Nome do projeto: Grupo 26

Alunos: Alyson Vieira - RM355327,

Eric Moraes - RM348492,

Leonardo Brambilla - RM355421,

Rodolfo Strunkis - RM350459,

Willi Ramon Sabino - RM355620.

História

O sistema de abertura de conta poupança é projetado para tornar o processo de poupança simples e rápido. A seguir, estão descritas as etapas do funcionamento do sistema desenvolvido:

Funcionamento do Sistema

Solicitação da Conta: O usuário solicita a abertura de uma conta poupança através do aplicativo ou em uma das agências.

Envio de Documentação: O banco solicita os documentos necessários para a abertura da conta, que podem ser enviados de forma totalmente digital.

Análise e Aprovação: A equipe do banco analisa a documentação e aprova a abertura da conta. Em poucos dias, a confirmação é enviada ao usuário.

Depósito Inicial: Para ativar a conta é necessário fazer um depósito inicial que pode ser um valor simbólico.

Recebimento do cartão poupança: Após a confirmação do depósito, o cartão e os dados da conta são enviados ao usuário.

Desbloqueio e Acesso: O usuário desbloqueia o cartão e acessa a conta pelo aplicativo ou em um caixa eletrônico.

Consulta de Extrato: O extrato pode ser solicitado pelo aplicativo, sendo enviado rapidamente ao usuário.

Saque de Valores: O usuário pode sacar dinheiro da conta poupança, com o valor debitado automaticamente e a notificação da transação sendo enviada.

Depósitos Adicionais: Para aumentar a poupança, basta fazer um depósito adicional. O valor é creditado na conta e o usuário é notificado da operação.

Transferências: O usuário pode transferir dinheiro para outra conta pelo aplicativo, sendo informado em cada etapa do processo.

O sistema atualiza o saldo ao final de cada operação e, periodicamente, adiciona os juros, garantindo o crescimento da poupança ao longo do tempo. Com este sistema de abertura de conta poupança, economizar se torna simples e acessível.

Motivação

O projeto de abertura de conta poupança foi escolhido com base em nossa experiência pessoal com o processo de abertura de contas poupança e no desejo de transformar essa experiência em um projeto concreto. Observamos que muitas pessoas enfrentam dificuldades e barreiras ao tentar iniciar sua poupança, seja pela complexidade do processo ou pela falta de orientação adequada. Nossa motivação principal é simplificar esse processo, tornando-o mais acessível e eficiente para todos.

Justificativa

Para tornar a poupança mais acessível e eficaz para todos, estamos melhorando nosso sistema de abertura de conta poupança.

Uma das melhorias mais notáveis foi a simplificação do procedimento de abertura de contas. Os usuários podem solicitar rapidamente uma conta poupança usando nosso aplicativo, evitando filas em agências ou preenchendo formulários pessoalmente. Mais pessoas estão começando a economizar, pois esse processo digital reduz significativamente as barreiras tradicionais.

Além disso, a comodidade permanece após a abertura da conta. Os usuários podem facilmente consultar saldos, saques, depósitos e transferências por meio do aplicativo. Atualizações de saldo e notificação de transações em tempo real melhoram a gestão financeira pessoal.

Um processo mais rápido de análise e aprovação da documentação garante a abertura da conta em poucos dias. A poupança é acessível a todos, mesmo que o depósito inicial seja mínimo. O kit de boas-vindas com os dados da conta e o cartão é enviado rapidamente após a confirmação do depósito. Isso permite o desbloqueio e acesso imediato à conta.

A implementação desses recursos melhora a forma como as contas de poupança são abertas e ensina as pessoas a planejar e economizar dinheiro. Nosso sistema incentiva os usuários a economizar de forma mais inteligente e eficaz ao facilitar o acesso e a gestão de suas poupanças.

Para resumir, é necessário tornar o processo de poupança mais fácil, conveniente e claro para que todos possam aproveitar os benefícios e aumentar o número de pessoas que o usam. Essas mudanças não apenas tornam o processo mais fácil, mas também promovem uma gestão financeira mais inteligente e eficaz.

Arquitetura

Descrição Geral

O sistema de abertura de conta poupança é uma aplicação web robusta, desenvolvida utilizando o framework Spring Boot. Ele é composto por diversas camadas que garantem a modularidade e a eficiência do processo, desde a recepção das requisições dos usuários até a persistência dos dados no banco de dados. A aplicação abrange operações essenciais de gestão de clientes, incluindo criação, atualização, deleção e consulta de informações.

