

Curso de Análise e Desenvolvimento de Sistemas

Projeto do Módulo de Desenvolvimento Back-End

Tema: Walbit – Carteira Digital

Aluno(a): Maria Clara Bastos

Documentação referente à entrega preliminar do módulo de desenvolvimento Back-End, desenvolvida com base no site Walbit – Carteira Digital, no curso de Análise e Desenvolvimento de Sistemas do Centro Universitário Augusto Motta.

Professor Orientador: Charles Bastos

Unidade: Campo Grande

Turno: Noite

Link do Repositório no Github:

Rio de Janeiro, Rj

Setembro, 2025

Sumário

[1. Introdução 3](#_Toc209653429)

[1.1. Objetivo 3](#_Toc209653430)

[2. Metodologias 4](#_Toc209653431)

[2.1. Levantamento de Requisitos 4](#_Toc209653432)

[2.1.1. Requisitos Funcionais do Sistema 4](#_Toc209653433)

[2.1.2. Requisitos Não Funcionais do Sistema 4](#_Toc209653434)

[2.2. Tecnologias e Ferramentas Utilizadas 5](#_Toc209653435)

[3. Cronogramas 6](#_Toc209653436)

[3.1. Cronograma de Tarefas 6](#_Toc209653437)

[3.2. Cronograma de Sprints 8](#_Toc209653438)

[4. Banco de Dados 9](#_Toc209653439)

[4.1 Funcionamento do Banco de dados 9](#_Toc209653440)

[4.2 Diagrama de Relacionamento 10](#_Toc209653441)

[5. Testes 11](#_Toc209653442)

[6. Mockups das Funcionalidades Atuais do Sistema 12](#_Toc209653443)

[7. Conclusão 17](#_Toc209653444)

## Introdução

O presente documento registra de forma detalhada o desenvolvimento do projeto WallBit, uma plataforma web voltada para gerenciamento de transações de moedas digitais, desenvolvida como parte da disciplina de Desenvolvimento Back-End. O WallBit busca proporcionar uma experiência segura e intuitiva aos usuários, permitindo o acompanhamento de transações, visualização de rankings, gerenciamento de moedas e funcionalidades administrativas.

Este documento apresenta todo o ciclo do projeto, incluindo levantamento de requisitos, metodologias aplicadas, tecnologias utilizadas, cronogramas de tarefas e sprints, estrutura de banco de dados, testes realizados e evidências visuais, servindo como referência formal e suporte para futuras melhorias e manutenção.

### Objetivo

O objetivo do projeto WallBit é desenvolver uma plataforma web funcional e segura, capaz de gerenciar transações de moedas digitais e oferecer funcionalidades administrativas, como gerenciamento de usuários e acompanhamento de desempenho das moedas.

Além disso, a documentação tem como finalidade:

* Registrar todo o processo de desenvolvimento;
* Detalhar os requisitos funcionais e não funcionais;
* Servir como guia de implementação para futuras iterações;
* Fornecer base para apresentação acadêmica e avaliação formal do projeto.

## Metodologias

Para garantir o desenvolvimento organizado e eficiente, foram adotadas as seguintes metodologias:

* Metodologia Ágil (Scrum): o projeto foi dividido em sprints, priorizando entregas incrementais e melhorias contínuas com base em feedback.
* Desenvolvimento Iterativo e Incremental: permite ajustes contínuos durante o desenvolvimento das funcionalidades.
* Documentação Técnica Estruturada: registra todas as decisões de projeto, requisitos, tecnologias e testes realizados, garantindo rastreabilidade.
* Controle de Versão (Git/GitHub): garante histórico de alterações, colaboração entre equipe e versionamento seguro do código.

### Levantamento de Requisitos

### Requisitos Funcionais do Sistema

* Cadastro de usuários com validação de dados (nome, e-mail, senha, CPF, telefone).
* Login e autenticação de usuários com verificação de credenciais.
* Tela de perfil de usuário com histórico de transações e saldo atual.
* Dashboard de ranking de moedas mais negociadas.
* Perfil master (administrativo) com:
  + Gerenciamento de usuários (criar, editar, excluir);
  + Gerenciamento de moedas (adicionar, remover, atualizar);
  + Visualização de indicadores de desempenho;
* Validação de formulários no front-end e back-end.
* Sistema de notificações de sucesso ou erro em operações.
* Visualização de gráficos de transações e rankings.

