

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ETEC DA ZONA LESTE

MTEC Desenvolvimento de Sistemas AMS

Erick Ferreira Lima

Gustavo Rodrigues Leite Da Silva

Hernandes Arthur Da Silva Santos

**ESTOK: automatização de controle de produtos em mercados
autônomos com RFID**

São Paulo

2025

Erick Ferreira Lima
Gustavo Rodrigues Leite Da Silva
Hernandes Arthur Da Silva Santos

**ESTOK: automatização de controle de produtos em mercados
autônomos com RFID**

**Trabalho de Conclusão de Curso
apresentado à Etec [nome da escola],
como requisito parcial para a obtenção
do título de Técnico em [nome do
curso], sob orientação do(a)
Professor(a) [nome do orientador].**

São Paulo

2025

ERICK FERREIRA LIMA

GUSTAVO RODRIGUES LEITE DA SILVA

HERNANDES ARTHUR DA SILVA SANTOS

**ESTOK: automatização de controle de produtos em mercados
autônomos com RFID**

**Trabalho de Conclusão de Curso apresentado à Escola Técnica Estadual
[Nome da Etec], como requisito parcial para a obtenção do título de Técnico
em [nome do curso], sob a orientação do(a) Prof.(a) [nome do orientador(a)].**

Aprovado em ____/____/____.

Banca Examinadora:

Prof.(a) [Nome do Orientador(a)]

Etec [Nome da Etec]

Prof.(a) [Nome do membro 1 da banca]

Etec [Nome da Etec]

Prof.(a) [Nome do membro 2 da banca]

Etec [Nome da Etec]

DEDICATÓRIA

Dedico este trabalho a Deus, que me deu força nos momentos mais difíceis, à minha família, pelo apoio incondicional, e aos meus amigos, que sempre estiveram ao meu lado durante essa jornada.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me conceder saúde e sabedoria para concluir mais essa etapa. Ao meu orientador(a) [Nome do Professor], pela paciência, dedicação e orientações ao longo da realização deste trabalho. Aos meus colegas e professores da Etec [Nome da sua Etec], pelo companheirismo, apoio e conhecimento compartilhado. À minha família, por todo o suporte emocional, incentivo e amor.

“Escolhe um trabalho de que gostes, e não terás que trabalhar nem um dia na tua vida.”

– Confúcio

AULAS PRÁTICAS NO ENSINO REMOTO: CONVERGÊNCIAS E DISRUPTURAS

Resumo

Este artigo busca levantar quais as metodologias utilizadas pelos professores das escolas técnicas nas aulas práticas oferecidas a distância, estabelecendo interlocução entre as necessidades de aprendizagem em oficinas e laboratórios transpostas para as ferramentas disponíveis no ensino remoto. Para tanto, realizou-se uma pesquisa on-line que levantou os pontos convergentes para as dificuldades impostas pela impossibilidade da oferta de aulas práticas presenciais, ao ponto que também apresentou as possibilidades de ensino e de aprendizagem propiciadas por essas ferramentas. Cabe destacar que os resultados obtidos advêm de práticas e experiências docentes em construção, por isso fica em evidência a sua importância no que diz respeito à validação do cenário atual, que precisa ser encarado como uma oportunidade de ruptura de padrões, conceitos e concepções sobre ensino e aprendizagem em aulas práticas. Nesse sentido, entende-se que o presente artigo pode contribuir para futuros debates e estudos sobre a oferta de Educação Profissional no modelo híbrido.

Palavras-chave: aulas práticas; metodologias ativas; ensino híbrido.

PRACTICAL CLASSES IN REMOTE TEACHING: CONVERGENCES AND DISRUPTIONS

ABSTRACT

This article aims to identify the methodologies employed by technical school teachers in remote practical classes, establishing a dialogue between the learning needs typically addressed in workshops and laboratories and the tools available in distance education. To this end, an online survey was conducted to highlight both the convergences regarding the difficulties imposed by the impossibility of offering in-person practical classes and the teaching and learning possibilities afforded by these tools. It is important to note that the results stem from ongoing teaching practices and experiences, underscoring their significance in validating the current scenario. This situation should be viewed as an opportunity to disrupt established patterns, concepts, and understandings of teaching and learning in practical classes. In this sense, this article seeks to contribute to future debates and studies on the delivery of Vocational Education in a hybrid model.

Keywords: practical classes; active methodologies; hybrid teaching

LISTA DE ILUSTRAÇÕES

Figura 1 — Diagrama de casos de uso sistema de biblioteca.....	15
Figura 2 — Diagrama de atividade empréstimo de livro.....	17
Figura 3 — Diagrama de sequência empréstimo de livro.....	19
Figura 4 — Placa ESP32 com anotações de componentes.....	22
Figura 5 — Código que soma dois números em C++.....	24
Figura 6 — Resultado do código que soma dois números em C++.....	25
Figura 7 — Estrutura básica de uma página em HTML.....	27
Figura 8 — Exemplo de uma tela HTML para cadastro de produtos.....	27
Figura 9 — Resultado do código HTML de uma página de cadastro de produtos.....	28
Figura 10 — Exemplo de conexão entre HTML e CSS.....	30
Figura 11 — Exemplo de escrita na CSS.....	31
Figura 12 — CSS da tela de cadastro de produtos.....	32
Figura 13 — Resultado da tela de cadastro HTML e CSS em conjunto.....	33
Figura 14 — Função juntar nome e sobrenome NodeJS.....	35
Figura 15 — Função juntar nome e sobrenome NodeJS.....	36

LISTA DE ABREVIATURAS E SIGLAS

ASCII – American Standard Code for Information Interchange (Código Padrão ‘Americano para o Intercâmbio de Informação)

CSS – Cascading Style Sheets (Folhas de Estilo em Cascata)

DNS – Domain Name System (Sistema de Nomes de Domínio)

FTP – File Transfer Protocol (Protocolo de Transferência de Arquivos)

HTML – HyperText Markup Language (Linguagem de Marcação de Hipertexto)

HTTP – HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)

HTTPS – HyperText Transfer Protocol Secure (Protocolo Seguro de Transferência de Hipertexto)

I/O – Input/Output (Entrada e Saída de Dados)

IDE – Integrated Development Environment (Ambiente de Desenvolvimento Integrado)

IoT – Internet of Things (Internet das Coisas)

JS – JavaScript (Linguagem de Programação para Web)

Node.js – Plataforma para execução de JavaScript no servidor

RFID – Radio-Frequency Identification (Tecnologia de Identificação por Rádio Frequência)

UHF – Ultra High Frequency (Frequência Ultra Alta)

URL – Uniform Resource Locator (Localizador Uniforme de Recursos)

SUMÁRIO

1 INTRODUÇÃO.....	12
2 REFERENCIAL TEÓRICO.....	13
2.1 Automação No Controle De Estoque.....	13
2.2 UML, Linguagem De Modelagem Gráfica.....	13
2.3 Diagrama de casos de uso.....	14
2.4 Diagrama de atividade.....	16
2.5 Diagrama de sequência.....	18
2.6 Internet Das Coisas (IoT).....	20
2.7 Etiqueta RFID.....	20
2.8 Leitor RFID.....	21
2.9 ESP32.....	22
2.10 ArduinoIDE.....	22
2.11 C++.....	23
2.12 HTML.....	26
2.13 CSS.....	29
2.14 JAVASCRIPT.....	34
2.14.1. NodeJS.....	34
2.15 Firebase.....	36
REFERÊNCIAS.....	37
GLOSSÁRIO.....	41

1 INTRODUÇÃO

O aumento da popularidade de mercados autônomos no Brasil fez do país um dos protagonistas nesse setor. Esse aumento de popularidade é atribuído à facilidade e segurança promovidas por esse modelo de negócio, sendo um meio tanto de fonte de renda para o dono/franqueado, quanto uma fonte de comodidade para os clientes (LEÃO, 2024).

No entanto, esse aumento de popularidade pode gerar também problemas para o gerente do mercado se essa automatização não for feita para garantir um real autonomia, problemas como dependência de verificação constante da mercadoria disponível e da quantidade em estoque por parte do gerente do mercado podem surgir, afetando não somente o gerente do mercado ao diminuir sua comodidade, como também os clientes que podem sofrer com a falta de produtos, o que é chamado de ruptura de estoque, resultado de um produto indisponível, porém existente no estoque.

Problemas como esse devem ser contornados com o uso de novas tecnologias, buscando trazer um aumento de eficiência, principalmente na parte logística (ROVAROTO, 2024). Esse aumento da rastreabilidade de produtos será possível devido à tecnologia de identificação por radiofrequência (RFID), com suas etiquetas emissoras de radiofrequência e leitores de radiofrequência, que poderão ser utilizados para identificar a movimentação de saída e entrada de produtos das prateleiras de forma automática, disponibilizando essa informação para fácil uso dos gestores.

A tecnologia de rádio frequência vem sendo utilizada no setor varejista, promovendo um aumento de eficiência atrelado ao aumento da rastreabilidade de produtos. Uma das pioneiras na implementação dessa tecnologia, as lojas Renner, que já desfrutam de 100% de suas lojas utilizando essa tecnologia, registrou melhoria de 64% na acuracidade dos estoques, agilizando o trabalho de reposição e evitando a ruptura de estoque (INFORCHANNEL, 2022).

