

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

**Escola Técnica Estadual da Zona Leste
MTEC Desenvolvimento De Sistemas AMS**

Kauan dos Santos

Miguel Yudi Baba

Nicolas Sinelli Pereira das Neves Hilário

Kyntact: Sistema de tradução de texto para braile em tempo real

São Paulo

2025

Kauan dos Santos

Miguel Yudi Baba

Nicolas Sinelli Pereira das Neves Hilário

Kyntact: Sistema de tradução de texto para braile em tempo real

Trabalho de Conclusão de Curso apresentado ao MTEC Desenvolvimento de Sistemas AMS da Etec da Zona Leste, orientado pelo Prof. Esp. Jeferson Roberto de Lima, como requisito parcial para a obtenção do título de técnico em Desenvolvimento de Sistemas.

São Paulo

2025

Resumo

Palavras-Chave: Lorem ipsum dolor sit amet consectetur adipiscing elit. Quisque faucibus ex sapien vitae pellentesque sem placerat. In id cursus mi pretium tellus dui convallis. Tempus leo eu aenean sed diam urna tempor. Pulvinar vivamus fringilla lacus nec metus bibendum egestas. laculis massa nisl malesuada lacinia integer nunc posuere. Ut hendrerit semper vel class aptent taciti sociosqu. Ad litora torquent per conubia nostra inceptos himenaeos.

Abstract

Keywords: Lorem ipsum dolor sit amet consectetur adipiscing elit. Quisque faucibus ex sapien vitae pellentesque sem placerat. In id cursus mi pretium tellus dui convallis. Tempus leo eu aenean sed diam urna tempor. Pulvinar vivamus fringilla lacus nec metus bibendum egestas. laculis massa nisl malesuada lacinia integer nunc posuere. Ut hendrerit semper vel class aptent taciti sociosqu. Ad litora torquent per conubia nostra inceptos himenaeos.

LISTA DE ILUSTRAÇÕES

Figura 01 — Palavras escritas em Braile.	13
Figura 02 — Exemplo de um Diagrama de Caso de Uso.....	14
Figura 03 — Exemplo de um Diagrama de Classe.	15
Figura 04 — Exemplo de um Diagrama de Sequência.	16
Figura 05 — Exemplo de um Diagrama de Componentes.	17
Figura 06 — Baterias de Íons de lítio.	19
Figura 07 — Raspberry Pi 3.....	19
Figura 08 — Módulo de Câmera para Raspberry Pi Rev 1.3	20
Figura 09 — Exemplo de código em Python.	21
Figura 10 — Exemplo de uso do PIP.	21
Figura 11 — ESP8266.	23
Figura 12 — Exemplo de código em C++.	24
Figura 13 — Circuito integrado MCP23017.....	26
Figura 14 — Solenoide push pull.	26
Figura 15 — Modelo tridimensional.....	27
Figura 16 — Exemplo de impressão 3D.....	28

LISTA DE ABREVIATURAS E SIGLAS

Instituto Brasileiro de Geografia e Estatística (IBGE)

Inter-Integrated Circuit (I2C)

Internet of Things (IoT)

Light-Emitting Diode (LED)

Optical Character Recognition (OCR)

Open Source Computer Vision Library (OpenCV)

Package Installer for Python (PIP)

Pessoa com Deficiência (PcD)

Random Access Memory (RAM)

Sistema de Gerenciamento de Banco de Dados (SGBD)

Sistema Operacional (SO)

Structured Query Language (SQL)

Tridimensional (3D)

Unified Modeling Language (UML)

User Experience (UX)

You Only Look Once (YOLO)

LISTA DE SÍMBOLOS

KB Kilobyte

MHz Megahertz

P Varredura progressiva

SUMÁRIO

1	INTRODUÇÃO	10
2	REFERENCIAL TEÓRICO	12
2.1	Deficiência Visual	12
2.1.1	Dificuldades enfrentadas após a perda de visão	12
2.1.2	Braile	12
2.2	UML	13
2.2.1	Diagrama de Caso de Uso.....	13
2.2.2	Diagrama de Classe	14
2.2.3	Diagrama de Sequência.....	15
2.2.4	Diagrama de Componentes.....	16
2.3	Internet of Things (IoT).....	17
2.4	User Experience (UX).....	18
2.5	Tecnologias Utilizadas	18
2.5.1	Baterias íon-lítio.....	18
2.5.2	Raspberry Pi.....	19
2.5.2.1	Debian	20
2.5.2.2	Módulo de Câmera para Raspberry Pi	20
2.5.3	Python.....	20
2.5.3.1	Package Installer for Python (PIP).....	21
2.5.4	MySQL.....	22
2.5.5	Open Source Computer Vision Library (OpenCV)	22
2.5.6	Optical Character Recognition (OCR)	22
2.5.6.1	Tesseract	22
2.5.6.2	PyTesseract.....	23
2.5.7	You Only Look Once (YOLO)	23

2.5.8	ESP8266.....	23
2.5.9	Linguagem de Programação para Embarcados	24
2.5.10	Jumpers	25
2.5.11	Resistores.....	25
2.5.12	Transistores	25
2.5.13	Interruptor.....	25
2.5.14	Circuito integrado MCP23017	25
2.5.15	Solenoides.....	26
2.5.16	Push buttons	26
2.5.17	Modelagem 3D.....	27
2.5.18	Impressão 3D	27
3	DESENVOLVIMENTO	29
4	CONSIDERAÇÕES FINAIS.....	30
	REFERÊNCIAS.....	31

1 INTRODUÇÃO

As pessoas com deficiência visual ainda são muito negligenciadas no cenário atual, tendo problemas como falta de adaptação em ambientes movimentados, dificuldade em estudar e ler livros, placas e coisas que consideramos básicas no dia a dia, como cita a federação APAES (2024). Pensando nisso, pode-se idealizar o desenvolvimento de um projeto, que se trata de uma solução tecnológica baseada em Internet das Coisas (IoT) voltada à acessibilidade para pessoas com deficiência visual total. O projeto consiste em um dispositivo portátil capaz de captar textos por meio de uma câmera, processá-los em um minicomputador e reproduzi-los em braile com solenoides, permitindo que o usuário entenda o conteúdo por meio de relevos táteis.

Este projeto tem como objetivo principal auxiliar pessoas com deficiência visual total a compreender textos ao seu redor, utilizando reconhecimento de caracteres e transcrição imediata para o sistema braile, e tornar o projeto acessível e portátil.

Tendo esse objetivo em vista, busca-se resolver a dificuldade que pessoas com deficiência visual enfrentam para acessar informações escritas em locais públicos, onde, na maioria das vezes, não há sinalização ou materiais previamente adaptados para braile.

