**1. Introdução**

**1.1. Descrição do Projeto**

Este projeto é um simulador de banco simples desenvolvido como parte de uma avaliação do curso Formação Acelerada em Programação (FAP). O programa foi construído utilizando o framework Django, com o banco de dados MySQL, acessado e gerenciado por meio do MySQL WorkBench. O objetivo do projeto é permitir a criação, edição e manipulação de contas bancárias (Conta Corrente e Conta Poupança), além de permitir movimentações financeiras e a visualização de extratos e gráficos.

**1.2. Objetivos**

* Desenvolver um sistema de gerenciamento de contas bancárias.
* Implementar funcionalidades de criação, edição e manipulação de contas e usuários.
* Permitir operações de depósito, saque e transferência (implementação pendente) entre contas.
* Fornecer recursos de visualização de extratos, movimentações da conta e gerencia de usuários e contas.
* Aprofundar o conhecimento em Django, Python, Programação Orientada a Objeto (POO) e Bancos de Dados (MySQL).

**1.3. Público-Alvo**

Este sistema é destinado a ser uma ferramenta educacional para demonstrar a aplicação prática de conceitos de desenvolvimento web, gerenciamento de banco de dados e programação em Django.

**2. Tecnologias Utilizadas**

**2.1. Linguagem de Programação**

* **Python (Django)**: Utilizado como o principal framework para o desenvolvimento do backend e das funcionalidades do sistema.
* **HTML, CSS (Bootstrap 5)**: Utilizado no front-end para exibição do sistema.

**2.2. Banco de Dados**

* **MySQL**: Banco de dados relacional utilizado para armazenamento e manipulação de dados. As operações de banco de dados foram gerenciadas por meio do MySQL WorkBench.

**2.3. Ferramentas de Desenvolvimento**

* **Django Admin**: Utilizado para gerenciar modelos, usuários e dados no ambiente administrativo.
* **MySQL WorkBench**: Ferramenta utilizada para modelagem, administração e gerenciamento do banco de dados MySQL.

**3. Estrutura do Sistema**

**3.1. Modelos**

Os principais modelos do sistema representam usuários e contas bancárias e movimentações das contas bancárias. Cada modelo foi definido de forma a refletir a lógica de negócio do sistema.

**3.1.1. Usuário**

* **Campos**:
  + username: Nome de usuário único para login.
  + email: Endereço de e-mail.
  + password: Senha criptografada para autenticação.
  + first\_name: Nome.
  + last\_name: Sobrenome
  + is\_superuser: Administrador (abstração de Gerente).
  + last\_login: Data / hora do última sessão cadastrada.
  + date\_joined: Data / hora da criação da conta.
* **Funcionalidades**:
  + Criação e edição de usuários.
  + Autenticação e controle de acesso ao sistema.

**3.1.2. Conta Bancária**

* **Campos**:
  + numero\_da\_conta: Número único com 6 dígitos gerados randomicamente no momento da criação da conta, serve para identificação da conta bancária.
  + tipo\_de\_conta: Tipo da conta (Corrente ou Poupança).
  + saldo: Saldo atual da conta.
  + limite\_especial: Valor de crédito extra para a movimentação da conta.
  + id\_user\_id: Relacionamento com o modelo Usuário (chave estrangeira).
  + data\_abertura: Data / hora da criação da conta.
  + ativa: Flag para encerramento da conta (*soft delete*).
* **Funcionalidades**:
  + Criação, edição e encerramento de contas.
  + Operações de depósito, saque e transferência (não implementado).
  + Visualização de resumo de movimentação, extratos e gráfico de movimentação e média de saldo da conta.

**3.2. Diagramas de Modelagem**

A seguir, são apresentados os diagramas que ilustram as diferentes fases de modelagem do banco de dados deste projeto. Estes diagramas servem para documentar e visualizar a estrutura dos dados e suas relações, desde a concepção inicial até a implementação física no banco de dados MySQL.

**3.2.1. Diagrama Conceitual**

O Diagrama Conceitual define a estrutura inicial do banco de dados, focando nos principais objetos do sistema e suas relações de forma abstrata. Ele não se preocupa com detalhes técnicos, mas sim com a representação das entidades e suas interações.



Figura 1

**3.2.2. Diagrama Lógico**

O Diagrama Lógico refina a modelagem conceitual, adicionando mais detalhes sobre os atributos das entidades e especificando os tipos de relacionamento. Este diagrama é mais técnico e prepara o terreno para a implementação física do banco de dados.

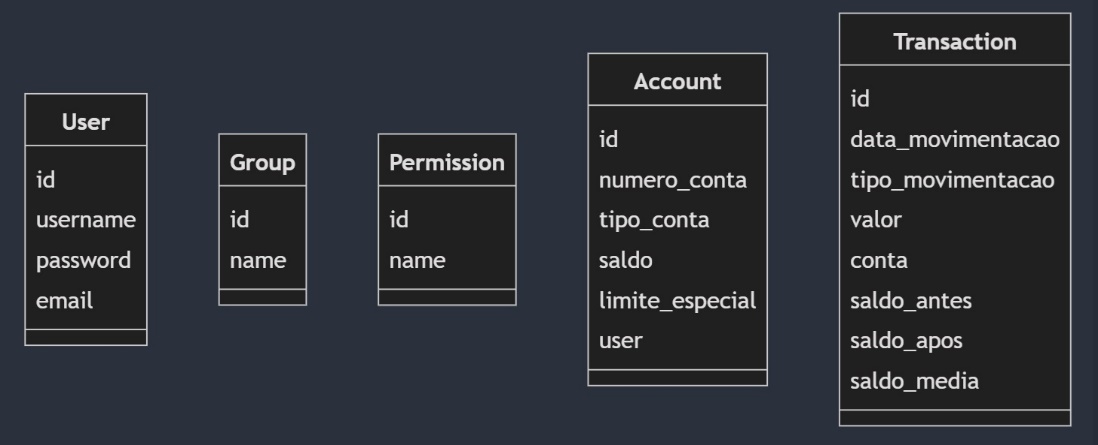


Figura 2

**3.2.3. Diagrama Físico**

O Diagrama Físico traduz o modelo lógico para uma representação real, mostrando como o banco de dados será implementado no MySQL. Ele inclui a estrutura das tabelas, colunas, tipos de dados e índices, refletindo a configuração exata do banco de dados no ambiente de produção.

