

UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
ENGENHARIA DE COMPUTAÇÃO

**ETIQUETAS ELETRÔNICAS INTELIGENTES: UMA SOLUÇÃO IOT COM E-INK
PARA MODERNIZAÇÃO DOS VAREJOS**

MANAUS – AMAZONAS

DEZEMBRO – 2025

EDUARDO DE SOUZA NOGUEIRA

**ETIQUETAS ELETRÔNICAS INTELIGENTES: UMA SOLUÇÃO IOT COM E-INK
PARA MODERNIZAÇÃO DOS VAREJOS**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de Engenharia de Computação, da Escola Superior de Tecnologia, da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro de Computação.

Orientador(a): Prof. Dr. Edgard Luciano Oliveira Da Silva

MANAUS – AMAZONAS

DEZEMBRO – 2025

Universidade do Estado do Amazonas – UEA

Escola Superior de Tecnologia – EST

Reitor: André Luiz Nunes Zogahib

Vice-Reitor: Kátia do Nascimento Couceiro

Diretor da Escola Superior de Tecnologia:

Jucimar Maia Da Silva Júnior

Coordenador do Curso de Engenharia de Computação:

Áurea Hiléia da Silva Melo

Banca Avaliadora composta por:

Data da Defesa: 04/12/2025

Prof. Dr. Edgard Luciano Oliveira da Silva (Orientador)

Prof. Dr. Walter Charles Sousa Seiffert Simões

Prof. M.Sc. Osvaldo Tavares Viana Junior

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).
Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

N778e	Nogueira, Eduardo de Souza Etiquetas eletrônicas inteligentes: Uma solução IoT com E-ink para modernização dos Varejos / Eduardo de Souza Nogueira. Manaus : [s.n], 2025. 50 f.: il., color.; 21.0 cm. TCC - Graduação em Engenharia de Computação- Universidade do Estado do Amazonas, Manaus, 2025. Orientador: da Silva, Edgard Luciano Oliveira. 1. Internet das coisas. 2. e-ink. 3. Automação no varejo. 4. Esp32. I. da Silva, Edgard Luciano Oliveira (Orient.) II. Universidade do Estado do Amazonas. III. Título
CDU(1997)62:004	



FOLHA DE APROVAÇÃO

Etiquetas Eletrônicas Inteligentes: Uma solução IoT com e-ink para modernização dos varejos

Eduardo Souza Nogueira

Trabalho de Conclusão de Curso defendido e aprovado pela banca avaliadora constituída pelos professores:

Prof. Dr. Edgard Luciano Oliveira da Silva
Presidente

Prof. Dr. Walter Charles Sousa Seiffert Simões
Avaliador

Prof. M.Sc. Osvaldo Tavares Viana Junior
Avaliador

Resumo

Atualmente, com o avanço constante das tecnologias voltadas à captura e análise de dados, a capacidade de responder rapidamente às flutuações do mercado tornou-se essencial para manter a competitividade no setor varejista. Diante desse cenário, grandes estabelecimentos comerciais podem enfrentar gargalos operacionais na atualização dos preços de seus produtos em pontos de venda físicos, uma vez que parte dos sistemas utilizados ainda depende de etiquetas aplicadas manualmente, que precisam ser substituídas a cada alteração de preço ou mudança na disposição dos produtos. Como alternativas a esse modelo tradicional, algumas redes têm adotado soluções intermediárias, como impressoras térmicas com códigos de barras atualizáveis, sistemas baseados em *QR Codes* vinculados a bases de dados de preços ou até etiquetas eletrônicas configuráveis. Este trabalho propõe uma solução centralizada baseada na substituição das etiquetas impressas em papel por *displays* digitais com tecnologia *e-ink*, visando automatizar o processo de etiquetagem, reduzir custos operacionais e minimizar erros humanos, além de proporcionar maior agilidade, precisão e confiabilidade à gestão de preços no ambiente varejista.

Palavras-Chave: Internet das Coisas; *e-ink*; automação no varejo, ESP32.

Abstract

Currently, with the constant advancement of technologies aimed at data acquisition and analysis, the ability to respond quickly to market fluctuations has become essential to maintain competitiveness in the retail sector. In this context, large commercial establishments may face operational bottlenecks in updating product prices at physical points of sale, since part of the systems in use still relies on manually applied labels, which must be replaced with each price change or modification in product arrangement. As alternatives to this traditional model, some retail chains have adopted intermediate solutions, such as thermal printers with updatable barcodes, systems based on *QR Codes* linked to price databases, or even configurable electronic labels. This work proposes a centralized solution based on replacing printed paper labels with digital *displays* using *e-ink* technology, aiming to automate the labeling process, reduce operational costs and human errors, and provide greater agility, accuracy, and reliability in price management within the retail environment.

Keywords: Internet of things; e-ink; retail automation; ESP32.

Sumário

Lista de Tabelas	vii
Lista de Figuras	ix
1 Introdução	1
1.1 Objetivos	2
1.2 Justificativa	3
1.3 Metodologia	4
2 Fundamentação Teórica	5
2.1 Internet das Coisas (IoT)	5
2.1.1 Arquitetura de Soluções em IoT	6
2.1.2 Impactos da IoT	8
2.2 Família ESP32	9
2.2.1 Tecnologia de Tinta Eletrônica (E-ink)	11
2.3 Framework Spring	14
2.4 Framework Angular	16
2.5 Arquitetura de Software Model-View-Controller (MVC)	17
2.6 Trabalhos Relacionados	19
3 Solução Proposta	21
3.1 Visão Geral	21
3.2 Diagramas de Caso de Uso	23

3.2.1	Módulo de Gerenciamento de <i>Tags IoT</i>	24
3.2.2	Módulo de Gerenciamento de Produtos	25
3.2.3	Módulo de Gerenciamento de Categorias	26
3.2.4	Módulo de Gerenciamento de Corredores	26
3.3	Modelo de Banco de Dados	27
3.4	Projeto do Hardware	28
3.5	Projeto de Software	32
3.5.1	Software Embarcado no Módulo de Exibição	32
3.5.2	Aplicação Web	34
4	Resultados	40
4.1	Implementação do SyncShelf	40
4.2	Análise dos Resultados	42
5	Considerações Finais	46

Lista de Tabelas

2.1	Comparativo metodológico entre os trabalhos relacionados	19
3.1	Mapeamento de conexões entre o módulo <i>e-ink WeAct Studio 2.7</i> e o microcontrolador <i>ESP32 DevKit V1</i>	30
3.2	Custos dos componentes utilizados no protótipo.	32
4.1	Tempo de Execução – UC-01: Parear Nova Tag	43
4.2	Tempo de Execução – UC-09: Atribuir Produto à Posição	44
4.3	Tempo de Execução – UC-08: Atualizar Preço do Produto	44

Lista de Figuras

2.1	Microcontrolador <i>ESP32 DevKit</i> . Fonte: (ESPRESSIF, 2018).	10
2.2	Módulo <i>E-Ink</i> . Fonte: (Própria, 2025).	13
2.3	Arquitetura <i>MVC</i> . Fonte: (Própria, 2025).	18
3.1	Diagrama Superficial da Arquitetura da Solução. Fonte: (Própria, 2025).	23
3.2	Diagrama de Caso de Uso do Módulo de Gerenciamento de <i>Tags IoT</i> . Fonte: (Própria, 2025).	25
3.3	Diagrama de Caso de Uso do Módulo de Gerenciamento de Produtos. Fonte: (Própria, 2025).	26
3.4	Diagrama de Caso de Uso do Módulo de Gerenciamento de Categorias. Fonte: (Própria, 2025).	26
3.5	Diagrama de Caso de Uso do Módulo de Gerenciamento de Corredores. Fonte: (Própria, 2025).	27
3.6	Modelo Entidade Relacionamento da Solução. Fonte: (Própria, 2025).	28
3.7	Protótipo do sistema de etiquetagem eletrônica: conexão entre o ESP32 Dev-Kit V1 e o módulo e-ink WeAct Studio 2.7” montados em protoboard. Fonte: (Própria, 2025).	31
3.8	Caso de Uso 01: Pareamento de Nova <i>Tag</i> . Fonte: (Própria, 2025).	37
3.9	Caso de Uso 03: Atribuir <i>Tag</i> a Posição. Fonte: (Própria, 2025).	37
3.10	Caso de Uso 09: Atribuir Produto a Posição. Fonte: (Própria, 2025).	38
3.11	Caso de Uso 08: Atualizar Preço do Produto. Fonte: (Própria, 2025).	38
3.12	Caso de Uso 04: Enviar <i>Heartbeat</i> . Fonte: (Própria, 2025).	39

4.1	Tela de <i>Login</i> do <i>SyncShelf</i> . Fonte: (Própria, 2025).	40
4.2	Tela do Caso de Uso <i>UC-012</i> : Criação de Corredor. Fonte: (Própria, 2025). . .	41
4.3	Tela do Caso de Uso <i>UC-05</i> : Criação de Produto. Fonte: (Própria, 2025). . . .	41
4.4	Tela de <i>Dashboard</i> do <i>SyncShelf</i> . Fonte: (Própria, 2025).	42

Capítulo 1

Introdução

A constante evolução tecnológica nas últimas décadas tem impactado significativamente o setor varejista, especialmente no que diz respeito à automação de processos e à gestão eficiente de informações. A digitalização e o uso de tecnologias como a Internet das Coisas (IoT, do inglês *Internet of Things*) vêm transformando práticas tradicionais de operação comercial, promovendo maior controle, redução de custos e aumento da competitividade. No entanto, apesar desses avanços, muitos estabelecimentos ainda enfrentam gargalos operacionais em tarefas básicas, como a atualização de preços em pontos de venda físicos, que frequentemente permanecem atreladas a processos manuais e suscetíveis a falhas (OVANI; WINDASARI, 2022).

Apesar dos avanços recentes na digitalização de ambientes comerciais, ainda é comum observar, em escala global, a utilização de modelos de etiquetamento baseados em impressoras térmicas e etiquetas de papel. Mesmo grandes corporações como *Walmart*, *Carrefour*, *Target* e *Costco* mantêm esse formato em parte significativa de suas operações. A resistência à adoção de tecnologias mais modernas, como etiquetas eletrônicas, é sustentada por uma série de justificativas dentro do ecossistema varejista, entre as quais se destacam os elevados custos de instalação, a necessidade de infraestrutura específica e as preocupações com o consumo energético dos dispositivos (GARAUS; WOLFSTEINER; WAGNER, 2016).

Ainda assim, o processo de precificação manual apresenta uma série de limitações significativas. A dependência de intervenção humana pode acarretar atrasos operacionais, inconsistências entre os preços divulgados online, os exibidos nas gôndolas e os registrados nos caixas, além

de gerar desperdício de papel e outros insumos gráficos. Além disso, a ausência de um sistema centralizado e automatizado compromete a capacidade de resposta rápida às flutuações do mercado, afetando diretamente a experiência do consumidor e reduzindo a competitividade do negócio (VO; VO, 2023).

A proposta deste trabalho é desenvolver uma ferramenta capaz de garantir o controle remoto e centralizado de etiquetas eletrônicas baseadas em tecnologia *e-ink*, oferecendo uma alternativa moderna para a gestão de preços no varejo. O uso de *displays* eletroforéticos possibilita excelente legibilidade e baixo consumo energético, características fundamentais para aplicações em que a autonomia e a atualização eficiente das informações são essenciais (KHAN et al., 2021).

A solução consiste na substituição das etiquetas manuais por etiquetas inteligentes que utilizam microcontroladores de baixo custo e *displays e-ink*, conectados a um servidor central por meio de comunicação sem fio. Essa infraestrutura permite a atualização remota, rápida e confiável das informações de preço, automatizando o processo e viabilizando um gerenciamento totalmente centralizado e acessível em tempo real.

Com foco na redução de esforços operacionais e na melhoria da precisão durante a gestão de preços, a ferramenta proposta oferece simplicidade, eficiência e escalabilidade. A combinação entre *displays* de baixo consumo e um sistema centralizado de comunicação IoT oferece uma alternativa sustentável e alinhada às demandas atuais de modernização do varejo físico (CHANG et al., 2017).

1.1 Objetivos

Objetivo Geral

Desenvolver uma solução automatizada de etiquetagem eletrônica baseada em tecnologia *e-ink* para o setor varejista.

Objetivos Específicos

Para atingir o objetivo geral proposto, estabelecem-se os seguintes objetivos específicos:

- Investigar tecnologias disponíveis no mercado relacionadas a microcontroladores, comunicação IoT e displays com tecnologia *e-ink*.
- Projetar uma arquitetura de sistema que permita a atualização remota e centralizada das etiquetas de preços, utilizando comunicação sem fio entre o servidor e os dispositivos.
- Desenvolver um protótipo funcional composto por microcontrolador, módulo *e-ink* e interface de comunicação web, capaz de exibir e atualizar preços dinamicamente.

1.2 Justificativa

Apesar do crescimento expressivo da transformação digital no varejo, o processo de modernização ainda encontra diversos entraves práticos. Conforme apontado por estudos recentes, muitos estabelecimentos enfrentam dificuldades na adoção de tecnologias emergentes devido a fatores como resistência organizacional, infraestrutura limitada, custos de implementação e incertezas quanto ao retorno financeiro do investimento digital (SAGAR, 2024). Um dos reflexos mais evidentes desse cenário é a persistência de práticas operacionais analógicas, como a troca manual de etiquetas de preços em lojas físicas — uma tarefa repetitiva, propensa a falhas humanas e desalinhada com a dinâmica acelerada dos mercados atuais. Essa lacuna tecnológica impacta diretamente a eficiência operacional, a uniformidade entre canais de venda e a experiência do consumidor.

