



UNIVERSIDADE DE FORTALEZA
VICE REITORIA DE GRADUAÇÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Alice Rocha Jamil
Breno Guimarães Ribeiro
Daniel de Moura Mascarenhas
Larissa Evangelista Moreira
Matheus Silva Matos
Walther Oliveira Pires

DOCUMENTAÇÃO TÉCNICA

Projeto Recycle

FORTALEZA
2025

Alice Rocha Jamil
Breno Guimarães Ribeiro
Daniel de Moura Mascarenhas
Larissa Evangelista Moreira
Matheus Silva Matos
Walther Oliveira Pires

DOCUMENTAÇÃO TÉCNICA

Projeto Recycle

Este documento contém a documentação técnica do Sistema de Gestão de Estoque denominado Recycle, voltado a empresas de reciclagem, desenvolvido na componente curricular N393 - Projeto Aplicado Multiplataforma, como requisito para obtenção de nota.

Supervisor: Prof. Bruno Lopes, Me

FORTALEZA

2025

SUMÁRIO

1. INTRODUÇÃO.....	4
1.1. CONTEXTO E JUSTIFICATIVA.....	4
1.2. OBJETIVOS.....	4
1.3. ESCOPO E DELIMITAÇÃO.....	5
2. ENGENHARIA DE REQUISITOS.....	6
2.1. REQUISITOS FUNCIONAIS (RFs).....	6
2.2. REQUISITOS NÃO FUNCIONAIS (RNFs).....	6
3. PROJETO E ARQUITETURA DO SOFTWARE.....	8
3.1. ARQUITETURA GERAL.....	8
3.2. PROJETO DO BANCO DE DADOS.....	8
3.3. PROJETO DE API.....	9
4. TECNOLOGIAS E FERRAMENTAS.....	11
4.1. STACK DE TECNOLOGIAS.....	11
4.2. FERRAMENTAS DE DESENVOLVIMENTO.....	11
5. IMPLEMENTAÇÃO E RESULTADOS.....	13
5.1. TELAS DO SISTEMA.....	13
6. AMBIENTE E GUIA DE IMPLANTAÇÃO.....	14
6.1. REQUISITOS DO AMBIENTE.....	14
6.2. PROCESSO DE IMPLANTAÇÃO.....	14
6.3. ACESSO À APLICAÇÃO IMPLANTADA.....	15
7. CONCLUSÃO.....	16
7.1. TRABALHOS FUTUROS.....	16
7.2. LIÇÕES APRENDIDAS.....	16

1. INTRODUÇÃO

1.1. CONTEXTO E JUSTIFICATIVA

O setor de reciclagem desempenha um papel essencial na sustentabilidade ambiental, mas muitas empresas ainda realizam o controle de seus estoques de forma manual, utilizando planilhas ou anotações em papel. Essa prática torna o processo suscetível a erros, perda de informações e falta de atualização em tempo real, comprometendo a eficiência das operações e a tomada de decisões.

Diante desse cenário, o sistema *Recycle* foi idealizado como uma solução tecnológica voltada à digitalização e automação da gestão de estoque em empresas de reciclagem. O sistema permitirá controlar entradas e saídas, calcular automaticamente o preço médio dos materiais e oferecer uma visão geral do estoque, por meio de uma interface intuitiva e segura para usuários autenticados. Assim, o *Recycle* busca reduzir erros humanos, otimizar processos e contribuir para a eficiência e sustentabilidade do setor.

1.2. OBJETIVOS

O objetivo deste projeto é desenvolver um sistema de gestão de estoque voltado a empresas de reciclagem, com o intuito de otimizar o controle de materiais recicláveis por meio da automatização dos processos de entrada e saída, cálculo do preço médio e acompanhamento do saldo atual. O sistema visa proporcionar uma visão ampla e precisa do estoque, oferecendo interfaces web e mobile com acesso seguro e usabilidade intuitiva.

Dito isso, os objetivos específicos do projeto são:

- Implementar funcionalidades que possibilitem o cadastro de materiais recicláveis, bem como o registro das movimentações de entrada e saída, garantindo o cálculo automático do preço médio e do saldo atualizado.
- Desenvolver uma interface web administrativa que permita o gerenciamento completo do sistema, abrangendo o cadastro de administradores, operadores, materiais e movimentações de estoque.