O presente estudo visa aplicar tecnologias IoT e sistemas inteligentes para facilitar a eficiência por meio do aumento da rastreabilidade de produtos, com foco em estabelecimentos localizados na cidade de São Paulo.

2 REFERENCIAL TEÓRICO

A atual seção visa analisar o cenário atual de pesquisa sobre mercados autônomos em língua portuguesa, apresentando de forma clara e respaldada a proposta da integração de um sistema baseado na tecnologia de identificação por rádio frequência (RFID), solucionando a falta de automatização no monitoramento de estoque em mercados autônomos.

2.1 Automação No Controle De Estoque

No Brasil, o setor varejista enfrenta problemas de administração de estoque que acarretam prejuízos para as empresas e em perda de qualidade da experiência do consumidor (KPMG, 2024). Segundo Wenceslau (2024), esses problemas podem ser atribuídos a imprecisão de processos orientados à ação humana ou ao mau funcionamento de softwares.

A automação no controle de estoque se torna essencial para o desenvolvimento e a diminuição de problemas de imprecisão em todos os setores. Os tópicos a seguir buscam explicar uma das possíveis soluções para o problema anteriormente mencionado. Com foco em mercados autônomos, a solução aumentará a autonomia na reposição de estoque pelos gerentes.

É importante notar também a importância dessa pesquisa para o incentivo à busca e desenvolvimento de soluções e inovações na área de mercados autônomos. Em vista do desenvolvimento e crescimento dessa modalidade no Brasil, e da iminente mudança na atual interação do cliente com o varejo causada pela implementação de novas tecnologias, pesquisas referentes a esse setor da indústria se tornam primordiais para a criação de uma base sólida de conhecimentos em língua portuguesa (TORTORELLI, 2018).

2.2 UML, Linguagem De Modelagem Gráfica

Para que a ambição de construir esse projeto que aumentará a rastreabilidade dos produtos, primeiro precisaremos fazer um bom planejamento para determinarmos quais as funcionalidades que o nosso sistema terá e como ele interagirá com agentes externos. Uma boa forma de começar a visualização é por meio de diagramas, mais especificamente diagramas de UML.

A UML é uma forma de descrever o seu projeto de forma gráfica para facilitar a compreensão de outras pessoas. Ela ajuda a definir os processos a serem seguidos no desenvolvimento do software (FOWLER, 2000).

Independente da complexidade do projeto, é importante desenvolver diagramas, pois ajudará na escalabilidade desse software. Os diagramas contemplarão tanto a descrição geral do projeto quanto detalhes específicos do sistema (GUEDES, 2018).

É importante ressaltar que os diagramas suportam desde uma visualização mais simples quanto uma visualização extremamente detalhada, aumentando sua complexidade. A seguinte abordagem dos diagramas busca a explicação do que utilizaremos como padrão, poupando o leitor de explicações de características, componentes e funcionalidades que não contribuirão para o real uso no nosso projeto. Dada a característica de explicar desde o nível básico até especificações mais detalhadas do sistema, darei início à explicação dos diagramas em ordem de complexidade.

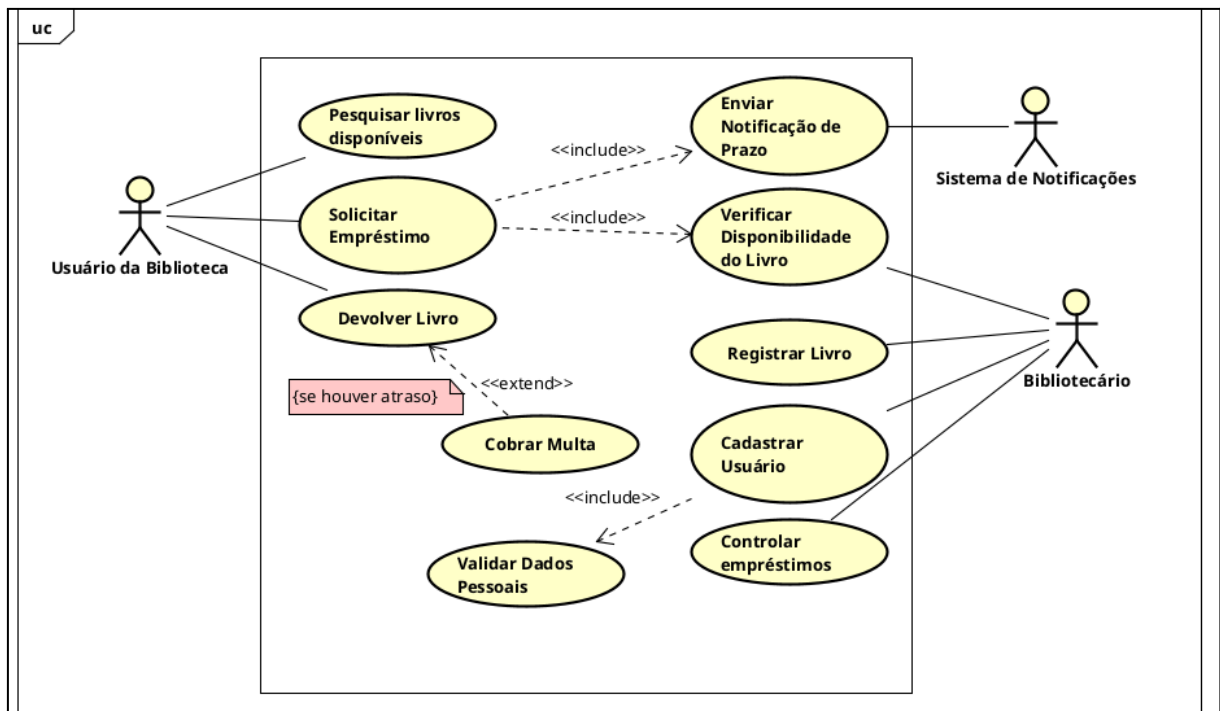
2.3 Diagrama de casos de uso

Um diagrama de caso de uso é a descrição dos cenários envolvendo sua aplicação. Sendo esses cenários as interações em que os usuários, ou outras entidades, terão com o seu sistema (FOWLER, 2000).

No nosso projeto, um diagrama como o de casos de uso serve para começar a visualizar as interações que existiram, possibilitando a delimitação e definição de como lidaremos com elas.

Para representar essas interações, utilizaremos os seguintes elementos. Atores, casos de uso e associações. Com esses três elementos é possível criar uma vasta possibilidade de interações, como demonstrado no exemplo a seguir.

Figura 1 — Diagrama de casos de uso sistema de biblioteca



Fonte: Do próprio autor, 2025

Para compreender esse diagrama de caso de uso, primeiro começaremos definindo cada componente dentro dele. Os principais componentes representados na figura são atores, casos de uso e associações de inclusão e de extensão.

De acordo com Guedes (2018), os atores representam os usuários do sistema, podendo também representar funções ou softwares específicos que interagem com o sistema. No caso do diagrama representado na figura, os atores são usuários da biblioteca, bibliotecário e sistema de notificação, sendo o primeiro ator primário, portanto representados do lado esquerdo do diagrama, e bibliotecário e sistema de notificação sendo atores secundários, tendo sua representação do lado direito.

Esses atores interagirão com o sistema no que chamaremos de casos de uso. Os casos de uso representam toda a ação que um ator tem para chegar em um determinado resultado observável. Cada caso de uso normalmente se inicia com um verbo e descreve a ação que o ator executa no sistema (IBM, 2021).

Na figura também pode-se observar que diferentes casos de uso se conectam. Essas conexões são chamadas de associações, sendo as representadas na figura as de inclusão e de extensão. Um relacionamento de inclusão,

representado por uma seta aberta com linha tracejada indo do caso de uso principal para o caso de uso incluído, se dá quando o acionamento do caso de uso que está sendo incluído é obrigatório ao acionamento do caso de uso principal (base) (GUEDES, 2018).

Já o caso de uso de extensão, representado por uma seta aberta com linha tracejada indo do caso de uso estendido para o caso de uso principal, não tem a obrigatoriedade de ser executado quando o caso de uso principal é acionado, tendo sua execução atribuída a validações, consistências ou condições, esses podendo ou não ser acompanhadas de restrições em associações, que serão representadas em formato de folha com orelha dobrada contendo uma descrição entre chaves (GUEDES, 2018).

No exemplo, o usuário da biblioteca se conecta com o caso de uso “solicitar empréstimo”, que tem uma relação de inclusão com o caso de uso “verificar a disponibilidade do livro”. Isso significa que sempre que o usuário da biblioteca solicitar empréstimo, obrigatoriamente o bibliotecário terá que verificar a disponibilidade do livro.