Contribuir para o desenvolvimento de soluções automatizadas e energeticamente eficientes voltadas à gestão de preços no varejo físico pode representar um avanço significativo na modernização desse setor, especialmente diante das limitações observadas em estabelecimentos que ainda operam com processos manuais ou infraestrutura tecnológica restrita.

A proposta de uso de etiquetas eletrônicas com tecnologia *e-ink* pode viabilizar a implementação de sistemas mais escaláveis, sustentáveis e conectados mesmo em redes de menor aporte financeiro. Tal abordagem apresenta potencial para reduzir significativamente os custos operacionais, otimizar processos internos e promover maior integração entre os canais físico e digital. Ressalta-se, no entanto, que no escopo deste trabalho de conclusão de curso não serão

realizadas implantações em ambientes comerciais reais, sendo as validações restritas ao protótipo funcional desenvolvido. Caso os resultados sejam satisfatórios, estudos futuros poderão contemplar testes em ambientes reais, mediante planejamento adequado e eventual parceria com instituições ou empresas do setor.

Por fim, sob o ponto de vista acadêmico, o desenvolvimento deste projeto possibilita a aplicação integrada de conhecimentos adquiridos ao longo da formação — como eletrônica embarcada, protocolos de comunicação IoT e desenvolvimento de sistemas distribuídos — em uma solução de impacto real.

1.3 Metodologia

O desenvolvimento deste projeto foi organizado em seis etapas fundamentais, planejadas para integrar a fundamentação teórica com a implementação prática, conduzindo à construção do protótipo final. Essa estrutura permitiu compreender o panorama tecnológico, definir requisitos e executar a solução de maneira coerente. As etapas conduzidas foram:

1. Elaborar a revisão teórica sobre os *frameworks Angular* e *Spring*, incluindo também os fundamentos de Internet das Coisas que sustentam a solução proposta;
2. Identificar e detalhar as funcionalidades necessárias ao sistema de gerenciamento dos visores eletrônicos;
3. Selecionar e analisar as tecnologias apropriadas para o desenvolvimento dos componentes de *hardware* e *software*, com ênfase em microcontroladores com *Wi-Fi* e telas *e-ink*;
4. Desenvolver, implementar e documentar tanto a parte física do dispositivo quanto a aplicação *web* e os serviços de *backend*;
5. Realizar a avaliação da solução implementada, por meio de testes e validações funcionais;
6. Disponibilizar a solução final, permitindo sua reprodução e evolução em trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os conceitos e as tecnologias que constituem a base para o desenvolvimento da solução proposta. A estrutura da fundamentação segue uma abordagem que parte do paradigma conceitual da Internet das Coisas, passa pelos componentes de hardware e software específicos utilizados no projeto, e conclui com o padrão de arquitetura que organiza a aplicação. Serão detalhados o conceito de Internet das Coisas (IoT), o microcontrolador *NodeMCU*, a tecnologia de tela *e-ink*, os *frameworks Spring* e *Angular* para o desenvolvimento do sistema de *back-end* e *front-end*, respectivamente, e a arquitetura *Model-View-Controller* (MVC) que guia a estruturação do *software*.

2.1 Internet das Coisas (IoT)

A Internet das Coisas, ou *Internet of Things* (IoT), refere-se a um paradigma tecnológico que integra objetos físicos ao ambiente digital por meio de sensores, atuadores, conectividade em rede e sistemas de processamento de dados. Esses objetos — que podem incluir desde eletrodomésticos, veículos, máquinas industriais, dispositivos vestíveis até sensores ambientais — tornam-se “inteligentes” ao adquirir a capacidade de coletar, transmitir, processar e reagir a informações sem necessidade de intervenção humana contínua (BERTE et al., 2018). Assim, a IoT expande a conectividade tradicional da internet, anteriormente restrita a computadores e *smartphones*, para uma diversidade de dispositivos cotidianos, estabelecendo um ecossistema

capaz de suportar novos modelos de automação, monitoramento e tomada de decisão digital (DOMÍNGUEZ-BOLAÑO et al., 2022).

Do ponto de vista conceitual, a IoT surge da convergência de avanços em diversas áreas tecnológicas, como sensoriamento miniaturizado, protocolos de comunicação de baixo consumo, computação em nuvem, *big data* e inteligência artificial. A redução do custo dos sensores e microcontroladores, associada ao crescimento da infraestrutura de redes sem fio, possibilitou a disseminação de dispositivos conectados em larga escala. Atualmente, a IoT é considerada um dos pilares da chamada Indústria 4.0, possibilitando a criação de sistemas ciberfísicos capazes de integrar o ambiente físico ao digital de forma contínua e inteligente.

2.1.1 Arquitetura de Soluções em IoT

Embora existam diferentes modelos de referência, como a arquitetura IoT-A e o modelo de múltiplas camadas da Cisco, uma das formas mais recorrentes de representar a IoT na literatura é por meio de uma arquitetura dividida em quatro camadas principais (DOMÍNGUEZ-BOLAÑO et al., 2022).

Camada de Sensoriamento

A primeira camada, também chamada de *camada de percepção*, compreende os dispositivos físicos responsáveis pela coleta e atuação no ambiente. Inclui sensores capazes de medir variáveis como temperatura, luminosidade, umidade, movimento ou localização, bem como atuadores que respondem aos estímulos processados pelo sistema, como motores, relés ou *displays* eletrônicos. Essa camada estabelece o elo entre o mundo físico e o digital, sendo essencial para a tradução de fenômenos físicos em dados computacionais.

Camada de Rede

A camada de rede, ou de conectividade, garante o transporte dos dados coletados até servidores locais ou plataformas em nuvem. Diversas tecnologias podem ser empregadas, dependendo dos requisitos de alcance, energia e largura de banda, como *Wi-Fi*, *Bluetooth Low Energy*

(BLE), *LoRaWAN*, *ZigBee* e redes celulares (3G, 4G, 5G e NB-IoT). Em termos de protocolos, destacam-se o *MQTT*, voltado para comunicação leve e eficiente; o *CoAP*, destinado a dispositivos extremamente limitados; e o *HTTP*, mais comum em aplicações web tradicionais (BURHAN et al., 2018). A escolha adequada dessa camada é determinante para garantir interoperabilidade e eficiência.

Camada de Processamento de Dados

A terceira camada compreende os servidores, *gateways* e plataformas em nuvem responsáveis por armazenar, processar e analisar os dados provenientes dos dispositivos. Serviços de computação em nuvem fornecem ferramentas para armazenamento escalável, análise de dados, detecção de padrões e aplicação de algoritmos de aprendizado de máquina. Tecnologias como bancos de dados orientados a séries temporais, sistemas de *event streaming* e mecanismos de análise distribuída são amplamente utilizadas nesse contexto.

Camada de Aplicação

A camada de aplicação é responsável pela interação com o usuário final, apresentando os dados processados por meio de interfaces gráficas, *dashboards*, aplicativos móveis ou sistemas supervisórios. Além de visualização, essa camada permite o controle remoto de dispositivos, a configuração de regras de automação e a integração com outros sistemas corporativos. Em ambientes comerciais, a camada de aplicação também pode realizar monitoramento em larga escala, emissão de alertas e atualizações remotas de *firmware* (*Over-the-Air Updates*).

Além desses aspectos estruturais, soluções baseadas em IoT também demandam atenção a questões de segurança, especialmente em ambientes conectados onde dispositivos físicos interagem continuamente com serviços remotos. A comunicação entre etiquetas eletrônicas e servidores precisa adotar mecanismos de autenticação e criptografia para evitar interceptação ou manipulação de comandos. Da mesma forma, a exposição direta do endereço IP dos dispositivos pode representar um vetor de risco, permitindo tentativas de acesso não autorizado ou mapeamento indevido da rede interna. Assim, práticas como isolamento de rede, uso de canais

seguros e controle rigoroso de credenciais tornam-se essenciais para garantir a integridade e a confiabilidade das aplicações IoT.

2.1.2 Impactos da IoT

A utilização da IoT tem provocado transformações profundas em setores como manufatura, saúde, transporte e agricultura, possibilitando automação avançada, otimização de recursos e criação de novos modelos de negócio (MAGOMEDOV; BAGOV; ZOLKIN, 2020).

Com a popularização da IoT, uma série de produtos e serviços que antes eram considerados complexos ou financeiramente inacessíveis passaram a se tornar modelos de negócio concretos. Soluções como monitoramento remoto de ativos, automação residencial e industrial, rastreamento inteligente, agricultura de precisão, sistemas de iluminação pública conectada e gerenciamento automatizado de estoques tornaram-se acessíveis tanto para pequenas empresas quanto para grandes organizações. Essa transformação ocorre porque a IoT permite a coleta contínua de dados e sua análise em tempo real, possibilitando decisões mais rápidas, maior eficiência operacional e redução de custos.

Além disso, a IoT impulsiona novos paradigmas de monetização baseados em serviços, como o *Device-as-a-Service* (DaaS), manutenção preditiva, assinaturas de dispositivos inteligentes e modelos baseados em consumo (*pay-per-use*). Empresas passam a comercializar não apenas produtos, mas ecossistemas completos, integrando hardware, software e serviços em nuvem. Essa mudança favorece a inovação contínua, a diferenciação competitiva e a criação de mercados antes inexistentes.

Nesse contexto, a IoT não apenas aprimora processos já existentes, mas também viabiliza ideias que antes eram consideradas inviáveis, seja por limitações tecnológicas, seja pelo elevado custo de implementação. A capacidade de conectar objetos cotidianos, monitorá-los à distância e automatizar ações abre espaço para soluções disruptivas e para o surgimento de novos mercados orientados por dados, consolidando a IoT como um dos pilares centrais da transformação digital contemporânea.

No contexto deste trabalho, esse paradigma viabiliza a integração entre a plataforma web

e as etiquetas eletrônicas, permitindo a atualização centralizada e automática das informações de preço no ambiente varejista.

2.2 Família ESP32

A família *ESP32* representa uma das linhas mais avançadas de microcontroladores desenvolvidos pela empresa *Espressif Systems*, sendo amplamente adotada em projetos de *Internet das Coisas* devido à sua combinação de alto desempenho, múltiplas interfaces de conectividade e baixo consumo energético. Assim como ocorre na família *ESP8266*, o objetivo fundamental desses dispositivos é permitir que objetos, sensores e sistemas embarcados sejam integrados à internet, possibilitando a coleta, transmissão e processamento de dados em tempo real. Todavia, o *ESP32* destaca-se por apresentar melhorias substanciais em capacidade de processamento, segurança e variedade de periféricos, expandindo significativamente o potencial de desenvolvimento de soluções *IoT* (PLAUSKA; LIUTKEVIČIUS; JANAVIČIŪTĖ, 2022).

Com o crescimento da *IoT*, a demanda por microcontroladores capazes de integrar diversos sensores, operar com conectividade estável e executar algoritmos mais complexos tornou-se cada vez maior. Nesse cenário, o *ESP32* consolidou-se como uma das plataformas preferidas tanto na comunidade acadêmica quanto industrial. Esses microcontroladores possibilitam que objetos físicos se conectem à *Internet* de forma eficiente, permitindo aplicações que envolvem automação residencial e industrial, dispositivos vestíveis, sistemas de monitoramento, etiquetas eletrônicas e painéis inteligentes.

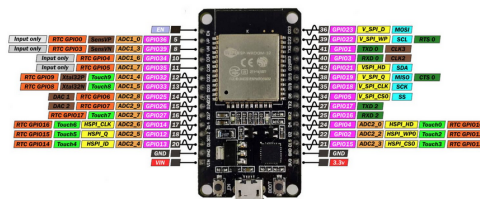
A família *ESP32* compreende diversos módulos e *SoCs*, como *ESP32-WROOM*, *ESP32-WROVER*, *ESP32-S2*, *ESP32-C3* e *ESP32-H2*, cada um projetado para atender necessidades específicas de conectividade, segurança ou consumo energético. Alguns desses modelos incluem suporte integrado a *Wi-Fi*, *Bluetooth Clássico*, *Bluetooth Low Energy (BLE)*, *Zigbee*, *Thread* ou redes *802.15.4*, permitindo ao desenvolvedor selecionar o módulo mais adequado para o contexto do projeto (RAI; REHMAN, 2019). Esses microcontroladores podem ser encontrados tanto em módulos individuais quanto em placas de desenvolvimento como o *ESP32 DevKitC*, *NodeMCU ESP32* e variações produzidas por diferentes fabricantes.

Em termos de características técnicas, o *ESP32* apresenta avanços significativos em relação à família *ESP8266*. Entre os principais recursos, destacam-se:

- **Processamento:** CPU *dual-core Xtensa LX6* ou *RISC-V* (dependendo da versão), com frequência de até 240 MHz.
- **Memória:** *RAM* de alta velocidade integrada e, em alguns modelos, suporte à *PSRAM* externa.
- **Conectividade:** *Wi-Fi* 802.11 b/g/n, *Bluetooth* 4.2/*BLE* e, em versões mais recentes, *Zigbee*, *Bluetooth 5*, *Thread* e *IEEE 802.15.4*.
- **Segurança:** criptografia por *hardware*, armazenamento seguro de chaves, *boot* seguro e aceleração criptográfica.
- **Periféricos:** múltiplos *ADCs*, *DACs*, interfaces *UART*, *I2C*, *SPI*, *PWM*, *I2S*, *RMT*, sensores *touch* capacitivos e suporte a *CAN*.
- **Energia:** modos de baixo consumo (*Deep Sleep* e *Light Sleep*), adequados para projetos alimentados por bateria.
- **Software:** suporte nativo ao *FreeRTOS*, pilha *TCP/IP* completa e compatibilidade com protocolos como *MQTT*, *HTTP(S)* e *CoAP*.