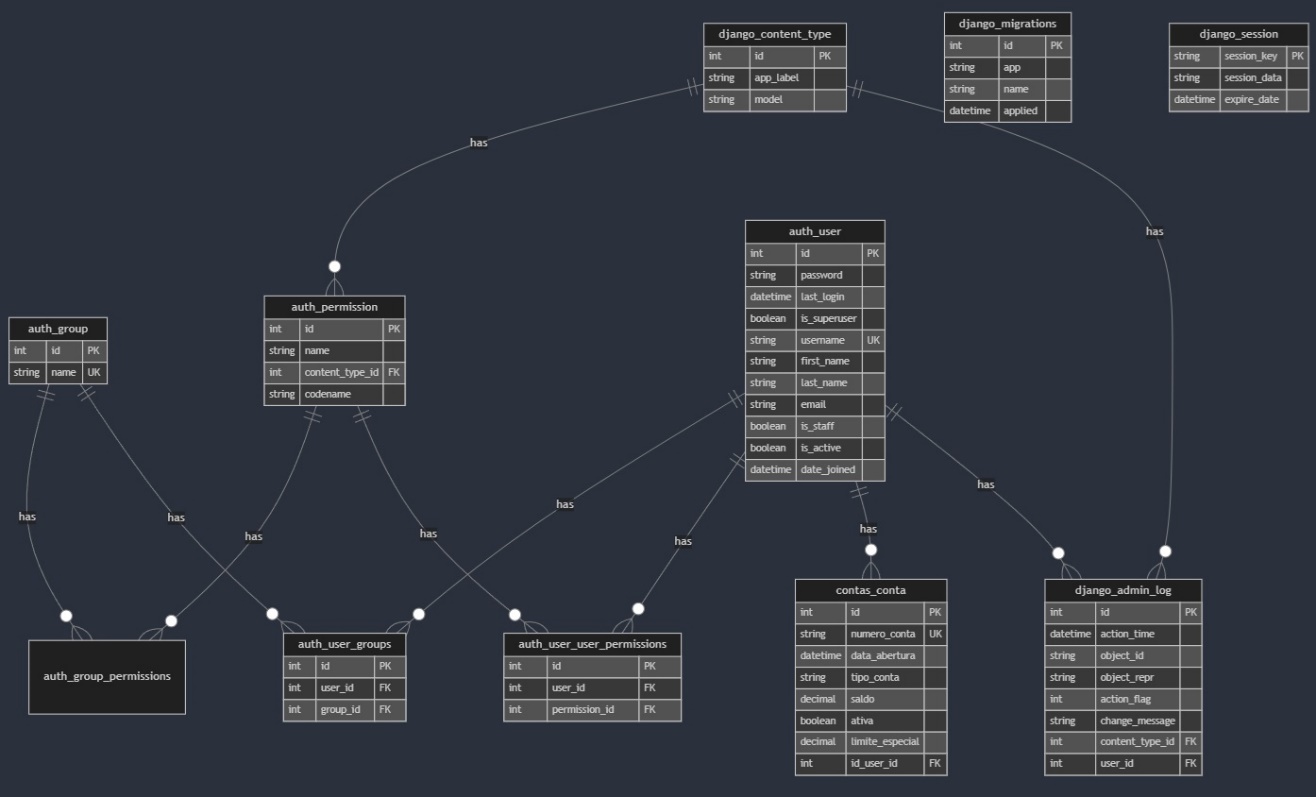


Figura 3

**3.3. Funcionalidades Principais**

**3.3.1. Criação de Usuários e Contas**

* O sistema permite a criação de novos usuários e sua edição, que podem gerenciar suas contas bancárias. Somente o Gerente pode realizar a criação e edição de contas bancárias para os usuários.

**3.3.2. Movimentação Financeira**

* As operações de depósito e saque são realizadas diretamente nas contas dos usuários.
* A transferência entre contas ainda está em fase de implementação.

**3.3.3. Visualização de Extratos**

* Os usuários podem visualizar extratos de suas contas, com filtros por período e tipo de movimentação.
* Existe a possibilidade de imprimir os extratos utilizando o próprio navegador.
* A implementação da geração de PDF para download está pendente.

**3.3.4. Gráficos e Relatórios**

* Gráfico é gerado para mostrar as movimentações e a média do saldo das contas ao longo do tempo.

**4. Interface do Usuário**

**4.1. Design e Navegação**

O sistema foi desenvolvido com uma interface amigável, utilizando os recursos de templates do Django para renderizar as páginas HTML. As páginas foram estilizadas com CSS puro e também com o framework de CSS Bootstrap 5, além da biblioteca Materials UI do Google.

**4.2. Diagramas de Operações e Sequência**

Nesta seção, são apresentados os diagramas que ilustram os fluxos de operações e a sequência de interações dentro do sistema. Esses diagramas são fundamentais para entender como as funcionalidades são implementadas e como os diferentes componentes do sistema interagem entre si durante a execução de operações específicas.

**4.2.1. Diagrama de Operações: Depósito**

O Diagrama de Operações para a funcionalidade de depósito descreve o fluxo de atividades desde a solicitação do usuário até a atualização do saldo da conta bancária. Este diagrama detalha os passos envolvidos, incluindo a verificação de dados, processamento da transação e geração do registro de movimentação.

