

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ESCOLA TÉCNICA ESTADUAL 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.

“Quando vires um homem bom,
tenta imitá-lo; quando vires um
homem mau, examina-te a ti
mesmo.”

– 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++.....	26
Figura 7 — Estrutura básica de uma página em HTML.....	27
Figura 8 — Exemplo de uma tela HTML para cadastro de produtos.....	28
Figura 9 — Resultado do código HTML de uma página de cadastro de produtos.....	29
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.....	33
Figura 13 — Resultado da tela de cadastro HTML e CSS em conjunto.....	34
Figura 14 — Função juntar nome e sobrenome Node.js.....	36
Figura 15 — Resultado função Node.js.....	37

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)

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.....	11
2 REFERENCIAL TEÓRICO.....	13
2.1 Automatização 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 Diagrama de classe.....	20
2.7 Internet Das Coisas (IoT).....	21
2.8 Identificação por rádio frequência.....	22
2.9 ESP32.....	23
2.10 Arduino IDE.....	24
2.11 C++.....	24
2.12 HTML.....	26
2.13 CSS.....	30
2.14 JavaScript.....	35
2.15 React.....	35
2.16 Node.js.....	35
2.17 MySQL.....	37
3 DESENVOLVIMENTO.....	38
3.1 Especificação do Sistema.....	38
REFERÊNCIAS.....	39
GLOSSÁRIO.....	43
ANEXOS.....	47
ANEXO A - Detalhamento Hello World C++.....	47
ANEXO B - Principais etiquetas HTML.....	47

1 INTRODUÇÃO

O Estok consiste em uma ferramenta que utiliza a tecnologia de identificação por radiofrequência (RFID) para aumentar a rastreabilidade de produtos em prateleiras de mercados autônomos, diminuindo perdas por ruptura de estoque e vencimento. O objetivo geral é desenvolver um dispositivo para detectar a entrada e saída de produtos da prateleira, otimizando a reposição e evitando a perda de produtos próximo por vencimento ou a falta de produtos, o que é chamado de ruptura de estoque, resultado de um produto indisponível na prateleira, porém existente no estoque.

Os objetivos específicos para a concretização deste projeto consistem em produzir um dispositivo capaz de identificar a quantidade de produtos em uma prateleira, essa identificação possibilitará a notificação de produtos acabando ou perto da data de vencimento, notificações essas que deverão ser dispostas em uma página web para a visualização dos gerentes e funcionários.

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, se essa automatização não for feita para garantir um real autonomia, problemas como dependência de verificação constante da mercadoria disponível nas prateleiras e da quantidade em estoque por parte do gerente do mercado podem surgir, afetando não somente a comodidade do gerente, como também a dos clientes que poderão sofrer com ruptura de 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). Pesquisas serão feitas utilizando o rigor científico, pois, como afirma Lakatos e Marconi (2003), para estabelecer a realidade do nosso cenário e se de fato nosso projeto é eficiente, devemos seguir o tratamento científico para conhecer a realidade. Neste estudo, a principal hipótese se dá na produção e implementação de um aparelho que aumente a rastreabilidade dos produtos, buscando mitigar os problemas já mencionados, causados pela falta de rastreabilidade de produtos em mercados autônomos.

O presente estudo visa resolver a problemática dos clientes que sofrem com a ruptura de estoque e, portanto, responder como a tecnologia RFID pode aumentar a rastreabilidade e reduzir rupturas de estoque em mercados autônomos de São Paulo.

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).

Diante do cenário apresentado, o uso de tecnologias na área logística para o aumento da eficiência se faz crucial para um desenvolvimento contínuo do modelo de mercados autônomos e de outras evoluções no setor varejista.

2 REFERENCIAL TEÓRICO

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

2.1 Automatização 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, R\$34,9 bilhões de prejuízo foram registrados em 2024 (KPMG, 2024). Segundo Wenceslau (2024), esses problemas podem ser atribuídos à 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 do setor e para diminuição de problemas de imprecisão. A seguir buscaremos explicar uma das possíveis soluções para esse problema. Com foco em mercados autônomos, a solução aumentará a autonomia na reposição de estoque, resultando na diminuição da ruptura de estoque que beneficiará os clientes dos estabelecimentos.

É importante notar também a contribuição da pesquisa nessa área para o incentivo à busca e desenvolvimento de soluções e inovações em mercados autônomos (TORTORELLI, 2018). 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 (LEÃO, 2024), pesquisas referentes a esse setor da indústria se tornam primordiais para a criação de uma base sólida de conhecimentos.

2.2 UML, Linguagem De Modelagem Gráfica

Para iniciar esse projeto, primeiro precisaremos de um bom planejamento para determinarmos quais as funcionalidades que o nosso sistema terá e como ele interagirá com agentes externos. Uma boa forma de visualizar essa interação é por meio de diagramas, mais especificamente diagramas de Linguagem de Modelagem Unificada (UML).

A UML é uma forma de descrever o seu projeto de forma gráfica para facilitar a compreensão para 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 explicar o que utilizaremos, poupando o leitor de explicações de características, componentes e funcionalidades que não contribuíram para o real uso no nosso projeto. Dada a característica de esclarecer diferentes níveis de abstração, explicarei os diagramas em ordem de complexidade.