- Criar uma aplicação mobile voltada ao uso operacional em campo, possibilitando o registro de materiais e movimentações de forma prática e acessível.
- Estabelecer níveis de acesso diferenciados, assegurando que administradores possuam controle total sobre as funcionalidades, enquanto funcionários disponham de permissões restritas às suas atribuições.
- Proporcionar uma experiência de uso segura e eficiente, por meio de autenticação de usuários e design responsivo, favorecendo a confiabilidade e a acessibilidade do sistema em diferentes dispositivos.

1.3. ESCOPO E DELIMITAÇÃO

- **Escopo:**
 - Cadastro, edição, visualização e exclusão de materiais recicláveis.
 - Registro de entradas e saídas de materiais, com atualização automática do saldo e cálculo do preço médio.
 - Controle de acesso de usuários baseado em perfis (gerente, operador).
 - Autenticação de usuários, garantindo acesso seguro às funcionalidades do sistema.
 - Interface web destinada à gestão completa do sistema, voltada aos administradores.
 - Aplicativo mobile voltado à operação prática, permitindo o registro de materiais e movimentações.
 - Interface intuitiva e responsiva, proporcionando facilidade de uso em diferentes dispositivos.
- **Delimitação (Fora do Escopo):**
 - *O sistema não possui a funcionalidade de recuperação de senha ("Esqueci minha senha").*
 - *Não é possível suspender, desativar ou bloquear o acesso de um operador após o cadastro.*
 - *Não é possível alterar os dados cadastrados (ex.: e-mail, nome completo, CNPJ da empresa,...).*

- *Não há geração de relatórios sobre movimentações ou saldos de estoque.*
- *O sistema não realiza integração com sistemas externos, como plataformas financeiras ou de contabilidade.*
- *Não há suporte a notificações automáticas (por e-mail ou aplicativo) relacionadas a movimentações de estoque.*

2. ENGENHARIA DE REQUISITOS

2.1. REQUISITOS FUNCIONAIS (RFs).

ID	Nome do Requisito	Descrição
RF01	Autenticação de Usuário	O sistema, tanto web quanto o mobile, deve permitir que o usuário se autentique com e-mail e senha. O acesso deve ser permitido somente após validação das credenciais.
RF02	Controle de Perfis de Acesso	O sistema deve permitir a diferenciação de permissões entre usuários Gerentes e Operadores, garantindo que cada perfil visualize e execute apenas funcionalidades compatíveis com seu nível de acesso.
RF03	Cadastro de Materiais	O sistema deve permitir o cadastro de materiais, contendo informações como nome, categoria e unidade de medida.
RF04	Edição e Exclusão de Materiais	O sistema deve permitir editar e excluir materiais previamente cadastrados, desde que o usuário possua permissão.
RF05	Visualização de Materiais Cadastrados	O sistema deve permitir que os usuários visualizem a lista de materiais cadastrados, incluindo suas informações atualizadas.
RF06	Registro de Entrada de Materiais	O sistema deve permitir registrar entradas de materiais no estoque, atualizando automaticamente o saldo e contribuindo para o cálculo do preço médio.
RF07	Registro de Saída de Materiais	O sistema deve permitir registrar saídas de materiais do estoque, atualizando automaticamente o saldo disponível.
RF08	Interface Responsiva e Intuitiva	O sistema deve fornecer uma interface responsiva e de fácil navegação, permitindo uso eficiente tanto em web quanto mobile.

RF09	Cálculo Automático do Preço Médio	Ao registrar movimentações de entrada de materiais, o sistema deve recalcular automaticamente o preço médio do material com base no valor e quantidade adicionados.
RF10	Exibição do Saldo Atual de Estoque	O sistema deve apresentar ao usuário o saldo atual de cada material no estoque, considerando todas as movimentações registradas.
RF11	Interface Web Administrativa	O sistema deve disponibilizar uma interface web destinada a administradores, permitindo o gerenciamento completo de materiais, usuários e movimentações.
RF12	Aplicativo Mobile Operacional	O sistema deve disponibilizar um aplicativo mobile que permita aos funcionários registrar materiais e movimentações de estoque de forma prática durante o trabalho em campo.
RF13	Exibição das Movimentações Realizadas	O sistema deve permitir que o usuário visualize todas as movimentações registradas, exibindo o tipo da operação (entrada ou saída), a quantidade movimentada e o valor total da operação.
RF14	Visualização dos Funcionários Cadastrados	O sistema deve permitir que usuários do tipo “Gerente” visualizem a lista de funcionários cadastrados, incluindo o histórico atualizado de movimentações realizadas por cada funcionário (entradas e saídas).