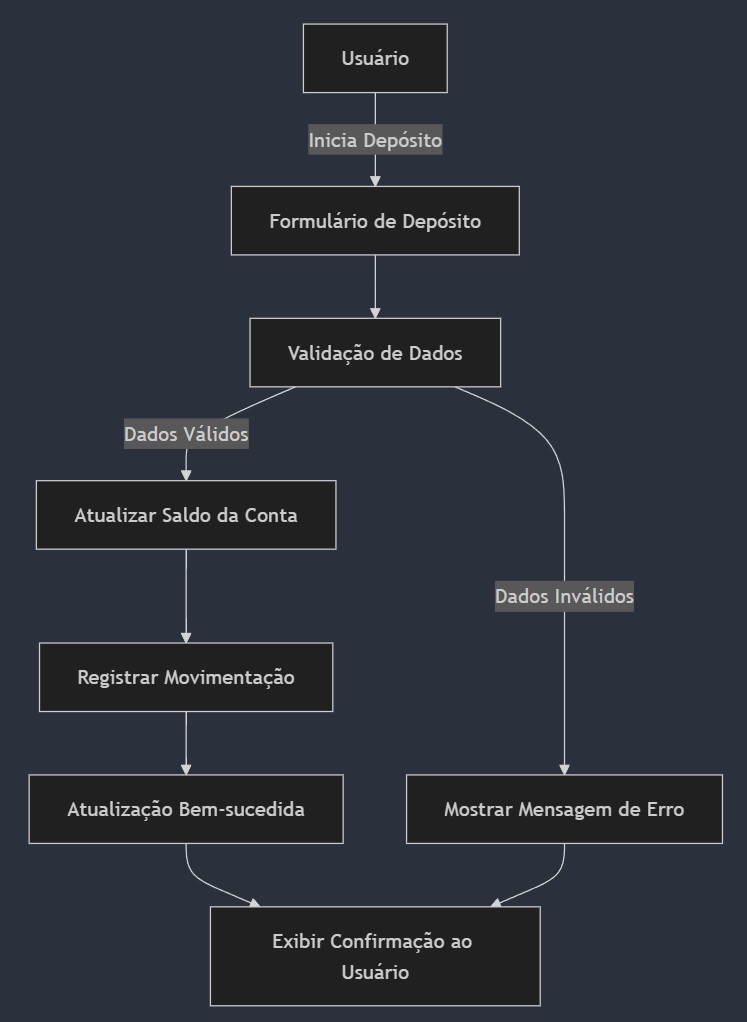


Figura 4

**4.2.2. Diagrama de Operações: Saque**

O Diagrama de Operações para a funcionalidade de saque descreve o processo pelo qual o sistema gerencia as solicitações de saque, verifica a disponibilidade de fundos, atualiza o saldo da conta e registra a transação. Este diagrama ajuda a visualizar o fluxo completo de uma operação de saque.

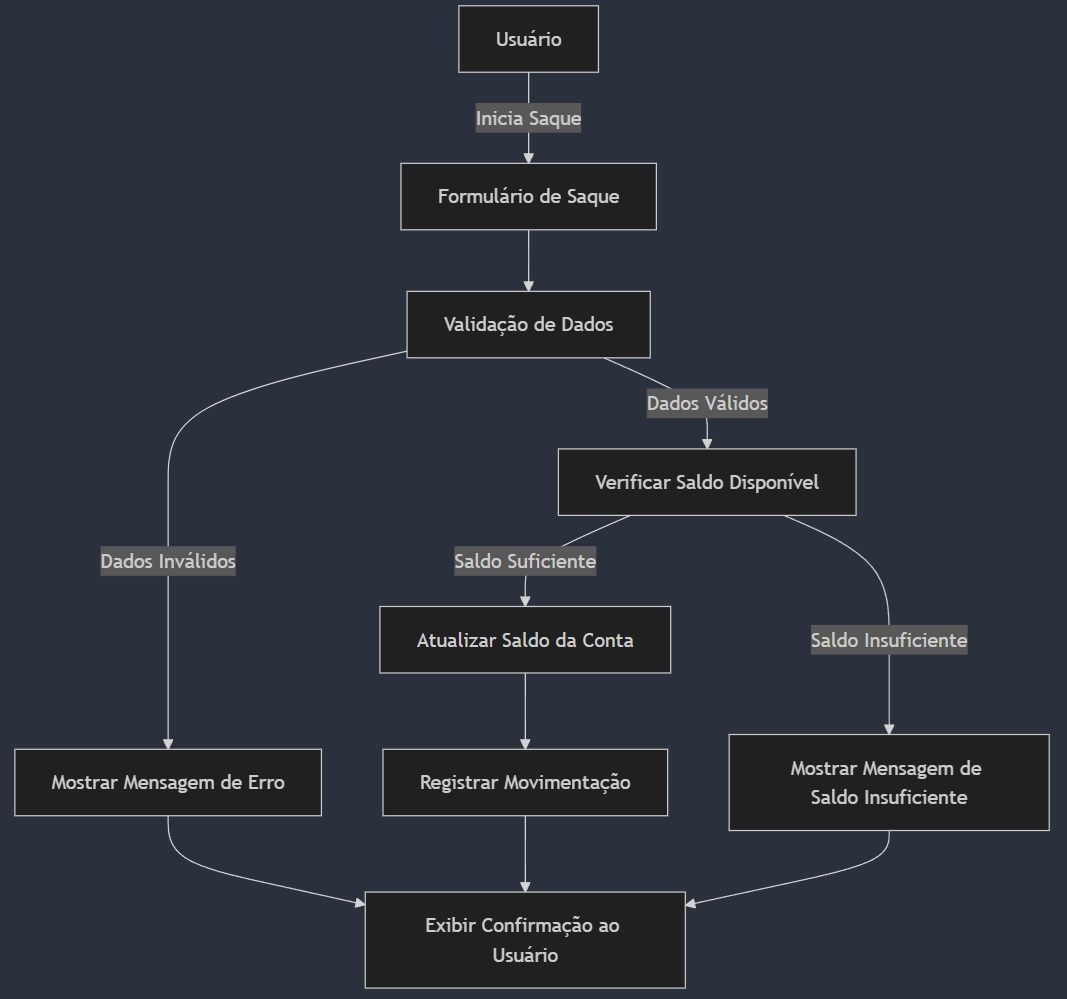


Figura 5

**4.2.3. Diagrama de Sequência: Depósito**

O Diagrama de Sequência para a funcionalidade de depósito apresenta as interações entre os objetos do sistema durante a execução da operação de depósito. Ele mostra a ordem em que as mensagens são trocadas, desde o início da solicitação até a conclusão da transação.