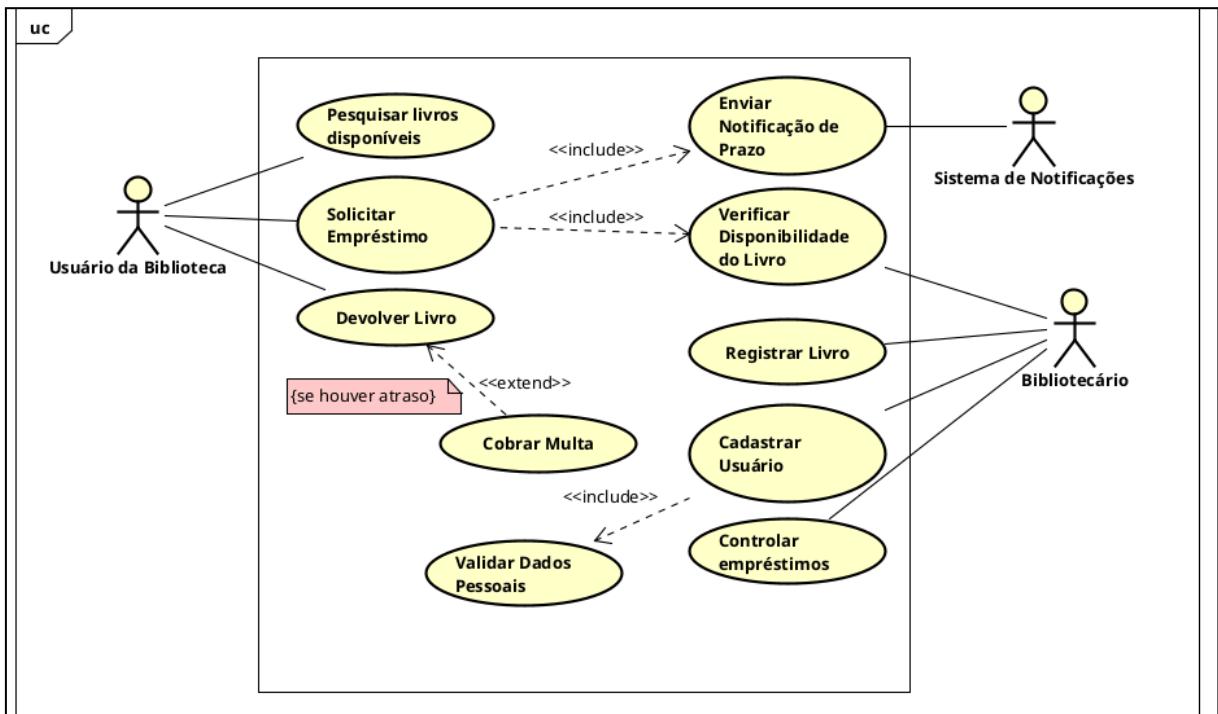
2.3 Diagrama de casos de uso

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).

O diagrama de casos de uso serve para começar a visualizar as interações que existirão, 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 na figura 1.

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.

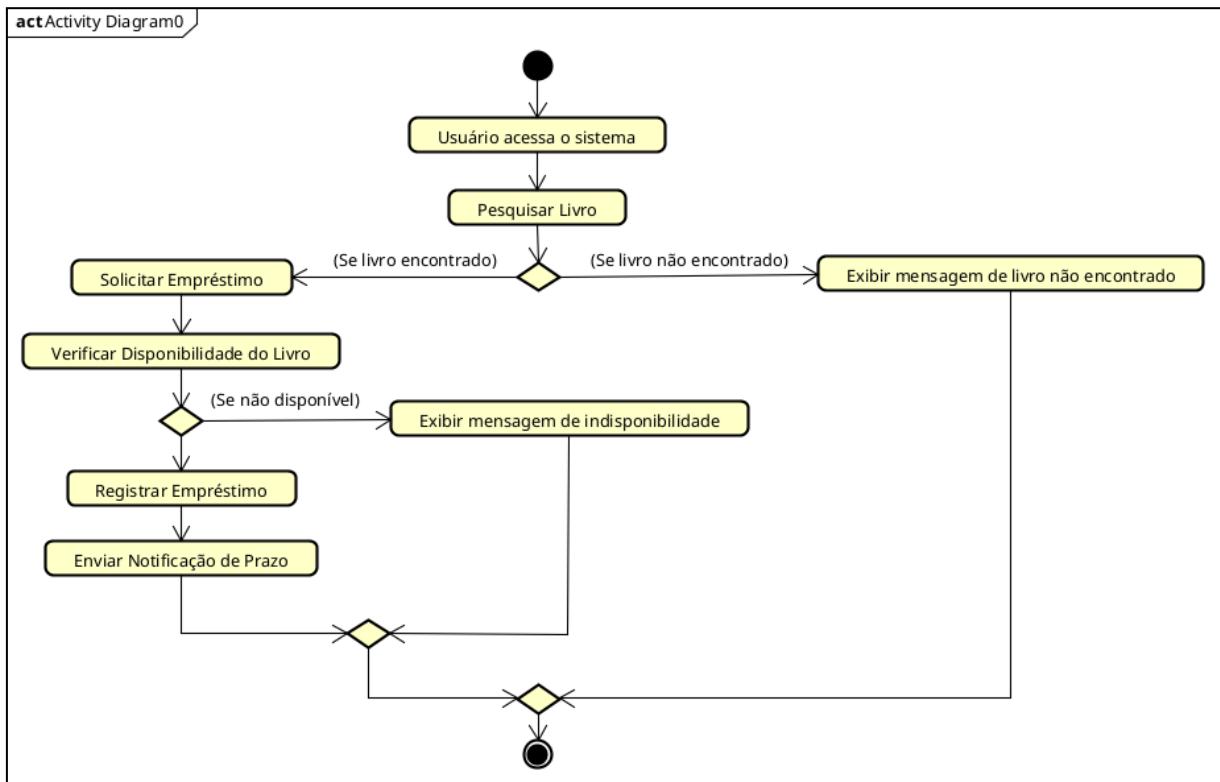
Já um exemplo de extensão se dá na figura quando o usuário da biblioteca aciona o caso de uso “devolver livro”, o caso de uso “cobrar multa” é estendido, mas somente se houver atraso, essa condição é descrita em uma restrição.