Estrutura do Sistema

- **Camada de Apresentação (Controller)**:
  - **ClientController**: Responsável por receber e gerenciar as requisições HTTP relacionadas às operações dos clientes, como criação, atualização, deleção e consulta.
- **Camada de Serviço (Service)**:
  - **ClientService**: Contém a lógica de negócio para manipulação dos dados dos clientes. Interage com a camada de persistência para realizar operações no banco de dados.
- **Camada de Persistência (Repository)**:
  - **ClientRepository**: Interface que extende JpaRepository, fornecendo métodos para operações CRUD (Create, Read, Update, Delete) no banco de dados.
- **Camada de Domínio (Domain)**:
  - **Client**: Entidade JPA que representa a tabela de clientes no banco de dados.
  - **ClientRequest**: Objeto de transferência de dados utilizado para receber dados do cliente nas requisições.
  - **ClientDTO**: Objeto de transferência de dados utilizado para enviar dados do cliente nas respostas.
- **Configuração**:
  - **SwaggerConfig**: Configuração do Swagger para documentação e testes das APIs.
  - **application.yml**: Arquivo de configuração da aplicação.
  - **docker-compose.yml**: Arquivo de configuração para containerização da aplicação usando Docker.
- **Scripts de Inicialização**:
  - **data.sql**: Script SQL para inserção de dados iniciais no banco de dados.

Componentes da Arquitetura

- **Aplicação Web (Backend)**:
  - **Framework**: Spring Boot
  - **Endpoints**: Definidos no ClientController para operações de CRUD (Create, Read, Update, Delete).

- **Banco de Dados**:
  - **Tipo**: Relacional (PostgreSQL)
  - **Scripts**: data.sql para dados iniciais.
- **Documentação e API**:
  - **Swagger**: Utilizado para documentar e testar a API.

Fluxo de Operações

- **Criação de Cliente**:
  - **Requisição**: O usuário envia uma requisição POST para /clients/create com os dados do cliente.
  - **Processamento**: O ClientController chama o ClientService para verificar se o cliente já existe e, se não, cria um novo cliente.
  - **Persistência**: O ClientService utiliza o ClientRepository para salvar o novo cliente no banco de dados.
  - **Resposta**: A resposta é enviada com os detalhes do cliente criado.
- **Atualização de Cliente**:
  - **Requisição**: O usuário envia uma requisição PUT para /clients/update/{id} com os dados atualizados do cliente.
  - **Processamento**: O ClientController chama o ClientService para atualizar os dados do cliente existente.
  - **Persistência**: O ClientService utiliza o ClientRepository para salvar as alterações no banco de dados.
  - **Resposta**: A resposta é enviada com os detalhes do cliente atualizado.
- **Deleção de Cliente**:
  - **Requisição**: O usuário envia uma requisição DELETE para /clients/delete/{id}.
  - **Processamento**: O ClientController chama o ClientService para deletar o cliente especificado.
  - **Persistência**: O ClientService utiliza o ClientRepository para remover o cliente do banco de dados.
  - **Resposta**: A resposta é enviada confirmando a deleção.
- **Consulta de Cliente por ID**:
  - **Requisição**: O usuário envia uma requisição GET para /clients/{id}.
  - **Processamento**: O ClientController chama o ClientService para buscar o cliente pelo ID.
  - **Persistência**: O ClientService utiliza o ClientRepository para recuperar o cliente do banco de dados.
  - **Resposta**: A resposta é enviada com os detalhes do cliente encontrado.

- **Consulta de Todos os Clientes**:
  - **Requisição**: O usuário envia uma requisição GET para /clients/all.
  - **Processamento**: O ClientController chama o ClientService para buscar todos os clientes.
  - **Persistência**: O ClientService utiliza o ClientRepository para recuperar todos os clientes do banco de dados.
  - **Resposta**: A resposta é enviada com a lista de todos os clientes.

Contexto de Implantação

- **Servidor de Aplicação**:
  - **Plataforma de Nuvem**: A aplicação pode ser implantada em serviços de nuvem como AWS, Azure ou Google Cloud para escalabilidade e alta disponibilidade.
  - **Docker**: Utilização de Docker para containerização, permitindo fácil implantação e gerenciamento dos serviços.
- **Banco de Dados**:
  - **H2**: Utilizada para simplificar o desenvolvimento e testes.
  - **PostgreSQL**: Hospedado na nuvem ou em um servidor dedicado, acessível pela aplicação web para operações de persistência de dados em ambiente de produção.
- **Documentação e Testes**:
  - **Swagger**: Integrado na aplicação para fornecer uma interface de teste e documentação das APIs.