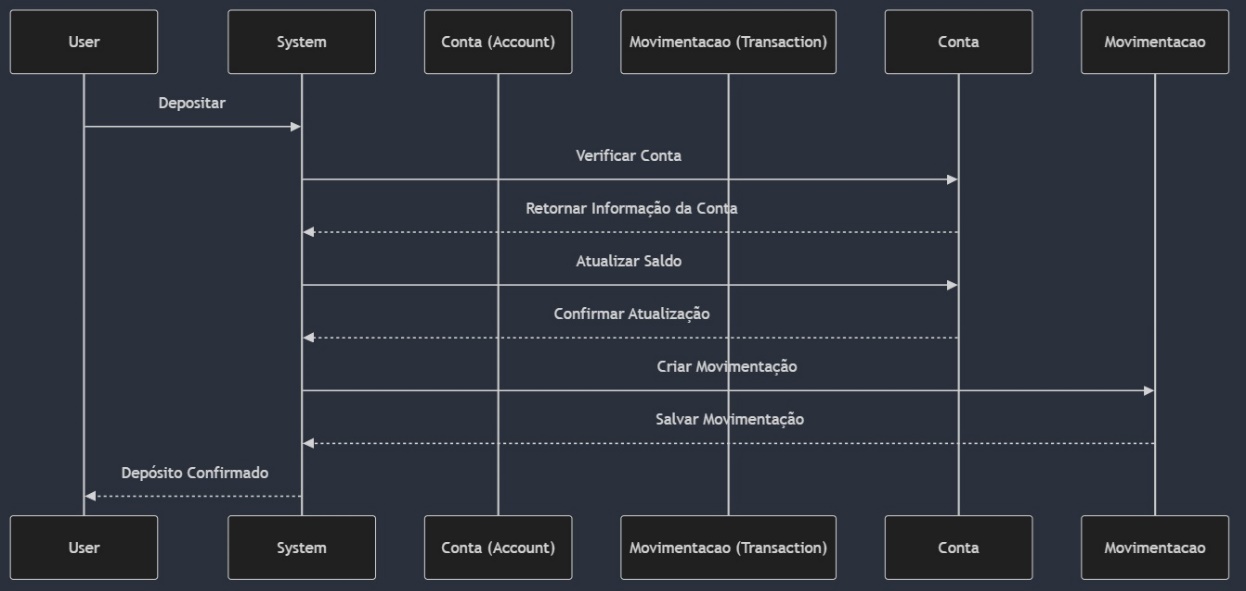


Figura 6

**4.2.4. Diagrama de Sequência: Saque**

O Diagrama de Sequência para a funcionalidade de saque detalha as interações entre os diferentes componentes do sistema, ilustrando a troca de mensagens e a sequência de chamadas de métodos necessários para completar a operação de saque.