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 2 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

O diagrama de atividade tem seu início no nó inicial, representado pelo símbolo 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).

Já o símbolo 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

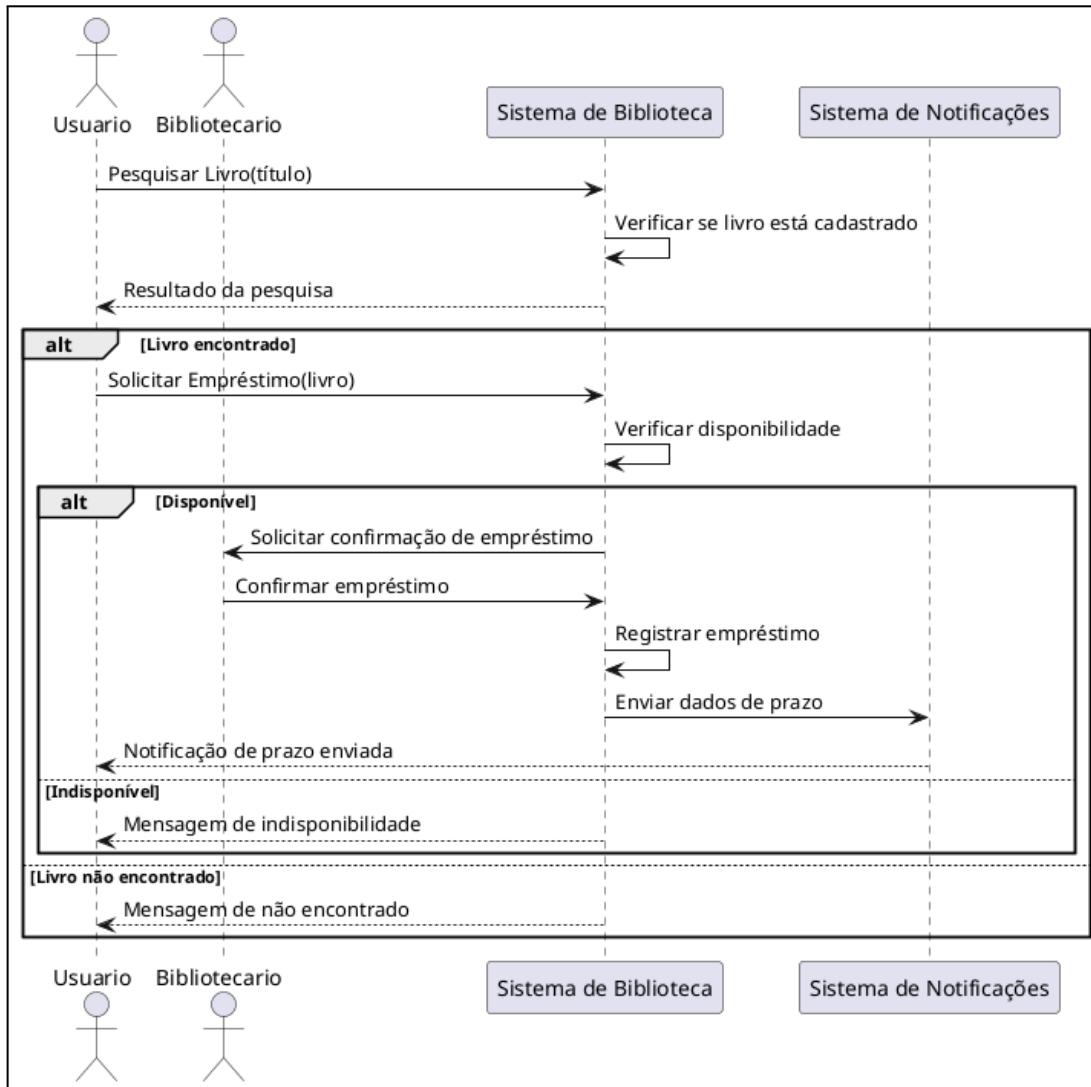
Uma maneira de representar as interações dos objetos que compõem o sistema de forma detalhada e cronológica é utilizando os diagramas de sequência. As interações são representadas por mensagens trocadas entre os objetos (MICROSOFT, 2025).

Todo o período da interação é atrelada a linhas de vida, representadas visualmente por linhas verticais tracejadas saindo do objeto. Dessa 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 3 mostra um diagrama de sequência baseado no exemplo de biblioteca em que víhamos 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 Diagrama de classe