2.2. REQUISITOS NÃO FUNCIONAIS (RNFs)

- **Desempenho:**

- RNF01: O sistema deve registrar entradas e saídas de materiais sem demora perceptível ao usuário.
- RNF02: As telas devem carregar em tempo adequado para uso cotidiano, evitando lentidão.

- **Usabilidade:**

- RNF03: A interface do sistema deve ser simples e fácil de entender.
- RNF04: As ações principais (como registrar entrada/saída) devem ser acessadas rapidamente pelo usuário.

- **Compatibilidade:**

- RNF05: A versão web deve funcionar nos navegadores mais comuns (Chrome, Edge e Firefox).
- RNF06: O aplicativo mobile deve funcionar em dispositivos Android utilizados pelos usuários.
- **Segurança:**
 - RNF07: O sistema deve permitir acesso apenas a usuários cadastrados.

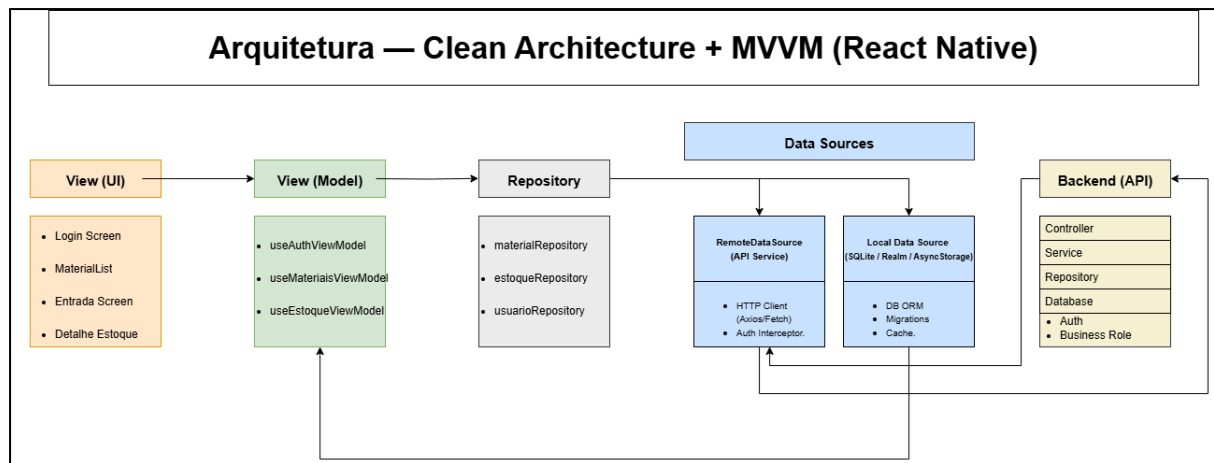
3. PROJETO E ARQUITETURA DO SOFTWARE

3.1. ARQUITETURA GERAL

O aplicativo foi desenvolvido seguindo o padrão **Clean Architecture** com adaptação para o modelo **MVVM (Model–View–ViewModel)**. Esse padrão foi escolhido por promover separação de responsabilidades, facilidade de manutenção, testabilidade e escalabilidade do projeto.

A organização interna do código é dividida em camadas:

1. **View (Apresentação):** Composta pelos componentes e telas desenvolvidas em React Native. É responsável por exibir os dados e capturar interações do usuário, sem conter a lógica de negócio.
2. **ViewModel:** Responsável por controlar o estado da interface e coordenar as ações iniciadas pela View. Toda lógica de apresentação fica nessa camada, que não depende diretamente da UI.
3. **Repository (Modelo / Regra de Negócio):** Centraliza o acesso aos dados. Decide se a informação deve ser buscada da API ou do banco de dados local, aplicando o padrão Single Source of Truth.
4. **Data Sources:** São as fontes de dados efetivas. Incluem a comunicação com a API REST (dados remotos) e a persistência local em SQLite / AsyncStorage (cache local).