### Requisitos Não Funcionais do Sistema

* Interface responsiva e compatível com diferentes dispositivos.
* Segurança na manipulação de dados sensíveis (senhas criptografadas).
* Alta disponibilidade e desempenho satisfatório.
* Implementação de acessibilidade, incluindo dark mode.
* Registro de logs de atividades críticas.
* Layout moderno, intuitivo e organizado.

### Tecnologias e Ferramentas Utilizadas

Para o desenvolvimento do WallBit foram utilizadas as seguintes tecnologias:

|  |  |
| --- | --- |
| Finalidade | Tecnologias e Ferramentas |
| Front-End | HTML5, CSS3, JavaScript, Bootstrap, Font Awesome (ícones). |
| Back-End | PHP (lógica de autenticação, CRUD, manipulação de dados). |
| Banco de Dados | MySQL (relacional, com tabelas para usuários, moedas, transações, rankings). |
| Controle de Versão: | Git, GitHub. |
| Testes e Acessibilidade: | Lighthouse, testes de responsividade, validação de formulários. |
| Design e Mockups | Figma e Canva. |

## Cronogramas

### Cronograma de Tarefas

O cronograma do projeto WallBit foi dividido nas seguintes etapas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Tarefa | Data | Status | Sprint |
| 01 | Levantamento de Requisitos e funcionalidades do site | 02/09 – 08/09 | Concluído | Sprint 1 |
| 02 | Definição da identidade visual (cores, tipografia, layout inicial) | 02/09 – 08/09 | Concluído | Sprint 1 |
| 03 | Criação do repositório no GitHub | 02/09 – 08/09 | Concluído | Sprint 1 |
| 04 | |  | | --- | |  |  |  | | --- | | Desenvolvimento inicial do front-end (Login, Cadastro, Home) | | 09/09 – 15/09 | Concluído | Sprint 1 |
| 05 | Criação do backlog detalhado do site | 09/09 – 15/09 | Concluído | Sprint 1 |
| 06 | Início da documentação do projeto | 09/09 – 15/09 | Concluído | Sprint 1 |
| 07 | Desenvolvimento do Diagrama ER | 16/09 – 22/09 | Concluído | Sprint 1 |
| 08 | Finalização da documentação | 16/09 – 22/09 | Concluído | Sprint 1 |
| 09 | Entrega Preliminar do Projeto e 1ª Sprint | 24/09 – 08/10 | Concluído | Sprint 1 |
| 10 | Finalização do front-end da página inicial (home) | 08/10 – 20/10 | Pendente | Sprint 2 |
| 11 | Ajustes e refino do front-end (responsividade + feedbacks) | 08/10 – 20/10 | Pendente | Sprint 2 |
| 12 | Revisão completa do front-end | 08/10 – 20/10 | Pendente | Sprint 2 |
| 13 | Entrega do front-end da Sprint 2 | 22/10 | Pendente | Sprint 2 |
| 14 | Início do Back-End (cadastro, autenticação, transações e integração da carteira) | 23/10 – 28/10 | Pendente | Sprint 3 |
| 15 | Desenvolvimento da carteira do usuário e integração com transações | 23/10 – 28/10 | Pendente | Sprint 3 |
| 16 | |  | | --- | |  |  |  | | --- | | Integração do banco de dados com dashboards e ranking de moedas | | 29/10 | Pendente | Sprint 4 |
| 17 | Finalização do Back-End e testes inicias/finais | 30/10 – 11/11 | Pendente | Sprint 4 |
| ID | Tarefa | Data | Status | Sprint |
| 19 | Entrega da 4° Sprint | 12/11 | Pendente | Sprint 4 |
| 20 | Entrega Final – Conclusão das Sprints | 19/11 | Pendente | Sprint 4 |
| 20 | Entrega Final do Projeto | 03/12 | Pendente | Sprint 4 |