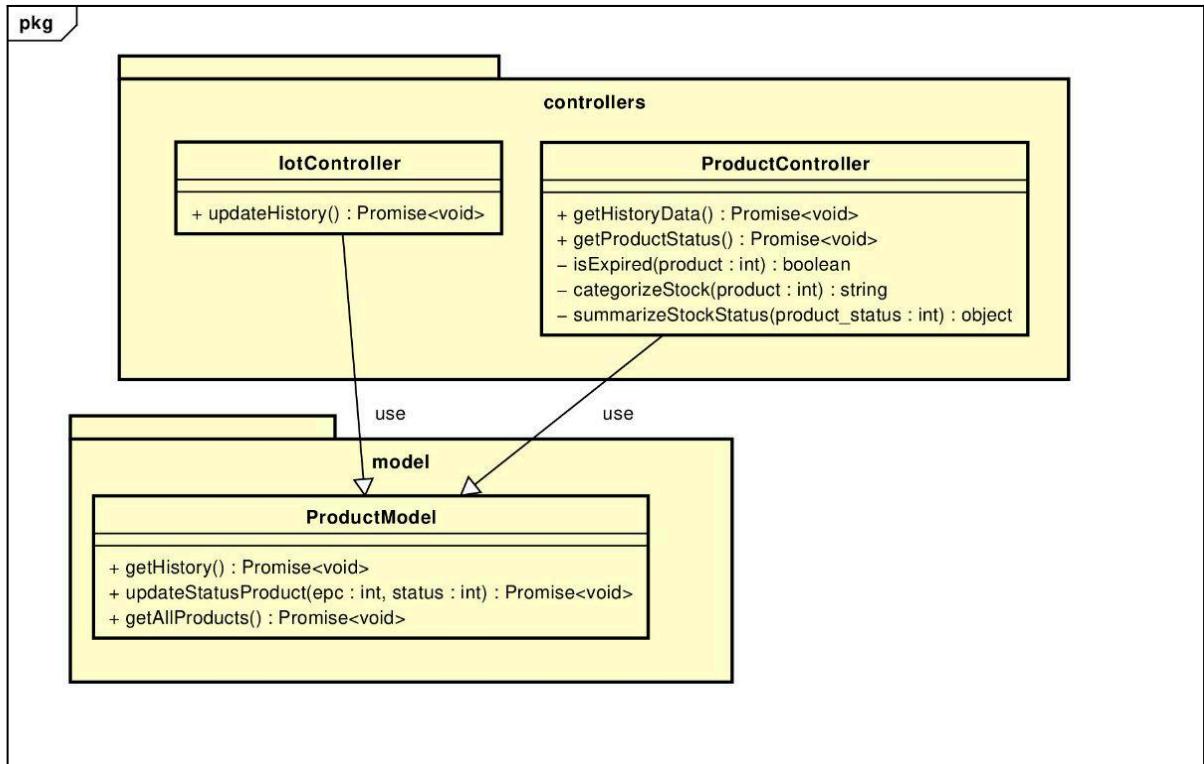
Por último, explicaremos o diagrama de classes. Esse diagrama pode ser visto como o diagrama principal na construção de projetos orientados ao objeto. A escolha de representá-lo por último se deve à forma como escolhemos abordar a construção do back-end, optando por uma forma menos engessada.

Para alcançar esse objetivo, optamos pela escolha do Model-View-Controller (MVC), uma arquitetura de software amigável às mudanças de paradigmas e de requisitos ao longo do projeto. Feita para funcionar em três camadas, o MVC lida com as interações primeiro recebendo-as na camada de controle (Controller), fazendo o processamento necessário na camada de modelo (Model), e por fim exibindo os elementos gráficos necessários para o usuário na camada de visualização (View) (LUCIANO; ALVES, 2011).

Embora a arquitetura do nosso sistema não seja orientada ao objeto, a representação gráfica com um diagrama de classes continua cabível ao projeto devido a sua versatilidade. É comum vermos o diagrama de classe sendo utilizado em mais de uma etapa na criação de um projeto. Como iniciamos o entendimento dos requisitos com o Diagrama de Casos de Uso, detalharei uma abordagem mais orientada ao modelo de domínio com Diagrama de Classe, ou seja, focando na solução do problema incluindo os detalhes de métodos e naveabilidade do nosso projeto.

Os principais componentes de um diagrama de classe são as classes, aqui lidas como as entidades que estamos identificando, os atributos dessa classe, que armazenam dados diversos, a visibilidade, que indica como essas diferentes classes podem interagir entre si, e os métodos, sendo as funções que essas classes podem executar .

Figura 4 - Diagrama de classe Estok



Fonte: Do próprio autor, 2025

Descrição do diagrama (complementar depois da correção)

2.7 Internet Das Coisas (IoT)

O Estok 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 projeto seja classificado como parte da Internet das Coisas (Internet of Things).

Segundo Magrani (2018), pode-se afirmar que dispositivos IoT 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.

Todas essas informações acumuladas com as leituras geram dados que beneficiam não somente o funcionamento dos mercados autônomos, mas geram dados que podem ser utilizados para estudar o padrão de consumo específicos, contribuindo para um conceito conhecido como Big Data, que é justamente esse amontoado de informações que produtos como o Estok produzem (Magrani, 2018).

Produtos como esse fazem parte da Indústria 4.0, sendo essa a grande nova transformação digital, tendo potencial de movimentar grandes alterações tanto em setores logísticos como industriais. Esses produtos são baseados na integração entre dispositivos físicos e a nuvem, onde os dados são armazenados e analisados para mapear padrões de uso (BOMJORNO; HERINGER; MADUREIRA, 2025).

2.8 Identificação por rádio frequência

Os padrões que poderão ser analisados são os de entrada e saída, que no nosso sistema será possibilitado por meio da tecnologia de identificação por radiofrequência (RFID).

Conforme os estudos de Ferreira (2021), a etiqueta RFID possui uma antena que armazena um código único de identificação, tornando cada tag única. Essa etiqueta contém as informações necessárias para identificação do produto, possibilitando toda a dinâmica de reconhecimento exato do produto que entrou na prateleira (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.

Como abordado por Ferreira (2017), as etiquetas RFID possuem um circuito integrado de memória digital, que além de armazenar o código de identificação único, podem armazenar outros caracteres. No nosso sistema 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. Devida essa característica, a posição do leitor deverá ser analisada para obtermos a melhor leitura possível dentro do espaço delimitado da prateleira.

A tecnologia de identificação por radiofrequência foi escolhida pela sua eficácia comprovada no caso de uso das lojas Renner. Nessa implementação, a ruptura de estoque caiu 87%. Já acurácia dos estoques aumentou em 64% (SENSORMATIC, 2025). Números como esse ressaltam a importância do aumento da rastreabilidade do estoque em setores varejistas, comprovando também a eficácia da tecnologia por nós escolhida.

