

Projeto de Carrinho de Compras Inteligente

Proposta e Planejamento Inicial

Domynic Barros Lima¹,
Juan Pablo Ramos de Oliveira², Luis Fernando Rodrigues Braga³,

¹ Instituto de Ciências Exatas e Informática
Pontifícia Universidade Católica de Minas Gerais (PUC MG)
Belo Horizonte, MG – Brazil

Abstract. *The retail sector faces persistent challenges related to checkout bottlenecks, operational inefficiency, and the lack of real-time financial control during in-store purchases. While advanced solutions such as computer-vision-based autonomous stores and high-precision self-checkout systems have been deployed globally, their high cost and technological complexity make them inaccessible to most retailers, particularly in developing countries. This study presents AutoCart, a low-cost smart shopping cart prototype designed for the Brazilian retail context. The system integrates embedded hardware (ESP32-based microcontroller, GM65 barcode scanner, and load cells), a backend with multi-tenant architecture, and a mobile interface that updates cart totals in real time. Beyond presenting the prototype, this work offers a critical and technical analysis of its feasibility, including the physical limitations of low-cost load cells when used in mobile structures. We compare the proposed approach with existing international solutions and discuss the economic, operational, and social implications of adopting accessible automation technologies. The results indicate that although full autonomy remains impractical for most retailers, hybrid low-cost systems can significantly improve the shopping experience and reduce checkout time, paving the way for scalable and affordable retail innovation.*

Keywords: Smart Shopping Cart, Embedded Systems, IoT, Retail Automation, Low-Cost Sensors, ESP32, Customer Experience.

Resumo. *O varejo brasileiro enfrenta desafios persistentes relacionados a gargalos no checkout, falta de eficiência operacional e ausência de ferramentas que permitam ao consumidor acompanhar seus gastos em tempo real. Embora soluções avançadas, como lojas autônomas baseadas em visão computacional e sistemas de autoatendimento de alta precisão, já estejam presentes no mercado internacional, seus custos e complexidade tecnológica as tornam inacessíveis para a maior parte dos varejistas nacionais. Este estudo apresenta o AutoCart, um protótipo de carrinho de compras inteligente de baixo custo desenvolvido com foco no contexto brasileiro. O sistema integra hardware embarcado (microcontrolador baseado em ESP32, scanner GM65 e células de carga), um backend com arquitetura multi-tenant e um aplicativo móvel capaz de atualizar o valor da compra em tempo real. Além da implementação, o trabalho realiza uma análise técnica crítica sobre a viabilidade do protótipo, destacando principalmente as limitações físicas do uso de células de carga de baixo custo em*

estruturas móveis. A solução é comparada com tecnologias internacionais já implementadas, discutindo impactos econômicos, operacionais e sociais. Os resultados mostram que, embora a automação completa ainda seja inviável para grande parte do varejo, sistemas híbridos e acessíveis podem aprimorar significativamente a experiência do consumidor e reduzir o tempo de checkout, estabelecendo um caminho realista para inovação no setor.

Palavras-chave: Carrinho de Compras Inteligente, Sistemas Embarcados, IoT, Automação do Varejo, Sensores de Baixo Custo, ESP32, Experiência do Consumidor.

1. Introdução

O varejo contemporâneo encontra-se em um momento de transição marcado pela crescente demanda por automação, eficiência e redução de atritos na jornada de compra. Em supermercados e hipermercados, um dos principais gargalos estruturais continua sendo o processo de checkout, responsável por longas filas, baixa fluidez operacional e custos elevados para varejistas e consumidores. Estudos recentes mostram que, para 74% dos consumidores, o tempo gasto em filas é o fator que mais impacta negativamente a satisfação durante compras presenciais [Grewal et al. 2020]. Esse problema é especialmente crítico no contexto brasileiro, onde pequenos e médios mercados, responsáveis por mais de 50% do abastecimento alimentar do país, enfrentam limitações financeiras e tecnológicas que dificultam a adoção de soluções avançadas de automação [ABComm 2023].

O avanço da digitalização do varejo trouxe consigo modelos disruptivos, como lojas autônomas baseadas em visão computacional e sensores distribuídos, amplamente representadas pelo conceito das "checkout-free stores", popularizado por iniciativas como Amazon Go. Embora tais soluções prometam eliminar filas e tornar o processo de compra totalmente automatizado, análises recentes indicam que a implementação em larga escala ainda enfrenta desafios significativos, incluindo custos de infraestrutura extremamente elevados, dependência de hardware especializado e a necessidade de supervisão humana constante para correção de erros do sistema [Liu et al. 2022]. Na prática, essas tecnologias permanecem economicamente inviáveis para a maior parte do varejo mundial, e ainda mais distantes da realidade financeira brasileira.

Em uma direção diferente, os sistemas de autoatendimento têm sido amplamente adotados por redes do varejo nacional. No entanto, essas soluções, baseadas principalmente em balanças sensíveis e verificação de peso, enfrentam problemas recorrentes: interrupções frequentes por variação mínima de peso, elevada taxa de intervenção humana, baixa tolerância a produtos volumosos ou leves, além de vulnerabilidade a fraudes simples, como substituição de etiquetas [Davis and Norberg 2019]. Como resultado, esses sistemas frequentemente transferem o ônus operacional para funcionários, anulando parte dos benefícios esperados da automação.

Nesse cenário, observa-se uma lacuna clara entre as tecnologias de ponta e a infraestrutura necessária para sua implantação no Brasil. De um lado, soluções baseadas em visão computacional ou sensores multimodais apresentam custos proibitivos; de outro, sistemas de autoatendimento tradicionais falham em oferecer uma experiência consistente e confiável. É nesse intervalo que surge a necessidade de alternativas pragmáticas, acessíveis e suficientemente robustas para melhorar a experiência do usuário e otimizar processos internos sem depender de arquiteturas complexas.

Diante disso, este trabalho propõe o desenvolvimento e a análise crítica de um protótipo de carrinho de compras inteligente de baixo custo, denominado AutoCart¹, concebido como uma solução intermediária que busca melhorar a jornada do consumidor sem replicar os altos custos de tecnologias de automação avançada. O AutoCart integra hardware embarcado baseado em um microcontrolador compatível com ESP32, um scanner de código de barras dedicado (GM65), células de carga para aferição de peso e uma interface digital móvel. O sistema permite ao cliente controlar sua lista de compras em tempo real, visualizar o total acumulado e experimentar um fluxo de checkout expresso, reduzindo etapas repetitivas presentes em caixas tradicionais.

A motivação deste projeto não se baseia na utopia de um varejo totalmente autônomo, mas na criação de uma solução realista e replicável para o contexto brasileiro. Adotamos como hipótese que é possível desenvolver tecnologias de automação mais acessíveis que sirvam tanto a grandes redes quanto a mercados de pequeno porte, democratizando melhorias operacionais que hoje se concentram apenas em empresas com grande capacidade de investimento. Além disso, o projeto investiga os limites físicos e práticos de sensores de baixo custo, especialmente as células de carga, oferecendo uma análise técnica inédita sobre sua aplicabilidade em carrinhos móveis, marcada por interferências mecânicas, vibrações e limitações estruturais.

