Data: 12/12/2022

Pedro H. Rosário

Domínio Escolhido: Aplicativo de controle de gastos - Money.

**Introdução ao App:**

O Money é uma aplicação de controle de gastos, onde o usuário cadastraria seus gastos diários, que no app são chamados de transferências. Estás transferência possuem campos de quantia, data da transferência, tipo de gasto, forma de pagamento entre outros.

Além do CRUD de transferências(gastos), existe o cadastro de objetivos que seriam as metas que o usuário decidiria seguir.

Existe a função de poupança que o usuário pode criar similar a um objetivo para que possa ter o controle de metas. Porém a poupança existe uma relação entre as transferências que o usuário faz com uma poupança cadastrada. Quando o usuário tiver um criado uma poupança com informações como título, descrição, data de início, data final e quantia, ele poderá adicionar transferências com o tipo de gasto “POUPANÇA”. O usuário poderá executar o relatório para pode ver todas as transferências vinculadas a uma poupança.

Diferenças entre os objetivos e a poupança:

* A poupança só pode será criada uma vez com um período de datas (data início – A data do dia da criação da poupança e uma data fim – Qualquer data posterior a data atual).Enquanto o usuário tiver uma poupança ativa ele não poderá criar outra a não ser que delete a poupança atual.
* A poupança possui campos dois campos em sua tabela cujo nomes e funcionalidades são:

quantidade\_meses – Diferença de meses entre a data inicio da poupança e a data final.

quantia\_mes\_esperada – Calculada com base na quantia da poupança e dividida pelo número da quantidade\_meses, campo tem valor calculado em código.

quantidade\_minima\_transferencias- Valor informado pelo usuário, para seu próprio controle do seu progresso de juntar dinheiro

* Os objetivos pode ser criado para múltiplos tipos de gastos, ele não tem vinculo com a tabela de transferências, mas sim com a de usuário.

No APP, você pode cadastrar novos tipos de gastos como por exemplo: COMIDA, ACADEMIA, CONTA DE TELEFONE, etc. Semelhantemente será possível fazer isso para forma de pagamento, podendo o usuário inserir mais formas de pagamento. Entretanto, quando for passar o nome de uma forma de pagamento ou tipo de gasto cadastrado pelo usuário ao criar um objetivo ou uma transferência, o usuário deve digitar exatamente como cadastrou.

A tabela plano\_gasto foi criada para evitar a duplicidade de campo entre as tabelas de objetivo e poupança. Dessa forma, cada objetivo ou poupança criada terá um vinculo único com um plano de gasto.

**Conexão com o BD Shared-Nothing**

Banco escolhido: Mongo DB

Para execução da parte dois do trabalho foi escolhido o banco Mong DB, no qual está rodando pelo servidor online MongoDB Atlas.

**OBS:** O MongoDB Atlas permite usar um Cluster rodando o banco gratuitamente.

**Repositório**

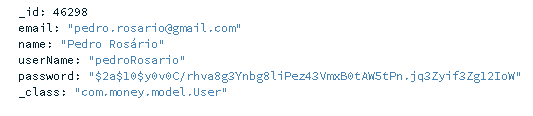
https://github.com/pdrohrosario/money/tree/v2.0.0

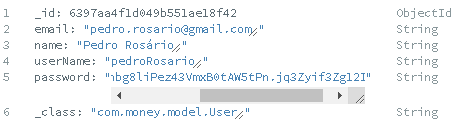
**Script de Criação do Banco de Dados:**

As duas tabelas escolhidas para serem mapeadas para o MongoDB foram **usuário** e **transferência**. Entretanto para que pudesse criar uma transferência tive que mapear a tabela de **tipo de gasto** e **forma de pagamento**, já que elas são tabelas que tem uma relação um para um com a transferência.