A partir disso, levantaram-se duas hipóteses principais após uma análise inicial: a primeira é como a baixa presença de textos em braile em espaços públicos está relacionada à falta de infraestrutura e investimento em acessibilidade. A segunda é que a ausência de soluções portáteis e acessíveis contribui para a exclusão informacional dessas pessoas, limitando sua autonomia e participação social.

Essas constatações reforçam como o desenvolvimento de um projeto com o potencial de ampliar a inclusão social e a independência de pessoas com deficiência visual total é essencial para as circunstâncias dos dias de hoje. Ao permitir que o usuário tenha acesso rápido a informações escritas no seu entorno, o dispositivo promove a autonomia, facilita a locomoção em espaços urbanos e amplia as oportunidades de participação em diferentes contextos sociais e profissionais.

O presente projeto tem como foco a aplicação e validação do dispositivo na região leste da cidade de São Paulo, a fim de avaliar sua funcionalidade em situações reais de uso, como a leitura de placas, avisos e documentos simples, considerando as condições e desafios específicos desse ambiente urbano.

Para fundamentar a construção e aplicação do dispositivo, recorreu-se a diferentes referenciais teóricos e técnicos. Na modelagem do sistema, utilizou-se a abordagem de Gilleanes T. A. Guedes (2018) sobre UML; para o processamento de dados, adotou-se a linguagem Python conforme descrita por José Augusto N. G. Manzano (2018); e, no controle do hardware, aplicaram-se conceitos apresentados por Stephen R. Davis (2016) sobre programação em C++

O trabalho foi organizado em quatro capítulos, de forma que cada um cumpre um papel específico dentro da construção da pesquisa. No primeiro, é feita a introdução, na qual se apresenta o contexto do tema, os objetivos a serem alcançados e a justificativa que sustenta sua relevância. Em seguida, o segundo capítulo reúne as bases teóricas e as referências que dão suporte ao projeto, abordando os conceitos, métodos e tecnologias relacionadas que foram utilizadas. Já o terceiro capítulo se volta para o desenvolvimento da proposta, descrevendo as fases de implementação e a aplicação prática das ideias discutidas anteriormente. Por fim, o quarto capítulo apresenta as considerações finais, trazendo uma síntese dos resultados alcançados, reconhecendo as limitações encontradas ao longo do percurso e apontando caminhos possíveis para futuras pesquisas e aprimoramentos.

2 REFERENCIAL TEÓRICO

Esta seção tem como objetivo listar e explicar os principais conceitos, linguagens e tecnologias que serão utilizadas no desenvolvimento do projeto.

2.1 Deficiência Visual

Deficientes visuais, segundo a Enciclopédia Barsa Universal (2009), são pessoas que não tem o sentido da vista, ou seja, não conseguem enxergar.

Existem dois tipos de deficiência visual; De acordo com Almeida e Araújo (2013), a congênita, a qual se refere a casos em que uma pessoa nasce com essa condição, e a adquirida, que é obtida ao longo da vida de um indivíduo.

Para o Instituto Brasileiro de Geografia e Estatística (IBGE, 2010), uma grande porcentagem da população brasileira, cerca de 3,2% dela, possui grande dificuldade para enxergar.

2.1.1 Dificuldades enfrentadas após a perda de visão

Como afirma o g1 (2019), os deficientes visuais enfrentam diversos desafios no seu cotidiano, como ter que andar em locais com o piso irregular ou embarcar no ônibus correto, já que não são capazes de ler o código e o destino dele.

PcDs visuais precisam de autonomia para ter seu direito à liberdade efetivamente exercido, a fim de alcançar essa independência, a ajuda de instituições e recursos assistivos é essencial, segundo Aciem (2011).

2.1.2 Braile

Braile, como afirma Borges (2009), é um sistema tátil de leitura e escrita, simples e de fácil reprodução, para pessoas com deficiência visual, o qual foi criado por um francês chamado Louis Braille no século XIX.

Essa linguagem se baseia em um conjunto de seis pontos, organizados em duas colunas iguais, que é usado para representar diversos elementos, como letras ou números, dependendo da combinação pontos presentes, segundo Oliveira (2018).

Figura 01 — Palavras escritas em Braille.



Fonte: g1, 2025.

2.2 UML

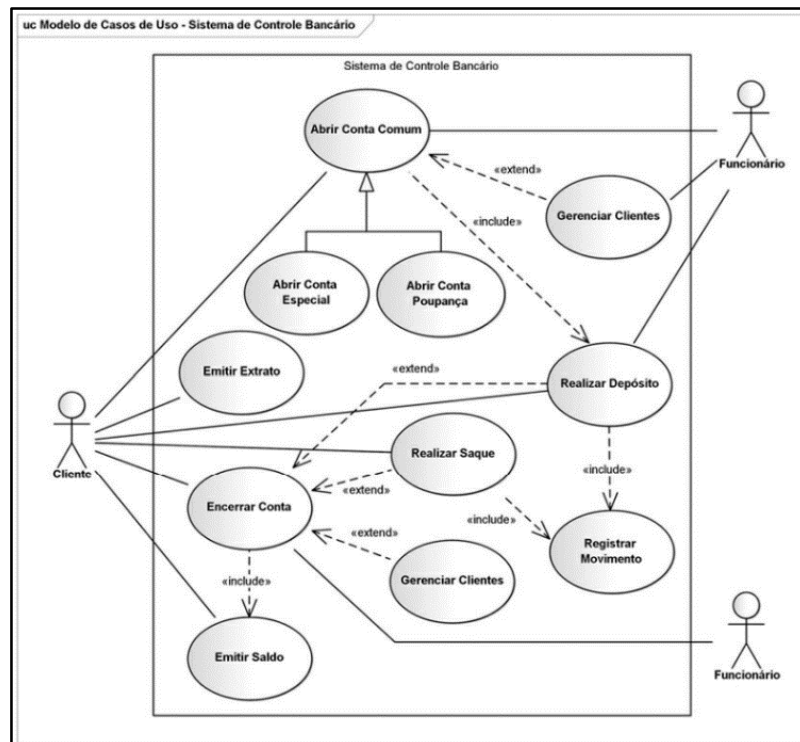
A Linguagem de Modelagem Unificada é, de acordo com Gilleanes Guedes (2018), uma linguagem de modelagem que tem como objetivo auxiliar na definição das características de um sistema antes de ele começar a ser de fato desenvolvido.

Para Martin Fowler (2005), a UML é particularmente útil como ferramenta de documentação e comunicação entre os membros de uma equipe. Ele destaca que o valor da UML é proporcional à sua clareza e utilidade das representações.

2.2.1 Diagrama de Caso de Uso

Como ressalta Booch (2006), os Diagramas de Caso de Uso são utilizados para identificar os usuários e os serviços que serão prestados a eles pelo sistema. Ele é especialmente útil nas fases iniciais do desenvolvimento.

