

1. Análise de Contexto

Atualmente, redes de franquias enfrentam desafios relacionados à falta de integração entre loja e estoque, dessa forma ocasionando:

- **Controle manual de inventário** (erros e perdas de produtos).
- **Baixa rastreabilidade** de vendas e estoque em tempo real.
- **Dificuldade na padronização** dos processos entre diferentes franquias.
- **Ausência de monitoramento automatizado**, o que reduz a eficiência operacional.

Oportunidade: implementar um sistema de automação integrada que permita gerenciar **loja + estoque + central da franquia** em tempo real.

2. Definição do Problema e Escopo

Problema: Falta de integração automatizada nos processos de venda e estoque em franquias.

Escopo do Projeto:

- Controle de estoque automático com sensores de entrada, saída e de peso.
- Dashboard centralizado para gestão da rede de franquias.
- Monitoramento remoto de lojas (estoque mínimo, alertas de reposição).
- Controle automático de ambiente, incluindo ar-condicionado e luzes.

Justificativa Técnica: O uso de sistemas embarcados, banco de dados centralizado e conectividade IoT garante eficiência, redução de erros, escalabilidade e padronização entre franquias.

3. Arquitetura Inicial da Solução

- **Módulo Loja:**
 - Ponto de Venda (PDV) conectado ao sistema.
 - **Sensores de peso nas prateleiras/gôndolas** para monitoramento contínuo de produtos.
- **Módulo Estoque:**
 - Entrada e saída de produtos monitorada.
 - Alertas automáticos quando o peso ficar abaixo do nível mínimo configurado.
- **Módulo Central:**
 - Banco de dados em nuvem.
 - Dashboard de gestão com relatórios de vendas, estoque em tempo real e sugestão de reabastecimento.

4. Topologia da Rede de Sensores e Atuadores

- Sensores de peso instalados em prateleiras e debaixo de palete, sensores de temperatura e luminosidade instalados pela loja para monitorar e automatizar o controle do ar-condicionado e das luzes.
- Microcontrolador (ex.: Arduino/DBX-MIO com Wi-Fi) coleta os dados.
- Comunicação via Wi-Fi para o servidor central, utilizando protocolo UDP para enviar os dados para o servidor, que depois vai enviar os dados para o banco de dados por protocolo HTTP.
- Atuadores: sistema envia alerta de reabastecimento para funcionário ou gestor.

5. Estratégia de Conectividade

- Sensores → Arduino/DBX-MIO → Wi-Fi/UDP → Servidor → HTTP → Banco de Dados na Nuvem.
- PDV também envia dados das vendas → sincronização com estoque.
- Dashboard central recebe ambos os dados para comparação (estoque físico x vendas registradas).

6. Product Backlog

- **Sprint 1:** Projetar/Implementar leitura e comandos com o sensor de temperatura e controle infravermelho do ar-condicionado.
- **Sprint 2:** Projetar/Implementar leitura e comandos com o sensor de luminosidade e controle de LED dimerizável.
- **Sprint 3:** Implementar leitura de sensores de peso (carga por produto), e criar lógica de cálculo para identificar quantidade em estoque a partir do peso.
- **Sprint 4:** Gerar alertas automáticos de reabastecimento.
- **Sprint 5:** Geração de logs, armazenamento de dados e tomadas de decisões mais difíceis com o DBX-MIO e o servidor, a partir do sensor de temperatura.
- **Sprint 6:** Geração de logs, armazenamento de dados e tomadas de decisões mais difíceis com o DBX-MIO e o servidor, a partir do sensor de luminosidade.
- **Sprint 7:** Desenvolvimento de sistema interativo de controle de estoque e prateleiras.
- **Sprint 8:** Geração de logs, armazenamento de dados e tomadas de decisões mais difíceis com o DBX-MIO e o servidor, a partir do sensor de peso.

7. Product Backlog estruturado com base no framework Scrum (tarefas priorizadas e descritas)

Sprint 1 — Controle de Ambiente (2 semanas)

História de Usuário:

“Como gestor da loja, quero monitorar e automatizar o ar-condicionado para manter o ambiente adequado para não estragar produtos e reduzir desperdício de energia.”