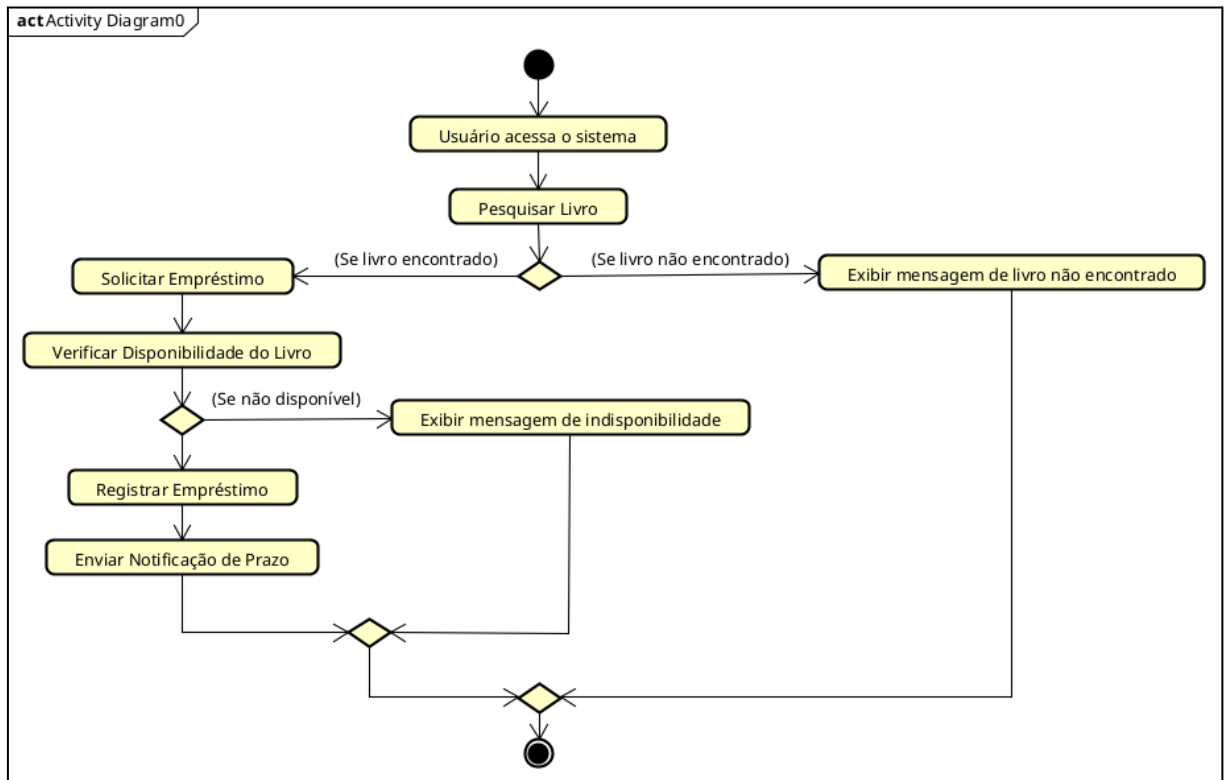
Já um exemplo de extensão se dá na figura quando o usuário da biblioteca aciona o caso de uso “devolver livro”, o caso de uso “cobrar multa” é estendido, mas somente se houver atraso, essa condição é descrita em uma restrição.

2.4 Diagrama de atividade

Agora abordaremos o diagrama de atividade, que descreverá as atividades, ou seja, as ações do usuário no sistema detalhadamente (FOWLER, 2000). Pode-se entender o diagrama de atividade como uma extensão dos principais casos de uso do projeto.

A figura a seguir representa o mesmo sistema de biblioteca que vínhamos abordando. Nela podemos observar o detalhamento das principais interações que o usuário terá com o sistema.

Figura 2 — Diagrama de atividade empréstimo de livro



Fonte: Do próprio autor, 2025

Seguindo o padrão utilizado no diagrama de casos de uso, primeiro explicarei os principais elementos do diagrama de atividades para em seguida explicar a figura.

O diagrama de atividade tem seu início no nó inicial, representado pela figura de um círculo preto. Ele representa o início da interação do usuário, quando uma atividade é invocada (GUEDES, 2018). Esse nó inicial será ligado a uma ação, que nada mais é do que uma unidade funcional, representada por um retângulo com a ação descrita, identificando uma ação do usuário ou do sistema (IBM, 2025).

A figura do losango representa uma decisão ou ramificação que pode transformar uma única ação em vários estados paralelos (MICROSOFT, 2025).

O diagrama de atividade será finalizado com o nome de final de atividade representado por um círculo preenchido em um círculo vazio, todo diagrama deve culminar nele (GUEDES, 2018).

Na figura, o usuário acessa o sistema e em seguida pesquisa o livro, essa ação é ramificada em solicitar empréstimo, caso o livro seja encontrado, ou exibir mensagem de livro não encontrado. Ao solicitar empréstimo, o sistema terá que verificar a disponibilidade do livro, prosseguindo para registrar o empréstimo do livro e enviar a notificação de prazo, caso o livro não estiver disponível o sistema exibirá a mensagem de indisponibilidade.

2.5 Diagrama de sequência

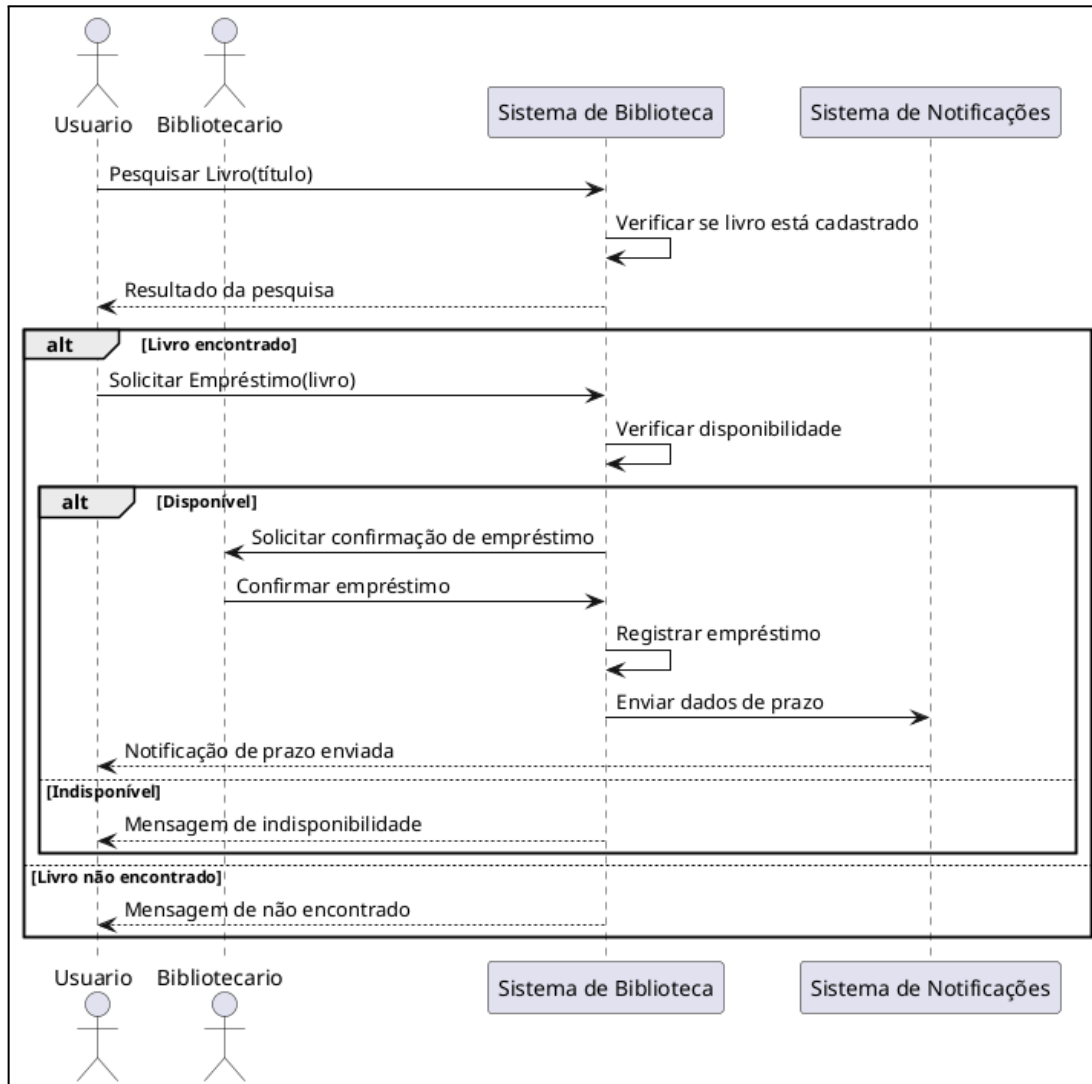
Uma maneira de representar as interações dos objetos que compõe o sistema de forma detalhada e cronológica é utilizando os diagramas de sequência. As interações são representadas por mensagens trocadas entre os objetos (MICROSOFT, 2025).

Todo o período da interação é atrelada a linhas de vida, representadas visualmente por linhas verticais tracejadas saindo do objeto. Dessas linhas de vida partem as mensagens trocadas entre os objetos, representadas por uma linha sólida com ponta preenchida. Mensagens de retorno, que respondem a uma mensagem anterior, são representadas por uma linha tracejada que retorna para a linha de vida a qual deverá receber a resposta (GUEDES, 2018).

Para representar condições, o diagrama de sequência utiliza os fragmentos combinados, representados por um quadrado que engloba todo o período da condição. Anotações comuns nesses fragmentos combinados são os de alternatividade (alt) e de loop, que são para casos em que o diagrama se ramifica em duas direções condicionais, ou repete o ciclo descrito, respectivamente (IBM, 2021).

A figura a seguir mostrara um diagrama de sequência baseado no exemplo de biblioteca em que vinhamos trabalhando.