Figura 02 — Exemplo de um Diagrama de Caso de Uso.



Fonte: UML 2: Uma Abordagem Prática, 2018.

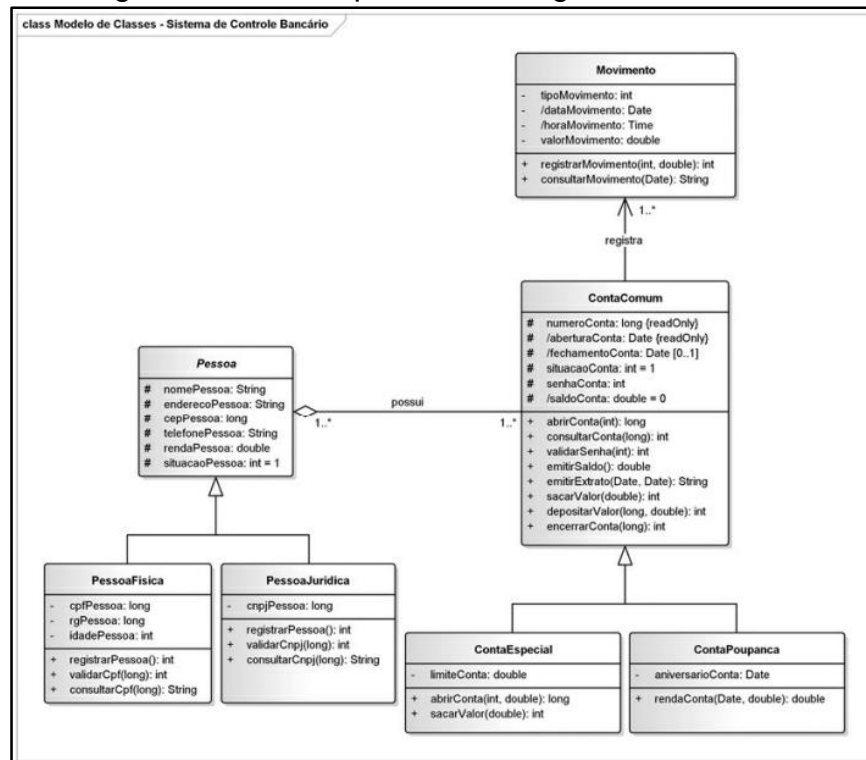
Explicação do Diagrama acima:

O diagrama mostra os principais serviços do Sistema de Controle Bancário e como Cliente e Funcionário interagem com o programa. O cliente pode, por exemplo, abrir contas, emitir extrato, realizar saques e encerrar conta. Já o funcionário gerencia clientes e registra movimentações.

2.2.2 Diagrama de Classe

Parafraseando Guedes (2018), o Diagrama de Classe serve para representar as funcionalidades do sistema e identificar os casos de uso, que são as interações entre os usuários e o sistema, descrevendo o comportamento esperado do programa.

Figura 03 — Exemplo de um Diagrama de Classe.



Fonte: UML 2: Uma Abordagem Prática, 2018.

Explicação do Diagrama acima:

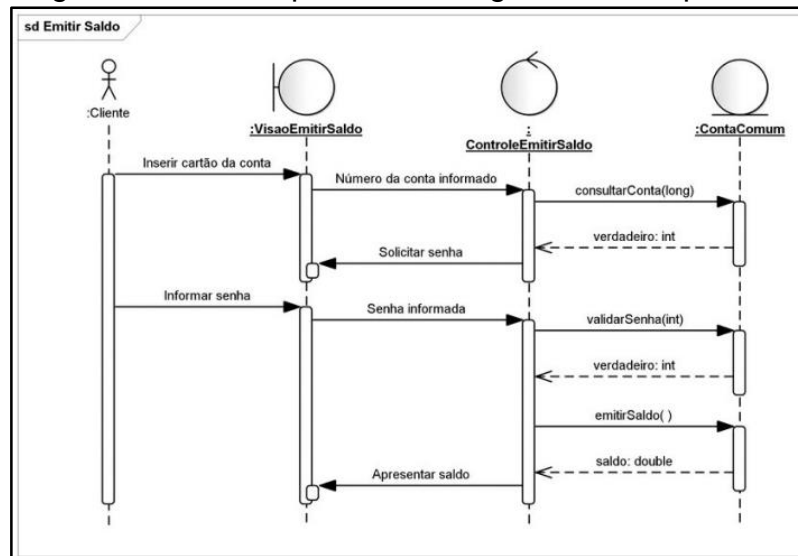
O diagrama modela a estrutura de um sistema bancário, a classe “ContaComum” contém os métodos usados no processo de emissão de saldo, como consultar conta, validar senha e emitir saldo. Cada pessoa, física ou jurídica, pode possuir uma ou mais contas, as quais podem ser especiais ou poupança, capazes de registrar diversas movimentações.

2.2.3 Diagrama de Sequência

Como aponta Fowler (2005), os Diagramas de Sequência são ideais para visualizar a interação entre objetos ao longo do tempo, esclarecendo a lógica do fluxo principal do sistema.

Complementando, para Booch (2006), esse diagrama foca nas ações realizadas pelo programa de forma sequencial, mostrando a ordem das mensagens trocadas para realizar uma funcionalidade específica.

Figura 04 — Exemplo de um Diagrama de Sequência.



Fonte: UML 2: Uma Abordagem Prática, 2018.

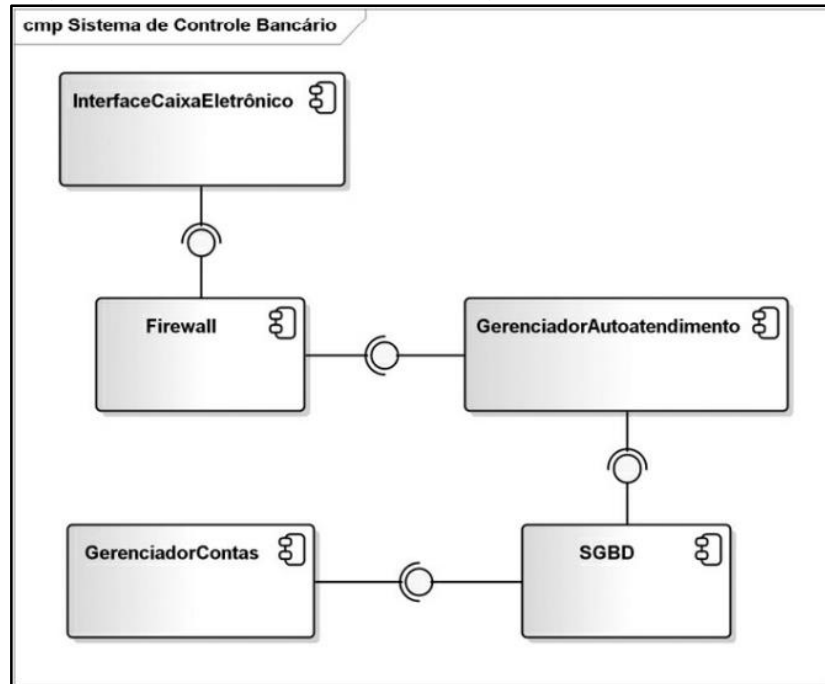
Explicação do Diagrama acima:

O cliente insere as informações do seu cartão por meio da interface “VisaoEmitirSaldo”, que em seguida, envia o número da conta ao “ControleEmitirSaldo”, o qual consulta os dados em “ContaComum” e caso a conta retorne o valor “verdadeiro”, o sistema solicita a senha, assim que o cliente a informar, o controle valida a senha da conta, caso ela esteja correta, o método “emitirSaldo” é chamado, retornando o valor, que é então apresentado ao cliente. O diagrama detalha a interação entre a interface, controle e lógica de negócio.

2.2.4 Diagrama de Componentes

Conforme Guedes (2018), este diagrama representa a organização e dependências entre os componentes físicos do sistema, além disso ajuda a visualizar a estrutura de implementação e as interações entre as diferentes partes do sistema.

Figura 05 — Exemplo de um Diagrama de Componentes.



Fonte: UML 2: Uma Abordagem Prática, 2018.

Explicação do Diagrama acima:

A “InterfaceCaixaEletrônico” conecta-se ao sistema por meio de um Firewall, que atua como barreira de segurança. Esse firewall se comunica com o “GerenciadorAutoatendimento”, o qual é o núcleo de controle das operações do usuário. O “GerenciadorContas” e o SGBD (Sistema de Gerenciamento de Banco de Dados) são acessados a partir do “GerenciadorAutoatendimento”, sendo responsáveis respectivamente pela lógica das contas e pelo armazenamento dos dados.

2.3 Internet of Things (IoT)

Oliveira (2017) afirma que o termo IoT se refere à ação de tornar objetos em equipamentos que têm capacidade de receber e enviar informações de forma inteligente, seja este um evento realizado com ou sem interação humana.

Segundo Santos *et al.* (s.d., p. 7), “A arquitetura básica dos objetos inteligentes é composta por quatro unidades: processamento/memória, comunicação, energia e sensores/atuadores.”

2.4 User Experience (UX)

UX é um termo que se tornou muito popular devido à preocupação das empresas em se destacar pela sua qualidade, e se refere à experiência que o usuário tem com um produto em todos os seus aspectos, conforme Sousa e Bertomeu (2015).

Essa área de estudo, para Teixeira (2022), envolve uma ampla gama de conhecimentos e é relevante para qualquer tipo de mercadoria, pois garante que seus consumidores se sintam satisfeitos, aumentando sua popularidade.

2.5 Tecnologias Utilizadas

Segundo Costa *et al.*, “A tecnologia é produto e processo nosso, nós as produzimos e as empregamos em nossas vidas, transformando-nos (e, constantemente, transformando-as) nesse fluxo do viver.” (COSTA *et al.*, 2020, p. 91). As seguintes tecnologias foram empregadas durante o desenvolvimento do projeto Kyntact:

2.5.1 Baterias íon-lítio

Como explicado por Santos (2018), as baterias de íons de lítio são dispositivos de armazenamento de energia recarregáveis que utilizam íons de lítio, movimentando-se entre os eletrodos durante os processos de carga e descarga.

Em concordância com Michelini (2020), o peso reduzido, longa vida útil e espessura fina fazem as baterias de lítio ótimas para sistemas portáteis que exigem alta densidade energética com baixo volume.

Figura 06 — Baterias de Íons de lítio.



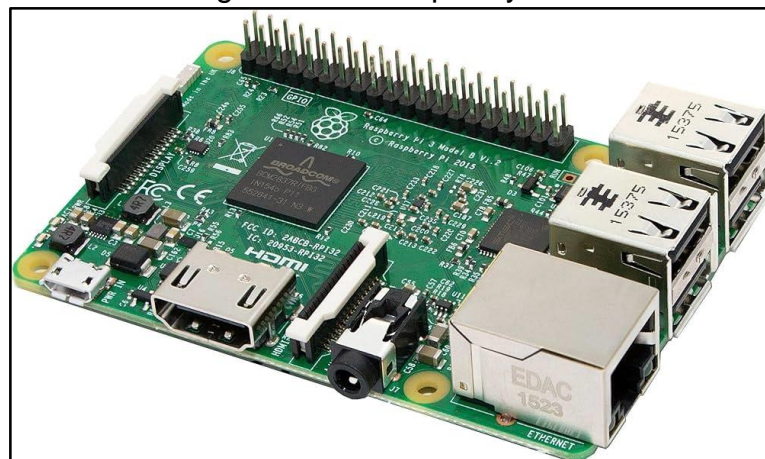
Fonte: Magazine Luiza, 2024.

2.5.2 Raspberry Pi

O Raspberry Pi, em concordância com Silva (2019), foi criado por uma fundação de mesmo nome; é um microcomputador que viabiliza diversas atividades técnicas em qualquer escala, ademais, comporta diversos SOs baseados em Linux.

Como lembra Bertonha (2018), um SO se resume como todos os programas que manipulam e controlam o hardware, ou seja, o conjunto dos softwares pré-instalados no sistema, os quais o usuário pode utilizar e manipular livremente.

Figura 07 — Raspberry Pi 3.



Fonte: Raspberry Pi, 2016.

2.5.2.1 Debian

Segundo a documentação oficial do Debian (2025), o Debian GNU/Linux é uma distribuição do SO popular Linux, mantida por uma comunidade global de voluntários e popularizada pela diversidade de pacotes disponíveis para instalação.

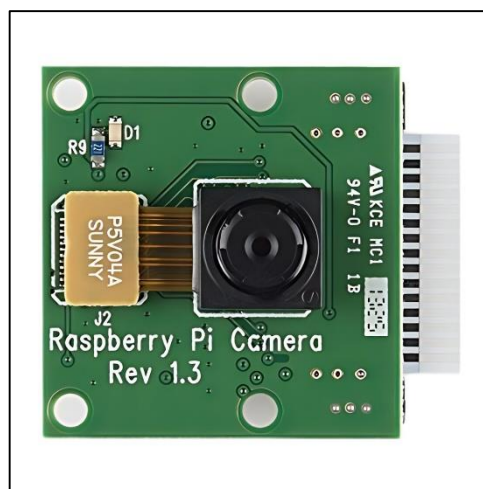
Hertzog e Mas (2017) também destacam que o Debian GNU/Linux não é somente mais uma distribuição do Linux, como também usufrui de um ótimo uso de software livre, e conta com um excelente sistema de gerenciamento de pacotes.