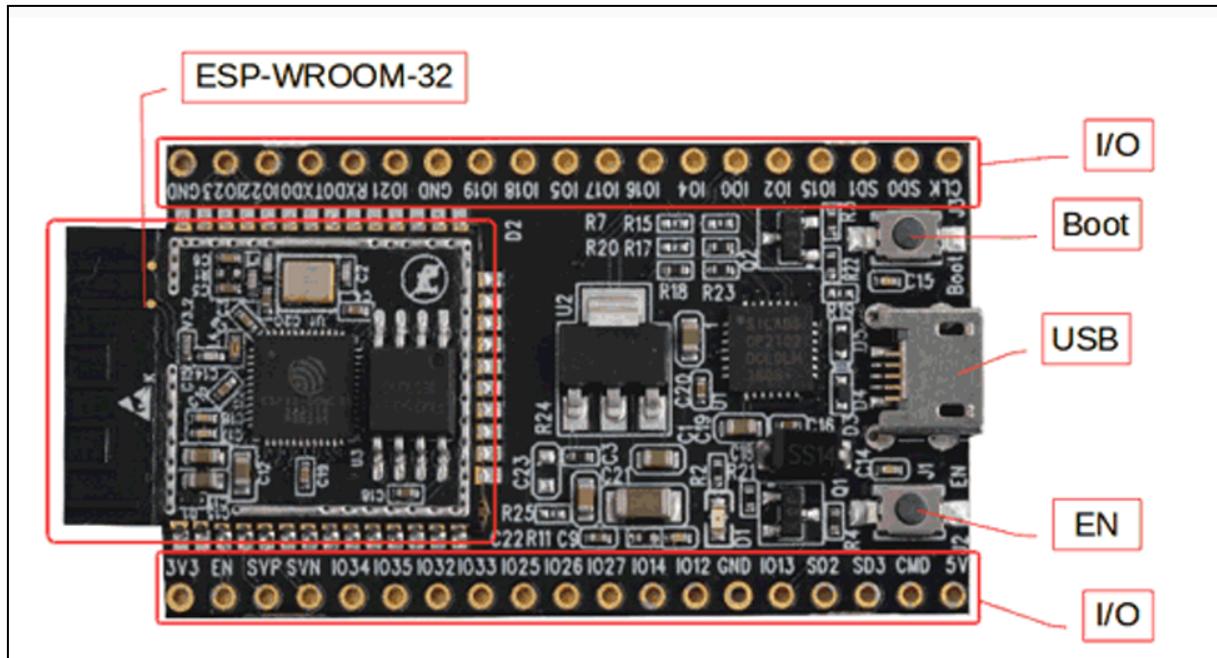
2.9 ESP32

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, pois possibilita maior escalabilidade na infraestrutura de leitores.

A figura 5 mostra o ESP32 visto de cima, onde é possível visualizar seus componentes de conexão onde serão feitas as comunicações com o leitor.

Figura 5 — Placa ESP32 com anotações de componentes



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

2.10 Arduino IDE

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 Carvalho (2023), o ESP32 é compatível com o Arduino IDE, ambiente primariamente utilizado para placas da família Arduino, porém tendo compatibilidade com a placa ESP32.

Essa IDE foi escolhida devido à sua interface clara e de bom entendimento, e a compatibilidade com extensões e bibliotecas que facilitaram o andamento do projeto.

2.11 C++

As instruções passadas para o ESP32 pelo Arduino IDE são feitas na linguagem de programação C++. Conforme a documentação oficial do C++ (2023), a linguagem desde o seu início se destaca fortemente pela sua capacidade de ser eficiente e poder atuar em diversos meios, como aplicativos de jogos, drivers e diversos programas.

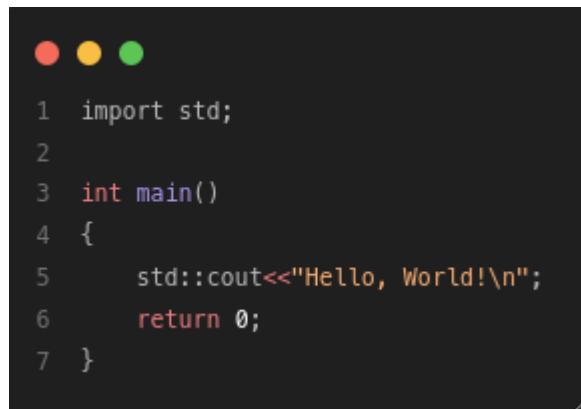
A escolha de C++ com ESP32 se deve à sua alta velocidade de processamento, utilizando pouca memória. Uma forte qualidade da linguagem quando usada em dispositivos IoT.

Como Wiener e Pinson (1991) afirmam, 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 Stroustrup (2025), o código feito em C++ é bem popular pela sua alta compatibilidade entre diferentes máquinas.

No exemplo a seguir demonstrado na figura 6, será exibido e explicado um código em C++ que exibe a mensagem “Hello world” no terminal.