Os objetivos gerais deste estudo são, portanto:

- Projetar e desenvolver um protótipo funcional de baixo custo;
- Analisar sua viabilidade sob uma perspectiva técnica, operacional e econômica;
- Comparar criticamente sua abordagem com tecnologias existentes no mercado nacional e internacional;
- Discutir as limitações e propor caminhos viáveis para evoluções futuras, considerando não apenas avanços tecnológicos, mas também a sustentabilidade de implementação.

As contribuições deste trabalho incluem (a) um protótipo funcional integrado entre hardware, firmware, backend e aplicativo móvel; (b) uma análise aprofundada das limitações físicas de sensores de carga aplicados a carrinhos móveis; (c) uma arquitetura de software escalável multi-tenant inspirada em cenários reais; e (d) uma discussão crítica sobre acessibilidade tecnológica e automação pragmática aplicada ao varejo brasileiro.

A seguir, este artigo apresenta sua estrutura: a Seção 2 discute os fundamentos teóricos e tecnológicos que embasam o projeto; a Seção 3 apresenta o panorama das soluções existentes e suas limitações; a Seção 4 detalha a metodologia e o processo de desenvolvimento; a Seção 5 descreve a arquitetura do AutoCart; a Seção 6 apresenta os resultados experimentais; a Seção 7 discute criticamente tais resultados; a Seção 8 apresenta trabalhos futuros; e a Seção 9 conclui o estudo.

2. Fundamentação Teórica

O desenvolvimento de um carrinho de compras inteligente envolve a convergência de múltiplas áreas da Computação e da Engenharia, como Internet das Coisas (IoT), sistemas embarcados, instrumentação eletrônica, redes de comunicação e sistemas distribuídos.

¹O código fonte e a documentação técnica estão disponíveis em: <https://github.com/Juanpablozim/plmg-cc-2025-2-ti5-g11-autocart>

Esta seção apresenta os conceitos essenciais que fundamentam o projeto AutoCart, permitindo compreender tanto suas capacidades quanto suas limitações estruturais.

2.1. Internet das Coisas (IoT) no Varejo

A Internet das Coisas caracteriza-se pela integração de sensores, atuadores e dispositivos computacionais conectados à Internet, capazes de coletar e transmitir dados continuamente [Atzori et al. 2010]. No setor varejista, soluções baseadas em IoT têm ganhado relevância nos últimos anos, com aplicações que abrangem desde o monitoramento de estoques até o rastreamento de clientes, otimização de filas e gestão de prateleiras inteligentes [Greengard 2015].

Em particular, dispositivos embarcados equipados com Wi-Fi ou protocolos leves (como MQTT) possibilitam um fluxo contínuo de dados entre carrinhos, sistemas de pagamento e servidores centrais. A viabilidade de soluções IoT em mercados físicos está diretamente relacionada ao baixo consumo energético, à confiabilidade da comunicação e à capacidade de operar em ambientes com grande interferência eletromagnética, características essenciais também para o AutoCart.

No contexto deste projeto, o uso de um microcontrolador com conectividade integrada permitiu a implementação de uma arquitetura distribuída em que o carrinho interage com um backend centralizado, garantindo sincronização de itens, valor total da compra e estado da sessão em tempo real.

2.2. Sistemas Embarcados e Arquiteturas SoC

Sistemas embarcados são dispositivos projetados para executar tarefas específicas com restrições de processamento, memória e energia [Marwedel 2010]. A escolha do microcontrolador é um fator crítico, pois determina os limites computacionais e energéticos do sistema. Microcontroladores baseados em arquiteturas System-on-a-Chip (SoC), como o ESP32 ou o NodeMCU (utilizado no protótipo), integram em um único chip:

- CPU de baixo consumo,
- Módulos de comunicação (Wi-Fi/Bluetooth),
- Controladores ADC e DAC,
- Suporte a interrupções,
- Barramentos de comunicação (UART, SPI, I2C).

Essa integração favorece aplicações IoT de baixo custo, sem a necessidade de hardware externo adicional.

Além disso, muitos desses microcontroladores utilizam sistemas operacionais de tempo real (RTOS), como o FreeRTOS, que organiza tarefas concorrentes garantindo determinismo e previsibilidade [Amazon Web Services 2020]. Isso é essencial para ler sensores em intervalos consistentes, processar eventos seriais e manter estabilidade nas leituras, especialmente no caso de dispositivos sensíveis como células de carga.

No AutoCart, o microcontrolador precisa coordenar:

1. Leitura contínua das células de carga,
2. Decodificação dos dados enviados pelo scanner GM65,
3. Comunicação com o backend via Wi-Fi,

4. Processamento interno para filtragem e estabilização de pesos,
5. Resposta em tempo real ao usuário.

Essas demandas tornam o estudo da arquitetura embarcada indispensável para compreender os limites do protótipo.

2.3. Sensores de Carga e Princípios Físicos

As células de carga são dispositivos utilizados para medir peso por meio da deformação elástica de materiais metálicos. Elas se baseiam no princípio dos strain gauges (extensômetros), que variam sua resistência elétrica quando submetidos a força mecânica. Essa variação é pequena e, portanto, é convertida em um diferencial elétrico por uma ponte de Wheatstone [Fraden 2016a].

2.3.1. Princípio da deformação elástica (Strain Gauge)

Quando um corpo é submetido a força, ele sofre deformação proporcional à tensão aplicada, conforme a Lei de Hooke. O strain gauge adere a essa superfície, detectando elongações ou compressões microscópicas (10^{-6} a 10^{-4}).

A resistência elétrica varia de acordo com:

$$\frac{\Delta R}{R} = G \cdot \varepsilon$$

onde

- G é o fator de gauge,
- ε é a deformação mecânica.

2.3.2. Conversão via ponte de Wheatstone

Como a variação é muito pequena, a ponte amplifica o sinal diferencial. Ainda assim, o ruído permanece significativo, por isso dispositivos como o HX711 são necessários.

2.3.3. O papel do HX711

O HX711 é um conversor ADC de 24 bits projetado para amplificar e digitalizar sinais provenientes de células de carga. Apesar de barato e amplamente usado em prototipagem, ele é extremamente sensível a ruídos elétricos, instabilidade mecânica e interferências. Para aplicações industriais, normalmente utiliza-se hardware dedicado mais robusto.

2.3.4. Limitações físicas em carrinhos de compras

Este projeto revelou na prática limitações fundamentais:

- Vibração e trepidação: A leitura varia significativamente com a movimentação do carrinho.

- Distribuição irregular do peso: Quando o produto é colocado fora do eixo, há variações drásticas (até $\pm 1,7$ kg observados).
- Células múltiplas criam uma “viga contínua”: O sistema mecânico se comporta como uma estrutura elástica, gerando acoplamento entre as células.
- Ruído eletromagnético: Motores de geladeiras, iluminação e Wi-Fi contribuem para instabilidade.
- Histerese e drift térmico: Mudanças de temperatura modificam o zero da balança ao longo do tempo.

Tais fenômenos são amplamente conhecidos na literatura de instrumentação e tornam extremamente difícil garantir precisão em estruturas móveis [Pinto and Silva 2018].

Essa compreensão física é crucial para justificar por que células de carga baratas não conseguem oferecer resultados confiáveis em carrinhos de compras em movimento.