2.5.2.2 Módulo de Câmera para Raspberry Pi

De acordo com Santos, o Raspberry Pi possui uma conector de interface serial que permite a comunicação com um módulo de câmera projetado para o mesmo.

Para Oliveira, o módulo tem uma resolução de 5 megapixels e consegue gravar vídeos em até 1080p, com uma taxa de quadros que varia de 30 até 90 frames por segundo, dependendo da qualidade.

Figura 08 — Módulo de Câmera para Raspberry Pi Rev 1.3



Fonte: Vavuyaan, 2025.

2.5.3 Python

Python é uma linguagem que usufrui de uma sintaxe intuitiva que acaba por economizar o tempo do desenvolvedor ao tirar proveito de diversos recursos como módulos reutilizáveis entre programas heterogêneos, como instrui a Python Software Foundation (2025).

Severance (2020) afirma que a linguagem Python se encontra como uma forma volátil de enviar e receber instruções – algo que permite automatização de tarefas por meio de comandos e técnicas provenientes da linguagem.

Figura 09 — Exemplo de código em Python.

```

1  # Função que realiza cálculo de média
2  def calcular_media(notas):
3      return sum(notas) / len(notas)
4
5  # Atribuição das notas e da média
6  notas_aluno = [7.5, 8.0, 6.5, 9.0, 5.5]
7  media = calcular_media(notas_aluno)
8
9  print(f'Média das notas: {media:.2f}')
```

Fonte: autoria própria, 2025.

O código exemplificado foi desenvolvido em Python; é um conjunto simples de instruções que calculam, atribuem e exibem a média das notas de um aluno. Segue uma explicação mais detalhada do código:

Linhas 2-3: Função que recebe como parâmetro as notas e calcula a média após somar as mesmas.

Linha 6: Atribuição das notas do aluno como 7,5, 8,0, 6,5, 9,0 e 5,5.

Linha 7: Envio das notas atribuídas à função criada anteriormente; resultado da função é armazenado na variável `media`.

Linha 9: Exibição da média usando o comando “print”, que exibe a informação armazenada na variável “media” dentro do terminal.

Nas palavras de Banin (2018), os desenvolvedores que usam Python têm cultivado uma comunidade cercada na linguagem, oferecendo recursos como bibliotecas que criam trilhas novas para o desenvolvimento.

2.5.3.1 Package Installer for Python (PIP)

De acordo com o guia de usuário oficial do empacotamento Python (2025), o PIP é uma ferramenta que gerencia pacotes de dados Python, manipulando os dados conforme necessário.

Figura 10 — Exemplo de uso do PIP.

```
PS C:\kyntact> pip install opencv-python
```

Fonte: autoria própria, 2025.

Na figura acima, encontra-se a instrução que deve ser executada no terminal de comandos para realizar a instalação do PIP.

2.5.4 MySQL

Um SGBD (Sistema de Gerenciamento de Banco de Dados), nas palavras de Florentino (2018), é um tipo de software que permite o usuário criar e gerenciar um banco de dados, foi feito para facilitar os processos entre os usuários e as aplicações.

Segundo Ferrari (2007), O MySQL é um SGBD que utiliza a linguagem SQL) para a manipulação e o controle dos dados. O modelo relacional e compatibilidade com diversas linguagens, tornam o MySQL acessível para vários projetos.

Em concordância com Carvalho (2015), o MySQL destaca-se não apenas por ser gratuito e de código aberto, mas também por oferecer qualidade, estabilidade, segurança e fácil manutenção.

2.5.5 Open Source Computer Vision Library (OpenCV)

Como observa Barelli (2019), a OpenCV, criado pela Intel em 2000, facilita a manipulação e o processamento de imagens, e se tornou uma das principais bibliotecas de visão computacional no mundo.

A OpenCV é uma biblioteca com diversas funções e estruturada em diferentes níveis de abstração, o que permite diversos sistemas com diferentes implementações, de acordo com Marengoni (2009).

2.5.6 Optical Character Recognition (OCR)

O OCR (Optical Character Recognition) é uma tecnologia de conversão de imagens para texto digitalmente manipulável, sendo fundamental em sistemas que precisam reconhecer textos a partir de imagens, como afirma Farias (2023).

Conforme Antonio (2019), o uso de OCR pode ser aprimorado se associado a técnicas de pré-processamento de imagem, o que melhora a legibilidade dos dados e, conseqüentemente, a precisão do texto digitalizado.

2.5.6.1 Tesseract

De acordo com Lederhans (2022), o Tesseract é uma ferramenta de código aberto que realiza o reconhecimento automático de caracteres a partir de imagens, transformando elementos visuais em texto digital.

Complementando, Neto (2023) demonstra que o Tesseract pode ser usado para reconhecer fontes específicas, como as usadas em placas, o que comprova a flexibilidade da ferramenta.

2.5.6.2 PyTesseract

Segundo Leite e Antonello (2017), o PyTesseract é uma biblioteca modificada do Tesseract, feita por diversos programadores, que permite que ele funcione da mesma forma, usando a linguagem de programação em Python.

2.5.7 You Only Look Once (YOLO)

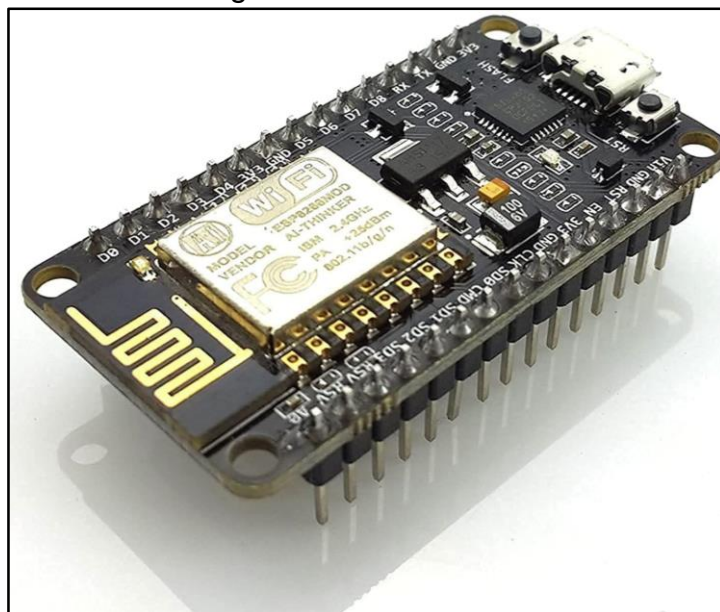
De acordo com a documentação oficial da Ultralytics (2025), YOLO é um modelo de ponta para a detecção de objetos, fundamentado em avanços tecnológicos que permitiram a integração em tempo real com aprendizado de máquina.