Figura 6 — Código que exibe um texto em C++



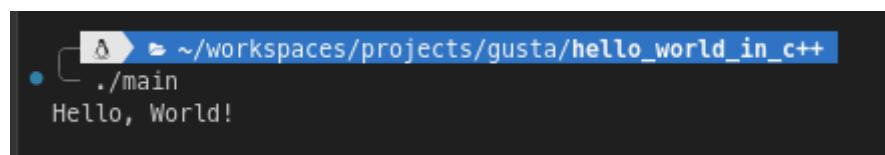
```

1 import std;
2
3 int main()
4 {
5     std::cout<<"Hello, World!\n";
6     return 0;
7 }
```

Fonte: Do próprio autor, 2025

O detalhamento dos comandos utilizados na figura 6 estão detalhados no Apêndice A. A figura 7 ilustra a mensagem que o programa retorna após a sua execução:

Figura 7 — Resultado do código que exibe um texto em C++



```

~/.main
Hello, World!
```

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 dashboard. Para construir a nossa dashboard utilizaremos o framework React, para dar mais profundidade a abordagem da criação do site, explicaremos as principais tecnologias envolvidas no seu funcionamento primeiro.

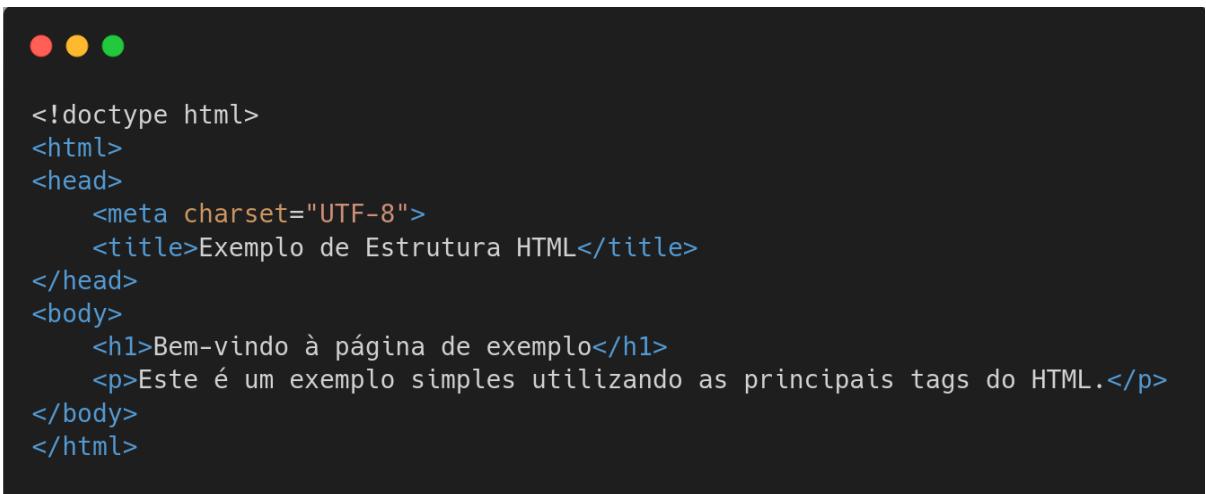
Segundo Silveira e Prates (2001) O Hyper Text Markup Language (HTML) é a tecnologia para criação da estrutura de WebSites. A navegação de forma dinâmica é possível por conta do HyperText que por links conecta as páginas entre si como explica Duckett (2016). Além disso, também destaca que Websites são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

Todo esse processo só é possível graças aos elementos HTML, cuja construção se baseia nas *tags* (etiquetas), nomes colocados entre colchetes angulares (< >), 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 as seguintes apresentadas na figura 8.

Figura 8 — Estrutura básica de uma página em HTML



The image shows a terminal window with three colored window control buttons (red, yellow, green) at the top. The terminal displays the following HTML code:

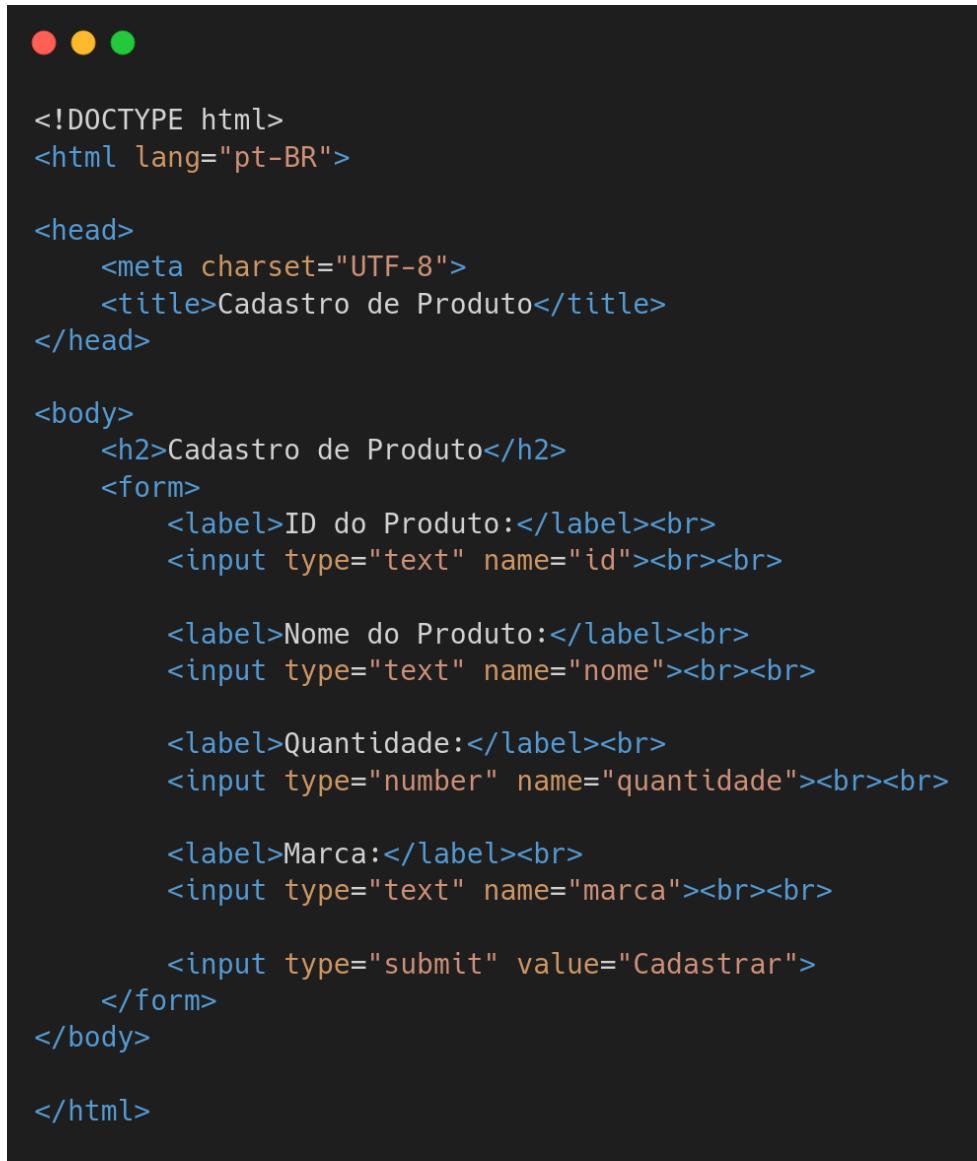
```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Exemplo de Estrutura HTML</title>
</head>
<body>
    <h1>Bem-vindo à página de exemplo</h1>
    <p>Este é um exemplo simples utilizando as principais tags do HTML.</p>
</body>
</html>
```