### Cronograma de Sprints

|  |
| --- |
| 1° Sprint - Documentação |
| Entregas:   * Levantamento de requisitos funcionais e não funcionais; * Criação da identidade visual do projeto (cores, fontes, logo); * Registro formal da documentação inicial do WallBit. |
| 2° Sprint – Front-End |
| Entrega de Páginas:   * Página Inicial: informações gerais do WallBit; * Tela de Cadastro: formulário para novos usuários; * Tela de Login: autenticação de usuários e direcionamento para perfil; * Perfil de Usuário: histórico de transações, saldo, gráficos simples; * Carteira: exibição do saldo e histórico de transações; * Dashboard de Ranking de Moedas: visualização das moedas mais negociadas.   Interações e Funcionalidades:   * Validação em tempo real de campos obrigatórios; * Feedback visual de sucesso ou erro nas operações; * Navegação entre páginas sem recarregamento (quando possível); * Responsividade para dispositivos móveis e desktop; * Preparação de placeholders para futura integração com back-end, * Visualização da carteira com saldo e histórico parcial (placeholder). |
| 3° Sprint – Back-End |
| Funcionalidades:   * Cadastro e autenticação de usuários, com criptografia de senhas; * Registro e consulta de transações; * Implementação da carteira**:** atualização de saldo conforme transações; * Cálculo de rankings e atualização automática de dados; * Gerenciamento administrativo de usuários e moedas; * Integração com dashboards e gráficos de performance; * Início da integração com o banco de dados relacional. |
| 4° Sprint - Testes Finais |
| Entregas:   * Testes de funcionalidade em todas as páginas, incluindo a carteira; * Testes de responsividade e compatibilidade entre dispositivos; * Implementação de acessibilidade; * Validação de todas as interações e feedbacks visuais; * Correção de bugs e ajustes finais antes da entrega oficial. |

## Banco de Dados

O banco de dados do WallBit foi modelado de forma relacional, contendo as tabelas de usuários, moedas, transações, ranking de moedas e carteira, estruturadas com identificadores únicos e relacionamentos através de chaves primárias e estrangeiras, garantindo consistência e integridade das informações.

### 4.1 Funcionamento do Banco de dados

O banco de dados do Walbit foi estruturado de forma simples e objetiva para garantir o funcionamento essencial da plataforma. Atualmente, ele é composto por cinco tabelas principais: usuários, moedas, transações, ranking de moedas e carteira.

O fluxo de funcionamento ocorre da seguinte forma:

1. Cadastro de Usuário: as informações pessoais ficam armazenadas na tabela usuarios, garantindo que cada cliente tenha um identificador único (id\_usuario).
2. Moedas Cadastradas: os dados das moedas disponíveis, como nome, símbolo e valor atual, ficam registrados na tabela moedas.
3. Carteira do Usuário: cada usuário possui uma carteira que armazena o saldo disponível e registra o histórico de movimentações financeiras, atualizando-se automaticamente a cada transação realizada.
4. Registro de Transações: sempre que um usuário realiza uma compra ou venda, a operação é registrada na tabela transacoes, que guarda o tipo (compra ou venda), quantidade, valor total, data e referência à carteira e à moeda envolvida.
5. Atualização do Ranking: a tabela ranking\_moedas é atualizada com base nas transações, refletindo o volume total de negociações e a posição de cada moeda no ranking.

Relacionamentos das Tabelas:

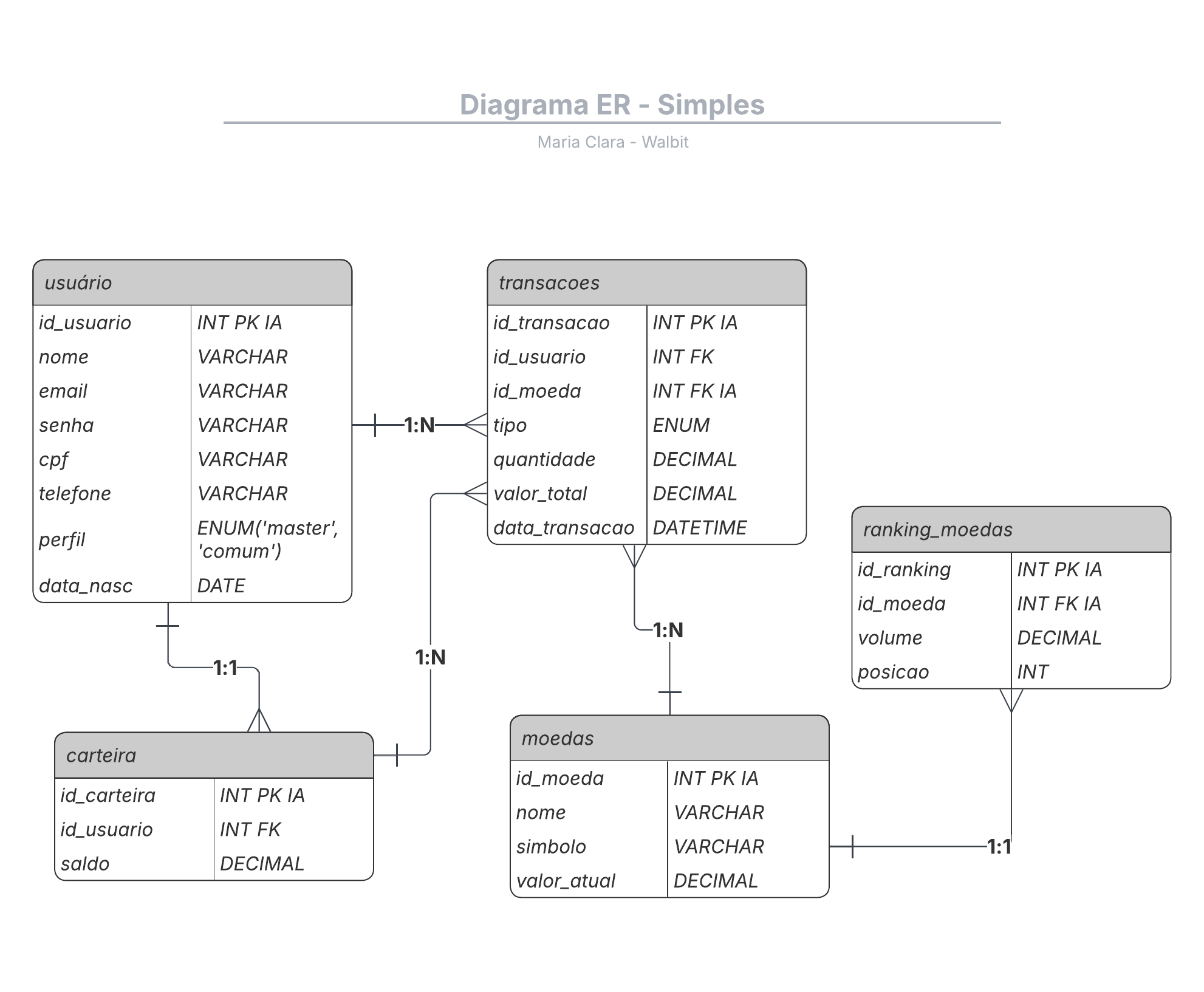
* Usuários → Transações (1:N): um usuário pode realizar várias transações;
* Usuários → Carteira (1:1): cada usuário possui apenas uma carteira;
* Carteira → Transações (1:N): cada transação está associada a uma carteira;
* Moedas → Transações (1:N): uma moeda pode estar associada a várias transações;
* Moedas → Ranking de Moedas (1:1): cada moeda possui apenas um registro no ranking.

Esse modelo garante uma estrutura leve, escalável e consistente, permitindo o funcionamento do sistema de ponta a ponta, desde o cadastro do usuário e controle de saldo até o monitoramento de desempenho das moedas.

Observação: Este é o modelo inicial, mas o banco de dados foi projetado de forma a permitir futuras expansões, como inclusão de tabelas para perfis de acesso (master/admin), logs de auditoria, histórico detalhado de preços das moedas e estatísticas avançadas.

### 4.2 Diagrama de Relacionamento

A seguir, é apresentado o Diagrama Entidade-Relacionamento do banco de dados do Walbit, representando graficamente as tabelas principais (usuarios, carteira, moedas, transacoes e ranking\_moedas) e seus respectivos relacionamentos. O modelo ilustra de forma clara a estrutura atual do sistema, destacando as ligações entre usuários, moedas, transações e a carteira. Ressalta-se que este é um modelo inicial, podendo ser expandido futuramente para contemplar novas entidades e funcionalidades, conforme a evolução do projeto.