Padrões de Design Utilizados

- **Arquitetura em Camadas (Layered Architecture)**: Esta arquitetura divide a aplicação em camadas distintas com responsabilidades específicas, facilitando a manutenção e escalabilidade.
  - **Camada de Apresentação**: Lida com a interface de comunicação com o usuário (ClientController).
  - **Camada de Serviço**: Contém a lógica de negócios (ClientService).
  - **Camada de Persistência**: Gerencia o acesso ao banco de dados (ClientRepository).
  - **Camada de Domínio**: Define as entidades do sistema (Client, ClientRequest, ClientDTO).
- **Repository Pattern**: O padrão Repository abstrai a lógica de acesso aos dados, permitindo operações CRUD através de uma interface (ClientRepository).
- **DTO (Data Transfer Object)**: Utilizado para transferir dados entre as camadas de forma eficiente (ClientDTO).
- **Dependency Injection**: Facilitado pelo Spring Framework, que injeta dependências promovendo baixo acoplamento e maior testabilidade.
- **Singleton**: Implícito no gerenciamento dos beans pelo Spring Framework, garantindo que uma única instância dos serviços seja usada.

Diagrama de Arquitetura

Abaixo está um diagrama simplificado da arquitetura do sistema:

LINK PARA OS DIAGRAMAS

Este diagrama ilustra a interação entre os diferentes componentes da aplicação, destacando a comunicação entre o cliente, a aplicação web, e o banco de dados.

Justificativa da Arquitetura

A arquitetura proposta é adequada para aplicações bancárias devido à sua modularidade, escalabilidade e segurança. Utilizando Spring Boot, garantimos uma estrutura robusta e familiar para desenvolvimento ágil. A integração com bancos de dados como H2 para desenvolvimento e PostgreSQL ou MySQL para produção assegura que o banco de dados seja confiável e capaz de lidar com transações complexas. Além disso, a documentação e teste de APIs com Swagger facilitam o desenvolvimento e manutenção contínua do sistema. Por fim, a containerização com Docker permite implantações consistentes e escaláveis, atendendo às demandas crescentes dos usuários.

Com essa arquitetura, a solução proporcionará um sistema de abertura de conta poupança eficiente, seguro e de fácil manutenção, preparado para crescer conforme as necessidades dos usuários e do mercado.

Ferramentas/Framework

Linguagem: Java

Framework: Spring Boot

Versão: Java 17, Spring Boot 2.9.2.

Base de Dados: H2 (utilizada para simplificar o desenvolvimento e testes; para produção, considere uma base de dados robusta como PostgreSQL ou MySQL)

Link para o repositório no GitHub: **Projeto no GitHub**

Essa estrutura apresenta claramente as tecnologias e recursos utilizados no projeto, facilitando o entendimento técnico.

Domínio/Subdomínios

Domínio: Sistema de Abertura de Conta Poupança

Subdomínios:

1. Solicitação de Conta: Processos relacionados à solicitação da abertura de uma conta poupança através do aplicativo ou em agências.
2. Envio e Verificação de Documentação: Módulo que lida com o envio digital e a análise da documentação necessária para a abertura da conta.
3. Análise e Aprovação: Gestão da análise rápida e aprovação da documentação submetida, e envio da confirmação de abertura de conta.
4. Ativação de Conta: Processos que incluem o depósito inicial e a ativação da conta poupança.
5. Envio do Cartão: Gerenciamento do envio do cartão e dados da conta após a confirmação do depósito inicial.
6. Acesso e Desbloqueio de Conta: Procedimentos para o desbloqueio do cartão e acesso à conta via aplicativo ou caixa eletrônico.
7. Gestão de Serviços Bancários:
    - Consulta de Saldo: Funcionalidade para consulta de extrato e saldo via aplicativo.
    - Saque de Valores: Processos relacionados ao saque de dinheiro da conta poupança.
    - Depósitos Adicionais: Módulo para realizar depósitos adicionais na conta.
    - Transferências: Gerenciamento de transferências de dinheiro para outras contas.
8. Notificações e Atualizações: Sistema de notificações em tempo real sobre transações e atualização de saldo.
9. Atualização de Saldo e Juros: Processos que garantem a atualização do saldo e a adição periódica de juros na conta poupança.

Event Storming

Link público para acessar o Event Storming