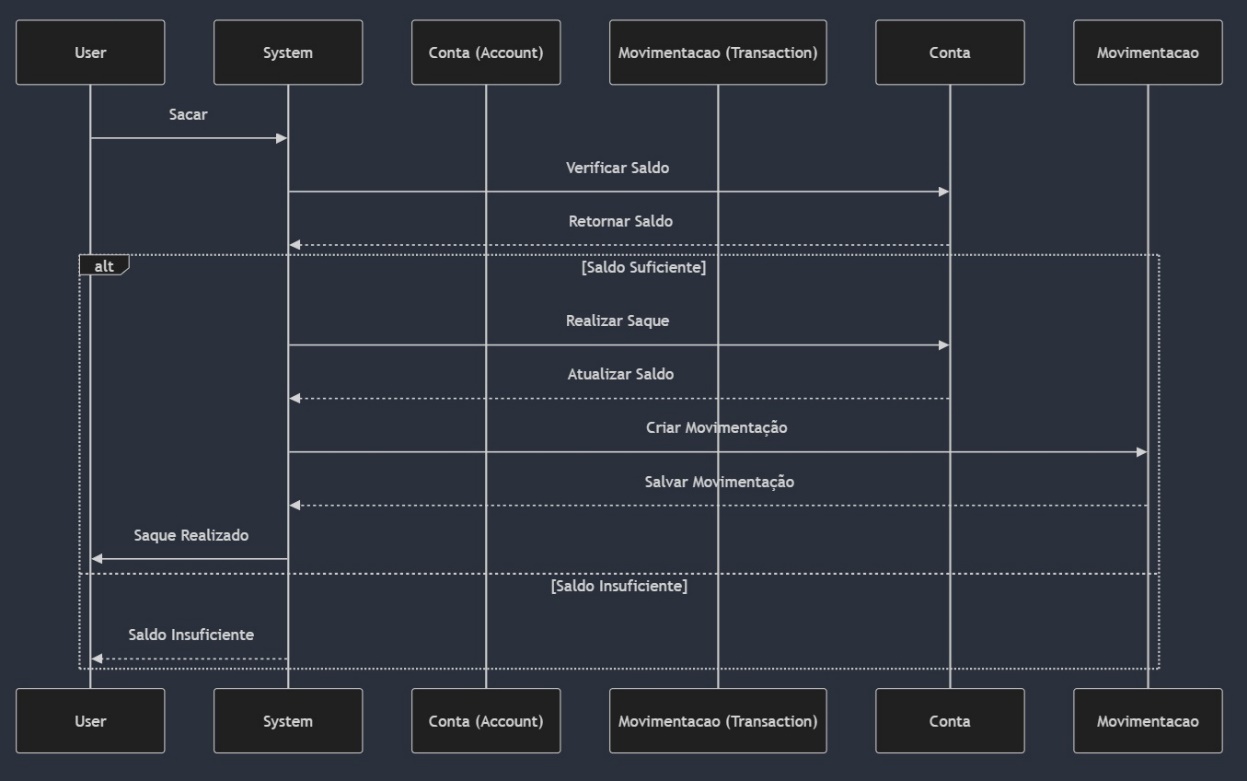


Figura 7

**4.3. Diagramas de Usuário**Os diagramas de usuário descrevem as interações dos diferentes perfis de usuário, como gerente e correntista, com o sistema. Estes diagramas demonstram as funcionalidades acessíveis a cada perfil e como eles navegam e utilizam as diferentes partes do sistema.

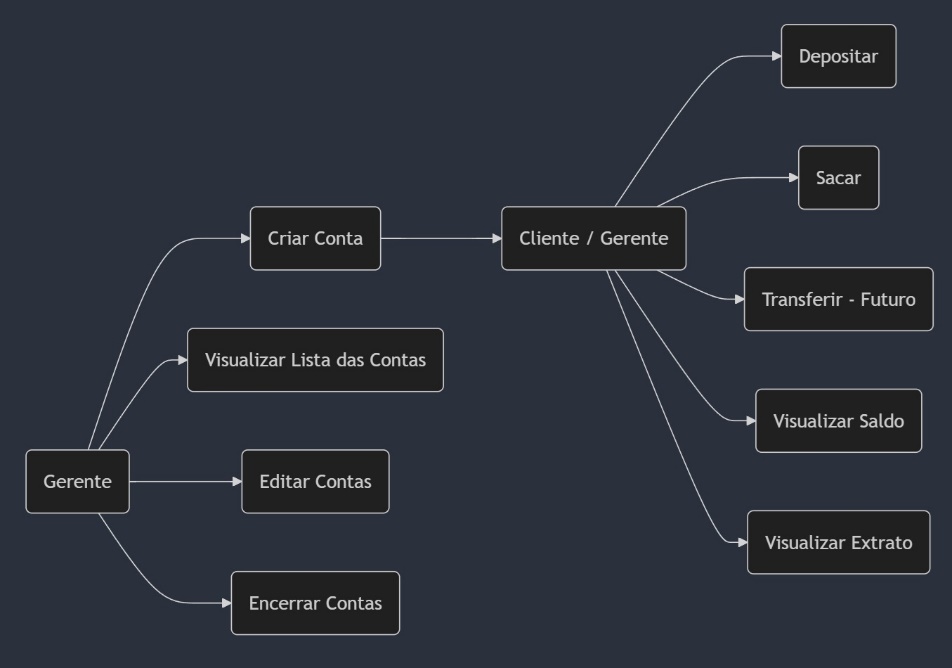


Figura 8

**4.4. Diagramas de Tempo**

Os diagramas de tempo mostram a sequência temporal das operações dentro do sistema. Estes diagramas são úteis para visualizar como as ações ocorrem ao longo do tempo, permitindo uma melhor compreensão dos prazos envolvidos em cada operação.



Figura 9

**4.5. Capturas de Tela da Interface**

Esta seção apresenta capturas de tela do sistema, oferecendo uma visão visual das principais páginas e funcionalidades. Cada captura de tela é acompanhada de uma descrição detalhada sobre sua função e o contexto de uso.

**4.5.1. Tela de Login**A tela de login é a interface inicial do sistema onde os usuários inserem suas credenciais para acessar o sistema. É a primeira página acessada após a abertura do aplicativo.

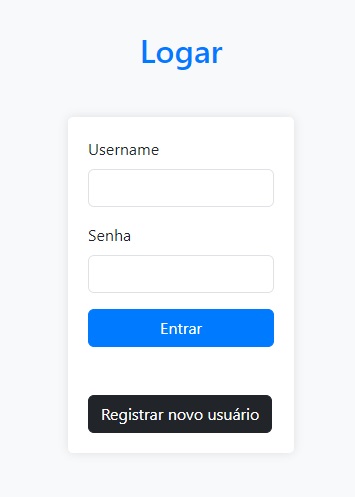


Figura 10

**4.5.2. Tela de Cadastro de Usuário**Esta tela permite a criação de novos usuários no sistema, fornecendo campos para a inserção de informações como nome, e-mail e senha.

