



Controle de Estoque

**Documentação do Sistema de Gerenciamento de Estoque
(MOBILE)**

**Dupla: Maria Fernanda Sombra Costa (Nº23)
Pâmela Esther Silva Pereira (Nº28)
Desenvolvimento de Sistemas 3**

**Professores: Carlos Renan Moreira
Eduardo Alves Almeida**

Sumário

1. Objetivo do Software	3
2. Público-alvo	3
3. Funcionalidades Implementadas	4
4. Execução do projeto	4
5. Estrutura do Banco de Dados	5
6. Integração	5
Fluxo de Integração	6
Autenticação com Laravel Sanctum	6
7. Projeto	6

1. Objetivo do Software

O sistema mobile tem como principal objetivo facilitar o controle de estoque em cosmeterias, com foco especial em perfumarias, otimizando a gestão de produtos como perfumes, cosméticos, maquiagens e itens de higiene pessoal. A plataforma visa oferecer uma solução prática, intuitiva e acessível, permitindo que os responsáveis pelo estoque registrem entradas e saídas de produtos, acompanhem o nível de reposição e gerenciem informações detalhadas sobre os itens comercializados.

Além disso, o sistema contribui para evitar perdas por vencimento, excesso ou falta de produtos, auxiliando o empreendedor na tomada de decisões mais assertivas, aumentando a eficiência operacional e, consequentemente, a lucratividade do negócio.

2. Público-alvo

O público-alvo do sistema são:

- Pequenos e médios empreendedores do ramo de cosméticos, especialmente proprietários de perfumarias e lojas de produtos de beleza.
- Gerentes e responsáveis pelo estoque que necessitam de uma ferramenta ágil para controlar inventário em tempo real, diretamente pelo celular.
- Microempreendedores Individuais (MEIs) que atuam na revenda de cosméticos e desejam um controle mais profissional e organizado de seus produtos.
- Equipes de vendas ou colaboradores que auxiliam no controle de mercadorias em estabelecimentos físicos.

O aplicativo foi pensado para atender um público que, muitas vezes, não possui familiaridade com sistemas complexos de gestão, e por isso, valoriza a simplicidade, mobilidade e eficiência.

3. Funcionalidades Implementadas

O sistema mobile de controle de estoque para perfumarias está em desenvolvimento com uma estrutura sólida e moderna, utilizando tecnologias eficientes e integradas. A seguir, estão listadas as funcionalidades e integrações já implementadas até o momento:

- Interface mobile pronta com FlutterFlow: O front-end do sistema foi desenvolvido com o FlutterFlow, garantindo uma experiência visual amigável, responsiva e intuitiva para o usuário final.
- Banco de dados configurado no Supabase: Toda a estrutura de dados está hospedada e operacional no Supabase, permitindo um armazenamento seguro, escalável e de fácil gerenciamento.
- Back-end em Laravel estruturado: A API REST está sendo construída com Laravel, framework robusto e seguro, garantindo a comunicação eficiente entre o app e o banco de dados.
- Primeiro CRUD integrado com sucesso: A primeira funcionalidade de cadastro, leitura, atualização e exclusão de dados (CRUD) já foi conectada com sucesso, validando a integração entre o front, o back-end e o banco de dados.
- Integração em andamento do segundo CRUD e login com token via Laravel Sanctum:
 - Está em fase de implementação o segundo CRUD, seguindo o mesmo padrão de integração e segurança.
 - Também está em desenvolvimento o sistema de autenticação com geração de token utilizando middleware com Laravel Sanctum, o que permitirá o login seguro de usuários e o acesso autenticado às funcionalidades do sistema.

4. Execução do projeto

1. Abrir o frontend pelo FlutterFlow, através do link:
2. Clone o repositório que armazena o backend do projeto através do link:
https://github.com/Pereirapam/api_mobile2.git
3. Instale as dependências do backend no seu computador, com os comandos:
 - a. `npm install`

- b. composer install
 - c. php artisan key:generate
- 4. Executa a API com os comandos
 - a. php artisan serve
 - b. npm run dev
 - c. colocar o comando do ngrok estático

5. Estrutura do Banco de Dados

- 1. Tabela: users
 - a. id(int);
 - b. name(varchar);
 - c. email(varchar);
 - d. email_verified_at(timestamp);
 - e. password(varchar);
 - f. remember_token(varchar);
 - g. created_at(timestamp);
 - h. updated_at(timestamp);
- 2. Tabela: marks
 - a. id(int);
 - b. markName(text);
 - c. created_at(timestamp);
 - d. updated_at(timestamp);
- 3. Tabela: products
 - a. id(int);
 - b. name(varchar);
 - c. description(varchar);
 - d. price(numeric);
 - e. expiration_date(date);
 - f. quantity(int);
 - g. markName_id;