3.2. PROJETO DE DADOS: INTEGRAÇÃO COM API E PERSISTÊNCIA LOCAL

A arquitetura de dados do sistema foi projetada considerando desempenho, disponibilidade e integridade das informações. A API REST atua como a fonte principal dos dados, garantindo consistência e atualização centralizada, enquanto o banco de dados local funciona como um cache de persistência, permitindo operação constante mesmo quando o dispositivo estiver sem conexão.

O fluxo de funcionamento segue o padrão Single Source of Truth, onde o aplicativo sempre exibe os dados diretamente do banco de dados local, enquanto a API é utilizada para manter essas informações sincronizadas.

● Fluxo de Dados

- **Solicitação de Dados:** Quando o usuário acessa seções como lista de materiais ou movimentações de estoque, a camada de aplicação solicita os dados ao Repositório.
- **Consulta à API:** O Repositório tenta obter a versão mais atualizada dos dados através da API REST, garantindo que alterações feitas por outros usuários ou sistemas sejam consideradas.
- **Persistência Local (Cache):** Após a resposta da API, os dados são salvos no banco de dados local, substituindo registros antigos. Esse armazenamento evita acessos desnecessários à API.
- **Exibição pela Camada Local:** A interface do usuário exibe sempre os dados armazenados localmente. Quando há atualização, a interface é atualizada automaticamente sem necessidade de nova requisição à API.

- **Benefícios da Estratégia**

- Performance: A leitura local é mais rápida do que requisitar a API constantemente.
- Operação Offline: Caso não haja conexão, o sistema continua funcional, exibindo os últimos dados salvos.
- Sincronização Confiável: Atualizações acontecem sempre que houver conectividade disponível.
- Menos carga na API: Diminui o número de requisições e reduz o consumo de rede.

- **Persistência Local**

O armazenamento local mantém tabelas essenciais para o funcionamento do sistema, incluindo:

- Usuários
- Materiais
- Estoque atualizado
- Registros de entradas e saídas

Esse modelo garante que o sistema de controle de estoque continue operando de forma consistente mesmo em ambientes desconectados ou com baixa internet, cenário comum em operações logísticas, almoxarifados e canteiros de obra.

Dicionário de Dados

- Tabela **Materiais**:

Nome do campo	Tipo de dados	Chave (PK/FK)	Nulo?	Descrição
data_atualizacao	DATETIME(6)		Não	Timestamp de quando o dado foi atualizado.
data_criacao	DATETIME(6)		Não	Timestamp de quando o dado foi criado.
id	INT	PK	Não	Chave Primária da tabela Materiais.
usuario_id	INT	FK	Não	Foreign Key da tabela Usuarios.
descricao	VARCHAR(255)		Não	Descrição do material cadastrado.
nome	VARCHAR(255)		Não	Nome do Material.
unidade	VARCHAR(255)		Não	Unidade de Medida do Material.

- Tabela **Empresas**:

Nome do campo	Tipo de dados	Chave (PK/FK)	Nulo?	Descrição
id	int	PK	Não	Chave Primária da tabela Empresas.
cnpj	VARCHAR(255)		Não	CNPJ da Empresa
nome_fantasia	VARCHAR(255)	PK	Não	Nome Fantasia da Empresa

- Tabela **Entradas**:

Nome do campo	Tipo de dados	Chave (PK/FK)	Nulo?	Descrição
preco	FLOAT		Não	Valor unitário do material no momento da entrada.
quantidade	FLOAT		Não	Quantidade de unidades do material adicionadas ao estoque nesta entrada.
data	DATETIME(6)		Não	Timestamp de quando o material foi adicionado no sistema.
id	INT	PK	Não	Chave Primária da tabela Entradas.
material_id	INT	FK	Não	Foreign Key da tabela Material.
usuario_id	INT	FK	Não	Foreign Key da tabela Usuarios.