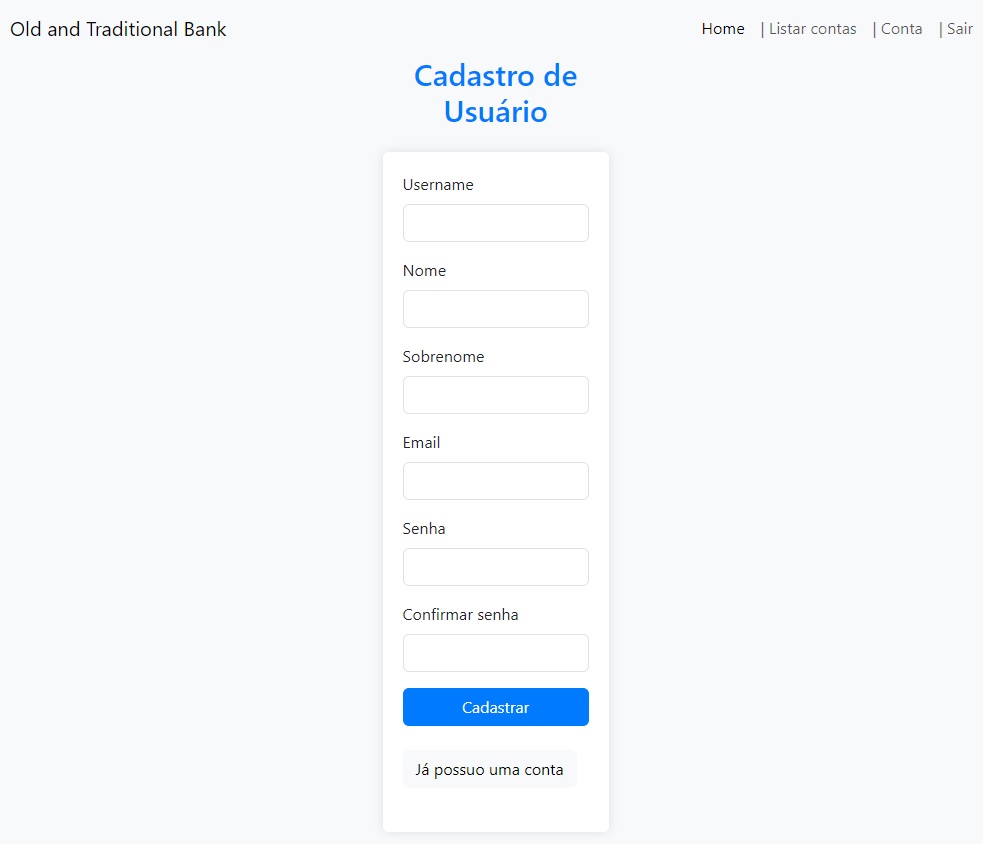


Figura 11

**4.5.3. Tela de Conta do Cliente**A tela de conta do cliente exibe detalhes sobre a conta do usuário, incluindo saldo e informações da conta, resumo de movimentação e é onde o cliente pode realizar as movimentações de depósito e saque. É acessada após o login do usuário caso ele já possua uma conta no banco.

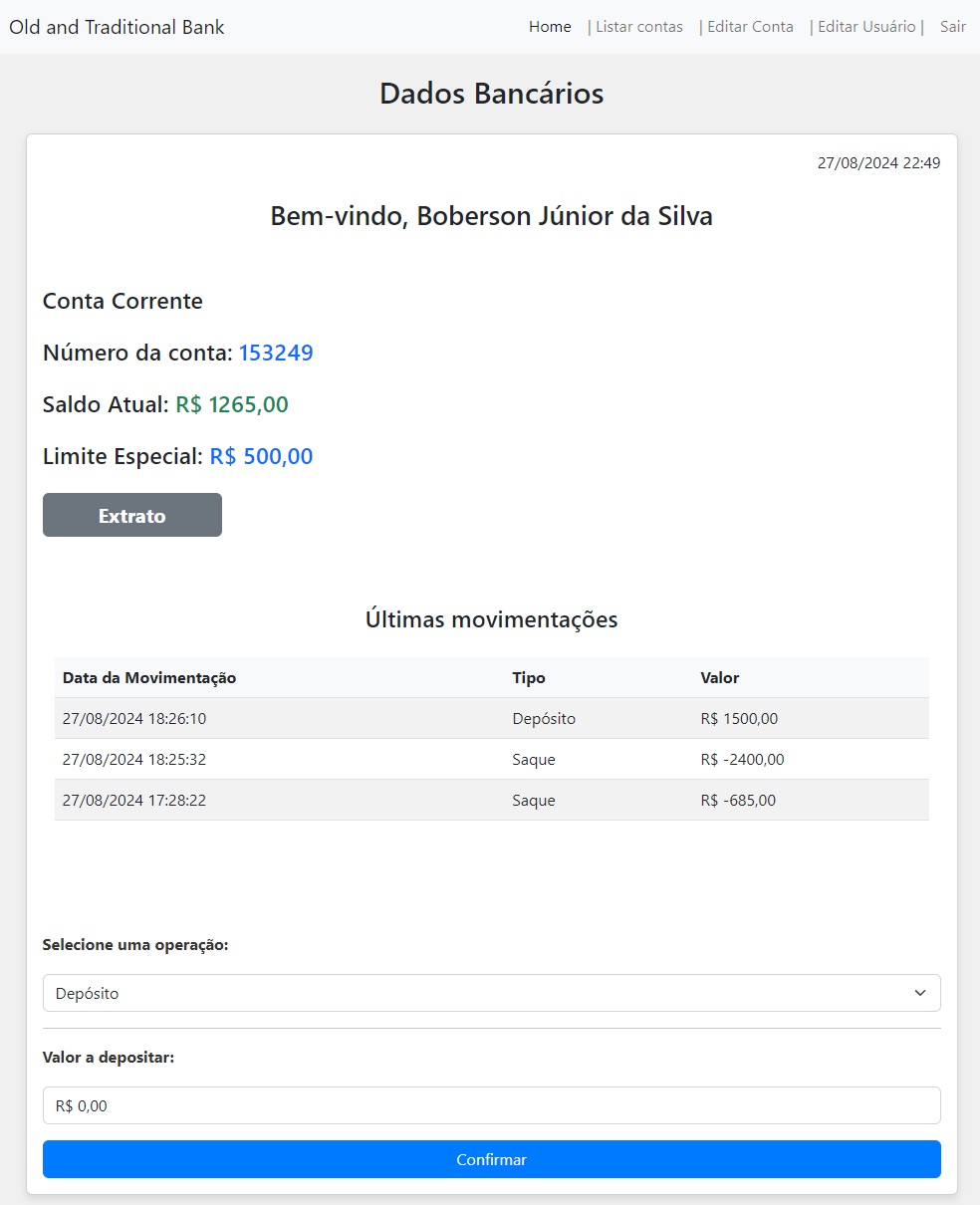


Figura 12

**4.5.4. Tela de Extrato de Conta**Esta tela mostra o extrato da conta do cliente, permitindo a visualização e filtragem das transações realizadas, assim como a média do saldo da conta e gráfico dessas informações.