2.5.8 ESP8266

Em concordância com Oliveira (2021), o ESP8266 é um microcontrolador amplamente empregado em projetos IoT. Ele oferece suporte ao protocolo TCP/IP, possibilitando conexão direta à rede sem a necessidade de módulos adicionais.

Em conformidade com Batrinu (2018), sua programação pode ser realizada por meio do SDK oficial da Espressif ou da IDE do Arduino, utilizando C ou C++, o que simplifica o desenvolvimento e a prototipagem de sistemas conectados.

Figura 11 — ESP8266.



Fonte: Amazon, 2015.

2.5.9 Linguagem de Programação para Embarcados

Utilizamos durante o projeto a linguagem C++ para programação de embarcados, que, de acordo com Bueno (2002), é uma linguagem de programação criada por Bjarne Stroustrup, a fim de incorporar os conceitos de orientação a objeto, como herança e classes, à linguagem C, para estendê-la.

A linguagem C++ foi baseada em C e criada com o intuito de fornecer uma forma eficiente de gerenciar programas complexos, entretanto ela pode ser usada para desenvolver variados tipos de projetos, parafraseando Schildt (2002).

Figura 12 — Exemplo de código em C++.

```
1  int led = 13;
2  void setup()
3  {
4      pinMode(led, OUTPUT);
5  }
6
7  void loop()
8  {
9      digitalWrite(led, HIGH);
10     delay(500); // Espera 500 milissegundos
11     digitalWrite(led, LOW);
12     delay(500); // Espera 500 milissegundos
13 }
```

Fonte: autoria própria, 2025

O código mostrado acima foi desenvolvido em C++, com o intuito de apagar e acender uma lâmpada de LED, por meio de um microcontrolador. Para garantir um melhor entendimento, leia a explicação abaixo sobre o funcionamento programa:

Linha 1: Declara uma variável do tipo inteiro, chamada “led”, que armazena o valor 13.

Linha 2: Cria uma função com o nome de “setup”, que é executada no momento em que o código é iniciado.

Linha 4: Configura a porta digital número 13, definindo que ela será responsável por realizar a saída de informações, uma vez que ela energizará o led.

Linha 7: Cria uma função denominada “loop”, a qual é executada constantemente até que o programa seja interrompido.

Linha 9: Envia energia ao LED.

Linha 10: Espera 500 milissegundos antes de realizar a próxima ação.

Linha 11: Interrompe o fluxo de energia que estava acendendo a lâmpada.

Linha 12: Aguarda 500 milissegundos novamente.

2.5.10 Jumpers

Um jumper é um equipamento que conecta os componentes de um sistema: “Os jumpers funcionam como pontes elétricas que permitem a transferência de dados e sinais entre os componentes, garantindo a comunicação adequada entre a protoboard e o Arduino.” (MEDEIROS; JÚNIOR, 2024, p. 3)

2.5.11 Resistores

Segundo Gussow (2009), um resistor diminui a corrente elétrica de um circuito, sendo essencial para controlar o fluxo de carga elétrica — podem ser definidos com precisão para certas aplicações.

2.5.12 Transistores

Em conformidade com Boylestad e Nashelsky (2005), o transistor é um semicondutor que pode atuar como amplificador ou chave, ele possui três terminais, sendo eles: base, coletor e emissor. Suas principais configurações são NPN, que em via sinal negativo e PNP, que envia sinal positivo. Eles operam, nos modos corte, saturação e ativo, conforme Nilsson (2016).

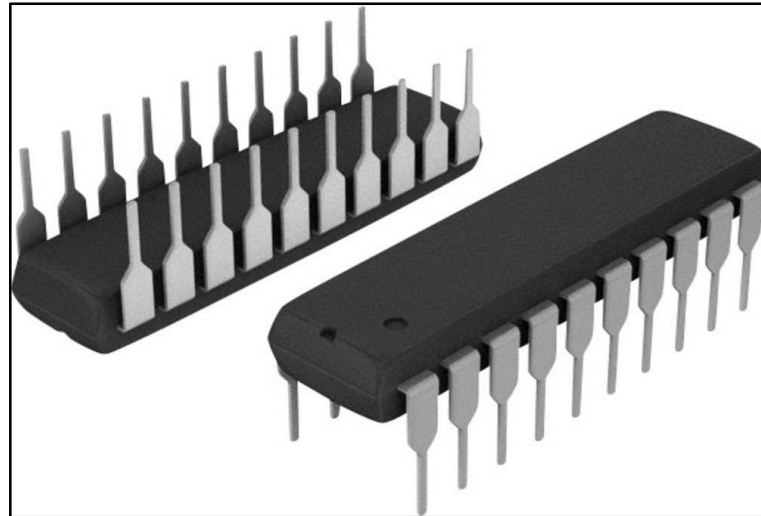
2.5.13 Interruptor

De acordo com Barros (2017), o interruptor é um dispositivo eletromecânico usado para abrir ou fechar circuitos elétricos, permitindo ou interrompendo o fluxo de corrente.

2.5.14 Circuito integrado MCP23017

Em concordância com Leite (2016), MCP23017 é um chip usado para aumentar o número de entradas de um dispositivo, já que possui 16 portas digitais — que podem ser definidas como entrada ou saída e utiliza o protocolo de comunicação I2C.

Figura 13 — Circuito integrado MCP23017.



Fonte: Amazon, 2018.

2.5.15 Solenoides

Solenoides são compostos por um fio longo enrolado, capaz de gerar um campo magnético sempre que é percorrido por uma corrente elétrica, segundo Koroishi (2013).

Em cordato a Pires (2007), solenoides são componentes simples, pois possuem um custo acessível, tem um método fácil de acionamento, trabalham numa grande gama de tamanhos e, ainda assim, possibilitam grandes deslocamentos.

Figura 14 — Solenoide push pull.



Fonte: Amazon, 2023.