- Tabela **Estoque**:

Nome do campo	Tipo de dados	Chave (PK/FK)	Nulo?	Descrição
preco_medio	FLOAT		Não	Preço médio do material no estoque, calculado com base nas entradas realizadas.
quantidade	FLOAT		Não	Quantidade atual disponível do material no estoque.
valor_total	FLOAT		Não	Valor total do estoque desse material, resultante da multiplicação entre preco_medio e quantidade.

material_id	INT	PK	Não	Foreign Key da tabela Material.
-------------	-----	----	-----	---------------------------------

- Tabela **Saidas**:

Nome do campo	Tipo de dados	Chave (PK/FK)	Nulo?	Descrição
quantidade	FLOAT		Não	Quantidade de unidades do material removidas do estoque nesta saida.
data	DATETIME(6)		Não	Timestamp de quando o material saiu do sistema.
id	INT	PK	Não	Chave Primária da tabela Saidas.
material_id	INT	FK	Não	Foreign Key da tabela Material.
usuario_id	INT	FK	Não	Foreign Key da tabela Usuarios.

- Tabela **Usuarios**:

Nome do campo	Tipo de dados	Chave (PK/FK)	Nulo?	Descrição
id	INT	PK	Não	Chave Primária da tabela Usuarios.
data_criacao	DATETIME(6)		Não	Timestamp de quando o dado foi criado.
nome	VARCHAR(255)		Não	Nome do Usuario
email	VARCHAR(255)		Não	Email do Usuario
senha	VARCHAR(255)		Não	Senha do Usuario
role	ENUM: ('GERENTE','OPERADOR')		Não	Define o nível de acesso do usuário no sistema.
empresa_id	INT	FK	Não	Foreign Key da tabela Empresa.
data_atualizacao	DATETIME(6)		Não	Timestamp de quando o dado foi atualizado.

3.3. CONSUMO DA API E FLUXO DE NAVEGAÇÃO

- **URL da Documentação da API:**

<https://github.com/SantoGuru/Recycle/tree/main/docs>

- **Diagrama de Fluxo de Navegação do Aplicativo**

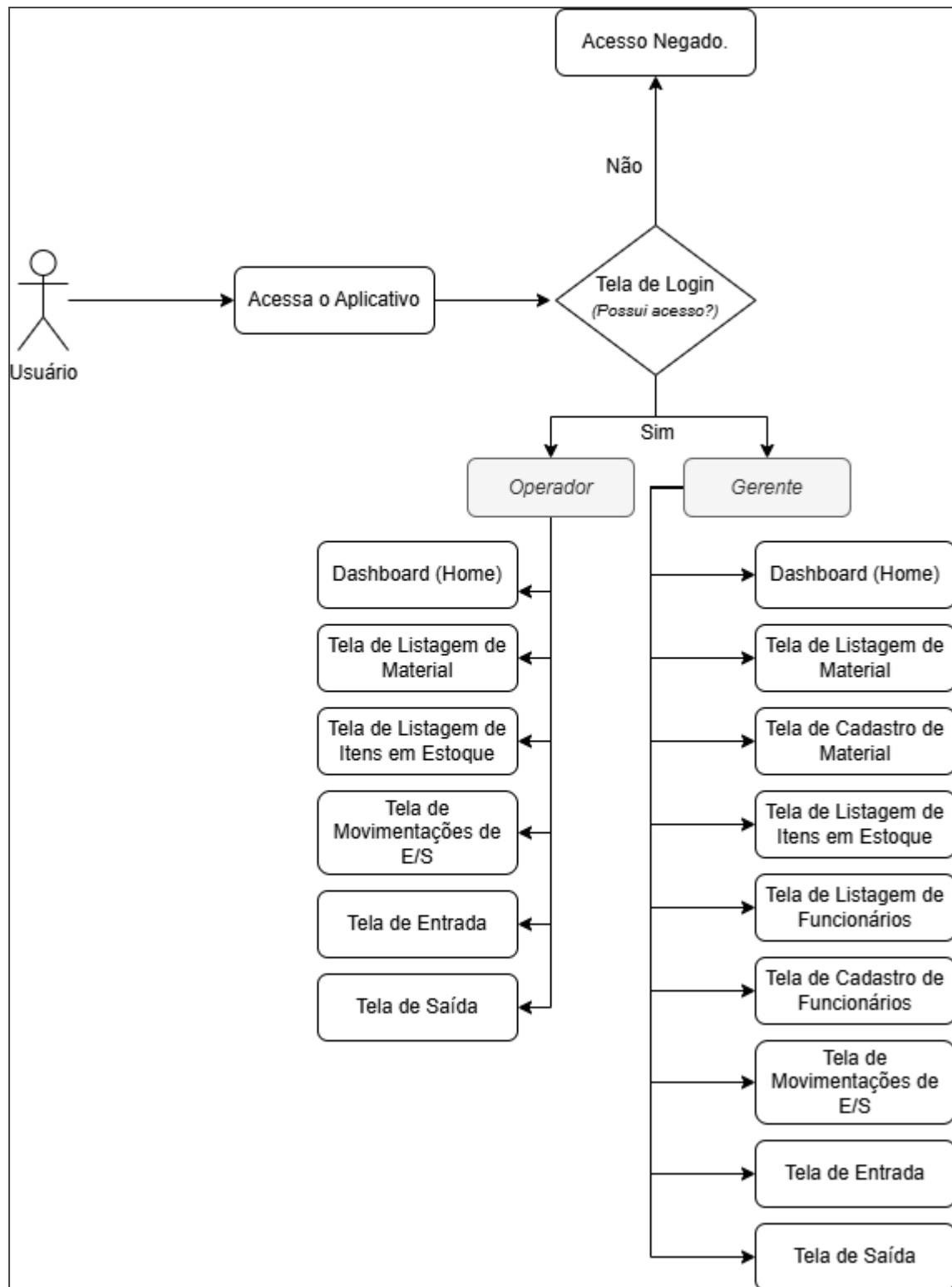
O fluxo de navegação do usuário no aplicativo foi projetado para ser simples, direto e alinhado aos perfis de acesso definidos no sistema. A interação começa na tela inicial, que exibe o botão “*Começar*” juntamente com a identidade visual do projeto Recycle. Ao selecionar essa opção, o usuário é direcionado para a tela de Login, onde deve informar seu e-mail e senha para acessar o sistema.