Esses recursos tornam o *ESP32* altamente versátil, permitindo sua utilização desde aplicações simples até projetos comerciais que exigem desempenho elevado, confiabilidade e comunicação contínua (KURNIASYAH; RAKHMAWATI, 2024).

Figura 2.1: Microcontrolador *ESP32 DevKit*. Fonte: (ESPRESSIF, 2018).



A Figura 2.1 apresenta o diagrama de pinos do módulo *ESP32*, no qual é possível observar suas interfaces distribuídas ao longo das duas fileiras laterais da placa. Os pinos são identificados por cores e legendas de acordo com suas funções — tais como alimentação, *GPIOs* multifuncionais, interfaces de comunicação (*UART*, *SPI*, *I²C*), entradas analógicas e pinos especiais de controle — conforme descrito na documentação técnica do fabricante (HAKKI, 2021). De acordo com essa documentação, o *ESP32* é amplamente apropriado para aplicações que demandam conectividade sem fio e processamento embarcado, incluindo automação residencial, sistemas inteligentes de acionamento, redes de sensores, dispositivos vestíveis, controle industrial, monitoramento ambiental, entre outros.

Neste trabalho, a escolha da família *ESP32* fundamenta-se em sua ampla adoção em projetos de *IoT*, graças à combinação de conectividade *Wi-Fi* integrada, baixo consumo energético e elevada capacidade de processamento. Neste trabalho, esse microcontrolador atua como elemento central do módulo embarcado, sendo responsável pela comunicação com o servidor e pelo controle das informações exibidas nas etiquetas eletrônicas.

2.2.1 Tecnologia de Tinta Eletrônica (E-ink)

A tecnologia de tinta eletrônica, ou *e-ink*, consiste em um tipo de tela desenvolvida para reproduzir a aparência da tinta impressa sobre o papel, proporcionando uma experiência de leitura natural, confortável e altamente eficiente em termos energéticos. Sua principal característica é a biestabilidade: o *display* consome energia apenas durante a atualização do conteúdo exibido. Uma vez que uma imagem ou texto é formado, ele permanece visível sem gasto adicional de energia, o que torna a tecnologia particularmente adequada para aplicações de baixo consumo, como as etiquetas eletrônicas de prateleira (*ESLs*), *e-readers*, relógios inteligentes e diversos dispositivos de *IoT* (RABBANI; AHMED; ZAHID, 2023).

O funcionamento da tecnologia *e-ink* baseia-se em uma estrutura conhecida como *display* eletroforético (*EPD – Electrophoretic Display*). Cada tela é formada por milhões de microcápsulas suspensas em um fluido transparente, com dimensões próximas ao diâmetro de um fio de cabelo humano. Dentro dessas microcápsulas encontram-se partículas de pigmento preto

carregadas negativamente e partículas de pigmento branco carregadas positivamente. A aplicação de um campo elétrico através de uma matriz de eletrodos posicionados abaixo da camada eletroforética controla a movimentação dessas partículas: as partículas pretas são atraídas para a superfície da tela para formar regiões escuras, enquanto as partículas brancas são deslocadas para o fundo, formando regiões claras. O processo inverso ocorre para exibir pontos brancos, permitindo a criação de caracteres, imagens e padrões complexos (IVANOVA et al., 2020).

Por ser uma tecnologia reflexiva, a *e-ink* utiliza a luz ambiente para a visualização, dispensando o uso de retroiluminação. Isso resulta em alto contraste, excelente legibilidade em ambientes iluminados e um ângulo de visão próximo a 180 graus. A ausência de luz de fundo contribui não apenas para o baixo consumo energético, mas também para a redução da fadiga ocular durante leituras prolongadas, proporcionando uma experiência mais confortável e semelhante à leitura em papel (YANG et al., 2020).

Além da biestabilidade, a tecnologia *e-ink* apresenta diversas vantagens relevantes para aplicações embarcadas e dispositivos *IoT*, incluindo:

- **Consumo energético extremamente baixo**, permitindo operação por longos períodos com fontes limitadas de energia, como baterias de pequena capacidade.
- **Durabilidade elevada**, uma vez que o *display* não depende de camadas emissivas sujeitas a desgaste rápido.
- **Flexibilidade física**, disponível em versões capazes de suportar curvaturas moderadas.
- **Leveza e robustez**, ideais para aplicações portáteis e ambientes comerciais.
- **Compatibilidade com microcontroladores** como *ESP8266*, *ESP32* e variantes, facilitando atualizações remotas via *Wi-Fi* ou *BLE*.

Figura 2.2: Módulo *E-Ink*. Fonte: (Própria, 2025).

A Figura 2.2 apresenta o módulo de *display* de tinta eletrônica utilizado no desenvolvimento deste projeto. Esse tipo de módulo, baseado em tecnologia eletroforética, é amplamente empregado em aplicações de baixo consumo energético devido à sua capacidade de manter a imagem exibida sem gasto contínuo de energia. O módulo apresentado integra a camada eletroforética composta por microcápsulas com pigmentos preto, branco e vermelho. Os módulos *e-ink* apresentam excelente legibilidade sob diferentes condições de luminosidade, incluindo ambientes de forte incidência de luz, característica essencial para dispositivos instalados em gôndolas de supermercados ou lojas de varejo, onde a iluminação varia ao longo do ambiente. Essas propriedades justificam sua aplicação neste projeto, no qual os displays são utilizados para substituir as etiquetas tradicionais, garantindo eficiência energética e maior durabilidade operacional.

Outro aspecto importante evidenciado pelo módulo é a interface elétrica simplificada, que permite sua integração direta com microcontroladores como a família *ESP32*. A comunicação com o *display* é realizada tipicamente por meio de interfaces *SPI*, facilitando a atualização de imagens ou textos via *software* e permitindo que o sistema funcione de maneira estável mesmo com baixa taxa de atualização — característica comum das telas *e-ink*, mas totalmente adequada ao cenário de atualização esporádica de informações, como preços e etiquetas de produtos.

2.3 Framework Spring

O *Spring Framework* é uma plataforma de código aberto, abrangente e robusta para o desenvolvimento de aplicações empresariais em *Java*. Criado por *Rod Johnson* no início dos anos 2000 como uma alternativa mais leve ao modelo de componentes *EJB*, o *Spring* rapidamente se consolidou como um dos *frameworks* mais utilizados no ecossistema *Java* devido à sua flexibilidade, modularidade e foco em boas práticas de engenharia de *software*. Seu objetivo principal é simplificar o desenvolvimento de aplicações complexas, fornecendo um modelo de programação que promove baixo acoplamento, alta coesão e separação clara de responsabilidades (JOHNSON et al., 2004).

Um dos conceitos centrais do *Spring* é a **Inversão de Controle** (IoC – *Inversion of Control*), implementada por meio do padrão de **Injeção de Dependência** (DI – *Dependency Injection*). Nesse paradigma, o contêiner *IoC* do *Spring* é responsável por criar, gerenciar e injetar as dependências nos componentes da aplicação, que passam a declarar apenas suas necessidades ao invés de instanciá-las diretamente. Esse mecanismo reduz o acoplamento entre módulos, facilita a realização de testes unitários, aumenta a modularidade do código e contribui para um desenvolvimento mais seguro e sustentável ao longo do ciclo de vida da aplicação (WALLS, 2022). O *Spring* também oferece suporte à Programação Orientada a Aspectos (*AOP*), permitindo encapsular preocupações transversais como *logging*, segurança e transações.

O ecossistema *Spring* é vasto e organizado de forma modular, permitindo que desenvolvedores utilizem apenas os componentes necessários. Entre seus principais módulos, destacam-se:

- **Spring Core:** Base fundamental do *framework*, fornecendo o contêiner *IoC*, mecanismos de *DI* e suporte a *AOP*.
- **Spring MVC:** *Framework* baseado no padrão *Model-View-Controller* para o desenvolvimento de aplicações *web* e *APIs RESTful*, com mapeamento flexível de rotas, conversão e validação de dados.
- **Spring Data:** Simplifica a persistência de dados, oferecendo abstrações para *JPA*, *JDBC* e bancos *NoSQL*, além do sistema de *repositories*, que reduz código *boilerplate*.

- ***Spring Security***: Módulo especializado em autenticação e autorização, oferecendo suporte a autenticação tradicional, *tokens JWT*, *OAuth2*, *OpenID Connect*, entre outros mecanismos de controle de acesso.

Com o objetivo de acelerar ainda mais o desenvolvimento, surgiu o *Spring Boot*, um projeto complementar projetado para eliminar grande parte da configuração manual típica das aplicações *Java*. O *Spring Boot* fornece um conjunto de configurações automáticas, convenções opinativas e servidores embutidos (como *Tomcat*, *Jetty* ou *Undertow*), permitindo que aplicações sejam executadas de forma autônoma e com mínima configuração inicial. Essa abordagem facilita a criação de sistemas prontos para produção, com inicialização rápida e estrutura padronizada.

O *Spring Boot* tornou-se uma das principais escolhas para arquiteturas de microsserviços, nas quais cada serviço é implementado como uma aplicação *Spring Boot* independente, com seu próprio ciclo de vida, configurações e ambiente de execução. Em conjunto com o *Spring Cloud*, o ecossistema oferece recursos essenciais para sistemas distribuídos, como descoberta de serviços, configuração centralizada, *circuit breakers*, tolerância a falhas e *gateways API*.

Neste projeto, o *framework* será utilizado para construir os módulos lógicos pertinentes à regra de negócio da aplicação, fornecendo uma base estável e padronizada para a implementação dos serviços responsáveis pelo gerenciamento dos dispositivos, atualização das informações exibidas nas etiquetas e integração com os demais componentes do sistema. A utilização do *Spring Boot* permite estruturar essas funcionalidades de forma modular, escalável e facilmente testável, garantindo que a aplicação *web* resultante seja capaz de atender aos requisitos de desempenho, manutenção e extensibilidade esperados em um ambiente de *IoT*. Além disso, sua compatibilidade com padrões modernos de arquitetura, como *APIs RESTful*, torna-o especialmente adequado para o desenvolvimento de um *backend* que servirá de ponte entre os microcontroladores e a interface administrativa da solução. Dessa maneira, o uso do *Spring Boot* não apenas agiliza o desenvolvimento, como também assegura robustez e confiabilidade à camada de serviços da aplicação.

2.4 Framework Angular

O *Angular* é uma plataforma de desenvolvimento e um *framework* de código aberto, mantido pelo *Google*, voltado para a criação de aplicações de página única (*SPAs* – *Single-Page Applications*) modernas, eficientes e escaláveis. Após sua reescrita completa em 2016, o *Angular* passou a utilizar o *TypeScript* como linguagem base, possibilitando o uso de tipagem estática, interfaces, decoradores e outros recursos avançados que favorecem a criação de aplicações robustas e de fácil manutenção (BHASKAR, 2020).

A arquitetura do *Angular* é centrada em componentes. Cada componente encapsula três elementos fundamentais: um *template HTML*, responsável pela estrutura visual; uma classe escrita em *TypeScript*, que contém a lógica de funcionamento e o estado; e uma folha de estilo associada. Essa abordagem modular e declarativa permite a construção de interfaces complexas a partir de blocos independentes e reutilizáveis, facilitando o desenvolvimento incremental e a manutenção do sistema (GEETHA et al., 2022).

Além dos componentes, o *Angular* fornece conceitos fundamentais que estruturam sua arquitetura:

- **Componentes:** unidades básicas que controlam pequenas partes da interface, organizadas hierarquicamente.
- **Templates:** arquivos *HTML* enriquecidos pela sintaxe do *Angular*, permitindo interpolação de dados e uso de diretivas estruturais como `*ngIf` e `*ngFor`.
- **Data Binding:** mecanismo que sincroniza dados entre o modelo (classe *TypeScript*) e a visão (*template*), podendo ser unidirecional ou bidirecional (*two-way data binding*) (LIU et al., 2021).
- **Serviços e Injeção de Dependência:** serviços encapsulam lógica de negócio ou operações comuns, sendo injetados em componentes através do contêiner de dependências do *Angular*, promovendo baixo acoplamento (EMMANNI, 2024).

- **Roteamento:** módulo dedicado à navegação entre visões, permitindo transição entre componentes sem recarregar a página, além de suportar rotas aninhadas, *lazy loading* e proteção de rotas.

O *framework* também oferece um ecossistema rico que inclui suporte a formulários reativos, validação, internacionalização, comunicação via *HTTP*, testes automatizados e modularização estruturada por *NgModules*. Esses recursos tornam o *Angular* especialmente adequado para aplicações *web* de médio e grande porte que exigem organização, escalabilidade e facilidade de manutenção (EMMANNI, 2023).

No contexto deste projeto, o *Angular* é utilizado para a construção da interface administrativa, responsável por permitir que o usuário visualize, edite e atualize informações que serão posteriormente enviadas aos microcontroladores e aos módulos de tinta eletrônica. Dessa forma, o *framework* contribui diretamente para a criação de uma aplicação *web* responsiva, modular e alinhada às melhores práticas de desenvolvimento *front-end* (BHASKAR, 2020; EMMANNI, 2024).

2.5 Arquitetura de Software Model-View-Controller (MVC)

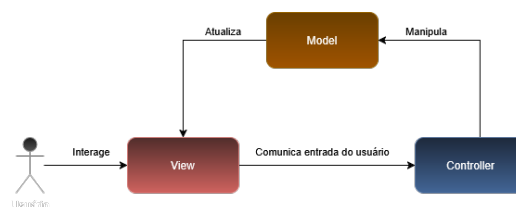
A arquitetura de *software Model-View-Controller (MVC)* é um padrão de projeto amplamente adotado na indústria de desenvolvimento de *software* desde sua sintetização em meados de 1980. Originada a partir da pesquisa de doutorado de *Trygve Mikkjel H. Reenskaug*, essa arquitetura tomou forma no âmbito do paradigma de programação orientada a objetos e foi inicialmente concebida para resolver desafios de interação homem-computador e melhorar a manutenibilidade do código (NECULA, 2024).

