicionais e um ajuste no design do sistema, acreditei que era a melhor opção disponível no momento para garantir o progresso contínuo do projeto.

A integração da nova câmara USB Logitech exigiu ajustes no código e na configuração do sistema, mas, felizmente, ela mostrou-se compatível e funcional. Com a nova câmara em funcionamento, pude retomar o desenvolvimento do projeto, explorando as suas capacidades para captura de vídeo em tempo real e reconhecimento de objetos.

Embora tenha sido necessário lidar com esse contratempo, essa experiência destacou a importância da flexibilidade e da capacidade de adaptação durante o desenvolvimento de projetos técnicos. A capacidade de identificar e implementar soluções alternativas foi fundamental para superar esse desafio e avançar com sucesso no projeto.

Essa experiência também enfatizou a importância de realizar uma pesquisa abrangente de soluções alternativas e de considerar diferentes opções antes de tomar uma decisão. Embora imprevistos como este possam surgir durante o desenvolvimento de projetos, a habilidade de encontrar soluções criativas e eficazes é essencial para garantir o sucesso final do projeto.

## **Conclusão**

Ao longo deste projeto, pude desenvolver um robô avançado que incorpora uma variedade de tecnologias e funcionalidades inovadoras. O robô é capaz de ser controlado remotamente através de um website intuitivo, onde os utilizadores podem interagir com ele de várias formas.

A funcionalidade de controlo por teclado no website permite aos utilizadores direcionar o robô de forma precisa e intuitiva. Além disso, a capacidade de capturar vídeo em tempo real e exibi-lo no site proporciona uma experiência imersiva, permitindo que os utilizadores vejam o que o robô está a ver em tempo real.

A função de tirar fotografias e realizar a identificação de objetos é outro destaque do robô. Os utilizadores podem solicitar que o robô tire uma fotografia e, em seguida, utilize algoritmos avançados de visão computacional para identificar e classificar objetos na imagem. Isso torna o robô útil para uma variedade de aplicações práticas, desde vigilância até análise de ambientes.

Além disso, a capacidade de interação com o robô através de chat, utilizando reconhecimento de voz e resposta em texto, adiciona uma experiência de interatividade e usabilidade. Os utilizadores podem conversar com o robô de forma natural e eficiente, tornando-o acessível até mesmo para aqueles com pouca experiência técnica.

No entanto, nem tudo correu como esperava, ao longo do desenvolvimento do projeto, enfrentei uma série de desafios e dificuldades. Desde problemas com cartões SD até a integração da base de dados e a configuração do servidor, cada etapa do processo apresentou os seus próprios obstáculos. A manipulação manual de arquivos, o controlo por teclado dos motores do robô, os problemas com o cooler e a câmara do RaspberryPi também exigiram soluções criativas e trabalho árduo para serem superados.



Apesar dessas dificuldades, o resultado final é um testemunho do meu compromisso com a excelência e a minha capacidade de superar obstáculos. O robô que desenvolvi não só demonstra capacidades técnicas avançadas, como também apresenta um potencial significativo para várias aplicações no mundo real, desde a automação doméstica até a segurança e monitoramento ambiental.

Estou entusiasmado com as possibilidades futuras deste projeto e confiante de que ele continuará a inspirar e capacitar outros entusiastas e desenvolvedores.

## Webgrafia

Informação sobre Visual Studio Code encontrada aqui “[https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)” - última vez visitado em: 4 de junho 2024

Informação sobre Thonny encontrada aqui “<https://en.wikipedia.org/wiki/Thonny>” - última vez visitado em: 4 de junho 2024

Informação sobre SQL encontrada aqui “<https://pt.wikipedia.org/wiki/SQL>” - última vez visitado em: 4 de junho 2024

Informação sobre JavaScript encontrada aqui “<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>”, “<https://pt.wikipedia.org/wiki/JavaScript>” - última vez visitado em: 4 de junho 2024

Informação sobre WebSockets encontrada aqui “[https://developer.mozilla.org/pt-BR/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/pt-BR/docs/Web/API/WebSockets_API)”, “<https://pt.wikipedia.org/wiki/WebSocket>” - última vez visitado em: 4 de junho 2024

Informação sobre OpenCV encontrada aqui “<https://opencv.org/about/>” - última vez visitado em: 5 de junho 2024

Informação sobre YOLO encontrada aqui “<https://pjreddie.com/darknet/yolo/>” - última vez visitado em: 5 de junho 2024

Informação sobre Node.js encontrada aqui “<https://nodejs.org/en/about>” - última vez visitado em: 5 de junho 2024

Informação sobre Apache encontrada aqui “[https://pt.wikipedia.org/wiki/Servidor\\_Apache](https://pt.wikipedia.org/wiki/Servidor_Apache)” - última vez visitado em: 5 de junho 2024

Informação sobre Python encontrada aqui “<https://pt.wikipedia.org/wiki/Python>” - última vez visitado em: 6 de junho 2024

Informação sobre HTML encontrada aqui “<https://pt.wikipedia.org/wiki/HTML>” - última vez visitado em: 6 de junho 2024

Informação sobre PHP encontrada aqui “<https://pt.wikipedia.org/wiki/PHP>” - última vez visitado em: 6 de junho 2024

Informação sobre CSS encontrada aqui “[https://pt.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://pt.wikipedia.org/wiki/Cascading_Style_Sheets)” - última vez visitado em: 6 de junho 2024

Informação sobre MariaDB encontrada aqui “ <https://pt.wikipedia.org/wiki/MariaDB> “ -  
última vez visitado em: 6 de junho 2024