Fonte: Do próprio autor, 2025

O detalhamento dos comandos utilizados na figura 8 estão detalhados no Apêndice B.

A variedade de tags dentro do HTML é vasta, abordaremos a seguir na figura 9 algumas tags exemplificadas na criação de um formulário de cadastro de produto.

Figura 9 — 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>
        <label>ID do Produto:</label><br>
        <input type="text" name="id"><br><br>

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

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

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

        <input type="submit" value="Cadastrar">
    </form>
</body>

</html>
```

Fonte: Do próprio autor, 2025

O código mostrado na figura 9, resulta na renderização visual feita pelo navegador apresentada na figura 10.

Figura 10 — Resultado do código HTML de uma página de cadastro de produtos.

The screenshot shows a web page with a light gray background and a white central content area. At the top, the title 'Cadastro de Produto' is displayed in a bold, dark font. Below the title, there are four input fields with labels: 'ID do Produto:' (with an empty input box), 'Nome do Produto:' (with an empty input box), 'Quantidade:' (with an empty input box and a small dropdown arrow icon to its right), and 'Marca:' (with an empty input box). At the bottom of the form is a single button labeled 'Cadastrar'.

Fonte: Do próprio autor, 2025

O detalhamento das tags utilizadas na criação da tela representada na figura 10 estão detalhados no Apêndice C.

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

Para complementar o visual dos sites, é utilizado o CSS. Como destaca Knight (2018) a Cascading Style Sheets (CSS) é uma tecnologia usada para estilizar páginas 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 código a seguir na figura 11 mostra a conexão entre os dois arquivos.

Figura 11 — Exemplo de conexão entre HTML e CSS.



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

A screenshot of a terminal window with a black background and white text. At the top left are three colored window control buttons: red, yellow, and green. Below them, the terminal displays the following HTML code:

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

The code shows the opening and closing tags for the head section of an HTML document. Inside the head section, there is a title tag with the text "Document" and a link tag that specifies a CSS file named "style.css".

Fonte: Do próprio autor, 2025

O Detalhamento dos papéis de atributo na conexão estão no ANEXO D. Observa-se que ao especificar o tipo do arquivo e o diretório do mesmo, o navegador é capaz de atribuir os estilos declarados às marcações.

Além disso, Duckett (2016) 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 apontam qual tag estará recebendo as regras de estilo.

A figura 12 faz parte da estilização do site e apresenta como escrever regras na folha de estilo.

Figura 12 — 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 explicação dos seletores e das declarações se encontram no Anexo E.

A figura 13 exibe a solução completa para estilização do formulário de cadastro de produtos da nossa dashboard.

Figura 13 — CSS da tela de cadastro de produtos.

```
● ● ●

body {
    font-family: sans-serif;
    background: #121212;
    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;
    cursor: pointer;
}
```

Fonte: Do próprio autor, 2025

Somado a isso, com a união de todas as tecnologias apresentadas, o resultado da página de cadastro de produtos representado na figura 14.

Figura 14 — Resultado da tela de cadastro HTML e CSS em conjunto.

The image shows a screenshot of a web page with a light gray background. In the center, there is a white rectangular form with a thin black border. At the top left of the form, the text "Informações do Produto" is displayed. Below this, there are four input fields, each preceded by a label: "ID do Produto:", "Nome do Produto:", "Quantidade:", and "Marca:". Each label is followed by a horizontal input field with a thin gray border. At the bottom of the form is a blue rectangular button with the white text "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.

2.15 React

O react

2.16 Node.js

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, representado na figura 15.

Figura 15 — Função juntar nome e sobrenome Node.js



```

● ● ●
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 figura 16 ilustra o que é exibido no terminal após a execução do código.

Figura 16 — Resultado função Node.js

```
Ananda Holanda
```

Fonte: Do próprio autor, 2025

2.17 Banco de dados

```
mysql
```

3 DESENVOLVIMENTO

Já explicado a teoria em que baseamos todo o nosso projeto, agora partiremos para a parte prática. Nesse capítulo abordaremos questões de planejamento e execução do protótipo do projeto.