Figura 3 — Diagrama de sequência empréstimo de livro



Fonte: Do próprio autor, 2025

O diagrama de sequência apresentado possui dois atores, sendo eles usuário e bibliotecário. Outros objetos também representados são de sistema de biblioteca e sistema de notificação.

A interação se inicia quando o usuário pesquisa livro, como parâmetro é passado o título do livro. Esses parâmetros, segundo Guedes (2018), são informações compartilhadas entre objetos, e devem ser utilizados para melhorar o entendimento do diagrama, podendo ser omitidos caso o autor entenda que beneficiará a compreensão.

A interação do diagrama prossegue com o sistema da biblioteca verificando se o livro está cadastrado e respondendo para o usuário o resultado da pesquisa. Usuário então, em uma condicional em que o livro foi encontrado, solicita o empréstimo do livro, ação essa prosseguida pela verificação de disponibilidade pelo sistema da biblioteca. Caso o livro esteja disponível o sistema da biblioteca solicitar a configuração do empréstimo para bibliotecário, que confirmará fazendo com que o sistema de biblioteca registre o empréstimo e envie os dados de prazo para o sistema de notificação, que por sua vez enviará essa informação para o usuário, caso o livro esteja indisponível o sistema de notificação retornará a mensagem de indisponibilidade do livro. O sistema da biblioteca informará o usuário caso o livro não seja encontrado na pesquisa inicial.

2.6 Internet Das Coisas (IoT)

O nosso projeto interagirá tanto com o meio digital quanto com o meio físico, identificando e registrando interações com a prateleira do mercado autônomo, essa interação resultará na obtenção de dados, essa característica faz com que o nosso projeto seja classificado como Internet das Coisas (Internet of Things).

A definição da Internet das Coisas não tem uma definição exata, porém, segundo Magrani (2018), pode-se afirmar que dispositivos IoT (Internet of Things) possuem características de conectividade com a Internet e compartilhamento de informações entre si, possibilitando cenários de integração entre dispositivos e serviços.

Essas características, portanto, possibilitam a criação de dispositivos voltados para a melhoria de processos cotidianos e conectados ao contexto geral da internet.

2.7 Etiqueta RFID

No âmbito físico do nosso projeto, começaremos identificando essa interação com a prateleira do supermercado autônomo, esse registro de interação acontecerá por intermédio de leitores e etiquetas RFID.

Conforme os estudos de Santini (2008 apud OLIVEIRA, 2023), a estrutura de uma etiqueta RFID corresponde a um chip e uma antena envolvidos por algum material. A etiqueta faz contato com os sinais transmitidos por meio da antena do leitor RFID.

Essa etiqueta contém as informações necessárias para identificação do produto, possibilitando toda de dinâmica de reconhecimento exato do produto que entrou na prateleira. A formatação das informações que estão presentes nas etiquetas podem ser recebidas nas seguintes formas: caracteres hexadecimais, decimais ou ASCII (COSTA, 2018).

Conforme analisa Campos (2021), a etiqueta RFID passiva não possui alimentação própria e, por conta disso, a emissão de dados e a própria alimentação da etiqueta vêm das ondas magnéticas emitidas pela antena do leitor.

2.8 Leitor RFID

Segundo Santini (2008 apud PREDIGER; FREITAS; SILVEIRA, 2014), a comunicação entre o leitor RFID e a tag RFID é feita via uma antena que pode transmitir essas informações e processá-las para uso em outros sistemas. No nosso sistema é essa antena ficaria localizada nas proximidades da prateleira, identificando as interações da entrada e saída de produtos.

De acordo com Gonçalves (2025), ocorre a emissão de ondas de radiofrequência por meio da antena do leitor RFID. Essas ondas uma vez em contato com a etiqueta energizarão a mesma fazendo com que ela envie o sinal e as suas informações de volta ao leitor.

Conforme Costa (2018), o leitor recebe os dados quando as etiquetas entram da área de alcance do mesmo. Devido essa característica, a posição do leitor deverá ser analisada para obtermos a melhor leitura possível dentro do espaço delimitado da prateleira.

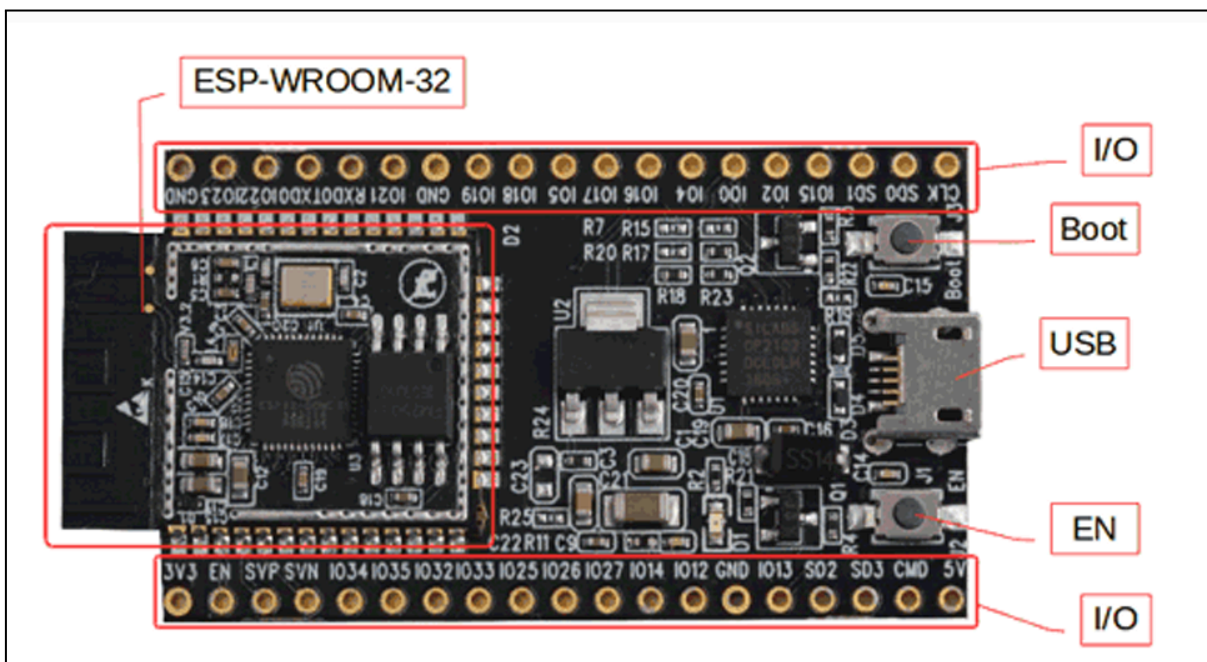
2.9 ESP32

Após as leituras das informações da etiqueta serem feitas pelo leitor RFID, o mesmo precisará comunicar essas informações para um microcomputador, para podermos tratá-las e enviá-las para a interface que será vista pelo gerente do mercado autônomo.

A comunicação entre o leitor e o sistema será feita pela placa ESP32, sendo esse um microcomputador desenvolvido pela empresa chinesa Espressif Systems, capaz de executar o processamento de dados e se comunicar com outros meios via

Wi-Fi ou Bluetooth (PET, 2021). Essa característica é importante para não ser necessário a conexão de internet cabeada, possibilitando uma maior escalabilidade na infraestrutura de leitores.

Figura 4 — Placa ESP32 com anotações de componentes



Fonte: PET. Introdução à Programação Embarcada. Minas Gerais, 2021

2.10 ArduinoIDE

Para podermos passar as instruções de como deve ser feita esse primeiro processamento dos dados lidos pelo leitor RFID para o microcomputador ESP32, utilizaremos um Ambiente de Desenvolvimento Integrado (IDE). Segundo Silva (2021 apud DAMACENO; FUNG; SARTORI, 2024), o ESP32 é compatível com o ArduinoIDE, ambiente primariamente utilizado para placas da família Arduino, porém tendo compatibilidade com a placa ESP32. Devido à sua interface clara que priorizo bom entendimento, e a compatibilidade com extensões e bibliotecas que facilitaram o andamento do projeto, essa IDE foi escolhida.

2.11 C++

As instruções passadas para o ESP32 pelo Arduino IDE são feitas na linguagem de programação C++. De acordo com a documentação oficial do C++ (2023), a linguagem desde o seu início se destaca fortemente pela sua capacidade de ser eficiente e abranger diferentes tipos de sistemas.