Conforme descrito no trabalho de *Reenskaug* (REENSKAUG, 2003), antes da introdução do *MVC*, a combinação indistinta de lógica de negócios e elementos de interface frequentemente resultava em um código desorganizado e de difícil manutenção, tornando árdua a tarefa de efetuar melhorias e introduzir novos recursos. A adoção do padrão *MVC* solucionou esses problemas ao estabelecer uma clara separação de responsabilidades por meio da criação de três

componentes essenciais: o *Model*, a *View* e o *Controller* (VOORHEES, 2020).

1. **Modelo (*Model*):** representa a camada de dados e a lógica de negócios do sistema. Ele encapsula as operações relacionadas aos dados, como recuperação, atualização e validação. Ao isolar a lógica de negócios do restante da aplicação, o *Model* torna-se independente das interfaces de usuário, o que facilita mudanças e manutenção;
2. **Visualização (*View*):** é responsável por apresentar os dados aos usuários de forma compreensível e interativa. Ela exibe a interface do usuário e coleta interações, como cliques e entradas. A *View* não contém lógica de negócios; em vez disso, reflete os dados do *Model* e as ações do usuário mediadas pelo *Controller*;
3. **Controlador (*Controller*):** atua como intermediário entre o *Model* e a *View*. Ele gerencia as interações do usuário e coordena as operações no *Model*. Quando o usuário interage com a interface, o *Controller* interpreta essa ação, atualiza o *Model* conforme necessário e solicita que a *View* reflita as mudanças (OZKAYA; FIDANDAN, 2020).

Figura 2.3: Arquitetura *MVC*. Fonte: (Própria, 2025).



A Figura 2.3 ilustra como a interação e integração entre esses elementos constituem um pilar essencial do padrão arquitetônico *MVC*. Quando o usuário interage com a camada de visualização, por meio de uma entrada de dados, o *Controller* assume a responsabilidade de processar essa entrada, executar ajustes pertinentes no *Model*, conforme a lógica previamente estabelecida, e atualizar a *View* em tempo real (THAKUR; PANDEY, 2019).

No contexto deste trabalho, o *MVC* sustenta a integração entre o *backend* em *Spring* e o *frontend* em *Angular*, garantindo uma comunicação organizada entre os módulos do sistema e os dispositivos *IoT*.

2.6 Trabalhos Relacionados

A pesquisa e o desenvolvimento de sistemas de etiquetagem eletrônica e automação para o varejo têm sido temas de crescente interesse na comunidade acadêmica e de desenvolvimento. Esta seção apresenta uma análise de trabalhos relacionados, contextualizando a proposta deste projeto em relação a soluções existentes e destacando seus diferenciais.

A literatura acadêmica tem explorado diversas abordagens para a implementação de sistemas de *ESL*. Le e outros (LE et al., 2023) propuseram um sistema de *ESL* de baixo consumo energético utilizando *displays e-paper* e a tecnologia de comunicação *LoRa*. O foco do trabalho foi a otimização da vida útil da bateria e o alcance da comunicação, demonstrando a viabilidade de sistemas de *ESL* com autonomia superior a cinco anos. Embora compartilhe o uso da tecnologia *e-paper*, o presente trabalho se diferencia ao optar pela conectividade *Wi-Fi*, mais prevalente em ambientes de varejo internos e que simplifica a integração com a infraestrutura de rede comumente existente nas grandes redes varejistas.

Liawatimena e Gunawan (LIAWATIMENA; GUNAWAN, 2024) focaram na camada de *software*, projetando um sistema automático de etiquetas de preço com uma interface de gerenciamento baseada na *web*, com o objetivo de reduzir imprecisões de precificação e otimizar a experiência de compra do cliente. A proposta dos autores se assemelha à deste trabalho no que tange à gestão centralizada via interface *web*.

Tabela 2.1: Comparativo metodológico entre os trabalhos relacionados

Trabalho	Desenvolvimento de Sistema <i>ESL</i>	Uso de Módulo <i>E-ink</i>
(LE et al., 2023)		X
(LIAWATIMENA; GUNAWAN, 2024)	X	
Este trabalho	X	X

A Tabela 2.1 apresenta um comparativo metodológico entre os principais trabalhos analisados e a proposta desenvolvida neste projeto. Observa-se que, enquanto algumas pesquisas concentram-se no uso de módulos de *e-ink* ou na implementação de sistemas de *ESL* com interface *web*, não foi identificado na literatura um trabalho que integrasse simultaneamente o

desenvolvimento completo de um sistema de *ESL* aliado ao uso direto de módulos de *e-ink*, conforme proposto nesta pesquisa. Dessa forma, este trabalho se diferencia ao reunir ambas as abordagens em uma única solução integrada.

Capítulo 3

Solução Proposta

Esta seção tem como finalidade descrever em detalhes os componentes que estruturam a solução desenvolvida para o cenário em estudo. Inicialmente, apresenta-se uma visão geral da proposta na Seção 3.1, seguida pelos diagramas de caso de uso elaborados para a elicitaco dos requisitos, descritos na Seção 3.2. Na sequncia, o projeto de hardware   abordado na Seção 3.3, enquanto os aspectos relacionados ao projeto de software so discutidos na Seção 3.4.

3.1 Viso Geral

A ideia geral da soluo proposta consiste essencialmente no desenvolvimento de um sistema eletrnico de precificao inteligente, voltado   automatizao do processo de atualizao de preos em ambientes varejistas. A proposta fundamenta-se na criao de mdulos de exibio autnomos, compostos por microcontroladores integrados a *displays* de tinta eletrnica (*e-ink*), capazes de apresentar informaes de preo e produto de forma dinmica, leg vel e energeticamente eficiente.

Cada mdulo, projetado e implementado neste trabalho na forma de prot tipo funcional,   destinado   substituio das tradicionais etiquetas de papel utilizadas nas prateleiras de supermercados e lojas. Esses dispositivos so instalados diretamente nas g ndolas ou suportes de exposio, exibindo os preos e descries dos produtos de acordo com os dados recebidos de uma plataforma *web* administrativa. Tal integrao permite a atualizao remota e auto-

matizada das informações, eliminando a necessidade de intervenção manual e reduzindo erros operacionais.

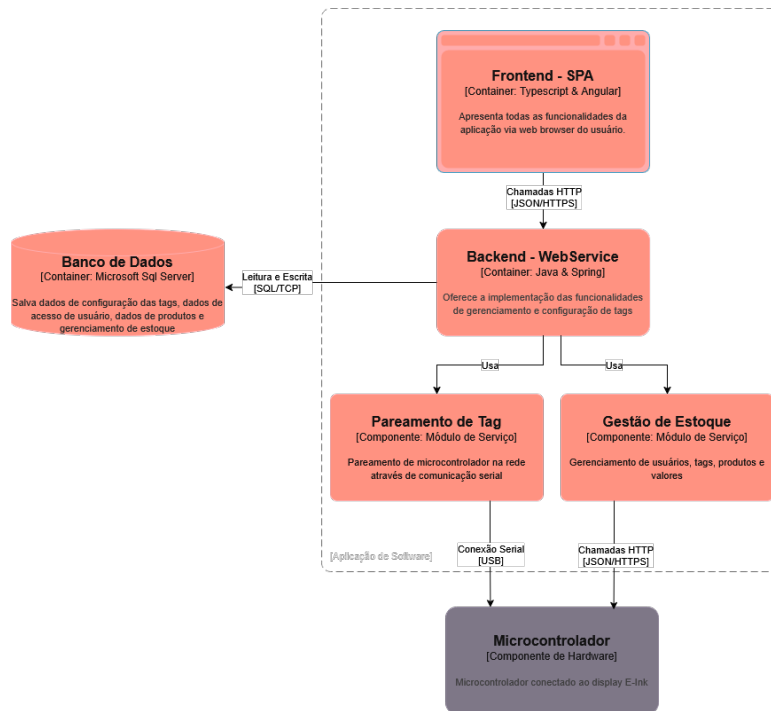
Os *displays e-ink* foram selecionados por suas características ideais para o contexto varejista: baixo consumo de energia, alta legibilidade em diferentes condições de iluminação e capacidade de manter o conteúdo exibido sem consumo contínuo de energia. Isso torna os módulos altamente eficientes e adequados para operação contínua, com autonomia prolongada e baixo custo de manutenção.

Cada unidade é controlada por um microcontrolador com conectividade de rede, responsável por estabelecer comunicação segura com o servidor central. Por meio desse componente, o dispositivo é capaz de receber comandos e dados via requisições *HTTPS*, processar o conteúdo recebido e atualizar automaticamente as informações exibidas em seu *display*. Essa abordagem garante uma comunicação direta, leve e confiável, adequada às necessidades de sincronização entre centenas de etiquetas eletrônicas espalhadas por um ambiente de varejo.

No âmbito do *software*, a solução é composta por uma plataforma *web* administrativa, desenvolvida com o *framework Angular*, que atua como núcleo de controle e gerenciamento dos dispositivos. Através dessa interface, administradores e operadores podem cadastrar produtos, associá-los a módulos específicos, definir preços, alterar descrições e emitir comandos de atualização de forma centralizada. Além disso, a aplicação *web* mantém um banco de dados de produtos e histórico de atualizações, garantindo rastreabilidade e integridade das informações exibidas nos módulos.

O fluxo de comunicação, como ilustrado na Figura 3.1, ocorre da seguinte maneira: ao realizar uma atualização de preço ou cadastro de produto na aplicação *web*, o servidor processa os dados e emite uma requisição *HTTPS* para o módulo correspondente. O microcontrolador interpreta a requisição, processa as informações recebidas e atualiza o conteúdo do *display e-ink*. Após a conclusão, o dispositivo envia uma resposta de confirmação à plataforma, assegurando a integridade da comunicação e o sucesso da operação.

Figura 3.1: Diagrama Superficial da Arquitetura da Solução. Fonte: (Própria, 2025).



A coexistência de diversos módulos inteligentes é contemplada desde o projeto, permitindo a expansão escalável do sistema em ambientes com grande quantidade de produtos e prateleiras. A padronização na comunicação e na estrutura de dados garante que novas unidades possam ser adicionadas de forma transparente, sem necessidade de reconfiguração manual ou alterações no código-fonte do sistema.

Dessa forma, a solução proposta oferece uma abordagem moderna e eficiente para a precificação eletrônica no varejo, integrando *hardware* de baixo consumo, tecnologia de tinta eletrônica e comunicação segura via *HTTPS*. O resultado é um sistema modular, escalável e sustentável, capaz de reduzir custos operacionais, minimizar erros humanos e tornar o processo de atualização de preços mais ágil e confiável.

3.2 Diagramas de Caso de Uso

Esta seção apresenta os diagramas de caso de uso do sistema desenvolvido, abrangendo todos os módulos que compõem a solução proposta para gerenciamento de etiquetas eletrônicas baseadas em microcontroladores com *display e-ink*, aplicadas ao ambiente varejista. Os diagramas

ilustram os principais atores envolvidos e suas interações com as funcionalidades do sistema, permitindo uma visão orientada a requisitos funcionais.

O sistema organiza-se em quatro módulos principais: (i) Gerenciamento de *Tags IoT*; (ii) Gerenciamento de Produtos; (iii) Gerenciamento de Categorias; e (iv) Gerenciamento de Corredores. Cada módulo é apresentado a seguir com seu respectivo conjunto de casos de uso.

3.2.1 Módulo de Gerenciamento de *Tags IoT*

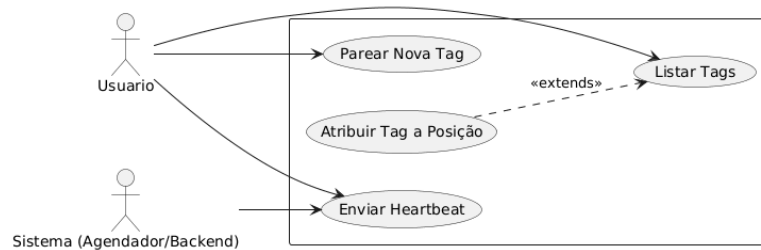
O módulo de Gerenciamento de *Tags IoT* contempla operações relacionadas aos dispositivos físicos instalados nas prateleiras — as *tags* eletrônicas. Essas *tags* se comunicam diretamente com o sistema via *HTTP*, possibilitando pareamento automático, atualização de conteúdo e envio periódico de informações de status (*heartbeat*).

Os casos de uso deste módulo são:

- **UC-01: Parear Nova *Tag*** — Registra um novo dispositivo no sistema, identificando-o por endereço *IP* e *MAC*.
- **UC-02: Listar *Tags* Não Atribuídas** — Exibe todas as *tags* pareadas que ainda não foram vinculadas a posições de prateleira.
- **UC-03: Atribuir *Tag* a Posição** — Permite associar uma *tag* a uma posição específica; este caso de uso **estende** o UC-02, pois depende da listagem prévia de *tags* não atribuídas.
- **UC-04: Enviar *Heartbeat*** — Caso de uso interno do sistema responsável por verificar periodicamente a conectividade das *tags*.

A Figura 3.2 apresenta o diagrama de caso de uso deste módulo.

Figura 3.2: Diagrama de Caso de Uso do Módulo de Gerenciamento de *Tags IoT*. Fonte: (Própria, 2025).