É importante destacar que o aplicativo e o site não permitem a criação de contas do tipo “Operador” diretamente pelo próprio usuário. O comportamento de cadastro funciona da seguinte forma:

- Se o usuário se cadastrar diretamente no site (Web/Desktop), sua conta será automaticamente vinculada ao perfil Gerente.
- Somente um Gerente pode cadastrar outros usuários, e ao fazer isso — seja pelo site ou pelo aplicativo mobile — a nova conta criada será do tipo Operador.

Depois que o usuário realiza o login, o sistema libera funcionalidades diferentes conforme o perfil:

- *Gerente*: Possui acesso completo às funcionalidades do sistema. Esse perfil pode cadastrar novos materiais, registrar entradas e saídas de materiais, cadastrar novos usuários (funcionários).
- *Operador*: Possui acesso restrito, podendo apenas registrar entradas e saídas, visualizar os materiais cadastrados e itens em estoque. Para ser Operador, o cadastro tem que ser feito pelo Gerente via desktop ou mobile.



4. TECNOLOGIAS E FERRAMENTAS

4.1. STACK DE TECNOLOGIAS

- **Linguagem:** *Typescript (react-native)* para o desenvolvimento mobile e *Java (Springboot)* para o backend.

- **Arquitetura:** Multicamadas.
- **UI Toolkit:** React Native Paper, que segue o padrão de UI do Google.
- **Consumo de API:** Requisições HTTP.
- **Banco de Dados Local:** MySQL.

4.2. FERRAMENTAS DE DESENVOLVIMENTO

- **IDE:** Visual Studio Code foi a IDE padrão para toda a equipe.
- **Controle de Versão:** Git, com o repositório hospedado no GitHub. Adotamos o fluxo de trabalho "GitFlow", com branches separadas para backend, frontend, documentação, features e main.
- **Ferramenta de API:** Postman foi usado para testar os endpoints da API durante o desenvolvimento.

5. IMPLEMENTAÇÃO E RESULTADOS

5.1. TELAS DO SISTEMA

- Visão Gerente:

Figura 1: Tela Inicial ([Clique aqui para ver.](#))

(Legenda: Tela de boas-vindas do aplicativo, apresentando a identidade visual do projeto e o botão "Começar", que direciona o usuário para o acesso ao sistema.)

Figura 2: Tela de Login ([Clique aqui para ver.](#))

(Legenda: A tela de login é responsável por controlar o acesso ao sistema. O usuário informa seu e-mail e senha previamente cadastrados. O aplicativo não permite criação de contas via mobile; esse processo ocorre exclusivamente no sistema Web.)