2.5.16 Push buttons

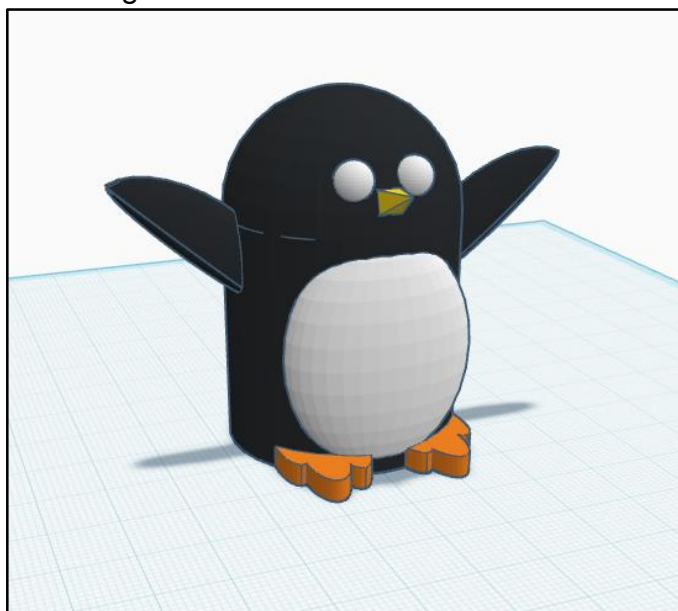
Em concordância com Costa (2022), push buttons são interruptores simples, que impedem ou permitem a passagem de uma corrente elétrica, caso eles estejam sendo pressionados.

2.5.17 Modelagem 3D

Consoante a Botolotto (2014), modelagem 3D é um processo no qual se analisa a superfície de um objeto tridimensional para criar uma representação matemática dele, chamada de modelo tridimensional, por meio de um software especializado.

Esses modelos 3D são elementos geométricos, os quais possuem uma estrutura formada por pontos dentro de um sistema, denominados vértices, que são interligados para formar a representação da figura desejada, segundo Ferreira (2015, apud BONATO, 2006).

Figura 15 — Modelo tridimensional.



Fonte: Autoria própria, 2025.

2.5.18 Impressão 3D

A impressão 3D é uma tecnologia que permite criar objetos físicos a partir de modelos digitais e que está sendo amplamente utilizada em ambientes educacionais, de acordo com a Secretaria de Estado da Educação do Paraná (2018). Como afirma Aguiar (2016), a impressão 3D pode ser aplicada na confecção de instrumentos didáticos personalizados, além da produção de materiais adaptados às necessidades específicas dos alunos.

Figura 16 — Exemplo de impressão 3D.



Fonte: UFSM, 2023.

3 DESENVOLVIMENTO

4 CONSIDERAÇÕES FINAIS

REFERÊNCIAS

- ACIEM, Tânia. **Autonomia Pessoal e Social de Pessoas Com Deficiência Visual Após a Reabilitação**. 2011. Dissertação (Mestrado em Políticas e formas de Atendimento em Distúrbios do Desenvolvimento) - Universidade Presbiteriana Mackenzie, São Paulo, 2011.
- AGUIAR, Leonardo De Conti Dias. **Um Processo para Utilizar a Tecnologia de Impressão 3D na Construção de Instrumentos Didáticos para o Ensino de Ciências**. 2016. Dissertação (Mestre em Educação para a Ciência) - Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru, 2016.
- ALMEIDA, T.; ARAÚJO, F. Diferenças Experienciais Entre Pessoas Com Cegueira Congênita e Adquirida: Uma Breve Apreciação. **Revista Interfaces: Saúde, Humanas e Tecnologia**, Ceará, jun. 2013.
- ANTONIO, David Vitor. **Implementação de protótipo baseado na tecnologia OCR aplicada ao reconhecimento de rótulos para busca em banco de dados**. 2019. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade do Extremo Sul Catarinense, Criciúma, 2019.
- BANIN, Sérgio Luiz. **Python 3 Conceitos e Aplicações: Uma Abordagem Didática**. 1. ed. São Paulo: Saraiva Educação, 2018.
- BARELLI, Felipe. **Introdução à visão computacional: Uma abordagem prática com Python e OpenCV**. 1. ed. São Paulo: Casa do Código, 2019.
- BARROS, José Carlos. **Fundamentos de Instalações Elétricas**. 3. ed. São Paulo: Editora Érica, 2017.
- BATRINU, Catalin. **Projetos de Automação Residencial com ESP8266**. São Paulo: Novatec, 2018.
- BERTONHA, Arian Fernandes. **Sistema de descrição de imagens com comandos por voz para auxílio a deficientes visuais**. 2018. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica – Ênfase em Eletrônica) – Universidade de São Paulo, Escola de Engenharia de São Carlos, São Carlos, 2018.
- BONATO, Raquel. **Revisão Bibliográfica em Realidade Virtual e Modelagem 3D Aplicada à Engenharia Civil e à Exploração e Produção de Petróleo e Gás Natural**. 2015. Trabalho de Graduação (Graduação em Engenharia Civil) - Universidade Estadual Paulista, Guaratinguetá, 2015.

BOOCH, G. *et al.* **UML: guia do usuário.** [s.l.] Rio De Janeiro Elsevier Campus, 2006.

BORGES, José. **Do Braille ao Dosvox – Diferenças nas Vidas dos Cegos.** 2009. Tese (Doutorado em Engenharia de Sistemas) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2009.

BORTOLOTTO, Mariana. **Compatibilização de projetos de uma habitação: verificação de incompatibilidades no sistema de projeção 2D e na modelagem 3D.** 2014. Trabalho de Conclusão de Curso (Graduação em Engenharia Civil) - Universidade Federal de Santa Catarina, Florianópolis, 2014.

BOYLESTAD, R. L.; NASHELSKY, L. **Dispositivos Eletrônicos e Teoria de Circuitos.** 8. ed. São Paulo: Pearson Prentice Hall, 2005.

BRASIL. Instituto Brasileiro de Geografia e Estatística (IBGE). **Censo Demográfico: População residente por tipo de deficiência permanente.** Rio de Janeiro: IBGE, 2010.

BUENO, André Duarte. *Apostila de Programação Orientada a Objetos em C++.* [S. l.], 2012. Disponível em: <https://pt.slideshare.net/lindomar2012/apostila-de-poo-em-c>. Acesso em: 27 maio 2025.

CARVALHO, Vinícius. **MySQL: Comece com o principal banco de dados open source do mercado.** 1. ed. São Paulo: Casa do Código, 2015. E-book.

COSTA, Alan Ricardo *et al.* **Paulo Freire Hoje na Cibercultura.** Porto Alegre: CirKula Ltda., 2020.

COSTA, Ingridy. **Análise da Viabilidade de Um Projeto de Segurança para Motociclistas: Uso Correto do Capacete.** 2022. Trabalho de Conclusão de Curso (Bacharelado em Ciência e Tecnologia) – Universidade Federal Rural do Semi-Árido, Rio Grande do Norte, 2022.

DEBIAN. **Documentação Debian.** 2025. Disponível em: <https://www.debian.org/>. Acesso em: 11 ago. 2025.

Enciclopédia Barsa Universal. Segunda edição. São Paulo: Planeta, 2009.