3.1 Especificação do Sistema

Para começarmos a desenvolver um protótipo para mitigar o problema da falta de rastreabilidade de produtos em mercados autônomos, primeiro precisaremos definir as nossas regras de negócio, elas nos guiarão durante todo o desenvolvimento para podermos seguir uma linha contínua que resulte no que planejarmos.

Para começarmos a definir essas regras de negócio, primeiro precisamos relembrar os pontos principais que devemos solucionar com esse projeto. O sistema deve registrar a entrada de produtos na prateleira, ou seja, se um leitor detectar uma nova etiqueta RFID ele

REFERÊNCIAS

BOMJORNO, Maria; HERINGER, Eudiman; MADUREIRA, Eduardo. **O IMPACTO DA INTERNET DAS COISAS (IoT) E DO BIG DATA NOS PROCESSOS LOGÍSTICOS: INOVAÇÕES NO PICKING E PACKING NA ERA DA INDÚSTRIA 4.0.** Paraná, Centro Universitário FAG. 2025.

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.

CARVALHO, Rafael et al. **IOT PARA CONTROLE E GERENCIAMENTO RESIDENCIAL.** Rio de Janeiro: RevistaFT, 2023.

COSTA, Alexander. **RFControl: sistema de gerência de estoque utilizando RFID Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação).** Minas Gerais: Universidade Federal de Ouro Preto, Instituto de Ciências Exatas e Aplicadas, 2018.

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.

FERREIRA, Juliana Borges. **Desenvolvimento de etiquetas planares passivas UHF RFID em superfícies metálicas e não metálicas com a proposta de uma figura de mérito para avaliação.** 2021. [Tese de Doutorado]. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2021. Orientador: Dr. Álvaro Augusto Almeida de Salles.

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 mai. 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 Standard Edition**. [S. I.]: IBM, 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>. Acesso em: 20 out. 2025.

IBM. **Rational Software Architect**. [S. I.]: 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. I.]: IBM, 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rational-soft-arch/10.0?topic=diagrams-use-case>. Acesso em: 10 ago. 2024.

INFORCHANNEL. **Lojas Renner conclui projeto de implementação de etiquetas RFID**. [S.I.]: 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.

KNIGHT, Indira. **Conectando o Arduino à web: Desenvolvimento de frontend usando JavaScript**. 1. ed. São Paulo: Novatec Editora, 2018.

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>. Acesso em: 27 mai. 2025.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Fundamentos de metodologia científica**. São Paulo: Atlas, 2003. E-Book.

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 INICIAINTES.** 1. ed. São Paulo: Novatec, 2018.

LUCIANO, Josué; ALVES, Wallison. **PADRÃO DE ARQUITETURA MVC: MODEL-VIEW-CONTROLLER.** Bebedouro, UNIFAFIBE. 2011.

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.** [S.I.]: Microsoft, 2023. 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-19745d-ae-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%A3cia-uml-19745d-ae-2872-4455-a906-13b736f01685#officeversion=windows](https://support.microsoft.com/pt-br/topic/criar-um-diagrama-de-sequ%C3%A3ncia-uml-19745d-ae-2872-4455-a906-13b736f01685#officeversion=windows). 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.I.]: Casa do Código, 2014.

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

PORDEUS, Igor Costa. **Desenvolvimento de um aplicativo para adoção de pets utilizando Flutter e Firebase.** 2021. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro Universitário Christus, Fortaleza, 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.

PUC-Rio. **UML: Diagrama de Classes.** Rio de janeiro: PUC-Rio, [s. d.]. Disponível em: https://moodle.unesp.br/pluginfile.php/25933/mod_resource/content/1/diagrama_classes.pdf. Acesso em: 20 ago. 2025

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

SENSORMATIC. **Renner reduz 87% da ruptura de seus estoques com tecnologia RFID da Sensormatic Solutions.** Disponível em: https://www.sensormatic.com/pt_br/resources/cs/2022/renner. Acesso em: 20 ago. 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.

STROUSTRUP, Bjarne. **Princípios e práticas de programação com C++.** 3. ed. Tradução de Eveline Vieira Machado. Revisão técnica de Daniel Antônio Callegari. Porto Alegre: Bookman, 2025.

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 Cataria: 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

Arduino IDE (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: 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.

ANEXOS

ANEXO A - Detalhamento Hello World C++

Stroustrup (2025) explica o funcionamento do código, da seguinte forma:

- “import std” está sendo solicitado para que o computador forneça as funcionalidades disponíveis na biblioteca padrão std.
- “main” todo projeto C++ deve ter a função main. Sendo ela quem mostra para o computador onde ele deve começar a executar o código.
- “int” significando “inteiro”, sendo o tipo de retorno que o programa espera receber após a execução do código.
- “std” usado para identificar o cout dentro da biblioteca padrão do C++ que foi definida no início do código.
- “cout” abreviado de "character output stream" (fluxo de saída de caracteres), cout junto com o operador de saída << exibirá caracteres na tela.
- “<<” operador de saída usado junto com cout para exibir caracteres na tela.
- “ "Hello, World!\n" ” a mensagem que será exibida junto com um marcador de nova linha (\n). Marcador de nova linha coloca o cursor na próxima linha.
- “return 0” o retorno do número 0 pela função main indica que a execução do código foi bem-sucedida.

ANEXO B - Principais etiquetas HTML

Principais tags que o HTML possui segundo Silveira e Prates (2001).

- “<!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.

ANEXO C - Etiquetas criação do formulário cadastro de produto

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.

ANEXO D - Detalhamento dos papéis de atributo na conexão entre HTML e CSS

- “<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.

ANEXO E - Explicação dos seletores e das declarações CSS

- “body” do HTML, indica tudo 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.