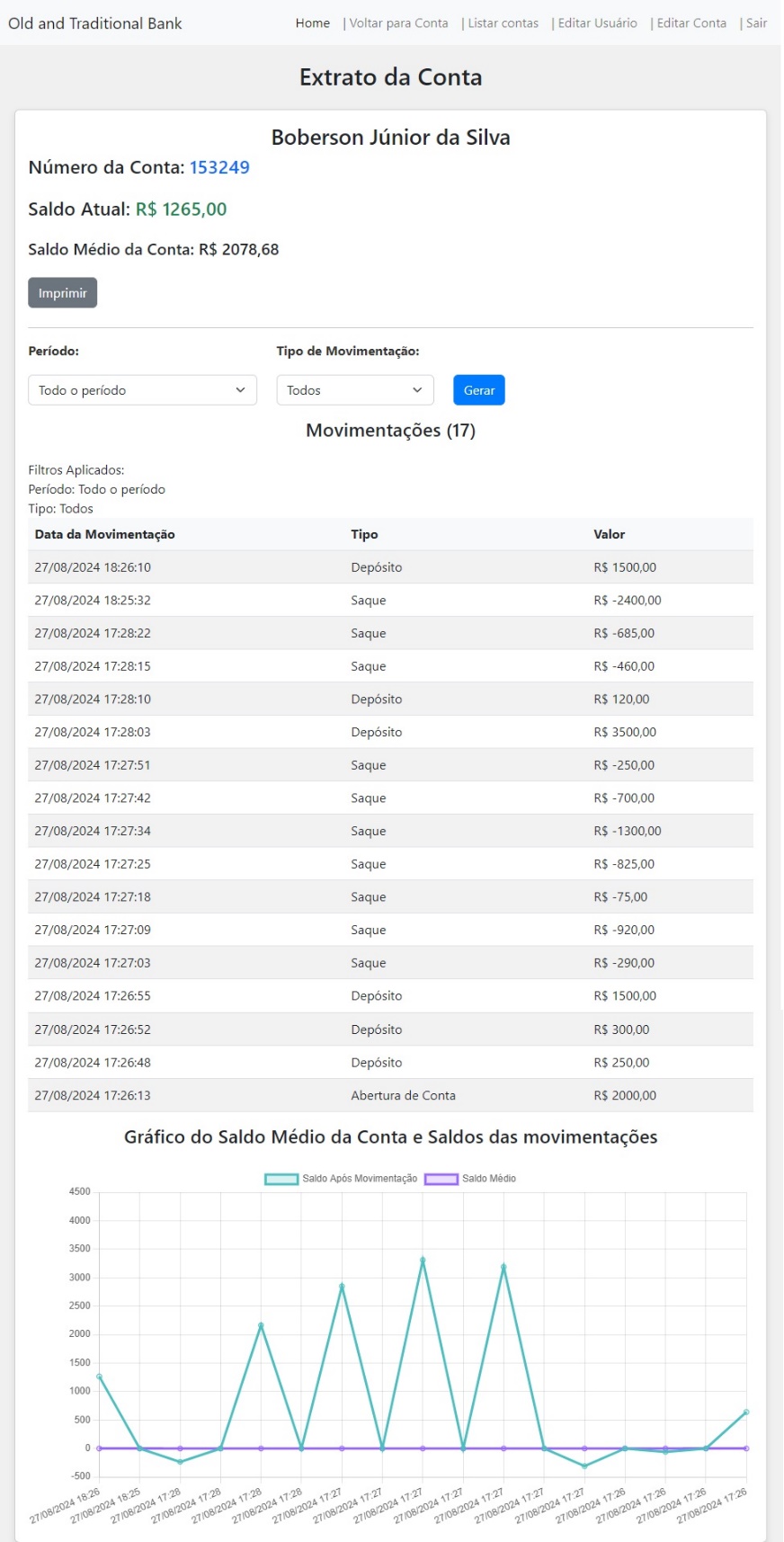


Figura 13

**4.5.5. Tela de Edição de Conta**Na tela de edição de conta, os Gerentes podem modificar informações relacionadas às contas, como o limite especial e sua ativação ou encerramento. O restante dos dados não pode ser editado diretamente no sistema. E o encerramento da conta só é possível caso o saldo da conta seja zero.



Figura 14

**4.5.6. Tela de Edição de Usuário**Esta tela é utilizada para atualizar os dados do perfil de usuário, como nome e e-mail e senha.