FARIAS, Walisson. **Aperfeiçoando o reconhecimento óptico de caracteres em imagens de documentos pessoais**. 2023. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande, Campina Grande, 2023.

FERRARI, Fabrício Augusto. **Crie Bancos de Dados em MySQL**. São Paulo: **Universo dos Livros**, 2007.

FLORENTINO, Plínio Tavares. **Gerenciamento e Desenvolvimento em Banco de Dados**. Londrina: Editora e Distribuidora Educacional S.A., 2018.

FOWLER, Martin. **UML Essencial: Um Breve Guia Para a Linguagem-Padrão de Modelagem Para Objetos**. 3. ed. Porto Alegre: Bookman, 2005.

G1. **Anda SP: Deficientes visuais com cão-guia enfrentam desafios de mobilidade em SP**. São Paulo, 2019.

GUEDES, Gilleanes. **UML 2: Uma Abordagem Prática**. 3. ed. São Paulo: Novatec, 2018.

GUSSOW, Milton. **Eletricidade básica**. 2. ed. Tradução de José Lucimar do Nascimento. Porto Alegre: Bookman, 2009.

HERTZOG, Raphaël; MAS, Roland. O Manual do Administrador Debian. 2025. Disponível em: <https://debian-handbook.info/browse/pt-BR/stable/>. Acesso em: 11 ago. 2025.

KOROISHI, Edson. **Controle de Vibrações Em Máquinas Rotativas Utilizando Atuadores Eletromagnéticos**. 2013. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal de Uberlândia, Minas Gerais, 2013.

LEDERHANS, Arnaldo Soares. **TRATAMENTO DE IMAGEM PARA AQUISIÇÃO DA INFORMAÇÃO DA DATA DE VALIDADE EM GARRAFAS DE REFRIGERANTES UTILIZANDO PYTHON OPENCV, TESSERACT E EASYOCR**. 2022. Projeto de Pesquisa (Bacharel em Engenharia de Controle e Automação) - Universidade do Vale do Rio dos Sinos, São Leopoldo, 2022.

LEITE, Daniel. **Dispositivo Robótico para Assistência à Locomoção de Pessoas Idosas em Ambientes Urbanos**. 2016. Dissertação (Mestrado em Engenharia Elétrica) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2016.

LEITE, Leonardo. ANTONELLO, Ricardo. **Identificação automática de placa de veículos através de processamento de imagem e visão computacional**. Semana da Educação Ciência, Tecnologia e Cultura (SECITEC), 2017.

MARENGONI, Maurício; STRINGHINI, Denise. Tutorial: Introdução à Visão Computacional usando OpenCV. **Revista de Informática Teórica e Aplicada**, [S. l.], v. 16, n. 1, p. 125–160, mar. 2010. Disponível em: https://seer.ufrgs.br/index.php/rita/article/view/rita_v16_n1_p125. Acesso em: 27 abr. 2025.

MEDEIROS, André Uesley; JÚNIOR, Idalmir. **Construção de um quadro de transferência automática**. 2024. Trabalho de Conclusão de Curso (Bacharelado em Ciência e Tecnologia) — Universidade Federal Rural do Semiárido, Rio Grande do Norte, 2024.

MENDES, Felipe Soares. **Sistema de monitoramento de parâmetros ambientais na piscicultura: redes IOT**. 2023. Dissertação (Mestrado em Engenharia de Sistemas Eletrônicos) – Universidade de São Paulo, São Paulo, 2023.

NETO, Aroldo de Sá Vargas. **RECONHECIMENTO AUTOMÁTICO DE PLACAS VEICULARES DO MERCOSUL UTILIZANDO O TESSERACT OCR**. 2023. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação) - Instituto Federal do Espírito Santo, Serra, 2023.

NILSSON, James W. *Circuitos Elétricos*. 10. ed. São Paulo: Pearson, 2016.

OLIVEIRA, Regina Fatima. **Braille!? O que é isso?**. São Paulo: Fundação Dorina Nowill, 2018.

OLIVEIRA, Sérgio. **Internet das coisas com ESP8266, ARDUINO e RASPBERRY PI**. São Paulo: Novatec, 2017.

OLIVEIRA, Tassany. **Desenvolvimento de um Dispositivo para Segurança Biométrica Facial Usando Câmera NoIR**. 2024. Monografia (Bacharelado em Engenharia da Computação) – Universidade Federal Rural da Paraíba, 2024.

PIRES, Diogo. **Bomba pulsátil de duto flexível**. 2007. Trabalho de Formatura (Graduação em Engenharia Mecânica) - Universidade de São Paulo, São Paulo, 2007.

PYTHON PACKAGING USER GUIDE. **Guia de Usuário para Empacotamento de Python**. 2024. Disponível em: <https://packaging.python.org/pt-br/latest/>. Acesso em: 14 jul. 2025.

PYTHON SOFTWARE FOUNDATION. **Documentação Python**. 2025. Disponível em: <https://docs.python.org/pt-br/3.13/>. Acesso em: 17 abr. 2025.

SANTOS, Célia. **Baterias de íons lítio para veículos elétricos**. *Revista IPT: Tecnologia e Inovação*, São Paulo, v. 2, n. 9, p. 62–82, dez. 2018. Disponível em: <https://www.researchgate.net/publication/330089304>. Acesso em: 26 maio 2025.

SANTOS, Wilka. **Exemplos De Utilização Do Raspberry Pi Para Auxílio No Ensino Fundamental**. 2018. Monografia (Bacharelado em Ciência da Computação) – Universidade Federal Rural do Semi-Árido, 2018.

SCHILDT, Herbert. **C++ guia para iniciantes**. Rio de Janeiro: Ciência Moderna, 2002.

SECRETARIA DE ESTADO DA EDUCAÇÃO DO PARANÁ. Superintendência da Educação. Departamento de políticas e Tecnologias Educacionais. **Impressora 3D: imaginar, planejar e materializar**. 2018.

SEVERANCE, Charles R. **Python para todos: Explorando dados com Python 3**. Tradução de Antonio Marcos *et al.* Porto Alegre: Bookman, 2016.

SILVA, Cássio Moreira. **Sistema de controle de apresentação por meio de Raspberry Pi**. 2019. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Universidade Federal de Uberlândia, Monte Carmelo, 2019.

SOUSA, Milene; BERTOMEU, João Vicente Cegato. UX Design na criação e desenvolvimento de aplicativos digitais. **Informática na educação: teoria & prática**, Porto Alegre, v. 18, n. 2, p.128, jul./dez. 2016.

TEIXEIRA, Fabricio. **Introdução e boas práticas em UX Design**. São Paulo: Casa do Código, 2022.

ULTRALYTICS. **YOLO**. Disponível em: <https://docs.ultralytics.com/pt/>. Acesso em: 11 ago. 2025.