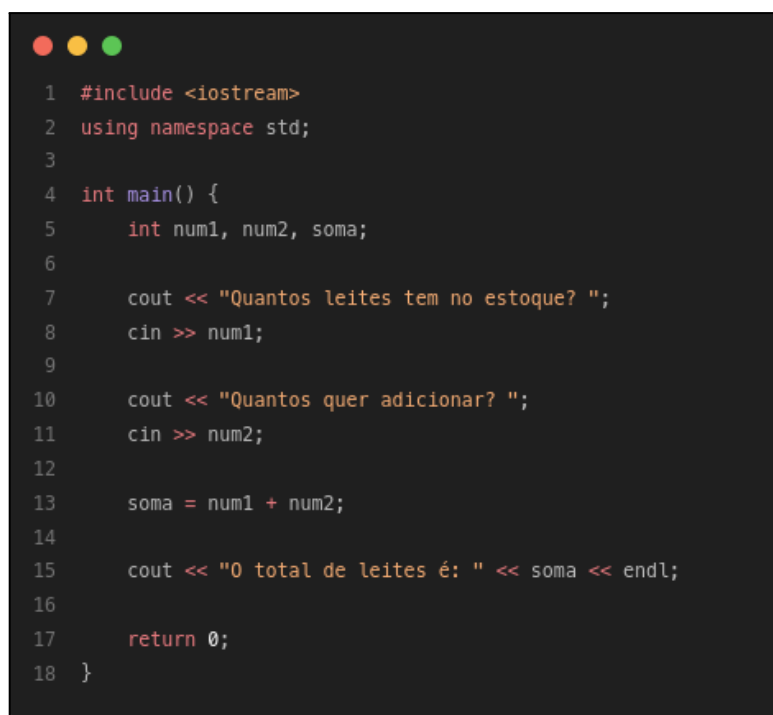
A escolha de C++ junto com um ESP32 se deve à sua alta velocidade de processamento, utilizando pouca memória. Uma forte qualidade da linguagem quando usada em dispositivos IoT.

Consoante com Wiener e Pinson (1991), C++, por ser orientada a objeto, é uma linguagem de programação extremamente eficaz quando se pensa em soluções mais humanizadas, que facilitam o entendimento do código. A linguagem também tem um acesso melhor ao hardware, parte física, que possibilitará a criação de melhores softwares, parte lógica.

Segundo Game (2018 apud CASTRO, 2019), C++ possui uma utilização de pouquíssimos requisitos computacionais, o que faz a linguagem ter um ótimo desempenho.

No exemplo a seguir, será exibido e explicado um código em C++ que recebe o número de um produto disponível em um estoque, recebe quantas unidades desse mesmo produto serão repostas e retorna a quantidade total desse produto no estoque.

Figura 5 — Código que soma dois números em C++

A screenshot of a code editor with a dark background and light-colored text. The code is written in C++ and is numbered from 1 to 18. It includes the <iostream> header, uses the std namespace, and defines a main function. Inside main, it declares three integer variables: num1, num2, and soma. It then prompts the user for two numbers using cout and cin, calculates the sum, and displays the result. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int num1, num2, soma;
6
7     cout << "Quantos leites tem no estoque? ";
8     cin >> num1;
9
10    cout << "Quantos quer adicionar? ";
11    cin >> num2;
12
13    soma = num1 + num2;
14
15    cout << "O total de leites é: " << soma << endl;
16
17    return 0;
18 }
```

Fonte: Do próprio autor, 2025

Abaixo segue uma descrição do funcionamento do código:

Linha 1: Aqui estamos incluindo a biblioteca “iostream” para conseguir exibir textos no terminal e receber valores do usuário.

Linha 2: Evita que precisamos escrever “std:” antes de qualquer elemento que venha da biblioteca “iostream”, como os “count” e “cin” que abordaremos mais em diante.

Linha 4: Nessa linha é feito a inicialização da nossa classe, chamada “main”. Por sua vez, uma classe é onde o computador começa a executar o código.

Linha 5: Estamos criando 3 variáveis do tipo “int”, que indica que podemos armazenar somente valores inteiros

Linha 7: É solicitado que o usuário insira o números de leites disponíveis no estoque, usando o “count” atrelado ao “<<” para imprimir a mensagem na tela do terminal.

Linha 8: Após o usuário inserir o valor, junto com “cin” e “>>” armazenamos ele na variável “num1”

Linha 10-11: Através dos mesmo recursos utilizados nas linhas 7 e 8, é solicitado que o usuário insira quantos leites ele quer repor e após a inserção do valor, esse número fica armazenado na variável “num2”

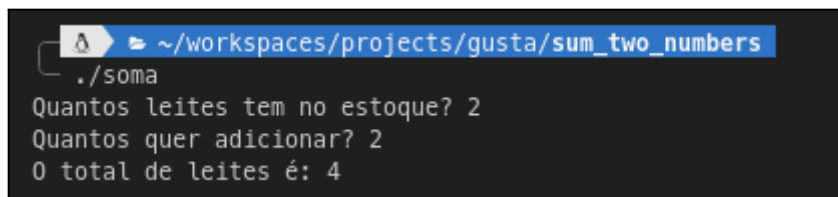
Linha 13: Ocorre a soma entre as variáveis “num1” e “num2” que resulta no valor total de leites que o estoque terá, por fim, esse valor total será armazenado na variável “Soma”

Linha 15: com o uso de “count” e “<<” é exibido a mensagem mostrando o número total de leites no estoque. Junto a isso o uso do “endl” que garante que a mensagem será exibida sem erros.

Linha 17: Indica que o programa terminou corretamente.

A imagem a seguir ilustra o que é exibido no terminal após a execução do código:

Figura 6 — Resultado do código que soma dois números em C++

A screenshot of a terminal window with a dark background. The title bar at the top shows a file icon, a folder icon, and the path `~/workspaces/projects/gusta/sum_two_numbers`. The terminal content shows the command `./soma` being executed, followed by two prompts: `Quantos leites tem no estoque? 2` and `Quantos quer adicionar? 2`. The final output line is `0 total de leites é: 4`.

```
~/workspaces/projects/gusta/sum_two_numbers  
./soma  
Quantos leites tem no estoque? 2  
Quantos quer adicionar? 2  
0 total de leites é: 4
```

Fonte: Do próprio autor, 2025

2.12 HTML

A interação com o usuário em nosso sistema RFID, é constituída via ferramenta visual que apresenta informações de forma concisa e organizada, a qual prosseguiremos chamando de dashboard, sendo sua estrutura construída em HTML.

Segundo Silveira e Prates (2001) O Hyper Text Markup Language (HTML) é a tecnologia para criação da estrutura de WebSites. A navegação de forma dinâmica é possível por conta do HyperText que através de links conecta as páginas entre si como explica Duckett (2016). Além disso, também destaca que Websites são compostos por diversos conteúdos como textos, links, imagens ou vídeos.

A partir disso, Cardoso (1999) afirma que para acessar aplicações web pelo navegador basta digitar o endereço (URL) na barra de pesquisa. Na sequência, Duckett (2016) ressalta que ao acessar um site, o navegador tem a função de traduzir o código HTML e projetar de forma visual para o usuário. Onde este mesmo fluxo ocorre ao acessar a dashboard do nosso sistema RFID. Todo esse processo só é possível graças aos elementos HTML, cuja construção se baseia em caracteres

colocados entre colchetes angulares (< >), conhecidos como tags, indicando o início e o fim do elemento. Eles são fundamentais na criação de telas, onde cada elemento desempenha uma função específica na estrutura e organização dos conteúdos na página.

Segundo Silveira e Prates (2001), as principais tags usadas para construção da estrutura de um documento HTML básico são:

- “<!doctype html>” indica o tipo de documento que está sendo escrito e a versão do HTML. No HTML5, essa declaração é obrigatória e deve ser a primeira linha do código.
- “<html>” Representa o elemento raiz de uma página HTML.
- “<head>” Armazena informações que não são diretamente exibidas nas páginas.
- “<body>” Representa o corpo da página, onde ficam todos os elementos visíveis ao usuário, como textos, imagens, vídeos, botões e demais conteúdos interativos.

Um exemplo de utilização dessas tags é apresentado na figura abaixo demonstrando sua estrutura básica.

Figura 7 — Estrutura básica de uma página em HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <!-- Todo Conteúdo desta Página -->
</body>
</html>
```

Fonte: Do próprio autor, 2025

Em nosso sistema, esses elementos foram utilizados para estruturar uma tela de cadastro de produtos para estoque.

Figura 8 — Exemplo de uma tela HTML para cadastro de produtos.

```
<!DOCTYPE html>
<html lang="pt-BR">

<head>
  <meta charset="UTF-8">
  <title>Cadastro de Produto</title>
</head>

<body>
  <h2>Cadastro de Produto</h2>
  <form action="#" method="post">
    <fieldset>
      <legend>Informações do Produto</legend>

      <label for="id">ID do Produto:</label><br>
      <input type="text" id="id" name="id"><br><br>

      <label for="nome">Nome do Produto:</label><br>
      <input type="text" id="nome" name="nome"><br><br>

      <label for="quantidade">Quantidade:</label><br>
      <input type="number" id="quantidade" name="quantidade"><br><br>

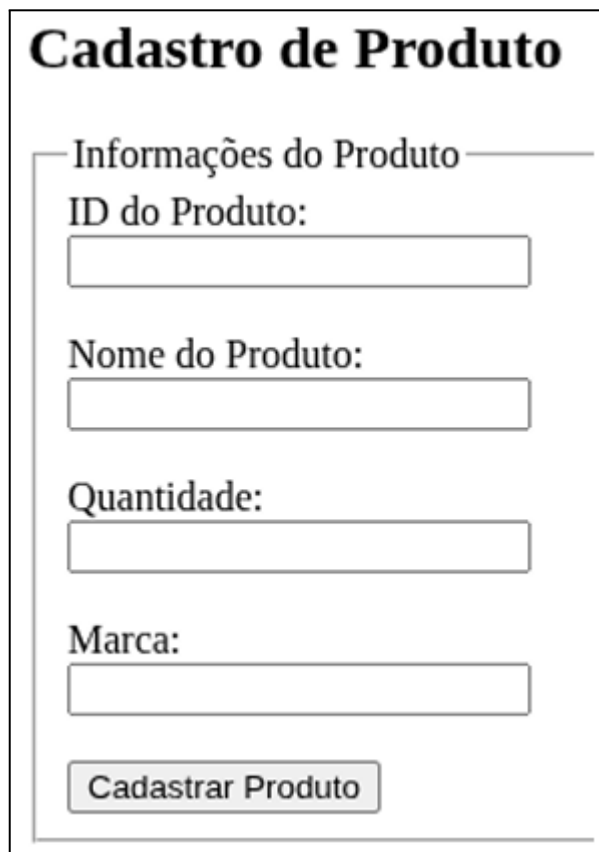
      <label for="marca">Marca:</label><br>
      <input type="text" id="marca" name="marca"><br><br>

      <input type="submit" value="Cadastrar Produto">
    </fieldset>
  </form>
</body>
</html>
```

Fonte: Do próprio autor, 2025

O código mostrado na figura acima, resulta na renderização visual apresentada na figura abaixo:

Figura 9 — Resultado do código HTML de uma página de cadastro de produtos.