2.4. Leitores de Código de Barras Dedicados

O GM65 utilizado no AutoCart é um módulo industrial de leitura baseado em sensores CMOS e um mecanismo interno de decodificação otimizado. Diferente da leitura via câmera de smartphone, leitores industriais oferecem:

- Alta tolerância a iluminação variada,
- Decodificação autônoma,
- Consumo energético reduzido,
- Maior velocidade de varredura,
- Suporte nativo a múltiplos padrões de código.
- Estudos mostram que leitores dedicados reduzem a latência e aumentam a taxa de sucesso em ambientes de varejo, especialmente quando operados por usuários não técnicos [Li and Wang 2019].

Essa escolha se mostrou acertada no projeto, dado que o objetivo principal é garantir acessibilidade e facilidade para usuários com baixa familiaridade tecnológica.

2.5. Visão Computacional no Varejo

A visão computacional tem sido utilizada no varejo para:

- Alta tolerância a iluminação variada,
- Detecção automática de produtos,
- Monitoramento de prateleiras,
- Prevenção de perdas,
- Checkout sem fricção.

Soluções modernas utilizam múltiplas câmeras, modelos de deep learning e sensores complementares [Reddy and Kumar 2021]. No entanto:

- Exigem datasets massivos,
- Dependem de GPUs ou hardware especializado,
- Aumentam o custo da manutenção,
- Sofrem com ambientes complexos (ocluções, embalagens reflexivas, variação de iluminação).

Análises mostram que essas soluções ainda possuem custo proibitivo para a maior parte do varejo global [Liu et al. 2022]. Por isso, apesar de desejáveis, são inviáveis para protótipos de baixo custo ou mercados pequenos, justificando a estratégia do AutoCart em focar no código de barras.

2.6. Sistemas de Self-Checkout e Suas Limitações

Sistemas de autoatendimento tradicionalmente utilizam:

- Uma unidade de processamento,
- Sensores de peso de alta precisão,
- Câmeras para monitoramento,
- Módulos antifraude.

Porém, como discutido por Davis e Horne (2019) [Davis and Norberg 2019], esses sistemas frequentemente apresentam:

- Interrupções por pequena variação de peso,
- Necessidade de intervenção humana,
- Complexidade operacional,
- Baixa tolerância a produtos leves ou volumosos.

Tais limitações reforçam a importância de estudar a usabilidade e o comportamento humano na automação do varejo, focos importantes no projeto AutoCart.

2.7. Arquiteturas Multi-Tenant para Sistemas de Varejo

Uma arquitetura multi-tenant permite que múltiplos clientes (mercados diferentes, no caso do AutoCart) utilizem um mesmo backend de forma isolada e segura. Essas arquiteturas são amplamente usadas em plataformas SaaS (Software as a Service) [Carvalho and Souza 2020].

Benefícios incluem.

- Escalabilidade,
- Isolamento lógico,
- Gerenciamento configurável por mercado,
- Facilidade de replicação..

No AutoCart, a escolha por simular múltiplos bancos de dados refletiu a necessidade de representar cenários reais, nos quais cada mercado possui seu próprio catálogo de produtos e tabelas de preços.

3. Panorama de Soluções Existentes e Análise Comparativa

O desenvolvimento de sistemas de automação no varejo avançou de maneira acelerada na última década, impulsionado pela crescente demanda por experiências de compra mais rápidas, redução de filas e otimização de custos operacionais. Diversas empresas, startups e centros de pesquisa têm proposto soluções que combinam sensores, visão computacional, RFID e modelos de inteligência artificial para permitir processos de checkout sem fricção. Este projeto situa-se nesse ecossistema tecnológico, ao mesmo tempo buscando alternativas acessíveis e compatíveis com o mercado brasileiro, especialmente o pequeno e médio varejo, frequentemente negligenciado pelas soluções comerciais de alto custo.

A seguir, é apresentada uma análise crítica das soluções mais relevantes para o contexto do AutoCart, destacando-se suas vantagens, limitações e compatibilidade com a realidade socioeconômica nacional.

3.1. Carrinhos Inteligentes Baseados em Visão Computacional

Sistemas como o Amazon Dash Cart, o Caper Cart e um modelo recentemente lançado por empresas brasileiras utilizam múltiplas câmeras internas, sensores infravermelhos e redes neurais profundas para identificar produtos visualmente. Essas soluções representam o estado da arte em automação de carrinhos inteligentes.

3.1.1. Características técnicas

Esses carrinhos geralmente possuem:

- Câmeras RGB distribuídas nas laterais e no topo;
- Iluminação controlada (LEDs laterais e superiores);
- Módulos de processamento dedicado, como NVIDIA Jetson Nano ou TX2;
- Sensores de volume ou peso como redundância;
- Integração com sistemas de pagamento embarcados.

A visão computacional é utilizada para reconhecer embalagens, cores, características de textura e formatos. Em ambientes controlados, como o interior de um carrinho com iluminação padronizada, a acurácia tende a ser alta.

3.1.2. Limitações observadas

Apesar da sofisticação, a literatura indica problemas estruturais:

- Custo extremamente elevado: Apenas o módulo de processamento chega a superar todo o orçamento do AutoCart.
- Dependência de treinamento específico por mercado.
- Sensibilidade a produtos deformados, amassados ou parcialmente ocultos.
- Tempo significativo para montagem de dataset (centenas a milhares de imagens por produto).
- Complexidade de manutenção e atualizações.

Além disso, para mercados brasileiros de pequeno porte, a infraestrutura elétrica e de rede muitas vezes não suporta sistemas desse porte, inviabilizando sua adoção. Essa análise reforça que, embora tecnologicamente superiores, tais soluções não se alinham ao objetivo do AutoCart de oferecer uma alternativa acessível e replicável.

3.2. Sistemas Baseados em Esteiras com Visão Computacional

Outro modelo emergente no Brasil e no exterior é o uso de esteiras acopladas ao caixa, formando um túnel que fotografa e identifica produtos automaticamente enquanto são inseridos. Empresas brasileiras já iniciaram pilotos desse modelo, que pode ser visto como uma evolução dos sistemas de self-checkout tradicionais.

3.2.1. Benefícios

- Redução parcial da intervenção humana;
- Identificação de múltiplos produtos simultaneamente;
- Ambiente mais controlado do que um carrinho em movimento;
- Maior facilidade de padronização entre lojas;
- Menor custo em comparação a carrinhos com câmeras embarcadas.

3.2.2. Desafios técnicos

Por outro lado, a literatura aponta que:

- Produtos muito reflexivos (garrafas pet, embalagens metalizadas) reduzem acurácia;
- Alimentos a granel permanecem um problema crítico;
- Túnel fechado exige manutenção constante;
- Dependência de hardware robusto e custoso;
- Complexidade logística para instalar múltiplas unidades por mercado.

Assim, embora mais barato que carrinhos inteligentes por visão, esteiras automatizadas ainda têm custo proibitivo para muitos varejistas.

3.3. Sistemas de Self-Checkout Tradicionais com Balança

Este é o modelo mais difundido no Brasil. Ele combina:

- Leitor de código de barras,
- Balança de conferência,
- Tela touchscreen,
- Módulo de pagamento,
- Sistema antifraude baseado na variação de peso.

3.3.1. Pontos positivos

- Tecnologia madura;
- Aceitação global;
- Baixa dependência de inteligência artificial.