Tarefas:

1. Pesquisar e escolher sensor de temperatura adequado.
2. Configurar microcontrolador (Arduino/DBX-MIO) para leitura do sensor.
3. Implementar código para coleta e envio dos dados via Wi-Fi/UDP.
4. Configurar módulo infravermelho para enviar comandos ao ar-condicionado.
5. Testar integração: sensor detecta temperatura → sistema aciona ar-condicionado.

Sprint 2 — Controle de Iluminação (2 semanas)

História de Usuário:

“Como gestor da loja, quero automatizar a iluminação para economizar energia e manter o ambiente agradável aos clientes.”

Tarefas:

1. Selecionar e configurar sensor de luminosidade.
2. Implementar leitura dos níveis de luz ambiente no microcontrolador.
3. Programar atuador para controle de LED dimerizável.
4. Integrar sistema de luminosidade com base em horários e presença de clientes.
5. Validar funcionamento e ajustes de intensidade automática.

Sprint 3 — Monitoramento de Estoque com Sensores de Peso (3 semanas)

História de Usuário:

“Como gestor, quero acompanhar o estoque em tempo real através de sensores de peso, para reduzir erros e otimizar o reabastecimento.”

Tarefas:

1. Selecionar load cells e módulo HX711 para prototipagem.
2. Calibrar sensores para diferentes tipos de produtos.
3. Implementar código para cálculo da quantidade em estoque com base no peso.
4. Enviar dados dos sensores via Wi-Fi para o servidor.
5. Validar precisão dos sensores em diferentes condições de carga.

Sprint 4 — Alertas e Dashboard Central (3 semanas)

História de Usuário:

“Como gestor de franquia, quero receber alertas de reabastecimento e visualizar relatórios centralizados para tomar decisões rápidas.”

Tarefas:

- 1. Implementar lógica para identificar quando o estoque está abaixo do limite mínimo.**
- 2. Desenvolver sistema de envio de alertas (notificação/app/e-mail).**
- 3. Estruturar banco de dados em nuvem para armazenar histórico.**
- 4. Criar dashboard central para monitoramento de lojas em tempo real.**
- 5. Testar integração entre PDV (vendas) e sensores (estoque físico).**

8. Definição dos papéis da equipe

- Bruno Costa → CIO
- Vitor Tokunaga → Gerente de TI/Sistemas
- João Vitor Leao → Desenvolvedores
- Felipe Martins → Desenvolvedores
- Alexsander Sudario → Desenvolvedores

9. Referencial técnico e benchmarking

1. Sensores e Hardware

- **Sensores de Peso (Load Cells + HX711):** amplamente utilizados em balanças eletrônicas e sistemas de monitoramento de carga. São baratos, precisos e já possuem bibliotecas consolidadas para Arduino/ESP32.
- **ESP32:** microcontrolador com Wi-Fi e Bluetooth integrados, ideal para IoT, com suporte a protocolos MQTT e HTTP.
- **Arduino (Uno/Nano):** opção mais simples para prototipagem inicial.

Padrões e bibliotecas:

- Biblioteca **HX711** para leitura de sensores de peso.
- Protocolos de comunicação: **MQTT** (para IoT em tempo real) e **HTTP/REST** (para integração com dashboards).

3. Benchmarking de Soluções Similares

- **Amazon Go:** utiliza sensores e visão computacional para monitorar estoque e vendas em tempo real. Embora complexo e caro, inspira a ideia de automação autônoma.

- **Vending Machines Inteligentes:** já utilizam sensores de peso para detectar retirada de produtos e calcular estoque automaticamente.
- **Franquias alimentícias (Starbucks, McDonald's):** fazem integração PDV ↔ Estoque, mas dependem de input manual, diferente da proposta com sensores automáticos.
- **Soluções IoT em supermercados (Ex.: Pão de Açúcar Mais):** usam etiquetas eletrônicas e RFID, mas têm custo elevado comparado aos sensores de peso.