Cadastro de Produto

Informações do Produto

ID do Produto:

Nome do Produto:

Quantidade:

Marca:

Cadastrar Produto

Fonte: Do próprio autor, 2025

Em sequência, iremos destacar em detalhe cada tag utilizada para a construção do formulário de cadastro de produtos de nosso sistema RFID. Conforme explica Duckett (2016), o funcionamento e a função de cada elemento HTML utilizado.

- “<form>” elemento que abrange todo formulário, utilizado para coleta das informações.
- “<h2>” elemento de títulos. As tags h1 a h6. Definem diversos níveis de títulos. Onde “<h1>” indica o título com maior importância e “<h6>” o com menor destaque.
- “<fieldset>” agrupa os campos do formulário, organizando o conteúdo com uma borda ao redor deles e simplificando a visualização das informações.
- “<legend>” concede um título para o grupo de campos do <fieldset>.

- “<label>” define um rótulo descritivo para um campo do formulário, ajudando na organização e acessibilidade.
- “<input>” criam campos interativos no formulário, como textos, números e botões de envio. Seu comportamento é definido no atributo type, que determina o tipo de dado aceito.
- “
” define uma quebra de linha.

Com a estrutura da página construída, é possível visualizar os elementos organizados, porém somente o HTML não garante a aparência da tela do nosso sistema. Por isso, se torna necessário uma folha de estilo (CSS), conforme abordaremos no próximo tópico.

2.13 CSS

Após a construção da estrutura da dashboard utilizando apenas HTML, a próxima etapa é definir suas propriedades de estilo por meio do CSS. Como destaca Knight (2018) a Cascading Style Sheets (CSS) é uma tecnologia usada para estilizar uma página web. Duckett (2016) explica que é possível criar regras que informam como determinado elemento se comporta em uma página web, controlando atributos como cores, tamanhos, fontes e cor de fundo da página.

O CSS é um dos principais fatores para uma experiência satisfatória ao usar uma página web, com o intuito de que seja consumida da melhor forma possível conforme apresentado por Eis (2012).

Como abordado por Duckett (2016), de início, para estilizar uma tela HTML, é necessário realizar a conexão entre os arquivos, HTML com o arquivo CSS. O intermediário desta ligação é a tag de ligação, quando inserida no código HTML referenciando a folha de estilo, o HTML e o CSS conseguem trabalhar em conjunto. O trecho de código contendo a conexão desses dois documentos:

Figura 10 — Exemplo de conexão entre HTML e CSS.

```
<head>
  <title>Document</title>
  <link rel="stylesheet" href="styles.css">
</head>
```

Fonte: Do próprio autor, 2025

Esses são os papéis de cada atributo para que essa conexão entre o arquivo de estruturação (HTML) e o de estilização (CSS) aconteça:

- “<link>” serve para comunicar ao navegador onde procurar o arquivo CSS.
- “href” mostra em específico onde está localizada a folha de estilo.
- “type” declara o tipo do documento que o link está referenciando.
- “rel” informa a relação entre a página HTML e o arquivo que referencia, quando o link aponta para um documento CSS, o dado declarado deve ser stylesheet.

Além disso, ressalta que ao escrever a CSS, uma regra CSS é a junção de duas partes, uma se chama seletor e a outra declaração. Para um elemento HTML ser estilizado, é utilizado os seletores, que falam qual tag estará recebendo as regras de estilo.

O bloco de código a seguir faz parte da estilização da dashboard RFID e apresenta como escrever regras na folha de estilo.

Figura 11 — Exemplo de escrita na CSS.

```
body {  
    font-family: sans-serif;  
    background: #f2f2f2;  
    padding: 30px;  
}  
  
form {  
    background: #fff;  
    padding: 20px;  
    max-width: 350px;  
    margin: auto;  
    border-radius: 6px;  
}
```

Fonte: Do próprio autor, 2025

A seguir, iremos enfatizar a função de cada uma das tags apresentadas com o intuito de entendermos seus papéis individualmente na estilização do formulário de cadastro:

- “body” do HTML, indica tudo que está no corpo da página.
- “font-family” usada para declarar a fonte que será utilizada no texto de qualquer elemento HTML em que a regra CSS está agindo.
- “padding” define uma distância entre o conteúdo e a borda. Foi utilizado para definir o espaçamento entre os campos na tela de cadastro.
- “background” determina a cor que será utilizada em determinado elemento.
- “width” relata a largura de um elemento.
- “margin” controla o espaço entre blocos e elementos.
- “border-radius” permite criar bordas arredondadas, seu valor declara o tamanho do raio, resultando em botões mais amigáveis.

- “color” possibilita definir a cor do texto dentro de um elemento, foi usado para definir a cor do nosso botão.

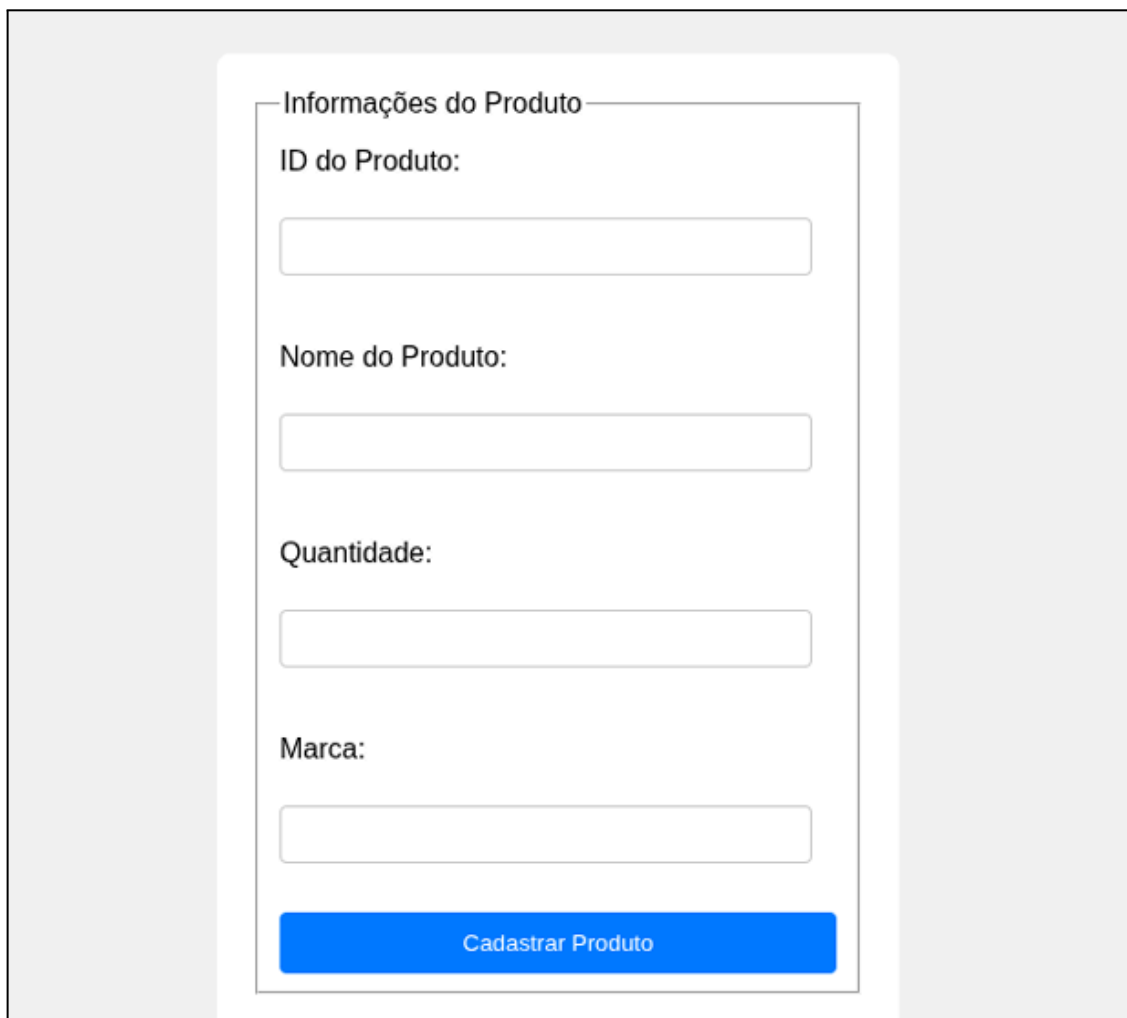
Prosseguindo, abaixo se encontra a solução completa para estilização do formulário de cadastro de produtos da nossa dashboard.