3.2.2 Módulo de Gerenciamento de Produtos

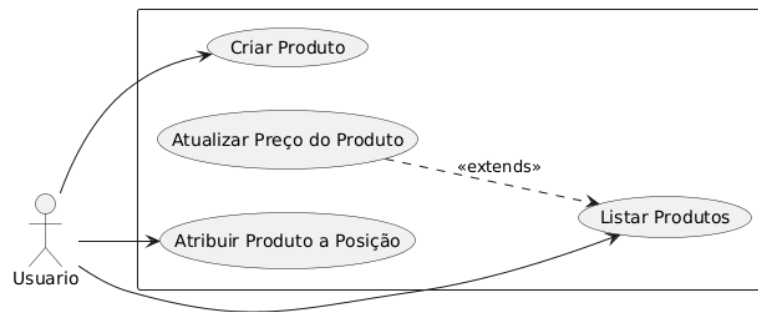
Este módulo centraliza todas as operações referentes aos produtos comercializados na loja. Além do cadastro e consulta, o módulo contempla a atualização automática das informações exibidas nas *tags* sempre que um preço é alterado.

Os casos de uso envolvidos são:

- **UC-05: Criar Produto** — Cadastra um novo produto com nome, categoria e preço.
- **UC-06: Listar Produtos** — Exibe todos os produtos registrados no sistema.
- **UC-07: Consultar Produto** — Obtém informações detalhadas de um produto específico.
- **UC-08: Atualizar Preço do Produto** — Modifica somente o preço e propaga a atualização às *tags* associadas.
- **UC-09: Atribuir Produto a Posição** — Relaciona um produto a uma posição física de prateleira, validando o tamanho do corredor.

O diagrama deste módulo é apresentado na Figura 3.3.

Figura 3.3: Diagrama de Caso de Uso do Módulo de Gerenciamento de Produtos. Fonte: (Própria, 2025).



3.2.3 Módulo de Gerenciamento de Categorias

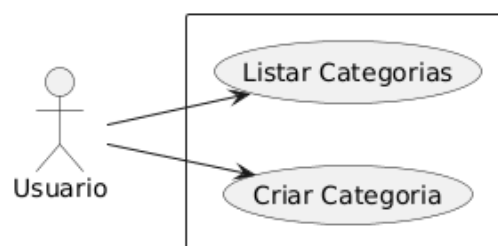
O módulo de categorias fornece suporte à classificação e organização dos produtos. Embora simples, é fundamental para estruturar semanticamente o catálogo de produtos.

Os casos de uso são:

- **UC-10: Criar Categoria** — Registra uma nova categoria garantindo unicidade do nome.
- **UC-11: Listar Categorias** — Recupera todas as categorias cadastradas.

O diagrama deste módulo encontra-se na Figura 3.4.

Figura 3.4: Diagrama de Caso de Uso do Módulo de Gerenciamento de Categorias. Fonte: (Própria, 2025).



3.2.4 Módulo de Gerenciamento de Corredores

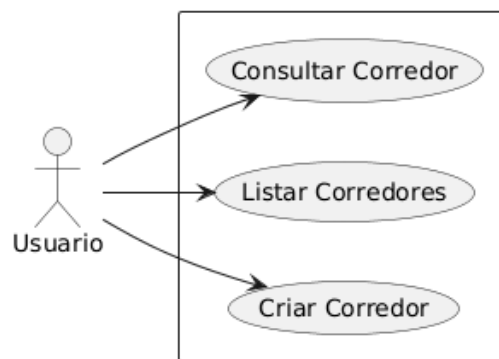
Este módulo trata da definição da estrutura física da loja, permitindo criar e consultar os corredores que organizam fisicamente as posições de prateleira e suas respectivas associações com produtos e *tags*.

Os casos de uso são:

- **UC-12: Criar Corredor** — Registra um novo corredor com número definido de posições.
- **UC-13: Listar Corredores** — Exibe todos os corredores cadastrados.
- **UC-14: Consultar Corredor** — Retorna informações específicas sobre um corredor como tamanho, produtos e *tags* associadas em cada posição.

A Figura 3.5 apresenta o diagrama desse módulo.

Figura 3.5: Diagrama de Caso de Uso do Módulo de Gerenciamento de Corredores. Fonte: (Própria, 2025).



3.3 Modelo de Banco de Dados

O banco de dados do sistema foi projetado de forma a refletir diretamente os principais elementos envolvidos na operação das etiquetas eletrônicas e na organização dos produtos no ambiente varejista. O modelo adota uma estrutura relacional composta por seis tabelas principais: *tag*, *heartbeat*, *product*, *category*, *shelf_pos* e *shelf_corridor*. Cada tabela desempenha um papel funcional específico, garantindo integridade referencial e suporte às operações essenciais do sistema.

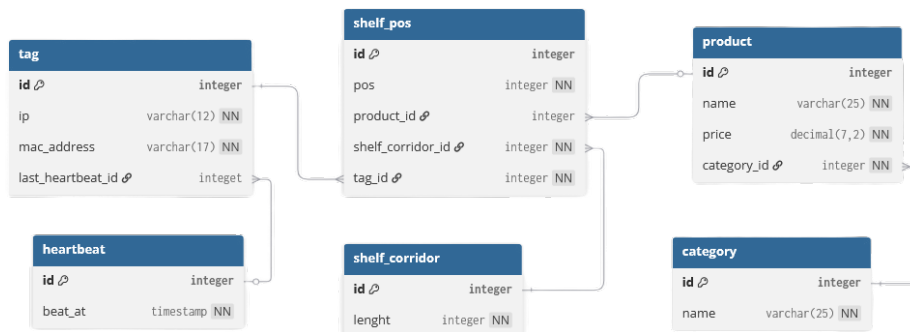
A tabela *tag* registra as informações de cada dispositivo instalado na prateleira, incluindo endereço IP, endereço MAC e referência ao último *heartbeat* recebido. A tabela *heartbeat*, por sua vez, armazena os registros de conectividade enviados pelas tags, permitindo monitorar o estado dos dispositivos em tempo real.

O módulo de produtos é representado pelas tabelas *product* e *category*. A primeira armazena nome, preço e categoria de cada item, enquanto a segunda mantém a relação de categorias existentes. A integridade é garantida por chave estrangeira que vincula cada produto à sua categoria correspondente.

A estrutura física da loja é descrita pelas tabelas *shelf_corridor* e *shelf_pos*. A tabela *shelf_corridor* contém a definição de cada corredor e seu comprimento (número de posições). Já a tabela *shelf_pos* relaciona posições específicas a produtos e tags, estabelecendo a ligação entre o ambiente físico e os dispositivos eletrônicos instalados.

O conjunto de chaves estrangeiras define as relações entre tags, produtos e estrutura física, permitindo ao sistema atualizar automaticamente o conteúdo exibido nas etiquetas conforme alterações de preço, posição ou disponibilidade de produtos. Essa modelagem garante consistência, facilita consultas e otimiza o desempenho das operações de gerenciamento no sistema.

Figura 3.6: Modelo Entidade Relacionamento da Solução. Fonte: (Própria, 2025).



3.4 Projeto do Hardware

A descrição do *hardware* do sistema de etiquetagem eletrônica consiste na definição e concepção do arranjo dos componentes responsáveis por realizar a exibição dinâmica das informações de preço e produto em ambientes varejistas. O conjunto foi projetado para garantir baixo consumo de energia, conectividade de rede estável e facilidade de integração com a plataforma *web* utilizada para o gerenciamento remoto dos dados.

Nesta seção, são apresentados os principais componentes utilizados na implementação do protótipo, bem como suas funções específicas dentro da composição da solução proposta:

- **ESP32 DevKit V1:** Trata-se do *microcontrolador principal* do sistema, responsável pelo processamento das informações recebidas via rede e pela comunicação com o módulo de exibição. O *ESP32* possui conectividade *Wi-Fi* integrada, eliminando a necessidade de módulos adicionais de comunicação, além de oferecer *duplo núcleo de processamento*, *memória suficiente* para execução de tarefas simultâneas e *baixo consumo energético*. No contexto do projeto, o *ESP32* atua como a *unidade central de controle*, responsável por receber os dados enviados pela plataforma *web* via requisições *HTTPS*, processá-los e atualizá-los no *display e-ink*.
- **Módulo de *Display e-Ink* – *WeAct Studio 2.7* (GDEH027W3):** Este componente é responsável pela *exibição das informações de preço e identificação do produto*. A escolha da tecnologia de tinta eletrônica justifica-se por sua *excelente legibilidade*, mesmo sob iluminação intensa, e por sua capacidade de manter o conteúdo exibido sem consumo contínuo de energia, o que é fundamental em aplicações de varejo onde os preços são alterados com pouca frequência. O módulo *WeAct Studio* é compatível com o controlador *IL91874*, amplamente utilizado em telas *e-paper*, e comunica-se com o microcontrolador por meio do protocolo *SPI* (*Serial Peripheral Interface*).

A escolha do microcontrolador *ESP32 DevKit V1* foi motivada por sua integração nativa com *Wi-Fi*, alto desempenho e versatilidade para aplicações *IoT*, características que o tornam mais adequado que microcontroladores como o *Arduino Uno*, que necessitaria de módulos de rede externos, ou o *Raspberry Pi*, cujo custo e capacidade de processamento seriam desproporcionais às demandas do sistema. O *ESP32*, portanto, oferece o melhor equilíbrio entre desempenho, custo e eficiência energética, atendendo plenamente aos requisitos de conectividade e controle deste projeto.

O conjunto de componentes descrito foi organizado de forma a permitir facilidade de montagem e escalabilidade, possibilitando que múltiplos módulos possam operar simultaneamente em um mesmo ambiente, cada um exibindo informações específicas recebidas da plataforma *web*.

Após a definição dos elementos que compõem o projeto de *hardware*, foi elaborado o *mapeamento de conexões físicas* entre o microcontrolador *ESP32* e o módulo de *display e-ink*, conforme apresentado na Tabela 3.1. Essa tabela representa a correspondência entre os pinos de ambos os dispositivos, destacando o canal de comunicação utilizado para a interface *SPI*.

Tabela 3.1: Mapeamento de conexões entre o módulo *e-ink WeAct Studio 2.7"* e o microcontrolador *ESP32 DevKit V1*.

Sinal do Módulo <i>E-Ink</i>	Descrição	Pino Correspondente no <i>ESP32</i>
VCC	Alimentação (3,3V)	3V3
BUSY	Status de operação do <i>display</i>	GPIO14
RES	Reinicialização (<i>reset</i>) do módulo	GPIO12
D/C	Controle de dados/comando	GPIO25
CS	<i>Chip Select</i> (seleção do dispositivo <i>SPI</i>)	GPIO21
SCL	<i>Clock</i> do barramento <i>SPI</i>	GPIO16
SDA	Dados do barramento <i>SPI</i> (<i>MOSI</i>)	GPIO17

A Tabela 3.1 demonstra que o módulo *e-ink* opera em nível lógico de 3,3 V, o que permite a conexão direta com o *ESP32* sem a necessidade de reguladores adicionais. O barramento *SPI* foi escolhido por sua alta velocidade de transmissão e confiabilidade, tornando-se a interface ideal para comunicação com o *display* de tinta eletrônica.

A fim de validar o funcionamento do *hardware* proposto, foi montado um protótipo físico em *proto-board*, integrando o microcontrolador *ESP32 DevKit V1* ao módulo de tinta eletrônica *WeAct Studio* de 2.7". Essa montagem permitiu a realização dos testes de comunicação *SPI*, alimentação e atualização de conteúdo no *display*, conforme ilustrado na Figura 3.7.

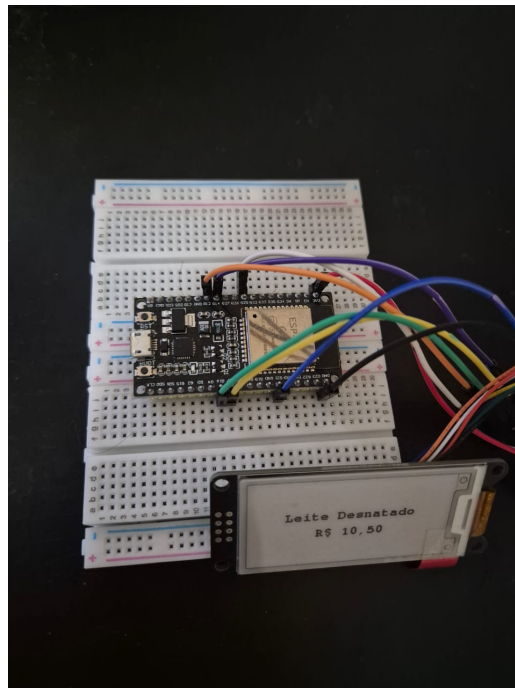


Figura 3.7: Protótipo do sistema de etiquetagem eletrônica: conexão entre o ESP32 DevKit V1 e o módulo e-ink WeAct Studio 2.7" montados em protoboard. Fonte: (Própria, 2025).

A Tabela 3.2 apresenta um levantamento dos componentes utilizados na implementação do protótipo, incluindo o microcontrolador *ESP32*, o módulo de *display e-ink*, as *protoboards* e os *jumpers* empregados na montagem do circuito. Além de caracterizar a composição física da solução, esse demonstrativo evidencia o baixo custo de aquisição dos elementos necessários, reforçando a viabilidade econômica do sistema proposto. O valor total investido na construção do protótipo confirma que é possível desenvolver uma etiqueta eletrônica funcional com tecnologia de baixo consumo energético utilizando componentes amplamente acessíveis no mercado, aspecto que contribui para a sustentabilidade e escalabilidade da solução em cenários reais de aplicação.

Tabela 3.2: Custos dos componentes utilizados no protótipo.