Figura 3: Tela Principal ([Clique aqui para ver.](#))

(Legenda: Após a autenticação, usuários com perfil Gerente têm acesso completo às funcionalidades do sistema, incluindo cadastro de materiais, registro de entradas, registro de saídas e cadastro de novos usuários.)

Figura 4: Tela Material ([Clique aqui para ver.](#))

(Legenda: Exibe a listagem de todos os materiais cadastrados na Empresa, permitindo ao Gerente visualizar quais itens já estão cadastrados no sistema.)

Figura 5: Tela de Cadastro de Material ([Clique aqui para ver.](#))

(Legenda: Permite o cadastro de novos materiais. Nesta tela, o Gerente pode informar nome, a descrição do material e a unidade de medida.)

Figura 6: Tela de Estoque ([Clique aqui para ver.](#))

(Legenda: Apresenta todos os materiais atualmente em estoque, permitindo ao Gerente visualizar rapidamente as quantidades disponíveis.)

Figura 7: Tela de Movimentações ([Clique aqui para ver.](#))

(Legenda: Exibe todas as movimentações registradas no sistema, incluindo o tipo da operação — entrada ou saída —, a quantidade movimentada e o valor total da operação.)

Figura 8: Tela de Entrada ([Clique aqui para ver.](#))

(Legenda: Permite registrar novas entradas de material, informando o tipo do material, a quantidade adicionada e o valor unitário.)

Figura 9: Tela de Saída ([Clique aqui para ver.](#))

(Legenda: Permite registrar saídas de material, informando o tipo do material e a quantidade retirada.)

Figura 10: Tela Funcionário ([Clique aqui para ver.](#))

(Legenda: Exibe a listagem de todos os funcionários cadastrados na Empresa, permitindo ao Gerente visualizar quem possui acesso ao sistema.)

Figura 11: Tela de Cadastro de Funcionário ([Clique aqui para ver.](#))

(Legenda: Permite que o Gerente cadastre novos colaboradores com o nome, email e senha, definindo quais funcionários terão acesso ao aplicativo para registrar entradas e saídas de materiais.)

- **Visão Operador:**

Figura 1: Tela Inicial [\(Clique aqui para ver.\)](#)

(Legenda: Tela de boas-vindas do aplicativo, apresentando a identidade visual do projeto e o botão “Começar”, que direciona o usuário para o acesso ao sistema.)

Figura 2: Tela de Login [\(Clique aqui para ver.\)](#)

(Legenda: A tela de login é responsável por controlar o acesso ao sistema. O usuário informa seu e-mail e senha previamente cadastrados. O aplicativo não permite criação de contas via mobile; esse processo ocorre exclusivamente no sistema Web.)

Figura 3: Tela Principal [\(Clique aqui para ver.\)](#)

(Legenda: Usuários com perfil Operador possuem interface simplificada, permitindo apenas registrar entradas e saídas de materiais, promovendo usabilidade e evitando ações não autorizadas.)

Figura 4: Tela Material [\(Clique aqui para ver.\)](#)

(Legenda: Exibe a listagem de todos os materiais cadastrados na empresa, permitindo ao Operador visualizar os itens já disponíveis no sistema)

Figura 5: Tela de Estoque [\(Clique aqui para ver.\)](#)

(Legenda: Apresenta todos os materiais atualmente em estoque, permitindo ao Operador visualizar rapidamente as quantidades disponíveis.)

Figura 6: Tela de Movimentações [\(Clique aqui para ver.\)](#)

(Legenda: Exibe todas as movimentações registradas no sistema, incluindo o tipo da operação — entrada ou saída —, a quantidade movimentada e o valor total da operação.)

Figura 7: Tela de Entrada ([Clique aqui para ver.](#))

(Legenda: Permite registrar novas entradas de material, informando o tipo do material, a quantidade adicionada e o valor unitário.)

Figura 8: Tela de Saída ([Clique aqui para ver.](#))

(Legenda: Permite registrar saídas de material, informando o tipo do material e a quantidade retirada.)