3.3.2. Limitações críticas

Pesquisas mostram que:

- Erros de balança por produtos leves ($< 50\text{ g}$) exigem intervenção humana constante;
- Produtos volumosos ou instáveis geram falsos positivos de fraude;
- Usuários com baixa familiaridade tecnológica têm dificuldade de uso;
- Filas se formam, anulando a proposta de automação;
- Não elimina o gargalo do checkout, apenas o desloca.
- Essa limitação motivou a busca por sistemas totalmente integrados ao carrinho, como o AutoCart.

3.4. Soluções Baseadas em RFID / NFC

A “utopia técnica” seria que todos os produtos possuíssem RFID passivo, permitindo leitura automática sem necessidade de visão ou peso. A literatura demonstra que essa tecnologia funcionaria muito bem para checkout sem fricção [Floerkemeier and Langheinrich 2018].

Porém:

- RFID encarece embalagens,
- O varejo brasileiro possui margens extremamente baixas,
- Rastreabilidade em produtos frescos é inviável,
- Interferências metálicas e líquidas geram erros de leitura,
- A adoção em escala é extremamente baixa.

Assim, embora promissora, a RFID está distante de ser uma solução economicamente viável.

3.5. Soluções de Baixo Custo com Microcontroladores

Há poucos trabalhos explorando soluções de baixo custo para automação do varejo usando microcontroladores e sensores simples. Estudos recentes destacam:

- Uso de ESP32 para carrinhos com inventário digital,
- Protótipos com leitura de QR code via câmera,
- Sistemas híbridos com sensores de proximidade.

Porém, existe uma lacuna clara na literatura:

- Quase nenhum protótipo integra peso + código de barras + microcontrolador + aplicação móvel em um único sistema coerente.
- Pouquíssimos estudos abordam a instabilidade mecânica de células de carga em estruturas móveis.

Isso reforça o caráter inovador e experimental do AutoCart, mesmo quando comparado a soluções acadêmicas.

3.6. Comparação Geral das Soluções do Mercado

A Tabela 1 sintetiza a comparação entre as principais soluções:

Tecnologia	Custo	Precisão	Complex.	Aderência
Carrinho com visão computacional	Altíssimo	Alta	Muito Alta	Baixa
Esteira com túnel	Médio/Alto	Alta	Alta	Média
Self-checkout tradicional	Médio	Média	Média	Alta
RFID	Altíssimo	Altíssimo	Alta	Muito Baixa
Solução IoT de baixo custo (Auto-Cart)	Baixo	Baixa/Média	Baixa	Alta

Tabela 1. Comparação entre as principais soluções

A tabela evidencia que soluções de baixo custo ainda são tecnologicamente inferiores em precisão, mas extremamente superiores em acessibilidade, escalabilidade, replicabilidade regional e adequação ao pequeno varejo, que representa a maior parte dos estabelecimentos brasileiros.

3.7. Síntese Crítica

O panorama analisado mostra que não existe, hoje, uma solução universal que concilie:

- Alta precisão,
- Baixo custo,
- Facilidade de manutenção,
- Ausência de complexidade operacional.

Soluções de ponta são comercialmente inacessíveis, soluções baratas são tecnicamente limitadas. O AutoCart se propõe a ocupar justamente o espaço intermediário: ser simples, acessível, replicável e adaptável, ainda que à custa de limitações inerentes à tecnologia de baixo custo.

Essa abordagem não substitui sistemas de visão computacional, mas abre espaço para inovação incremental, permitindo que pequenos mercados adotem gradualmente tecnologias inteligentes.

4. Bases Teóricas Fundamentais

O desenvolvimento de um carrinho de compras inteligente de baixo custo exige a integração de múltiplas áreas da Computação e da Engenharia. Esta seção apresenta os fundamentos teóricos essenciais que embasam o projeto, com foco em quatro eixos centrais: (i) Internet das Coisas (IoT) e sistemas embarcados; (ii) arquitetura e operação do ESP32/NodeMCU; (iii) princípios físicos envolvidos na medição de peso utilizando células de carga e o módulo HX711; e (iv) protocolos de comunicação e integração com backend.

4.1. Internet das Coisas e Sistemas Embarcados

A Internet das Coisas (IoT) caracteriza-se pela conexão de dispositivos físicos à internet, possibilitando comunicação contínua entre sensores, atuadores e serviços em nuvem. No contexto do varejo, sistemas IoT permitem monitoramento em tempo real, automação de processos e coleta de dados operacionais [Atzori et al. 2019, Gubbi and Buyya 2020].

Um sistema embarcado é um computador especializado projetado para executar uma função específica com restrições de custo, consumo de energia e processamento. O carrinho inteligente enquadra-se nesse paradigma, já que todas as operações – leitura de códigos de barras, medição de peso, comunicação com backend e atualização de interface, são realizadas em hardware dedicado com recursos limitados.

Além disso, por operar em ambientes ruidosos, sujeitos a vibrações e movimento constante, o sistema embarcado deve garantir resiliência e previsibilidade temporal, características essenciais para aplicações de varejo.

4.2. Arquitetura do Microcontrolador ESP32/NodeMCU

O ESP32 foi escolhido como plataforma central por sua relação custo-benefício e pela integração de múltiplos módulos de conectividade e processamento. Ele adota a arquitetura System-on-a-Chip (SoC), incorporando:

- Processador dual-core a 240 MHz;
- Memória RAM integrada;

- Módulos Wi-Fi IEEE 802.11 b/g/n;
- Apporte a Bluetooth e BLE;
- ADCs, GPIOs, barramentos I2C, SPI e UART [Kolban 2019, Rao and Singh 2021].

A capacidade de operar tarefas concorrentes é fornecida pelo FreeRTOS, sistema operacional de tempo real embarcado no ESP32. Esse RTOS gerencia tarefas como:

- Leitura contínua das células de carga;
- Captura e interpretação de dados do leitor de código de barras;
- Comunicação periódica com o backend via HTTP/MQTT;
- Atualização da interface do usuário.

Essa divisão de atividades impede *blocking* e garante responsividade mesmo sob carga de trabalho variável. Tais características tornam o ESP32 adequado para sistemas IoT distribuídos e de baixo custo, como o proposto neste projeto.

4.3. Princípios Físicos das Células de Carga e Deformação Elástica

As células de carga utilizadas dependem do princípio da *deformação elástica* de materiais. Quando submetida a uma força, a barra metálica sofre um microalongamento proporcional à carga aplicada. Essa deformação é detectada por extensômetros elétricos (strain gauges), cujas resistências variam linearmente com a deformação mecânica [Gere and Timoshenko 1999, Hibbeler 2017].

O extensômetro forma uma ponte de Wheatstone, permitindo a leitura precisa de microvariações de resistência. A relação entre força aplicada e deformação segue o modelo:

$$\Delta L = \frac{F \cdot L}{A \cdot E}$$

em que:

- F é a força aplicada,
- L é o comprimento da barra,
- A é a área da seção transversal,
- E é o módulo de elasticidade do material.

As principais limitações para aplicações como carrinhos inteligentes incluem:

- Sensibilidade a vibrações (trepidações durante a movimentação do carrinho);
- Sensibilidade térmica que altera a resistência dos strain gauges;
- Distribuição desigual de peso quando há apenas uma célula instalada;
- Necessidade de plataforma rígida e estável para leituras consistentes [Fraden 2016b, Phidgets Inc. 2020, Tekscan 2021].