Quantidade	Descrição do Componente	Custo Unitário (R\$)
1	Microcontrolador <i>ESP32 DevKit V1</i>	40,00
1	<i>Display e-ink 2.7"</i> (<i>WeAct</i>)	30,00
2	<i>Protoboard</i> 400 furos	15,00
12	<i>Jumpers</i> (macho-fêmea / macho-macho)	8,00
Total		108,00

3.5 Projeto de Software

3.5.1 Software Embarcado no Módulo de Exibição

O *software* embarcado no módulo de exibição foi desenvolvido com o propósito de estabelecer a comunicação entre o microcontrolador *ESP32 DevKit V1*, o módulo de tinta eletrônica *WeAct Studio 2.7"* e a plataforma *web* administrativa responsável pelo gerenciamento centralizado das informações de preço e produto. Sua função é garantir que cada módulo opere de forma autônoma e sincronizada, recebendo dados atualizados da aplicação *web* e exibindo-os no *display* de maneira confiável e energeticamente eficiente.

O desenvolvimento do *software* foi realizado no *Ambiente de Desenvolvimento Integrado Arduino IDE*, escolhido por sua ampla compatibilidade com a plataforma *ESP32*, suporte a bibliotecas de terceiros e facilidade de depuração durante o processo de prototipagem. O *firmware* foi estruturado de modo modular, contendo rotinas específicas para inicialização do sistema, conexão à rede sem fio, comunicação *HTTPs*, tratamento de dados em formato *JSON* e renderização das informações no *display e-ink*.

A seguir, são descritas as principais bibliotecas utilizadas e suas respectivas funções no contexto do projeto:

- ***WiFi.h***: Gerencia a conexão do microcontrolador *ESP32* à rede sem fio, fornecendo funções para autenticação, reconexão automática e verificação do estado da conexão;
- ***HTTPClient.h***: Implementa os métodos de envio e recebimento de requisições *HTTPs*.

É utilizada para que o microcontrolador receba as informações de preço e nome de produto diretamente do servidor da aplicação *web* e envie respostas de confirmação de atualização;

- ***ArduinoJson.h***: Responsável pela leitura, interpretação e formatação de mensagens no padrão *JSON*. Essa biblioteca é empregada na decodificação das requisições recebidas e na construção de respostas para o servidor;
- ***GxEPD2.h***: Biblioteca dedicada ao controle de *displays* de tinta eletrônica, permitindo a inicialização do módulo *WeAct Studio 2.7*”, atualização de tela, posicionamento de texto e renderização de caracteres com diferentes fontes;
- ***SPI.h***: Define o protocolo de comunicação serial síncrona utilizado entre o *ESP32* e o módulo de exibição. Essa biblioteca assegura a transferência confiável dos dados de imagem e comandos de controle entre os dispositivos;
- ***FS.h* e *SPIFFS.h***: Permitem o gerenciamento do sistema de arquivos interno do *ESP32*, possibilitando o armazenamento de informações locais, como o último preço exibido, garantindo que o *display* mantenha dados válidos mesmo em situações de reinicialização.

Após a inicialização, o microcontrolador configura as portas de comunicação *SPI*, estabelece a conexão com a rede *Wi-Fi* e inicia a comunicação com o servidor da aplicação *web*. Uma das principais rotinas do *firmware* consiste no processo de *pareamento periódico*, no qual o dispositivo envia requisições *HTTP* para o servidor, transmitindo seu endereço *IP* e identificador *MAC*. Esse mecanismo permite que a plataforma *web* identifique e registre dinamicamente todos os módulos de exibição conectados à rede, assegurando sua rastreabilidade e gerenciamento individual.

Além do pareamento, o *software* embarcado configura um *servidor HTTP local* que disponibiliza rotas específicas para comunicação direta com a aplicação *web*. Entre essas rotas destacam-se: a rota de verificação de integridade, que responde às requisições de *health check* confirmando o funcionamento do módulo; e a rota de atualização de conteúdo, que recebe e processa os dados de produto e preço enviados pela aplicação *web*. Os dados são transmiti-

dos em formato *JSON* e, ao serem recebidos, são interpretados e exibidos no *display e-ink*, substituindo o conteúdo anterior.

Durante o processo de atualização, o sistema converte os valores monetários recebidos para o formato brasileiro e organiza as informações na tela de modo centralizado, assegurando boa legibilidade e estética adequada ao ambiente de varejo. Após a atualização, o microcontrolador envia uma resposta *HTTP* ao servidor, confirmando a conclusão bem-sucedida da operação. Esse mecanismo de confirmação garante a integridade da comunicação e a sincronização das informações exibidas entre todos os módulos do sistema.

O *firmware* mantém a execução contínua dessas rotinas, alternando entre o envio periódico de mensagens de *pareamento* e o atendimento das requisições recebidas da plataforma *web*. Essa operação simultânea permite que o sistema mantenha conectividade estável e atualização constante, mesmo com múltiplos dispositivos operando em paralelo. Caso ocorra perda de conexão, o módulo continua exibindo o último conteúdo recebido, aproveitando a característica do *display* de tinta eletrônica de manter a imagem mesmo sem alimentação elétrica.

De forma geral, o *software* embarcado atua como o elo entre o sistema físico e a aplicação *web*, integrando conectividade *Wi-Fi*, comunicação via protocolo *HTTP* e exibição gráfica em tinta eletrônica.

3.5.2 Aplicação Web

A aplicação *web*, denominada *SyncShelf*, foi desenvolvida com o objetivo de gerenciar os dispositivos *IoT* utilizados no sistema de etiquetagem eletrônica, permitindo o controle centralizado e a atualização dinâmica das informações exibidas nos módulos de tinta eletrônica. Essa aplicação constitui o núcleo lógico e administrativo da solução, sendo responsável por armazenar, processar e propagar os dados de produtos e suas respectivas associações com as *tags* de exibição.

A aplicação *web* foi construída sobre uma base de tecnologias consolidadas, incluindo o *Angular*, utilizado na camada de apresentação para o desenvolvimento da interface administrativa do sistema, proporcionando uma experiência interativa e responsiva ao usuário. No *backend*,

empregou-se o *Spring Boot* para o gerenciamento de dependências e configuração da aplicação, o *Spring Data JPA* para persistência de dados, o *Flyway* para controle e migração de versões do banco de dados, e o *Docker Compose* para orquestração dos ambientes de desenvolvimento e produção. O banco de dados utilizado é o *Microsoft SQL Server*, responsável por armazenar as entidades de domínio apresentadas na seção onde o modelo de entidade foi discutido. A arquitetura adotada segue o modelo em camadas, promovendo separação de responsabilidades e facilitando a manutenção e evolução do sistema. As camadas principais são:

- **Camada de Apresentação (*Controller*):** Responsável por receber as requisições *HTTP* e direcioná-las às regras de negócio adequadas, implementando os *endpoints REST* da *API*;
- **Camada de Negócio (*Service*):** Contém a lógica central da aplicação, incluindo as validações e regras de negócio referentes ao gerenciamento de *tags*, produtos e posições de prateleira;
- **Camada de Persistência (*Repository*):** Realiza a abstração de acesso ao banco de dados por meio do *Spring Data JPA*, utilizando a especificação do *Hibernate* para operações de leitura e escrita;
- **Camada de Modelo:** Define as entidades de domínio e os objetos de transferência de dados utilizados na comunicação entre as camadas internas e externas da aplicação;
- **Camada de Integração:** Responsável pela comunicação com os dispositivos físicos, realizando chamadas *HTTP* diretas para atualização de conteúdo e verificação de conectividade.

O sistema implementa comunicação bidirecional com os módulos embarcados utilizando o protocolo *HTTP*, o que permite tanto o envio de requisições para atualização dos dados exibidos quanto o recebimento de respostas de confirmação e *heartbeats* de monitoramento. Durante o processo de pareamento, o sistema utiliza o protocolo *Address Resolution Protocol (ARP)* para identificar automaticamente as configurações de rede de cada *Tag IoT*, obtendo seu endereço

físico (*MAC address*) a partir do endereço *IP* informado na requisição inicial. Essa abordagem simplifica a integração com os microcontroladores, evitando dependências de intermediários e assegurando um fluxo direto e transparente entre o servidor e os dispositivos.

A partir do endereço *IP* de cada *tag* salvo no banco de dados, a aplicação constrói dinamicamente as *URLs* utilizadas na comunicação com os módulos embarcados. As requisições do tipo *POST* são empregadas para atualização das informações exibidas, como nome e preço do produto, enquanto o método *GET* é utilizado para consultas de disponibilidade e verificação de *health check*. Essa estratégia garante a sincronização contínua entre o sistema *web* e os dispositivos *IoT*, permitindo o monitoramento remoto do estado operacional de cada *tag* e a atualização em *tempo real* das etiquetas eletrônicas.

Para representar o fluxo das principais interações do sistema, foram elaborados diagramas de sequência *UML* que descrevem os casos de uso essenciais que compõem o núcleo funcional da aplicação. Tais diagramas, apresentados nas Figuras 3.8 a 3.12, ilustram as trocas de mensagens entre os componentes lógicos do sistema e os dispositivos *IoT*. Entre os principais casos de uso modelados, destacam-se:

- **Pareamento de *Tag IoT*:** Processo de registro inicial de um dispositivo no sistema, no qual o servidor identifica automaticamente o endereço *IP* e o *MAC address* da *tag* por meio da tabela *ARP* da rede;
- **Atribuição de *Tag* a Posição de Prateleira:** Associação de uma *tag* a uma posição física dentro de um corredor, com validação de limites e atualização automática do conteúdo da *tag*;
- **Atribuição de Produto a Posição:** Associação de um produto a uma posição de prateleira, propagando as informações correspondentes à *tag* vinculada, caso existente;
- **Atualização de Preço de Produto:** Alteração do valor de um produto, com atualização automática em todas as *tags* associadas a ele;
- **Envio de *Heartbeat*:** Verificação periódica da conectividade das *tags* cadastradas, registrando no banco de dados o *status* de cada dispositivo.

A Figura 3.8 exemplifica o diagrama de sequência correspondente ao caso de uso de pareamento de *tags IoT*, que representa o primeiro ponto de interação entre os módulos físicos e o sistema *web*, enquanto as Figuras 3.9, 3.10 e 3.11 demonstram, respectivamente, o fluxo de atribuição de *tags*, produtos e atualização de preços. Por fim, a Figura 3.12 ilustra o processo de monitoramento de *heartbeat*, responsável por garantir a rastreabilidade e o controle de disponibilidade dos dispositivos conectados.

Figura 3.8: Caso de Uso 01: Pareamento de Nova *Tag*. Fonte: (Própria, 2025).

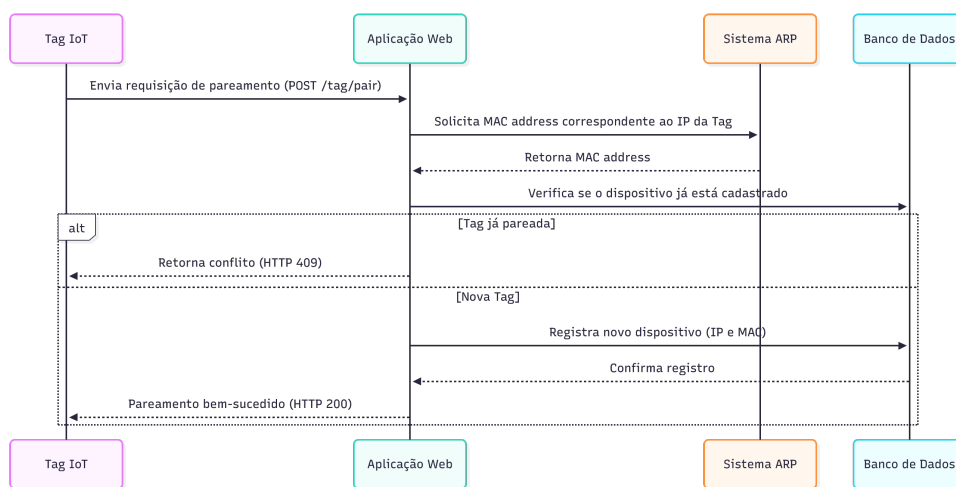


Figura 3.9: Caso de Uso 03: Atribuir *Tag* a Posição. Fonte: (Própria, 2025).

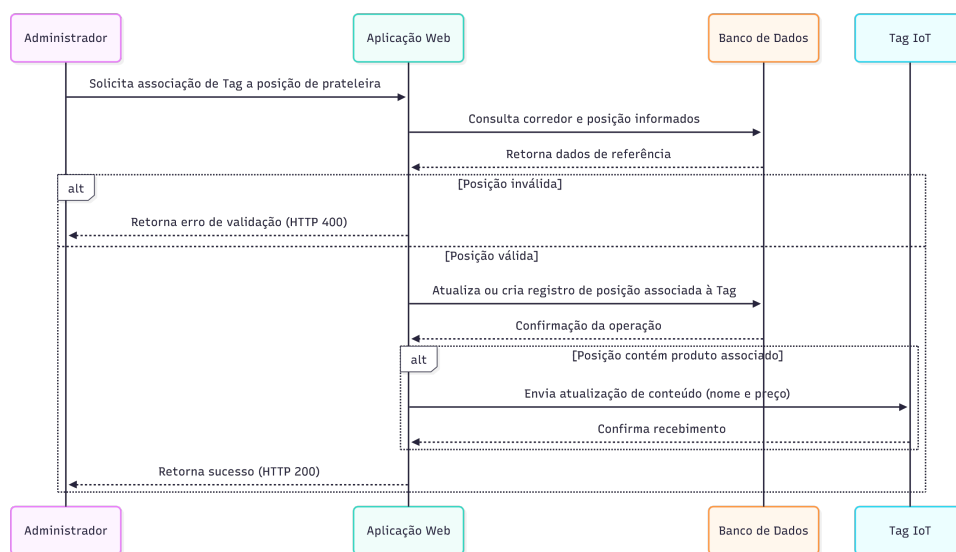


Figura 3.10: Caso de Uso 09: Atribuir Produto a Posição. Fonte: (Própria, 2025).