6. AMBIENTE E GUIA DE GERAÇÃO (BUILD)

6.1. REQUISITOS DO AMBIENTE

- **IDE:** Visual Studio Code
- **Node:** 21.0.0
- **JDK:** JDK 17

6.2. PROCESSO DE GERAÇÃO DE APLICATIVO

- **Pré-requisitos:**
 - Ter uma conta cadastrada na versão web;
 - Ter o backend funcionando;
- **Instruções:**
 1. Clone o repositório do projeto: `git clone https://github.com/SantoGuru/Recycle.git`
 2. Entre na pasta do projeto: `cd mobile`
 3. No Windows, execute o comando para instalar os pacotes: `npm i`
 4. Após isso, obtenha o ipv4 usando o comando no terminal: `ipconfig`
 5. Copie o ip e adicione no arquivo `config.ts`, em `API_URL`.
 6. Execute o projeto usando o comando no terminal: `npx expo start`
 7. Baixe o aplicativo Expo Go no seu dispositivo móvel.
 8. Escaneie o QR Code exibido no terminal do seu desktop para executar o aplicativo.
 9. Após o término do carregamento, o aplicativo deve aparecer no seu dispositivo.

6.3. ACESSO À APLICAÇÃO IMPLANTADA

O aplicativo não será disponibilizado via APK, pois a publicação exigiria etapas pagas que não estão dentro do escopo do projeto. Além disso, não há credenciais de acesso externas, já que o banco de dados é local e os usuários só funcionam na máquina em que foram cadastrados.

7. CONCLUSÃO

7.1. TRABALHOS FUTUROS

Com a base do aplicativo já estruturada e funcional, foram identificadas possíveis melhorias que podem ser implementadas em versões futuras. Entre as principais evoluções estão:

1. **Aplicativo Offline:** Expandir o suporte offline para permitir que operadores registrem entradas e saídas mesmo sem conexão com a internet. Assim que o dispositivo se reconectar, os dados seriam sincronizados automaticamente com o servidor, garantindo continuidade das operações.
2. **Notificações Push:** Implementar notificações para alertar o Gerente sobre atividades importantes, como novas movimentações registradas ou baixa de estoque de um material. Isso aumentaria a visibilidade e o controle sobre o sistema em tempo real.
3. **Relatórios:** Adicionar a geração de relatórios, como histórico de movimentações, materiais com maior saída, itens com risco de desabastecimento e desempenho operacional. Essa funcionalidade apoiaria a tomada de decisão e ampliaria a utilidade administrativa do sistema.

7.2. LIÇÕES APRENDIDAS

O desenvolvimento do sistema de controle de estoque, Recycle, foi uma experiência enriquecedora, trazendo aprendizados importantes em diversas áreas:

1. **Desafios Técnicos de Arquitetura:** Um dos principais desafios foi garantir que as movimentações (entradas e saídas) fossem registradas corretamente, com atualização consistente do estoque. A definição clara entre responsabilidades do backend e do aplicativo foi essencial, bem como a validação adequada de quantidades e valores. Isso mostrou que uma arquitetura organizada é indispensável para um sistema seguro e confiável.

2. **Integração entre Aplicativo e API:** A comunicação entre o Mobile e o sistema Web exigiu alinhamento constante, especialmente no envio de dados e interpretação das respostas da API. Esse processo destacou a importância da documentação atualizada, padronização de endpoints e validação precisa dos dados trafegados.
3. **Experiência do Usuário e Perfis de Acesso:** A divisão entre os perfis Gerente e Operador mostrou que entender quem usa o sistema e quais permissões cada um realmente precisa evita complexidade excessiva e melhora a segurança e a usabilidade. Esse aprendizado reforçou a importância de planejar fluxos simples, diretos e adequados ao contexto real de uso.
4. **Mensagens de Erro Claras e Consistentes:** Durante o desenvolvimento, aprendemos o valor de mensagens de erro bem formuladas. Houve situações em que o backend retornava mensagens técnicas ou pouco claras, e o frontend precisava interpretar, adaptar e exibir versões compreensíveis para o usuário. Esse aprendizado reforçou a importância da comunicação entre as equipes e da definição de um padrão unificado para erros e validações.

A principal lição deste projeto é que construir um sistema eficiente vai muito além de implementar telas: envolve compreender processos, validar regras de negócio, tratar erros de forma clara, colaborar entre equipes e garantir que cada parte funcione de maneira integrada e estável.