Figura 12 — CSS da tela de cadastro de produtos.

```
body {  
  font-family: sans-serif;  
  background: #f2f2f2;  
  padding: 30px;  
}  
  
form {  
  background: #ffffff;  
  padding: 20px;  
  margin: 0 auto;  
  width: 100%;  
  max-width: 350px;  
  border-radius: 8px;  
}  
  
label {  
  display: block;  
  margin: 10px 0 5px 0;  
}  
  
input[type="text"],  
input[type="number"] {  
  width: 90%;  
  padding: 8px;  
  margin-bottom: 10px;  
  border-radius: 4px;  
  border: 1px solid #ccc;  
}  
  
input[type="submit"] {  
  width: 100%;  
  padding: 10px;  
  background: #007bff;  
  color: white;  
  border-radius: 4px;  
  border: none;  
}
```

Somado a isso, com a união de todas as tecnologias apresentadas, o resultado da página de cadastro de produtos:

Figura 13 — Resultado da tela de cadastro HTML e CSS em conjunto.



Informações do Produto

ID do Produto:

Nome do Produto:

Quantidade:

Marca:

Cadastrar Produto

Fonte: Do próprio autor, 2025

2.14 JAVASCRIPT

Conforme explica Silva (2010), com a criação de HTML, responsável pela estrutura dos websites, e o CSS, com o papel de estilizar a página para melhor experiência do usuário final, houve a necessidade de uma tecnologia com a função de adicionar interações para tais páginas.

Na visão de Lepsen (2018), o JavaScript criado pela Netscape com auxílio da Sun Microsystems, é a ferramenta encarregada pela criação das interações da aplicação web com o consumidor final. É a tecnologia capaz de designar funcionalidades aos elementos web, por meio dos campos de formulários, configurações gerais de uma página e interações para salvar nossos dados nos navegadores, o JavaScript se comunica com o visitante deixando a página com ar mais dinâmico.

A dinamicidade do JavaScript proporciona ao usuário uma experiência mais agradável durante sua navegação pelo website. Atrelado ao painel (dashboard), essa característica pode gerar uma sensação ainda mais marcante.

Segundo Groner (2019), com essa ferramenta, é possível agir tanto na parte visual (Front-end) aplicando animações e manipulando elementos, quanto na parte lógica (Back-end) criando funcionalidades, que podem ser chamados de scripts. JavaScript é utilizado principalmente para rodar scripts no lado do cliente, os responsáveis pela interpretação são os navegadores, que entendem e executam as funcionalidades.

2.14.1. NodeJS

De acordo com Pereira (2014), Node.js, criado em 2009 por Ryan Dahl e com ajuda inicial de 14 colaboradores, se destaca especialmente em aplicações que possuem muitas entradas e saídas de dados (Input/Output - I/O), onde, nesse cenário, ele consegue usufruir o máximo do poder de processamento dos servidores de forma produtiva.

Em relação ao projeto atual, é necessário devido ao alto fluxo de dados recebidos pelos dispositivos IOT.

Segundo Moraes (2023), Node.js utiliza o motor V8 da Google e permite criar rotas web usando diversos protocolos de redes (regras de como dispositivos devem

se comunicar entre si) como , como HTTPS, DNS, FTP etc., permitindo a construção de sites e apps para sistemas operacionais iOS e Android.

Para Powers (2017), Node.js é uma ferramenta versátil e completa, podendo ser usada em diversos cenários e situações, graças ao seu ambiente com diversas funcionalidades e um bom alcance.

O exemplo a seguir é de um código que junta o nome e o sobrenome do usuário, exibindo ao final o nome completo. Muito usado em sistemas em que é necessário exibir o nome e sobrenome do usuário juntos. Abaixo segue uma descrição do funcionamento do código:

Figura 14 — Função juntar nome e sobrenome NodeJS



```
1 function JuntarNomes(nome, sobrenome) {
2     return console.log(nome + " " + sobrenome);
3 }
4
5 JuntarNomes("Ananda", "Holanda");
6
```

Fonte: Do próprio autor, 2025

Abaixo segue uma descrição do funcionamento do código:

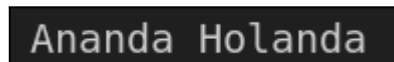
Linha 1: Declara uma função chamada “JuntarNomes” e, dentro dos parênteses, são solicitados dois parâmetros separados por vírgula (nome e sobrenome do usuário).

Linha 2: Retorna o nome e o sobrenome juntos através do sinal de “+”, com um espaço vazio entre eles, declarado entre aspas, que representa o espaçamento entre nome e sobrenome.

Linha 5: Chama a função passando como parâmetros o nome e o sobrenome separados por vírgula.

A imagem a seguir ilustra o que é exibido no terminal após a execução do código:

Figura 15 — Função juntar nome e sobrenome NodeJS

A dark rectangular box with the text "Ananda Holanda" in a light gray, monospaced font, representing the output of a NodeJS function in a terminal.

Fonte: Do próprio autor, 2025

2.15 Firebase

Conforme afirma Andrade (2018 apud SILVA; SANTOS JUNIOR, 2023), sendo parte da infraestrutura Google, o Firebase é uma plataforma que busca ajudar desenvolvedores a construírem aplicações de forma mais rápida, performática e fácil.

Segundo a documentação oficial do Firebase (2025), Firebase é uma plataforma que permite criar projetos que podem usufruir de múltiplas funcionalidades, como hospedagem, banco de dados, autenticação, entre outros serviços.

Consoante Machado (2021), o Firebase entrega uma infraestrutura rica em recursos, possibilitando que o responsável pelo projeto foque em outras atividades, enquanto a plataforma adapta a infraestrutura de forma automática, com qualidade e escalabilidade.

A escolha do Firebase para armazenamento das informações do sistema deve-se ao seu alto nível de desempenho, citado por Andrade (2018 apud SILVA; SANTOS JUNIOR, 2023), tendo como benefício uma aplicação capaz de lidar com alto tráfego de dados de forma rápida.

REFERÊNCIAS

CAMPOS, Matheus. **Análise da viabilidade de implementação de etiquetas RFID na construção civil**. 2021. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) – Universidade Tecnológica Federal do Paraná, Departamento Acadêmico de Construção Civil, Campo Mourão, 2021.

CARDOSO, Carlos. **HTML4**. Rio de Janeiro: Axcel Books, 1999.

CASTRO, Herik. **Um web framework para C++: projeto, implementação e avaliação**. 2019. Dissertação (Mestrado em Metodologia e Técnicas Computacionais) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2019.

COSTA, Alexsander. **RFControl: sistema de gerência de estoque utilizando RFID**. 2018. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Universidade Federal de Ouro Preto, Instituto de Ciências Exatas e Aplicadas, Departamento de Computação e Sistemas, João Monlevade, 2018.

DAMACENO, Donizeti; FUNG, Charles; SARTORI, Rodrigo. **Central de automação residencial de baixo custo com ESP32**. Curitiba: UNINTER, v. 4, n. 8, 2024.

DUCKETT, Jon. **HTML & CSS: projete e construa websites**. Rio de Janeiro: Alta Books, 2016.

EIS, Diego et al. **HTML5 e CSS3: com farinha e pimenta**. São Paulo, Brasil: Tableless, 2012.

FOWLER, Martin; SCOTT, Kendall. **UML Essencial Um breve guia para a linguagem-padrão de modelagem de objetos**. 2. ed. Porto Alegre: Bookman, 2000.

GONÇALVES, Luiz. **Sistema de Controle de Acesso Físico por Dispositivos com Identificação por RFID e Autenticação por Biometria de Impressão Digital**.

2025. Monografia (Graduação em Engenharia de Controle e Automação) – Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2025.

GOOGLE. **Entender os projetos do Firebase**. Disponível em:

<https://firebase.google.com/docs/projects/learn-more?hl=pt-br>. Acesso em: 20 maio 2025.

GRONER, Loiane. **Estruturas de dados e algoritmos com JavaScript**. 2. ed. São Paulo: Novatec, 2019.

GUEDES, Gilleanes. **UML2 UMA ABORDAGEM PRÁTICA**. 3. ed. São Paulo: Novatec, 2018.

IBM. **Rational Software Architect**. [S. l.]: IBM, 2025. Disponível em:

<https://www.ibm.com/docs/pt-br/rational-soft-arch/10.0?topic=diagrams-activity>.

Acesso em: 11 ago. 2024.

IBM. **Rational Software Modeler**. [S. l.]: IBM, 2021. Disponível em:

<https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-cases>. Acesso em: 10

ago. 2024.

INFORCHANNEL. **Lojas Renner conclui projeto de implementação de etiquetas RFID**. [S.l.]: InforChannel, 2022. Disponível em:

<https://inforchannel.com.br/2022/06/30/lojas-renner-conclui-projeto-de-implementacao-de-etiquetas-rfid/>. Acesso em: 11 ago. 2025.

KPMG. **Pesquisa Abrappe de Perdas no Varejo Brasileiro 2024**. [s.l.], 2024.

Disponível em:

https://kpmg.com/br/pt/home/insights/2024/11/pesquisa-abrappe-2024.html?utm_source=chatgpt.com. Acesso em: 27 mai. 2025.

