



Roadmap de Desenvolvimento e Implementação - App Dose Certa

Este roadmap foi estruturado utilizando a metodologia **SCRUM** para garantir um desenvolvimento ágil, iterativo e focado em entregar valor contínuo ao usuário. O projeto será dividido em Sprints de 2 semanas.



Sprint 0: Configuração e Planejamento

Objetivo: Preparar todo o ambiente de desenvolvimento, definir a arquitetura inicial e validar as ferramentas.

Tarefa	Branch Git	Descrição
Configuração do Ambiente	chore/setup-environment	Criação do pyproject.toml (ou requirements.txt), configuração do ambiente virtual e instalação das dependências iniciais (Flet, etc).
Estrutura de Pastas	chore/project-structure	Definição da arquitetura de pastas do projeto (ex: app/, database/, views/, tests/).
Integração com Git/GitHub	chore/git-setup	Configuração do .gitignore e definição das regras de branch e commit para o projeto.
Definição do Esquema do BD	feat/database-schema	Desenho do esquema inicial do banco de dados SQLite, incluindo tabelas para usuários, itens, locais de estoque, etc.

Sprint 1: Autenticação e Onboarding do Usuário

Objetivo: Implementar a tela de login e o fluxo inicial de cadastro do usuário e estabelecimento (versão SQLite).

Tarefa	Branch Git	Descrição
Tela de Login (UI)	feat/login-screen-ui	Criação da interface gráfica da tela de login, com campos para usuário/senha e botão de login com Google (visual).
Lógica de Autenticação (SQLite)	feat/sqlite-auth-logic	Implementação da lógica para criar um usuário e "logar" localmente. Se o banco não existir, ele será criado.
Pré-cadastro de Bebidas	feat/seed-database	Criação de um script para popular o banco de dados SQLite com a lista inicial de bebidas da planilha.
Tela de Onboarding (Cadastro)	feat/onboarding-flow	Após o primeiro login, criar a view para solicitar Nome do Usuário, Nome do Estabelecimento, E-mail, WhatsApp e Local de Contagem.

Sprint 2: Dashboard e Navegação Principal

Objetivo: Construir a tela principal (Dashboard) e a navegação entre as diferentes seções do app.

Tarefa	Branch Git	Descrição
UI do Dashboard	feat/dashboard-ui	Desenho da tela principal com os botões "Cadastros", "Contagem de Estoque", "Incluir Itens" e "Relatórios".
Header Personalizado	feat/custom-header	Exibição do Nome do Usuário e do

		Estabelecimento no topo da tela.
Alternador de Tema	feat/theme-switcher	Implementação do botão para alternar entre os temas claro e escuro.
Sistema de Roteamento/Views	feat/app-routing	Implementação da lógica de navegação para que os botões do Dashboard levem às suas respectivas telas (views).

Sprint 3: Módulo de Cadastros

Objetivo: Desenvolver todas as funcionalidades da seção de "Cadastros".

Tarefa	Branch Git	Descrição
View Principal de Cadastros	feat/registrations-view	Criação da tela que agrupa os botões de cadastro (item, local, unidade de medida, etc.) e o botão "Voltar".
CRUD - Cadastro de Item	feat/crud-items	Implementação da tela e lógica para Criar, Ler, Atualizar e Deletar itens no banco de dados.
CRUD - Local de Estoque	feat/crud-storage-locations	Funcionalidade completa para gerenciar os locais de contagem.
CRUD - Unidade de Medida	feat/crud-measurement-units	Funcionalidade completa para gerenciar as unidades de medida (Ex: Garrafa, Dose, Litro).

Sprint 4: Funcionalidades Core - Contagem e Compras

Objetivo: Implementar as funcionalidades centrais de contagem de estoque e lançamento de compras.

Tarefa	Branch Git	Descrição
View - Contagem de Estoque	feat/stock-count-view	Desenvolvimento da interface para selecionar um local e listar os itens para contagem.
Lógica de Contagem	feat/stock-count-logic	Implementação da lógica para salvar os dados da contagem no banco, registrando data, hora e usuário.
View - Incluir Itens (Compras)	feat/purchase-entry-view	Desenvolvimento da interface para lançamento de notas de compra.
Lógica de Lançamento	feat/purchase-entry-logic	Implementação da lógica para atualizar a quantidade dos itens no estoque após uma compra.

Sprint 5: Relatórios e Auditoria

Objetivo: Criar a seção de relatórios para fornecer insights ao usuário.

Tarefa	Branch Git	Descrição
View Principal de Relatórios	feat/reports-view	Criação da tela de seleção de relatórios (Contagens, Itens, Auditoria).
Relatório de Contagens	feat/report-stock-counts	Geração de um relatório que exibe o histórico de contagens realizadas.
Relatório de Itens	feat/report-items	Relatório de movimentação de itens (entradas e

		saídas).
Exportação de Relatórios	feat/export-reports	Funcionalidade para exportar os relatórios para formatos como CSV ou PDF.

Sprint 6: Integração com Firebase (Versão Paga)

Objetivo: Adaptar a aplicação para suportar o Firebase como backend.

Tarefa	Branch Git	Descrição
Configuração do Firebase	feat/firebase-setup	Integração do SDK do Firebase ao projeto Python.
Autenticação com Google	feat/firebase-google-auth	Implementação da lógica de login utilizando a autenticação do Firebase com contas Google.
Migração da Lógica de BD	refactor/firestore-logic	Refatoração do código para que as operações de CRUD sejam feitas no Firestore em vez do SQLite.
Sincronização de Dados	feat/data-sync	Lógica para garantir que os dados sejam recuperados da nuvem após o login.

Sprint 7: Testes, Deploy e Compilação

Objetivo: Garantir a qualidade da aplicação, publicá-la na web e compilar para Android.

Tarefa	Branch Git	Descrição
Testes Unitários e de Integração	test/full-coverage	Criação de testes automatizados para garantir que todas as funcionalidades operem

		como esperado.
Deploy na Fly.io	feat/deploy-fly-io	Configuração e publicação da versão web da aplicação na plataforma Fly.io.
Compilação para Android	feat/build-android-apk	Geração do pacote .apk da aplicação Flet para instalação em dispositivos Android.
Documentação Final	docs/final-documentation	Revisão e atualização de toda a documentação do projeto, incluindo o README.md.