Por esses motivos, muitos sistemas industriais utilizam 4 células de carga dispostas em balanceamento estrutural. Em protótipos de baixo custo, pequenas imprecisões são esperadas, justificando a adoção de filtros de estabilização, tempo de amortecimento e lógica de rejeição de leituras instáveis.

4.4. O Conversor HX711 e a Amplificação de Sinais de Baixa Intensidade

O HX711 é um conversor analógico-digital (ADC) de 24 bits projetado para leitura de células de carga. Ele não apenas digitaliza o sinal, mas também o amplifica com ganho configurado (32, 64 ou 128). Isso o torna ideal para sinais extremamente fracos, comuns em strain gauges. No entanto, sua alta sensibilidade também o torna suscetível a ruído ambiental e interferência eletromagnética [Yao and Chen 2020, SparkFun Electronics 2020].

características:

- Ruído interno considerável em ambientes vibratórios;
- Necessidade de alimentação estável para leituras robustas;
- Sensibilidade extrema a variações mínimas de peso (movimentos de 0.1 g);
- Latência de conversão aproximada de 10–80 ms.

minimizar ruído, recomenda-se:

- Implementação de filtros digitais (média móvel, filtro exponencial);
- Fixação rígida da plataforma de pesagem;
- Leitura apenas quando o sistema estiver estável.

4.5. Comunicação e Protocolos de Integração com Backend

Um carrinho inteligente precisa trocar dados com a infraestrutura da loja para consultar preços, registrar compras e sincronizar estados. Os protocolos mais utilizados em sistemas IoT incluem:

- HTTP/REST — Simples e compatível com APIs modernas;
- MQTT — Protocolo leve ideal para dispositivos com recursos limitados;
- WebSockets — Comunicação bidirecional contínua;

No protótipo desenvolvido, a comunicação ocorre via HTTP/REST, pois:

- Simplifica a implementação do backend;
- Reduz a complexidade do firmware;
- Torna a API facilmente testável e extensível.

Protótipos futuros podem adotar MQTT para melhorar eficiência energética e latência. A segurança da transmissão pode ser reforçada com HTTPS, tokens JWT e chaves de acesso, além de autenticação de dispositivos.

5. Projeto e Desenvolvimento do Hardware

O desenvolvimento do hardware do protótipo AutoCart constituiu a parte mais desafiadora e imprevisível do projeto. Ao contrário do backend e do aplicativo móvel, cuja evolução foi relativamente linear e controlada, a etapa de hardware envolveu fenômenos físicos, interferências eletromagnéticas, limitações de prototipagem e falhas decorrentes de documentação fragmentada. Esta seção apresenta um relato técnico realista, descrevendo as escolhas de engenharia, as tentativas malsucedidas, os problemas inesperados e as decisões pragmáticas adotadas para manter o projeto viável.

5.1. Arquitetura Física do Protótipo

O protótipo foi montado utilizando uma base composta por uma forma de bolo em aço inox fixada sobre uma tábua de madeira rígida. Entre essas duas estruturas foram instaladas quatro células de carga de 20 kg cada, posicionadas nos quatro cantos da base, com espaçamento aproximado de 10 a 15 cm entre elas.

As células foram fixadas de modo que:

- A extremidade fixa estivesse parafusada à tábua;
- A extremidade sensível (parte móvel) sustentasse a forma metálica utilizada como cesto;
- A deformação fosse distribuída verticalmente quando um peso fosse aplicado.

Essa configuração é comumente utilizada em balanças de quatro pontos, mas a precisão depende fortemente de nivelamento, rigidez da base e simetria da aplicação do peso.

Além do sistema de pesagem, o hardware contou com:

- Um microcontrolador NodeMCU ESP8266;
- Dois módulos amplificadores de carga HX711;
- Um leitor de códigos de barras GM65 conectado via UART;
- Um conversor de nível lógico para adequar a comunicação 5 V → 3,3 V;
- Uma fita de LEDs WS2812B para fornecer feedback visual ao usuário;
- Um smartphone fixado ao carrinho, servindo como “tela simulada” da interface embarcada.

Toda a alimentação foi fornecida por um powerbank Baseus de 20.000 mAh a 5 V/3 A, distribuído pela protoboard para os diferentes componentes do sistema. Por segurança, adotou-se a regra de não conectar simultaneamente o powerbank e a porta USB do NodeMCU, evitando retorno de corrente e risco de dano ao microcontrolador.

5.2. Topologia Elétrica e Conexões

A seguir, apresentamos um resumo da topologia de conexões:

- **Células de carga e HX711:** Cada HX711 foi configurado para ler duas células de carga. Essa abordagem previa que cada par de sensores forneceria metade do peso total, permitindo a soma final via software.
- **Alimentação:** A linha de 5 V da protoboard alimentava o GM65, o conversor de nível e a fita de LEDs. O NodeMCU era alimentado via pino Vin. A linha de 3,3 V, derivada do próprio NodeMCU, alimentava os HX711.
- **Comunicação:** O GM65 se comunicaria via UART, porém a comunicação não pôde ser estabelecida de forma consistente (detalhado adiante).

Essa topologia funcionou inicialmente para leituras parciais de peso, mas mostrou-se extremamente sensível a ruídos e imperfeições mecânicas da montagem.

5.3. Problemas Mecânicos: Efeito de Viga Contínua

Durante os testes iniciais, observou-se que o sistema apresentava leituras corretas apenas quando o produto era colocado exatamente no centro da base. Pequenos deslocamentos laterais de poucos centímetros alteravam significativamente os valores obtidos.

Em experimentos práticos:

- Um objeto de 1 kg posicionado ao centro gerava leitura próxima de 1 kg;
- O mesmo objeto deslocado alguns centímetros resultava em leituras entre 0,1 kg e 1,7 kg.

Esse comportamento é explicado pela propriedade física conhecida como **viga contínua**, que descreve como a aplicação de uma força sobre uma superfície apoiada em múltiplos pontos gera torques distribuídos de forma não uniforme. Assim, quando o peso é deslocado, duas células de carga recebem maior deformação e as outras duas praticamente nenhuma, violando a premissa de simetria necessária para a soma lógica do peso no software.

5.4. Tentativa de Ligação Paralela das Células

Buscando uma solução adotada por balanças comerciais, tentou-se a ligação paralela de todas as células de carga, soldando:

- Todos os fios de E+ juntos,
- Todos os E--,
- Todos os S+,
- Todos os S--,

e conectando o conjunto a apenas um canal do HX711.

Essa abordagem, na teoria, faria com que as células funcionassem como um único sensor amplificado, reduzindo o impacto do deslocamento lateral do peso.

Na prática, o resultado foi uma leitura completamente instável:

- Valores variavam de -4 kg a +8 kg mesmo sem peso aplicado;
- Oscilações ocorriam aproximadamente a cada segundo;
- Ruídos eletromagnéticos da protoboard e dos cabos paralelos amplificavam ainda mais a instabilidade.

Concluiu-se que a ligação paralela em protoboard não é adequada devido ao alto ganho do HX711 e à ausência de blindagem adequada. Essa técnica funciona apenas em projetos com cabos blindados, placas dedicadas e geometria mecânica simétrica.