Link do Banco de Dados alocado no Supabase:

[postgresql://postgres:\[YOUR-PASSWORD\]@db.ulephtxgwrbjgnqraufg.supabase.co:5432/postgres](postgresql://postgres:[YOUR-PASSWORD]@db.ulephtxgwrbjgnqraufg.supabase.co:5432/postgres)

6. Integração

O sistema mobile se comunica com o backend Laravel por meio de uma API REST, consumida diretamente pelo FlutterFlow. O Laravel, por sua vez, está integrado ao Supabase, onde o banco de dados PostgreSQL está hospedado. Essa arquitetura garante segurança, organização e escalabilidade.

Fluxo de Integração

FlutterFlow (aplicativo mobile)



Laravel (API REST com Sanctum)

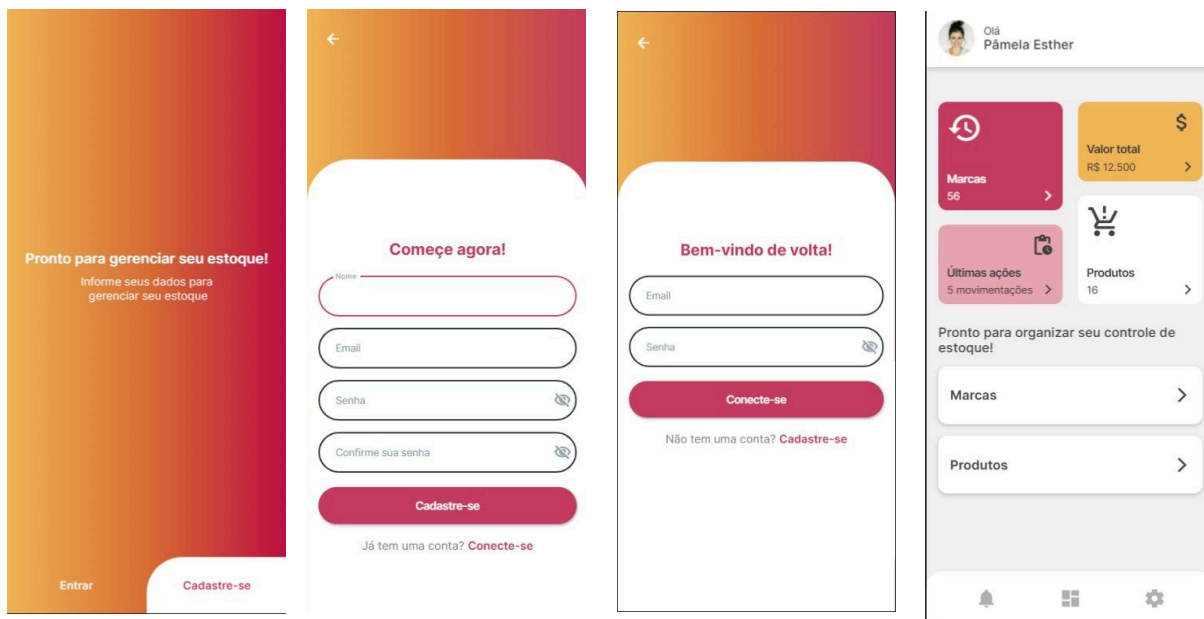


Supabase (banco de dados PostgreSQL)

Autenticação com Laravel Sanctum

O backend utiliza o Laravel Sanctum para autenticação. Quando um usuário realiza login, um token de acesso é gerado e enviado ao aplicativo. Esse token é utilizado nas requisições autenticadas subsequentes, garantindo segurança no acesso aos dados.

7. Projeto



←

Produtos cadastrados

Criar

Nome	Descrição	Preço	Quantidade
Malbec	Perfume	200.00	10
Lily	Perfume	299.90	5
Instance	Hidratante corporal	54.99	24

←

Edite o produto

Nome

Malbec

Descrição

Perfume

Preço

200

Quantidade

10

Atualizar

←

Marcas cadastradas

Criar

O Boticário

Natura

Eudora


←

Adicione uma nova marca

Marca

Adicionar

←



Pâmela Esther
pam@gmail.com

Edite sua conta

Sair da conta