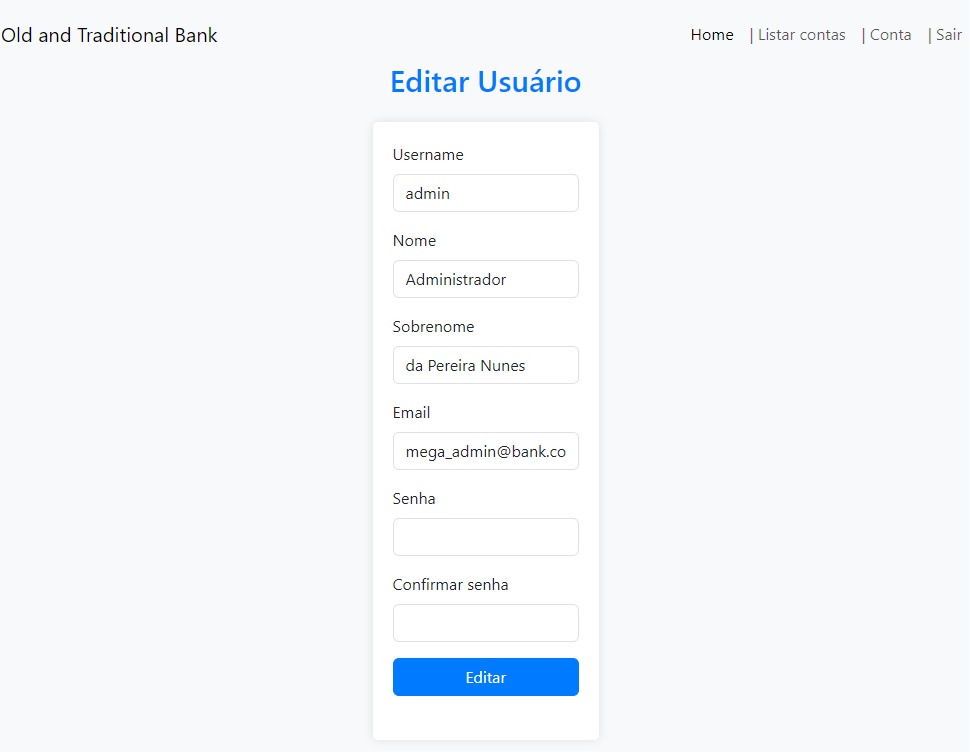


Figura 15

**4.5.7. Tela de Listagem de Contas**A tela de listagem de contas apresenta uma visão geral de todas as contas existentes no sistema, com opções para edição, encerramento e ativação das contas. Somente o Gerente (abstração do admin do sistema) tem permissão para acessar essa página, assim como realizar edições.

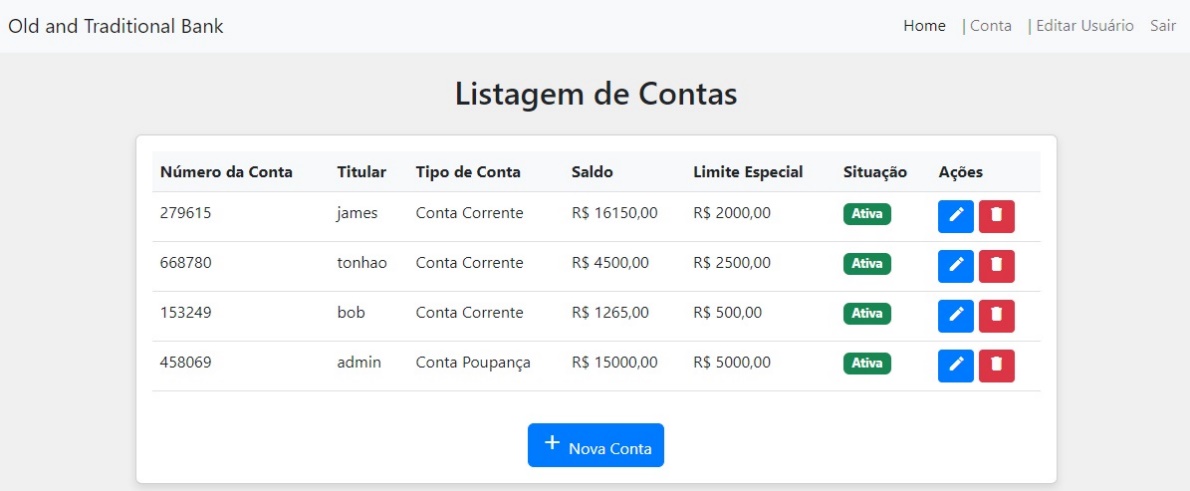


Figura 16

**4.5.8. Tela de Cadastro de Contas**

A tela de cadastro de contas é utilizada para criar contas bancárias. O formulário inclui campos para o número da conta, tipo de conta, saldo inicial, e outros detalhes necessários para o registro de uma nova conta bancária.



Figura 17

**5. Banco de Dados**

**5.1. Estrutura do Banco de Dados**

A base de dados foi projetada com as seguintes tabelas principais:

* **Usuarios**: Armazena informações sobre os usuários cadastrados. Aqui foi utilizada a estrutura disponibilizada pelo próprio Django.
* **Contas**: Armazena as contas bancárias, vinculadas aos usuários.
* **Movimentacoes**: Armazena todas as operações financeiras realizadas nas contas.

**5.2. Relacionamentos**

* **Usuários-Contas**: Um usuário pode ter várias contas (Relacionamento 1 – n).
* **Contas-Movimentações**: Uma conta pode ter várias movimentações (Relacionamento 1 - n).

**6. Instalação e Configuração**

**6.1. Pré-requisitos**

* **Python 3.x**: Linguagem de programação.
* **Django 5.x**: Framework web.
* **MySQL (WorkBench)**: Banco de dados.

**6.2. Configuração do Ambiente**

1. **Clone o repositório**:

git clone <url\_do\_repositorio>

1. **Instale as dependências**:

pip install -r requirements.txt

1. **Configuração do Banco de Dados**:
   * Crie um banco de dados MySQL (ou importe o arquivo de dump disponibilizado).
   * Atualize as configurações de conexão no arquivo settings.py.

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': '<nome\_do\_banco>',

'USER': '<usuario>',

'PASSWORD': '<senha>',

'HOST': 'localhost',

'PORT': '3306',

}

}

1. **Realize as migrações**:

python manage.py makemigrations

python manage.py migrate

1. **Execute o servidor de desenvolvimento**:

python manage.py runserver

**7. Testes**

**7.1. Testes Automatizados**

O sistema inclui testes automatizados para as principais funcionalidades, que podem ser executados utilizando o Django Test Framework:

python manage.py test

**7.2. Cobertura de Testes**

Os testes abrangem:

* Autenticação e controle de acesso.
* Criação e manipulação de contas e usuários.
* Realização de movimentações financeiras.

**8. Conclusão**

**8.1. Aprendizados e Desafios**

Este projeto proporcionou uma excelente oportunidade para aprofundar o conhecimento em desenvolvimento web utilizando Django e MySQL. O principal desafio foi garantir a integridade dos dados e a correta implementação das funcionalidades financeiras.

**8.2. Próximos Passos**

* Implementar a funcionalidade de transferência entre contas.
* Melhorar a interface gráfica para maior usabilidade.
* Implementar o uso de API utilizando o Django REST Framework (DRF).

**9. Referências**

* Documentação oficial do Django: <https://docs.djangoproject.com/>
* Documentação oficial do MySQL: <https://dev.mysql.com/doc/>