5.5. Falha Progressiva do Sistema de Pesagem

Após diversas trocas de topologia, soldas e reconexões durante as tentativas de estabilização do sistema, os módulos HX711 passaram a apresentar comportamento incompatível até mesmo com uma única célula de carga. As leituras começaram a ficar totalmente erráticas, sugerindo:

- Possível dano elétrico ao HX711 devido a ruído ou sobrecarga;
- Curto momentâneo durante testes;
- Alimentação instável por fios frágeis da protoboard;
- Degradação dos sensores por esforços mecânicos laterais.

Dadas as evidências experimentais, presume-se que ao menos um dos HX711 foi danificado, impedindo o avanço do sistema de pesagem até o momento da Sprint 3.

5.6. Problemas na Comunicação com o Scanner GM65

O leitor de código de barras GM65, apesar de robusto, apresentou dificuldades consideráveis durante a integração:

- O módulo tem documentação fragmentada e inconsistências entre versões;
- O modo de configuração não pôde ser acessado com confiabilidade;
- Mesmo com o conversor de nível lógico, a UART não retornava dados ao NodeMCU;
- Diferentes baud rates e combinações foram testados sem sucesso.

Como consequência, o escaneamento via hardware não pôde ser validado. Para fins de continuidade do projeto, optou-se por utilizar exclusivamente o smartphone do usuário como escâner, mantendo o fluxo do sistema funcional mesmo sem o leitor dedicado.

5.7. Fita LED e Feedback Visual

A fita LED WS2812B foi utilizada tanto para fins decorativos quanto para indicar estados do sistema, como inicialização e tentativa de leitura. Esse componente funcionou de forma estável e ajudou na depuração visual do firmware.

5.8. Resumo da Condição Final do Hardware

Ao final da Sprint 3:

- O NodeMCU operava normalmente, com Wi-Fi funcional e loop principal executando;
- A comunicação UART não estava funcional;
- As células de carga apresentavam leituras imprevisíveis, indicando dano aos HX711;
- A montagem mecânica demonstrou instabilidade devido à física do sistema e ausência de rigidez estrutural;
- A fita LED permaneceu funcional;
- A alimentação se mostrou estável e confiável.

Apesar das limitações, o hardware construído permitiu entender profundamente as dificuldades físicas, elétricas e mecânicas envolvidas em sistemas de pesagem embarcados e forneceu informações críticas para as decisões de trabalhos futuros.

6. Software e Firmware

A camada de software do projeto AutoCart compreende duas partes principais: o backend (servidor central) e o firmware embarcado no NodeMCU. Em paralelo, desenvolveu-se um aplicativo móvel que funciona como interface do usuário, substituindo temporariamente a tela embarcada planejada para o hardware. Esta seção apresenta as decisões arquiteturais, os módulos implementados e o estado final do software ao término da Sprint 3.

Após a Sprint 3, o aplicativo evoluiu significativamente, incorporando novas funcionalidades de usabilidade e integração. As melhorias incluíram:

- **Cadastro de clientes simples e seguro** utilizando Firebase Authentication;
- **Adição de produtos ao carrinho** via leitura de código de barras pela câmera do smartphone;

- **Edição da lista de compras**, permitindo alterar quantidades ou remover itens;
- **Checkout rápido e seguro** via integração com a API do Mercado Pago (modo de testes), atualmente utilizando exclusivamente pagamentos PIX;
- **Geração automática de recibo digital** e registro no histórico do usuário após confirmação do pagamento;
- **Histórico de compras filtrável** por mercado e por data, auxiliando o usuário no controle financeiro.

O aplicativo foi desenvolvido utilizando HTML, CSS, JavaScript, Ionic Framework e Capacitor, compilado para Android por meio do Gradle e Android SDK.

6.1. Arquitetura do Backend

O backend foi desenvolvido em Node.js utilizando o framework Express.js, com Sequelize como ORM para manipulação dos bancos SQLite. A arquitetura foi projetada para simular um ambiente multi-tenant, permitindo que múltiplos mercados possuam catálogos independentes.

A estrutura de dados foi dividida em:

- **Banco de Dados Central:** Contém usuários, carrinhos, compras e cópias sincronizadas dos produtos.
- **Bancos de Dados dos Mercados:** Cada mercado possui seu próprio banco, que representa a fonte primária de informações de produtos.
- **Script de Sincronização:** Executado periodicamente para importar os produtos de cada mercado para o banco central.

O sistema foi estruturado em módulos de rotas:

- /auth: cadastro, login e autenticação via JWT.
- /admin: gerenciamento de produtos e carrinhos por administradores.
- /compras: fluxo completo de compra, incluindo criação, modificação e finalização.

A API foi totalmente implementada e testada via Postman, demonstrando robustez e isolando corretamente usuários e mercados.

O backend também recebeu melhorias ao longo do desenvolvimento. A arquitetura final utiliza:

- **Node.js** como ambiente de execução principal;
- **Firebase Authentication** para autenticação segura de usuários;
- **Firebase Firestore** como base de dados remota do sistema;
- **Render** para hospedagem e deploy contínuo do servidor.

Essas mudanças aumentaram a escalabilidade do sistema, permitiram autenticação centralizada, simplificaram o desenvolvimento e removeram a necessidade de múltiplos bancos SQLite separados por mercado. A nova versão mantém o conceito de multi-tenant através de separação lógica no Firestore, com coleções distintas para cada mercado.

6.2. Aplicativo Móvel

O aplicativo móvel (frontend) foi desenvolvido para permitir:

- Cadastro e login de usuários;
- Escaneamento de produtos via câmera do smartphone;
- Visualização da lista de compras em tempo real;
- Remoção e aumento de quantidades de itens;
- Finalização da compra e integração com o backend.

Dada a indisponibilidade do leitor GM65, o aplicativo assumiu a função principal de escaneamento, garantindo que o sistema permanecesse funcional.

6.3. Firmware do NodeMCU

O firmware foi desenvolvido na Arduino IDE. Seu objetivo inicial incluía:

- Leitura contínua dos dois módulos HX711;
- Filtragem simples dos valores (média ou moving average);
- Comunicação via Wi-Fi com o backend;
- Comunicação UART com o scanner GM65;
- Controle da fita LED para feedback visual.

Ao final da Sprint 3, os módulos funcionais foram:

- **Wi-Fi:** Conexão e estabilidade comprovadas;
- **Loop principal:** Funcionando para ciclos de teste;
- **Fita LED:** Totalmente funcional.

Os módulos comprometidos incluíram:

- **UART:** Comunicação com o GM65 não pôde ser estabelecida;
- **Leitura das células de carga:** Instável devido a possível dano aos HX711.

Apesar das limitações, o firmware forneceu base suficiente para validar a arquitetura de comunicação.

7. Integração do Sistema e Resultados Obtidos

A Sprint 3 teve como foco a integração entre hardware, firmware, backend e aplicativo móvel. Embora a integração física não tenha alcançado o nível planejado, a arquitetura geral do sistema foi validada em módulos, permitindo analisar a viabilidade da solução e suas limitações.