## Testes

A fim de garantir a qualidade, confiabilidade e usabilidade do WallBit, diversos testes serão realizados em diferentes camadas do sistema:

1. Testes de Front-End

* Funcionalidade de formulários: validação de campos obrigatórios, formatos de dados (e-mail, CPF, senha), feedback visual de sucesso ou erro;
* Navegação entre páginas: checagem da consistência dos links, menus e redirecionamentos;
* Responsividade: verificação do layout em diferentes dispositivos (desktop, tablet e smartphone), garantindo adaptação de conteúdo e elementos visuais;
* Feedbacks visuais: alertas, mensagens de confirmação e de erro em operações de cadastro, login, transações e interação com dashboards;
* Interações com a carteira: atualização do saldo em tempo real, histórico de movimentações e integração com transações simuladas.

2. Testes de Back-End

* Cadastro e autenticação de usuários: verificação de criação de contas, login, criptografia de senhas e perfis de acesso (usuário/mestre);
* Transações e carteira: registro correto de transações, atualização automática do saldo da carteira e associação com usuários e moedas;
* Ranking de moedas: cálculo correto do volume negociado e atualização da posição no ranking;
* Integração com dashboards: conferência de dados exibidos em gráficos, tabelas e históricos de transações;
* Segurança e consistência de dados: validação de integridade referencial entre tabelas (usuarios, carteira, transacoes, moedas e ranking\_moedas).

3. Testes de Acessibilidade

* Modo escuro e contraste: verificação do contraste de cores, legibilidade de textos e elementos gráficos;
* Navegação via teclado: checagem da possibilidade de acessar todos os campos, botões e menus sem necessidade do mouse;
* Compatibilidade com leitores de tela: testes de leitura correta dos elementos interativos e conteúdos informativos.

Observação: Todos os testes serão documentados e relatados, permitindo ajustes e melhorias contínuas no sistema, garantindo um produto funcional, seguro e acessível a todos os usuários.

## Mockups das Funcionalidades Atuais do Sistema

Segue abaixo a apresentação dos mock-ups do WalBit, acompanhados de suas respectivas descrições, permitindo visualizar a estrutura atual e planejada do sistema:



* **Home Page (Atual)** – Mock-up da página inicial, apresentando a identidade visual do WalBit e os principais pontos de navegação do usuário.



**Tela de Cadastro (Atual)** – Formulário de registro de novos usuários, incluindo campos para informações pessoais e criação de senha.



**Tela de Login (Atual)** – Interface de autenticação do usuário, com campos para inserção de credenciais e botão de acesso.



**Tela de Erro (Atual)** – Mensagem informativa exibida quando ocorre um problema no acesso ou uso do sistema, orientando o usuário.



## Conclusão

O projeto WalBit foi desenvolvido com o objetivo de criar uma plataforma funcional, segura e escalável para gerenciamento de transações de moedas digitais, contemplando tanto o front-end, com interfaces intuitivas e responsivas, quanto o back-end, com banco de dados relacional estruturado e integração com carteira de usuários.

A documentação apresentou o levantamento de requisitos, a identidade visual, o planejamento das sprints, o diagrama de banco de dados e as estratégias de testes que asseguram a qualidade do sistema. Com a implementação das funcionalidades planejadas, incluindo cadastro, login, transações, carteira, dashboards e ranking de moedas, o WalBit garante uma experiência completa para o usuário, respeitando princípios de usabilidade, acessibilidade e segurança.

Além disso, o modelo do banco de dados foi concebido de forma a permitir futuras expansões, como inclusão de perfis administrativos, histórico detalhado de transações, estatísticas avançadas e logs de auditoria.

Portanto, o WalBit não apenas cumpre os objetivos propostos para este projeto acadêmico, mas também estabelece uma base sólida para evoluções futuras, demonstrando a aplicabilidade das metodologias, tecnologias e boas práticas adotadas durante todo o desenvolvimento.