LEÃO, Leonardo. **Minimercados autônomos ganham força no Brasil**. Belo Horizonte: Diário do Comércio, 2024. Disponível em: <https://diariodocomercio.com.br/negocios/minimercados-autonomos-ganham-forca-no-brasil/>. Acesso em: 24 mai. 2025.

LEPSEN, Edécio. **Lógica de Programação e Algoritmos com JavaScript: UMA INTRODUÇÃO À PROGRAMAÇÃO DE COMPUTADORES COM EXEMPLOS E EXERCÍCIOS PARA INICIANTES**. 1. ed. São Paulo: Novatec, 2018.

MACHADO, Kheronn. **Angular 11 e Firebase**: Construindo uma aplicação integrada com a plataforma do Google. Brasil: Casa do Código, 2021.

MAGRANI, Eduardo. **A Internet Das Coisas**. Rio de Janeiro: FGV Editora, 2018.

MICROSOFT. **Bem-vindo de volta ao C++ – C++ moderno**. Disponível em: <https://learn.microsoft.com/pt-br/cpp/cpp/welcome-back-to-cpp-modern-cpp?view=msvc-170>. Acesso em 11 ago. 2025.

Microsoft. **Criar um diagrama de atividade UML**. [S.I.]: Microsoft, 2025. Disponível em: <https://support.microsoft.com/pt-br/topic/criar-um-diagrama-de-atividade-uml-19745dae-2872-4455-a906-13b736f01685#officeversion=windows>. Acesso em: 11 ago. 2025.

Microsoft. **Criar um diagrama de sequência UML**. [S.I.]: Microsoft, 2025. Disponível em: <https://support.microsoft.com/pt-br/topic/criar-um-diagrama-de-sequ%C3%Aancia-uml-c61c371b-b150-4958-b128-902000133b26>. Acesso em: 12 ago. 2025.

MORAES, William. **Construindo aplicações com NodeJS**. 4. ed. São Paulo: Novatec Editora, 2023.

OLIVEIRA, Ana. **Estudo paramétrico e análise de impedância de uma etiqueta RFID UHF passiva**. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica). João Pessoa: Instituto Federal da Paraíba, 2023.

PEREIRA, Caio. **Aplicações web real-time com Node.js**. [S.l.]: Casa do Código, 2013.

PET. **Introdução à Programação Embarcada**. Minas Gerais: Itajubá, 2021.

POWERS, Shelley. **Aprendendo Node**. 1. ed. Brasil: Novatec Editora, 2017.

PREDIGER, Daniel; FREITAS, Edison; SILVEIRA, Sidnei. **Modelo de aplicabilidade de sistema RFID para rastreabilidade na indústria alimentícia**. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação). Rio Grande do Sul: Universidade Federal de Santa Maria, 2014. Disponível em: <http://repositorio.ufsm.br/handle/1/12810>. Acesso em: 25 maio 2025.

ROVAROTO, Isabela. **Como esta empresa de calçados deixou de perder vendas por falta de estoque em suas 340 lojas**. [S.l.]: Exame, 2024. Disponível em: <https://exame.com/negocios/como-esta-empresa-de-calcados-deixou-de-perder-ven-da-por-falta-de-estoque-em-suas-340-lojas/>. Acesso em: 24 mai. 2025.

SILVA, Antonio; SANTOS JUNIOR, João. **Vacmonitor: uma aplicação para o monitoramento de vacinas utilizando Flutter e Firebase**. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação). Paraíba: Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, 2023.

SILVA, Maurício Samy. **Javascript: Guia do Programador**. 1. ed. São Paulo: Novatec, 2010.

SILVEIRA, Marcelo; PRATES, Rubens. **HTML 4: guia de consulta rápida**. São Paulo: Novatec, 2001.

TORTORELLI, Henrique. **Estratégias de Negócios Internacionais: O caso da empresa Amazon.com**. Covilhã: Universidade Da Beira Interior, 2018.

WENCESLAU, Fernando. **Como a automação e a tecnologia preditiva estão mudando a gestão de estoque**. Santa Catarina: Economia SC, 2024. Disponível em: <https://economiasc.com/2024/07/25/como-a-automacao-e-a-tecnologia-preditiva-esta-o-mudando-a-gestao-de-estoque/>. Acesso em: 25 mai. 2025.

WIENER, Richard; PINSON, Lewis J. **Programação Orientada para Objeto e C++**. São Paulo: Makron Books, 1991.

GLOSSÁRIO

ArduinolDE (Integrated Development Environment): Ambiente de Desenvolvimento Integrado utilizado para programar microcontroladores como o ESP32, permitindo compilar e enviar código para o dispositivo.

ASCII (American Standard Code for Information Interchange): Código de caracteres amplamente utilizado para representar textos em dispositivos eletrônicos, podendo aparecer em etiquetas RFID como forma de formatação de dados.

Aulas práticas: Atividades presenciais ou remotas com foco na aplicação prática de conhecimentos teóricos, especialmente comuns em cursos técnicos.

Authentication: Serviço de autenticação de usuários oferecido por plataformas como o Firebase, utilizado para controle de acesso em aplicações web.

Automatização: Aplicação de tecnologia para aumentar a eficiência e reduzir a necessidade de ação humana em processos. Essencial no controle de estoque.

Back-end: Parte lógica de uma aplicação web, responsável pelo funcionamento interno, comunicação com banco de dados e execução de scripts do lado do servidor.

C++: Linguagem de programação de alto nível, de propósito geral, e é uma extensão da linguagem C. É utilizada para programar o ESP32 e possui características orientadas a objetos e acesso facilitado ao hardware.

Controle de estoque: Processo de administração de materiais e produtos armazenados. Pode sofrer perdas por erros humanos e falta de rastreabilidade; o projeto busca automatizá-lo usando RFID.

CSS (Cascading Style Sheets) é a ferramenta que se usa para controlar o visual de uma página, definindo elementos como cores, fontes e tamanhos.

Database: Serviço de banco de dados que armazena e organiza dados utilizados por aplicações web. No projeto, o Firebase oferece esse recurso.

Ensino híbrido: Modelo educacional que combina atividades presenciais e online, muito utilizado no contexto da Educação Profissional.

Escalabilidade: Capacidade de um sistema ou serviço (como o Firebase) de se adaptar ao aumento da demanda sem perda de desempenho.

ESP32: Microcontrolador da Espressif Systems com recursos de Wi-Fi e Bluetooth, compatível com o Arduino IDE. Utilizado para processar dados e controlar dispositivos físicos.

Etiqueta RFID: Dispositivo composto por um chip e uma antena, capaz de armazenar e transmitir dados quando ativado por um leitor RFID. Pode ser passiva (sem fonte de energia própria).

Firebase: Plataforma da Google para desenvolvimento de aplicações web e móveis. Oferece recursos como banco de dados em tempo real, hospedagem, autenticação e escalabilidade.

Front-end: É a tela com qual o usuário interage, a parte visual. Ela é criada com a junção das tecnologias como HTML, CSS e Javascript.

Hosting: Serviço de hospedagem que armazena e disponibiliza um site na internet. O Firebase oferece hosting integrado ao projeto.

HTML (HyperText Markup Language): Linguagem de marcação usada para estruturar páginas web. Utiliza elementos entre colchetes angulares para definir conteúdo como textos, links e imagens.

IDE (Integrated Development Environment): Ambiente de Desenvolvimento Integrado usado para escrever, compilar e enviar códigos para dispositivos como o ESP32.

IoT (Internet of Things): Conceito que envolve a conexão de objetos físicos à internet, permitindo a troca de dados e a automação de processos.

JavaScript (JS): Linguagem de programação utilizada para adicionar interatividade a páginas web. Pode ser executada tanto no front-end quanto no back-end.

Leitor RFID: Equipamento que emite sinais de rádio e lê dados das etiquetas RFID próximas. Responsável por iniciar a comunicação com as etiquetas.

Mercados autônomos: Estabelecimentos sem atendentes, com controle automatizado de entrada, saída e pagamento, geralmente com o uso de tecnologias como RFID.

Metodologias ativas: Estratégias de ensino centradas na participação ativa dos alunos, com foco em experimentação, resolução de problemas e protagonismo no aprendizado.

Node.js: Plataforma que permite a execução de JavaScript no lado do servidor. Utiliza o motor V8 da Google e é usada para desenvolver o back-end de aplicações web.

Protocolos de redes: Conjunto de regras que definem como dispositivos se comunicam em uma rede, como HTTP, HTTPS, FTP e DNS.

Rastreabilidade de estoque: Capacidade de identificar e acompanhar a movimentação de itens no estoque, o que reduz perdas e aumenta a eficiência da gestão.

RFID (Radio-Frequency Identification): Tecnologia que permite identificar objetos por meio de radiofrequência, facilitando rastreamento e controle de estoque de forma automática.

Scripts: Conjunto de instruções escritas em linguagens como JavaScript, executadas para automatizar tarefas ou criar funcionalidades em sistemas web.

UHF (Ultra High Frequency): Faixa de frequência de rádio entre 300 MHz e 3 GHz utilizada por algumas etiquetas RFID para transmissão de dados a médias e longas distâncias.

URL (Uniform Resource Locator): Endereço usado para acessar recursos na internet, como páginas web, imagens ou arquivos.