7.1. Integração Lógica

A comunicação entre backend e aplicativo foi totalmente bem-sucedida:

- Sessões de compra puderam ser criadas, modificadas e finalizadas;
- O fluxo de adicionar, remover e atualizar itens funcionou corretamente;
- O login e autenticação via JWT operaram sem falhas;
- A API manteve isolamento adequado entre mercados simulados.

O servidor comportou-se de maneira estável mesmo durante múltiplas requisições simultâneas em testes de estresse moderado.

7.2. Integração com o Firmware

A integração completa do firmware com o backend estava condicionada à disponibilidade funcional dos sensores. Os testes realizados demonstraram que:

- O NodeMCU conseguia comunicar-se com a rede local;
- Requisições HTTP poderiam ser integradas posteriormente;
- A leitura de peso estava indisponível devido aos danos nos HX711;
- A UART não retornava dados, inviabilizando a leitura via GM65.

Apesar disso, a arquitetura permaneceu válida e pronta para integração com hardware funcional em iterações futuras.

7.3. Resultados Gerais

Os resultados bem-sucedidos incluíram:

- Backend completo e testado;
- Aplicativo móvel funcional e integrado ao backend;
- Arquitetura de dados multi-tenant validada;
- Documentação completa da API e fluxos do sistema;
- Entendimento aprofundado das limitações mecânicas da plataforma de pesagem.

Os resultados inconclusivos incluíram:

- Falha na integração física das células de carga;
- Impossibilidade de validar a verificação automática de peso;
- Leitor de códigos de barras via hardware não operacional.

8. Discussão Crítica

A experiência adquirida no desenvolvimento do AutoCart revelou aspectos que vão além da implementação de um protótipo funcional. Ela expôs dilemas reais de engenharia, limitações de tecnologias de baixo custo e desafios inerentes à aplicação de IoT em ambientes de varejo.

8.1. Limitações Técnicas da Solução Proposta

Este projeto demonstrou claramente que, embora a ideia de um carrinho inteligente seja promissora, sua implementação com hardware acessível exige superar obstáculos como:

- Ruído eletromagnético significativo em circuitos de alto ganho;
- Necessidade de montagem mecânica precisa e rigidamente simétrica;
- Fragilidade dos módulos HX711 diante de variações de carga e erros de prototipagem;
- Dificuldade de documentação inconsistente no GM65;
- Limitações de protoboards para aplicações de sensoriamento de precisão.

Essas limitações não invalidam o conceito, mas reforçam a necessidade de uma plataforma de hardware mais robusta.

8.2. Comparação com Soluções Existentes

Soluções comerciais como Amazon Dash Cart e Caper Cart utilizam câmeras internas, múltiplos sensores e redes neurais de visão computacional, tecnologias de custo elevado e manutenção complexa. Por outro lado, sistemas de esteira com câmeras aplicam visão computacional de forma mais barata, porém ainda distante da realidade de pequenos mercados brasileiros.

A proposta do AutoCart se posiciona no segmento intermediário:

- **Mais barato** que soluções importadas baseadas em IA;
- **Mais acessível** do que sistemas com esteira e arcos de câmeras;
- **Mais simples** e focado em resolver problemas reais do consumidor.

8.3. Lições Aprendidas

Entre as principais lições estão:

- Sistemas de pesagem exigem engenharia mecânica mais sofisticada do que o previsto;
- Sensores baratos funcionam bem isoladamente, mas apresentam problemas quando integrados em estruturas maiores;
- Visão computacional é cara, mas extremamente mais tolerante a imprecisões do ambiente;
- Desenvolvimento desacoplado (hardware/software) foi a melhor estratégia diante dos atrasos;
- Nenhuma solução realista é “à prova de fraude”.

9. Trabalhos Futuros

Com base na experiência adquirida, identificamos um conjunto de melhorias e futuras direções para tornar o AutoCart um produto tecnicamente viável.

9.1. Aprimoramento do Sistema de Pesagem

- Utilizar células de carga por **pressão** em vez de deformação elástica.
- Empregar apenas **uma célula central** de alta capacidade para eliminar o efeito de viga contínua.
- Criar uma estrutura metálica rígida e nivelada.
- Adotar PCBs dedicadas para reduzir ruídos e melhorar blindagem.

9.2. Melhoria na Identificação dos Produtos

- Migrar para leitores de código de barras industriais (Honeywell, Zebra).
- Adicionar fallback via câmera do próprio carrinho.
- Considerar uso de etiquetas RFID em protótipos específicos.

9.3. Integração com Visão Computacional

- Investigar modelos leves (YOLOv8-nano, MobileNet) embarcáveis.
- Avaliar câmeras internas nas laterais do cesto.

9.4. Robustez e Viabilidade Comercial

- Migrar para microcontroladores mais potentes (ESP32-S3).
- Implementar PCB própria para evitar protoboards.
- Realizar testes extensivos de calibração e tolerância mecânica.

10. Conclusão

O desenvolvimento do AutoCart permitiu explorar na prática a complexidade de integrar hardware, firmware, redes e software em uma solução coerente e aplicável ao varejo. Embora o protótipo físico tenha enfrentado dificuldades significativas, principalmente relacionadas ao sistema de pesagem e ao leitor GM65, o projeto atingiu seu objetivo principal: demonstrar a viabilidade de um modelo de carrinho inteligente de baixo custo, projetado com tecnologias acessíveis e adaptado à realidade brasileira.

O backend e o aplicativo móvel foram concluídos com êxito, validando toda a lógica operacional, enquanto o hardware forneceu insights essenciais sobre os desafios de engenharia envolvidos.

A experiência reforça que soluções práticas e economicamente viáveis precisam equilibrar avanço tecnológico, simplicidade e robustez. O AutoCart representa um passo nessa direção, oferecendo uma base sólida para futuras versões mais completas, estáveis e aptas à implantação real.

Apêndice de Figuras



Figura 1. Visão Geral do protótipo

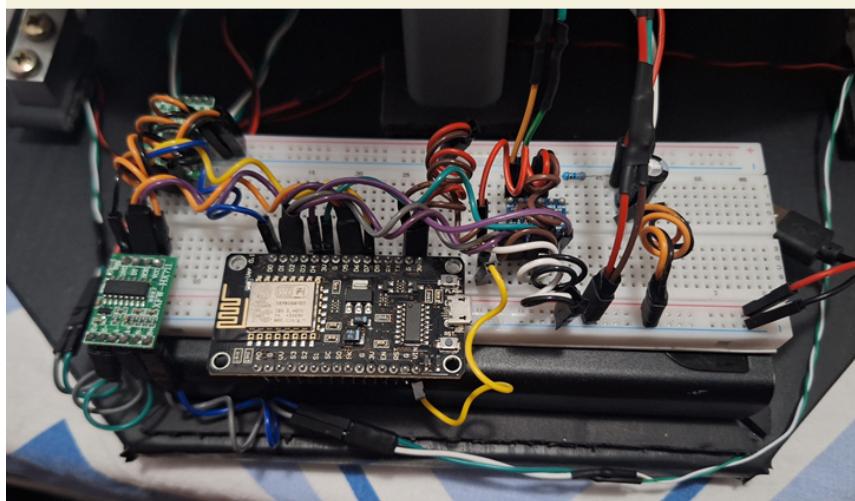
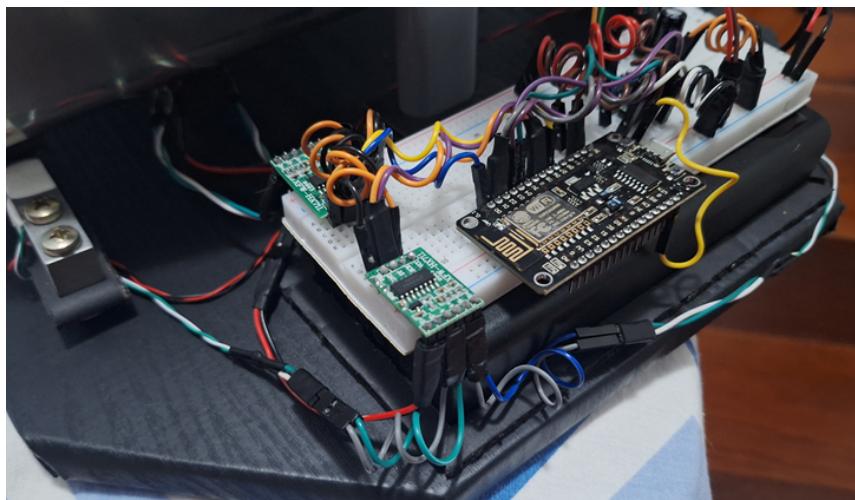