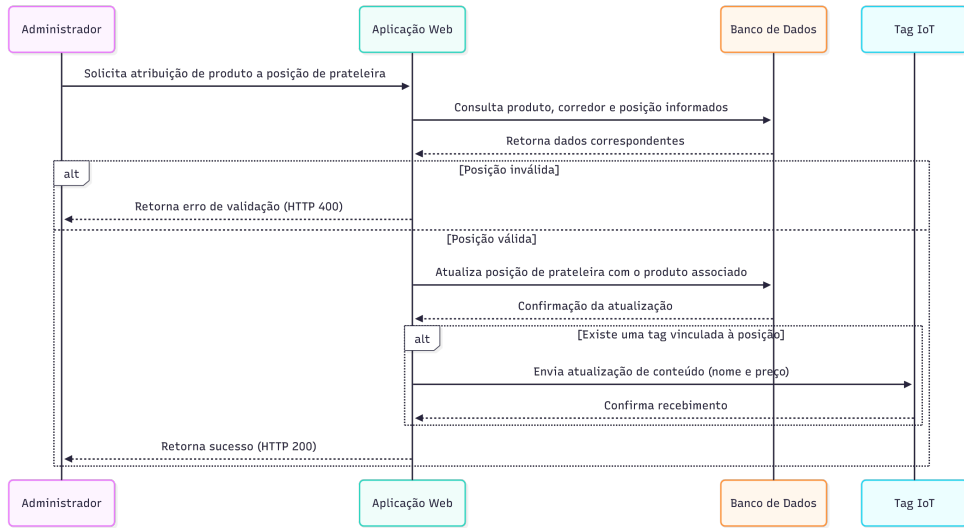


Figura 3.11: Caso de Uso 08: Atualizar Preço do Produto. Fonte: (Própria, 2025).

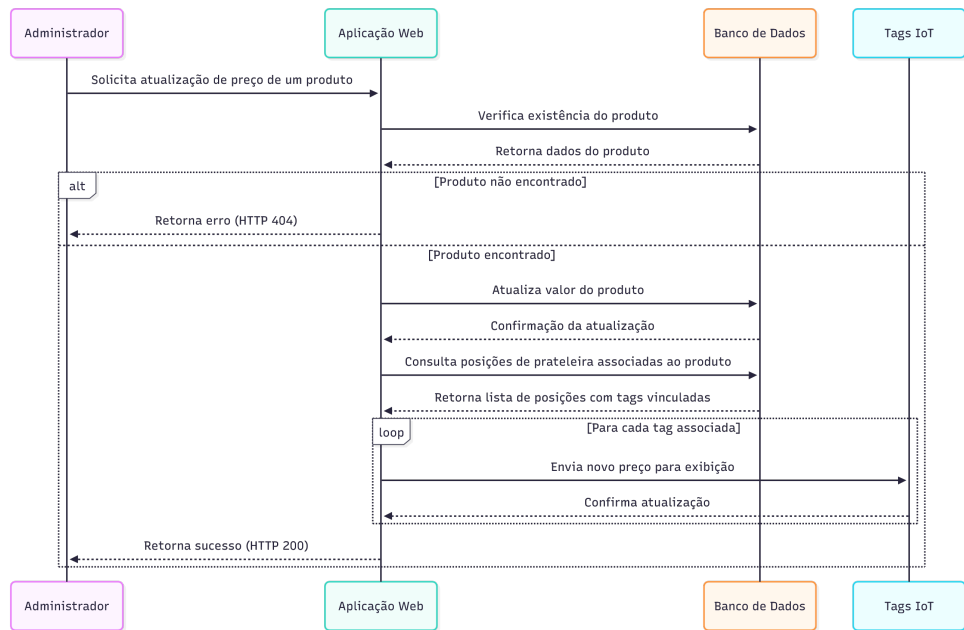
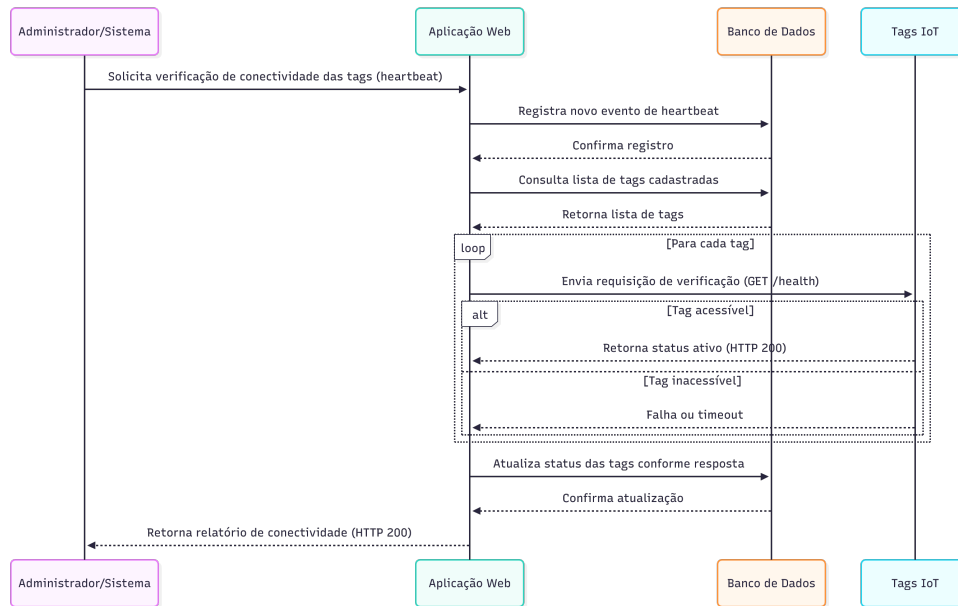


Figura 3.12: Caso de Uso 04: Enviar *Heartbeat*. Fonte: (Própria, 2025).



De maneira geral, a aplicação *SyncShelf* constitui o componente central da solução, atuando como intermediário entre a camada física e o gerenciamento lógico da informação. Sua arquitetura segue o modelo *Model-View-Controller*, que promove a separação de responsabilidades entre as camadas de apresentação, lógica de negócio e persistência de dados. Essa estrutura modular proporciona maior escalabilidade, facilita a manutenção e favorece o desacoplamento entre os componentes do sistema, permitindo a integração com diferentes tipos de *hardware IoT* e assegurando a evolução contínua da solução em contextos de varejo inteligente.

Capítulo 4

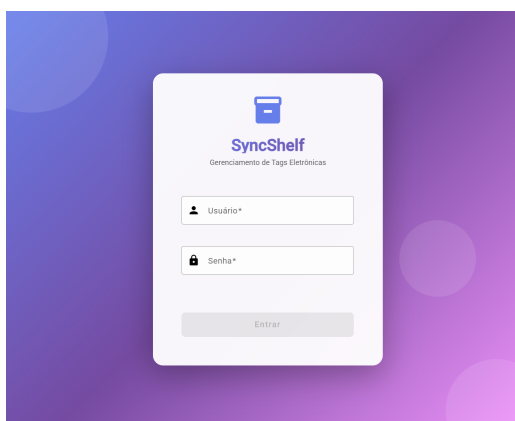
Resultados

4.1 Implementação do SyncShelf

A aplicação *web* segue uma arquitetura modular, organizada em componentes que contemplam autenticação, cadastro de entidades e gerenciamento das associações entre dispositivos e produtos. Cada um desses elementos foi documentado visualmente nas interfaces apresentadas ao longo deste trabalho.

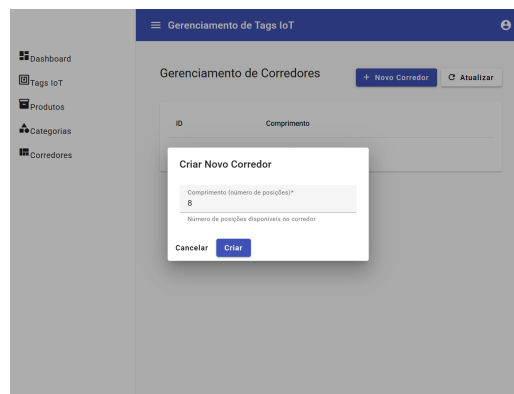
A interação com o sistema tem início na tela de *login*, ilustrada na Figura 4.1. Nessa etapa, são realizadas as validações de credenciais e o estabelecimento de sessão segura, garantindo que apenas usuários autorizados tenham acesso aos módulos administrativos. Esse mecanismo assegura a integridade das informações e a proteção das operações realizadas na plataforma.

Figura 4.1: Tela de *Login* do *SyncShelf*. Fonte: (Própria, 2025).



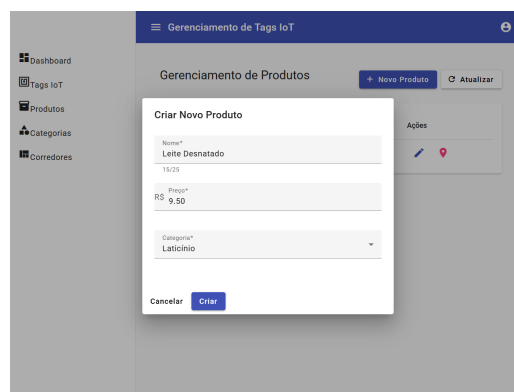
Após a autenticação, o usuário é direcionado ao ambiente administrativo, onde pode acessar o módulo de criação e gerenciamento de corredores, cuja interface está apresentada na Figura 4.2. Nesse módulo, é possível estruturar digitalmente a organização física do ambiente varejista, definindo o conjunto de corredores e suas respectivas posições de prateleira. Essa modelagem estabelece a base para o mapeamento posterior das etiquetas *IoT*.

Figura 4.2: Tela do Caso de Uso *UC-012*: Criação de Corredor. Fonte: (Própria, 2025).



Complementarmente, a plataforma oferece a seção de cadastro de produtos, representada na Figura 4.3. Nessa interface, o administrador pode registrar itens com nome, categoria e preço, permitindo que cada produto seja posteriormente vinculado a uma *tag* eletrônica. Esse cadastro centralizado assegura consistência nas informações exibidas nos dispositivos *e-ink*.

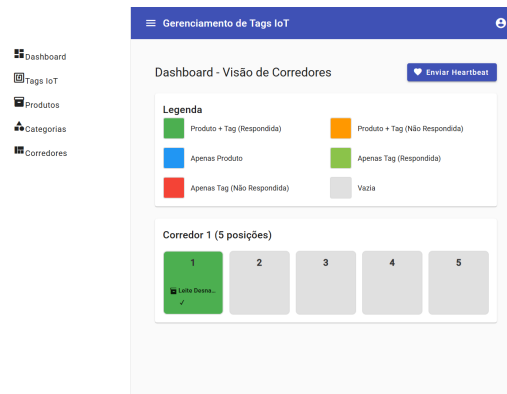
Figura 4.3: Tela do Caso de Uso *UC-05*: Criação de Produto. Fonte: (Própria, 2025).



A operacionalização dessas funcionalidades converge no módulo de *dashboard*, mostrado na Figura 4.4. Esse painel funciona como o núcleo de controle do sistema, proporcionando uma visão clara da organização dos corredores e permitindo associar cada *tag IoT* a uma posição

específica e ao produto correspondente. É nessa interface que ocorre o mapeamento efetivo entre o ambiente físico e o modelo lógico definido no sistema.

Figura 4.4: Tela de *Dashboard* do *SyncShelf*. Fonte: (Própria, 2025).



O *dashboard* também possibilita o envio de comandos de atualização às *tags*, desencadeando o processo de comunicação entre o *backend* e o módulo embarcado. Ao receber os dados enviados pela plataforma *web*, o microcontrolador renderiza de forma eficiente, no *display e-ink*, o nome, o preço e demais informações do produto. A biestabilidade da tecnologia *e-ink* garante baixo consumo energético e estabilidade visual, refletindo fielmente os dados mantidos na aplicação administrativa.

A integração entre os módulos — autenticação, criação de corredores, cadastro de produtos e *dashboard* de associações — consolidou a construção de um ecossistema funcional e coerente. O resultado é uma solução que automatiza a manutenção das informações exibidas nas etiquetas eletrônicas e estabelece uma base sólida e extensível para aprimoramentos futuros, como monitoramento em *tempo real*, análise de desempenho ou incorporação de novos padrões de comunicação *IoT*.

4.2 Análise dos Resultados

Os resultados obtidos neste trabalho referem-se à validação funcional do protótipo desenvolvido, cujo objetivo foi verificar se os componentes de hardware, software e comunicação integraram-se corretamente para executar as operações essenciais de atualização remota de etiquetas eletrô-

nicas. Assim, a validação concentrou-se no correto funcionamento dos fluxos implementados — pareamento de dispositivos, recepção e processamento de comandos, atualização do *display* e sincronização com a plataforma *web* — em um ambiente controlado de testes.

É importante destacar que não foram realizadas análises estatísticas, ensaios de desempenho quantitativo ou experimentos em larga escala, pois tais procedimentos extrapolam o escopo definido para este trabalho. A validação aqui apresentada possui caráter qualitativo e demonstrativo, voltado exclusivamente a comprovar a viabilidade técnica da solução proposta. Ainda assim, para complementar a análise do comportamento operacional, foram realizadas medições observacionais dos tempos de execução de três casos de uso considerados essenciais: o pareamento de nova *tag* (UC-01), a atribuição de produto à posição (UC-09) e a atualização de preço no *display e-ink* (UC-08). Os resultados das Tabelas 4.1, 4.2 e 4.3 demonstram que o sistema apresentou tempos estáveis e consistentes entre as execuções, variando em torno de 5,09s para o pareamento, 3,12s para a atribuição de produto e 4,34s para a atualização de preço. Esses valores, embora empregados apenas como indicadores qualitativos de responsividade, reforçam a adequação do protótipo às operações esperadas e evidenciam que os fluxos principais ocorrem sem atrasos significativos ou comportamentos irregulares.