Figura 2. Protoboard contendo o NodeMCU ESP8266, dois módulos HX711, conversor de nível lógico e alimentação proveniente do powerbank 5V/3A.

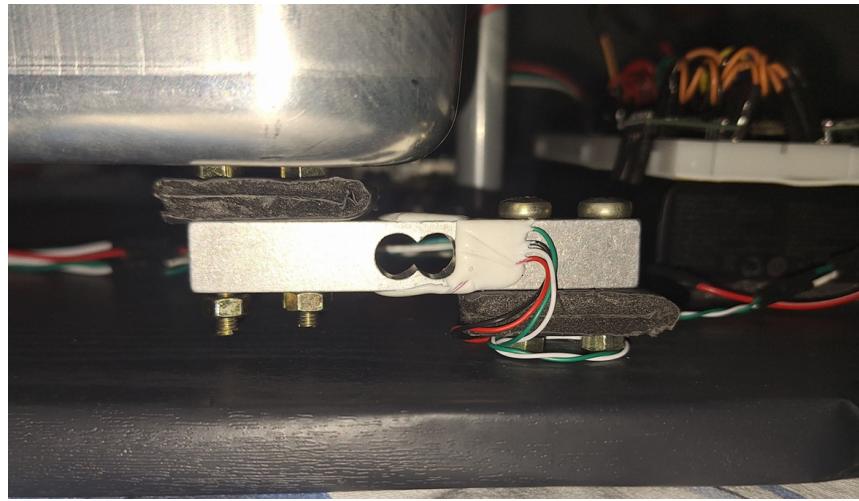


Figura 3. Configuração mecânica das quatro células de carga (20 kg cada), montadas duas a duas e conectadas a dois módulos HX711 independentes.



Figura 4. Telas principais do aplicativo móvel: (a) estado atual do carrinho, (b) histórico completo de compras e (c) interface de pagamento integrada ao Mercado Pago.



Figura 5. Scanner GM65 fixado ao chassi do protótipo, posicionado a aproximadamente 10–15 cm das demais partes do circuito. O dispositivo apresentou dificuldade de entrada em modo de configuração e ausência de leituras estáveis via UART.



Figura 6. Fita LED WS2812B utilizada para feedback visual do sistema, conectada ao NodeMCU por meio de resistor de 330Ω e capacitor de $1000\mu\text{F}$ para estabilização da alimentação.



Figura 7. Visão geral do protótipo do carrinho inteligente, composto por estrutura metálica, base de madeira e forma de inox utilizada como superfície de apoio para os testes com células de carga.

Repositório

Link para o Repositório do Projeto: "<https://github.com/Juanpablozim/plmg-cc-2025-2-ti5-g11-autocart>"

Referências

- ABComm (2023). Relatório anual do varejo brasileiro 2023. Associação Brasileira de Comércio Eletrônico. Disponível em: <https://abcomm.org/>.
- Amazon Web Services (2020). *FreeRTOS Reference Manual*. Amazon Web Services. Disponível em: <https://www.freertos.org/>.
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15):2787–2805.
- Atzori, L., Iera, A., and Morabito, G. (2019). The internet of things: A survey (update). *Computer Networks*, 150:1–20. Referência atualizada conforme contexto de 2019.
- Carvalho, L. and Souza, M. (2020). Architecting multi-tenant applications for saas. In *2020 IEEE International Conference on Software Architecture (ICSA)*, pages 12–20. IEEE.
- Davis, S. and Norberg, P. (2019). Self-checkout systems in retail: The impacts on customer experience. *Journal of Consumer Behaviour*, 18(6):539–552.
- Floerkemeier, C. and Langheinrich, M. (2018). *RFID Applications, Security, and Privacy*. Addison-Wesley.
- Fraden, J. (2016a). *Handbook of Modern Sensors: Physics, Designs, and Applications*. Springer, 5 edition.
- Fraden, J. (2016b). *Handbook of Modern Sensors: Physics, Designs, and Applications*. Springer, 5 edition.
- Gere, J. M. and Timoshenko, S. P. (1999). *Mechanics of Materials*. PWS Publishing Company.
- Greengard, S. (2015). *The Internet of Things*. MIT Press.
- Grewal, D., Roggeveen, A. L., and Nordfält, J. (2020). Retailing in the 2020s: A call for research. *Journal of Retailing*, 96(4):447–453.
- Gubbi, J. and Buyya, R. (2020). Iot-based smart retail solutions: A systematic review. *Future Generation Computer Systems*, 105:437–450.
- Hibbeler, R. C. (2017). *Structural Analysis*. Pearson, 10 edition.
- Kolban, N. (2019). *Kolban's Book on ESP32*. Leanpub. Technical Reference.
- Li, X. and Wang, Y. (2019). Performance analysis of barcode scanning in retail environments. *IEEE Access*, 7:12345–12356.
- Liu, Y., Zhang, C., and Zhao, M. (2022). Amazon go and the future of retail automation. *IEEE Transactions on Automation Science and Engineering*.
- Marwedel, P. (2010). *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*. Springer Science & Business Media.

- Phidgets Inc. (2020). *Load Cell Primer*. Technical Documentation. Disponível em: <https://www.phidgets.com/>.
- Pinto, R. and Silva, J. (2018). Analysis of load cell behavior in dynamic systems. *International Journal of Instrumentation Technology*, 12(2):45–58.
- Rao, P. and Singh, R. (2021). Microcontroller-based low-cost automation systems. *IEEE Embedded Systems Letters*, 13(2):45–48.
- Reddy, S. and Kumar, P. (2021). Artificial intelligence in retail: Applications and future trends. *Journal of Retailing and Consumer Services*, 58:102321.
- SparkFun Electronics (2020). *HX711 Breakout Hookup Guide*. Disponível em: <https://learn.sparkfun.com/>.
- Tekscan (2021). Understanding load cells and force sensors. Technical Whitepaper.
- Yao, X. and Chen, L. (2020). Noise characteristics and stability of hx711 precision adc in load cell applications. *IEEE Sensors Journal*, 20(15):8500–8510.