Tabela 4.1: Tempo de Execução – UC-01: Parear Nova Tag

Tentativa	Tempo (s)
1	5.12
2	4.98
3	5.31
4	5.07
5	4.89
6	5.22
7	5.15
8	5.08
9	4.95
10	5.18
Mínimo	4.89
Máximo	5.31
Média	5.09

Tabela 4.2: Tempo de Execução – UC-09: Atribuir Produto à Posição

Tentativa	Tempo (s)
1	3.12
2	3.05
3	3.21
4	3.18
5	2.98
6	3.14
7	3.09
8	3.22
9	3.11
10	3.07
Mínimo	2.98
Máximo	3.22
Média	3.12

Tabela 4.3: Tempo de Execução – UC-08: Atualizar Preço do Produto

Tentativa	Tempo (s)
1	4.42
2	4.18
3	4.27
4	4.51
5	4.39
6	4.29
7	4.34
8	4.22
9	4.41
10	4.36
Mínimo	4.18
Máximo	4.51
Média	4.34

Dentro desse propósito, o protótipo mostrou-se plenamente funcional: os módulos embarcados responderam de forma consistente aos comandos enviados pelo *backend*, os *displays e-ink* exibiram as informações de maneira estável e legível, e a plataforma *web* permitiu gerenciar produtos, categorias, corredores e posições de forma centralizada. A integração entre o *backend*, a aplicação *web* e o dispositivo físico demonstrou coerência arquitetural e operação harmoniosa entre os componentes.

O sistema desenvolvido contemplou todas as funcionalidades previstas, incluindo controle remoto das etiquetas, organização lógica da estrutura da loja e atualização automática das informações exibidas nas *tags*. Estudos recentes, como o trabalho de (LE et al., 2023), demonstram que etiquetas eletrônicas baseadas em *e-paper* podem alcançar autonomia superior a cinco anos e manter alta legibilidade em diferentes condições de iluminação, reforçando a adequação dessa tecnologia para ambientes varejistas onde longos períodos de exibição estática são predominantes. Esses achados corroboram os resultados funcionais observados neste protótipo, evidenciando que o uso de *e-ink* atende de maneira eficaz às demandas operacionais típicas de lojas físicas.

De forma geral, os resultados obtidos confirmam que a solução atende aos objetivos definidos no início do trabalho, oferecendo um sistema de etiquetagem eletrônica centralizado, escalável e energeticamente eficiente, fundamentado em princípios tecnológicos validados na literatura. O estudo de (LE et al., 2023), ao demonstrar baixo consumo energético, comunicação confiável e robustez do *display* em cenários reais de varejo, reforça a pertinência da abordagem adotada neste projeto, que também se mostrou funcional dentro das condições avaliadas. Assim, o protótipo desenvolvido estabelece uma base sólida para futuras evoluções, incluindo testes quantitativos, expansão em maior escala e aplicação prática em supermercados e centros de distribuição.

Capítulo 5

Considerações Finais

O desenvolvimento deste trabalho permitiu demonstrar a viabilidade técnica de um sistema de etiquetagem eletrônica baseado em tecnologia *e-ink*, integrado a uma plataforma *web* administrativa e a um módulo embarcado responsável pela exibição das informações de preço. A solução proposta cumpriu seu objetivo central de validar, de forma funcional, a operação coordenada entre hardware, software e comunicação, permitindo a atualização remota e confiável das etiquetas em um ambiente varejista simulado.

A validação realizada, de caráter qualitativo, evidenciou que o protótipo opera de maneira consistente nos fluxos essenciais do sistema: pareamento dos dispositivos, recepção e processamento de comandos, renderização gráfica no *display* e sincronização contínua com o *backend*. Embora não tenham sido conduzidas análises estatísticas ou testes de desempenho em larga escala, os experimentos realizados em ambiente controlado foram suficientes para confirmar a robustez arquitetural da solução desenvolvida e seu alinhamento com os requisitos funcionais inicialmente estabelecidos.

Os resultados obtidos reforçam ainda a adequação do uso da tecnologia *e-ink* em cenários de varejo físico, especialmente em razão do seu consumo energético reduzido, uma vez que requer energia apenas no momento da atualização das informações. Esse comportamento, observado no protótipo, está em consonância com achados da literatura, como aponta o estudo de (LE et al., 2023), que demonstra a capacidade de etiquetas eletrônicas baseadas em *e-paper* alcançarem elevada autonomia e manterem excelente legibilidade em diferentes condições de iluminação.

Assim, o presente trabalho contribui ao consolidar evidências experimentais que sustentam a viabilidade do uso dessa tecnologia em aplicações reais de precificação automatizada.

As contribuições alcançadas incluem: a implementação integral de um sistema funcional de etiquetagem eletrônica; a validação dos fluxos de comunicação entre plataforma *web* e dispositivos físicos; e a demonstração da aplicabilidade de *displays e-ink* em operações remotas de atualização de preços. Tais resultados configuram um avanço relevante para soluções de baixo consumo energético e gerenciamento centralizado, além de abrirem espaço para o desenvolvimento de aplicações mais amplas em ambientes varejistas.

Como limitações, destaca-se o escopo reduzido da avaliação, que não contemplou testes de desempenho em escala, análise de confiabilidade sob carga elevada ou experimentações prolongadas de autonomia energética em condições reais de uso. Além disso, abre-se espaço para a realização de avaliações com usuários, especialmente operadores de loja, a fim de observar a interação prática com o sistema e identificar oportunidades de refinamento na experiência de uso. Estudos de usabilidade da interface *web* também podem contribuir para aprimorar a eficiência operacional e a fluidez das rotinas administrativas, garantindo que a solução atenda não apenas aos requisitos técnicos, mas também às necessidades e expectativas dos profissionais que a utilizarão no cotidiano. Essas dimensões representam caminhos naturais para trabalhos futuros, que podem incluir a expansão do sistema para múltiplos dispositivos, medições quantitativas de consumo, integração com redes *IoT* especializadas e aplicação prática em estabelecimentos comerciais reais.

Em síntese, este trabalho demonstra que a arquitetura proposta é tecnicamente sólida e adequada às necessidades de automação de preços no varejo. A confirmação funcional do protótipo, associada ao respaldo da literatura, valida a viabilidade do uso da tecnologia *e-ink* e estabelece bases consistentes para evoluções posteriores, contribuindo para o avanço de soluções inteligentes e energeticamente eficientes no contexto do comércio moderno.

Referências Bibliográficas

BERTE, D.-R. et al. Defining the iot. In: PEARSON EDUCATIONAL. *Proceedings of the international conference on business excellence*. [S.l.], 2018. v. 12, n. 1, p. 118–128.

BHASKAR, A. *An Interpretation and Anatomization of Angular: A Google Web Framework*. 2020. <https://consensus.app/papers/an-interpretation-and-anatomization-of-angular-a-google-bhaskar/1047f13385625ffeb9f0caef3c2d4fa/?utm_source=chatgpt>.

BURHAN, M. et al. Iot elements, layered architectures and security issues: A comprehensive survey. *sensors*, MDPI, v. 18, n. 9, p. 2796, 2018.

CHANG, L. et al. Synthetic melanin e-ink. *Acs Applied Materials & Interfaces*, ACS Publications, v. 9, n. 19, p. 16553–16560, 2017.

DOMÍNGUEZ-BOLAÑO, T. et al. An overview of iot architectures, technologies, and existing open-source projects. *Internet of Things*, Elsevier, v. 20, p. 100626, 2022.

EMMANNI, P. S. Comparative analysis of angular, react, and vue.js in single page application development. *International Journal of Science and Research (IJSR)*, 2023.

EMMANNI, P. S. Architectural patterns and best practices for scalable enterprise applications with angular. *Journal of Mathematical & Computer Applications*, 2024.

GARAUS, M.; WOLFSTEINER, E.; WAGNER, U. Shoppers' acceptance and perceptions of electronic shelf labels. *Journal of Business Research*, Elsevier, v. 69, n. 9, p. 3687–3692, 2016.

GEETHA, G. et al. Interpretation and analysis of angular framework. In: *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*. [S.l.: s.n.], 2022. p. 1–6.

HAKKI, S. Esp8266 and esp32 series of soc microcontrollers. *Programmable Smart Microcontroller Cards*, v. 110, 2021.

IVANOVA, A. et al. Chemistry and manufacturing technology of electronic ink for electrophoretic displays (a review). *Russian Journal of Inorganic Chemistry*, Springer, v. 65, n. 13, p. 1985–2005, 2020.

JOHNSON, R. et al. *Expert One-on-One J2EE Development without EJB*. Indianapolis, IN: Wrox Press, 2004. ISBN 978-0764558313. Disponível em: <<https://www.springframework.org/>>.

- KHAN, M. A. A. et al. Research on key technologies of electronic shelf labels based on lora. *Journal on Big Data*, Tech Science Press, n. 2, p. 49, 2021.
- KURNIASYAH; RAKHMAWATI. Smart door lock innovation using integration of bluetooth low energy and mqtt iot protocol. *INAJEEE Indonesian Journal of Electrical and Electronics Engineering*, 2024. Disponível em: <https://consensus.app/papers/smart-door-lock-innovation-using-integration-of-bluetooth-kurniasyah-rakhmawati/2f55e858a88657beac4322c275ef5540/?utm_source=chatgpt>.
- LE, X.-M. et al. Research and design a low power electronic shelf label based on e-paper display and lora technology. In: *2023 International Conference on Advanced Technologies for Communications (ATC)*. [S.l.: s.n.], 2023. p. 356–360.
- LIAWATIMENA, S.; GUNAWAN, D. Design of an automatic price tag system for web-based retail business. In: IEEE. *2024 International Conference on Information Management and Technology (ICIMTech)*. [S.l.], 2024. p. 729–734.
- LIU, S. et al. Cross-platform drilling 3d visualization system based on webgl. *Mathematical Problems in Engineering*, v. 2021, p. 1–18, 2021.
- MAGOMEDOV, I.; BAGOV, A.; ZOLKIN, A. Internet of things: future business. In: *European Proceedings of Social and Behavioural Sciences EpSBS*. [S.l.: s.n.], 2020. p. 553–558.
- NECULA, S. Exploring the model-view-controller (mvc) architecture: A broad analysis of market and technological applications. Preprints, 2024.
- OVANI, D. J.; WINDASARI, N. A. The impact of electronic shelf label on customer well-being in the omnichannel smart retail. In: SPRINGER. *INFORMS International Conference on Service Science*. [S.l.], 2022. p. 31–46.
- OZKAYA, M.; FIDANDAN, I. Mvclang: A software modeling language for the model-view-controller design pattern. In: *Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering*. [S.l.: s.n.], 2020. p. 75–83.
- PLAUSKA, I.; LIUTKEVIČIUS, A.; JANAVIČIŪTĖ, A. Performance evaluation of c/c++, micropython, rust and tinygo programming languages on esp32 microcontroller. *Electronics*, 2022. Disponível em: <https://consensus.app/papers/performance-evaluation-of-cc-micropython-rust-and-tinygo-plauska-liutkevicius/6b375ddde7ab5aa3998646b0bcb884ea/?utm_source=chatgpt>.
- RABBANI, S. A.; AHMED, M.; ZAHID, A. H. E-ink; revolution of displays. In: *MATEC Web of Conferences*. [s.n.], 2023. Disponível em: <https://consensus.app/papers/eink-revolution-of-displays-rabbani-ahmed/74a3a8b9c9f8550ab8968dd3c400bc00/?utm_source=chatgpt>.
- RAI, P.; REHMAN, M. Esp32 based smart surveillance system. In: IEEE. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. [S.l.], 2019. p. 1–3.

- REENSKAUG, T. *The Model-View-Controller (MVC): Its Past and Present*. 2003. <https://consensus.app/papers/the-modelviewcontroller-mvc-its-past-and-present-reenskaug/919ab603aba55245bd2cc2d352773c88/?utm_source=chatgpt>. University of Oslo.
- SAGAR, S. The impact of digital transformation on retail management and consumer behavior. *Journal of Business and Management*, v. 26, n. 1, p. 06–14, 2024.
- THAKUR, R.; PANDEY, U. The role of model-view controller in object oriented software development. *Nepal Journal of Multidisciplinary Research*, 2019.
- VO, H. L.; VO, D. H. The purchasing power parity and exchange-rate economics half a century on. *Journal of Economic Surveys*, Wiley Online Library, v. 37, n. 2, p. 446–479, 2023.
- VOORHEES, D. P. Introduction to model-view-controller. In: *Architectural Patterns for Software Engineering*. [S.l.: s.n.], 2020. p. 175–179.
- WALLS, C. *Spring in Action*. 6th. ed. Shelter Island, NY: Manning Publications, 2022. ISBN 9781617297571. Disponível em: <<https://www.manning.com/books/spring-in-action-sixth-edition>>.
- YANG, B.-R. et al. Understanding the mechanisms of electronic ink operation. *Journal of the Society for Information Display*, v. 29, p. 38–46, 2020. Disponível em: <https://consensus.app/papers/understanding-the-mechanisms-of-electronic-ink-operation-yang-hu/cf5bf3e83fb55f9eb73787994a4ce9a1/?utm_source=chatgpt>.