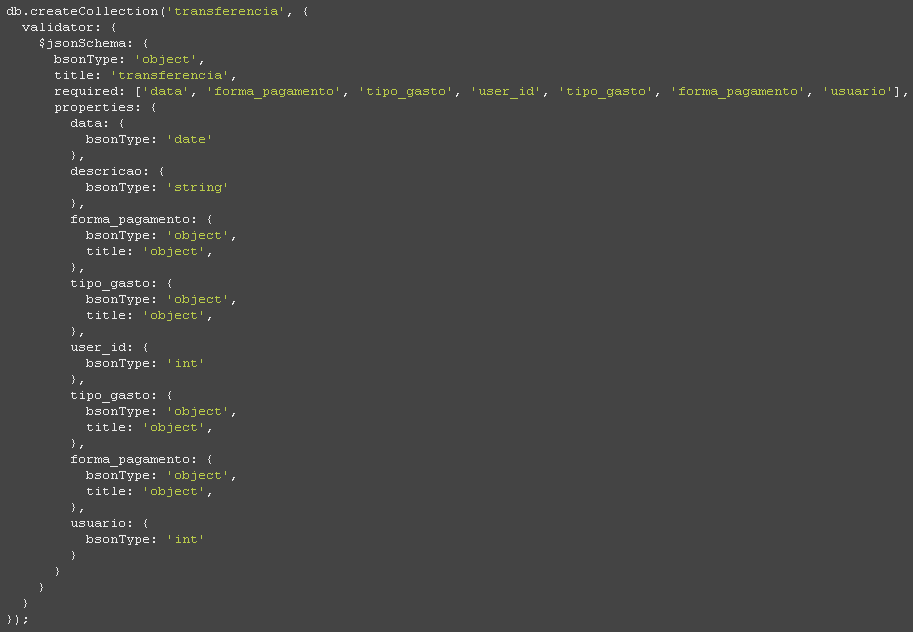
Collection Usuário:

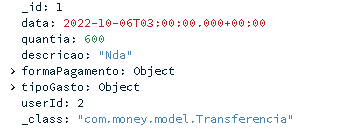


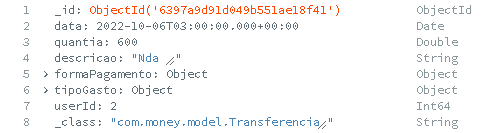




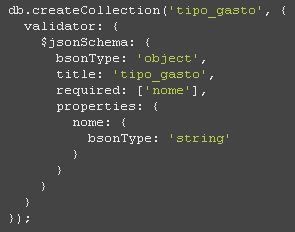
Collection Transferência:







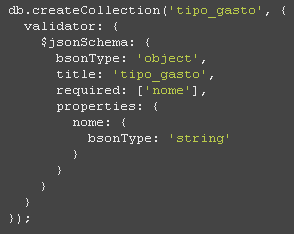
Collection Tipo Gasto:







Collection Forma Pagamento:







**Instruções para Rodar o Programa:**

O projeto foi estruturado em duas partes, BackEnd em Java com Spring Boot e FrontEnd com TypeScript e Angular.

Pois isso será necessário rodar ao mesmo tempo o Frontend e o Backend.

Baixe o projeto no Github.

Dentro da pasta principal teram as seguintes pastas:



**Backend:**

1. Primeiro abra a pasta backend em uma IDE compativel com o Java.
2. Após abrir a pasta backend na sua IDE vá no arquivo application.propreties que esta dentro da pasta resources e cole o texto abaixo para a aplicação possa connectar com o Banco.

spring.data.mongodb.uri = mongodb+srv://admin:admin@cluster0.dgs1wil.mongodb.net/money?retryWrites=true&w=majority  
spring.data.mongodb.database=Money

1. Após isso, pode iniciar a aplicação.

OBS: Caso não inicie por erro, tente atualizar as dependências do projeto usando o Maven

**Frontend:**

1. Abra a pasta **money** que está dentro da pasta **frontend** em um editor de código compatível com TypeScript, Html e CSS.



1. Execute os seguintes comando abaixo para pode rodar o frontend:

**npm install -g @angular/cli**

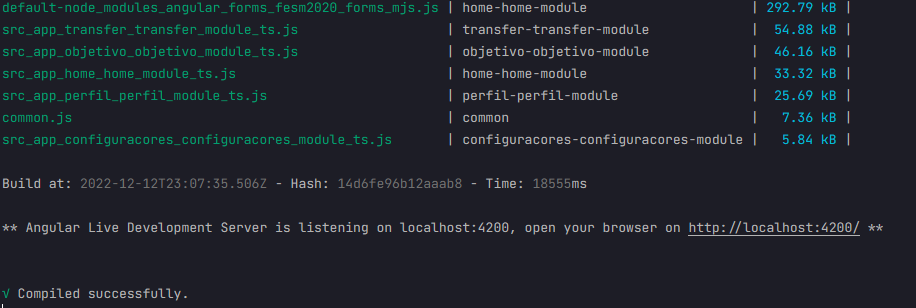
**npm install**

OBS: Para executar os comandos acima é necessário ter o node instalado em sua máquina, segue link de instalação caso necessário.

<https://balta.io/blog/node-npm-instalacao-configuracao-e-primeiros-passos>

1. Após executar os procedimentos acima, abra o terminal dentro da pasta money e execute o seguite comando abaixo e espere aparecer a seguinte tela:

**ng serve**

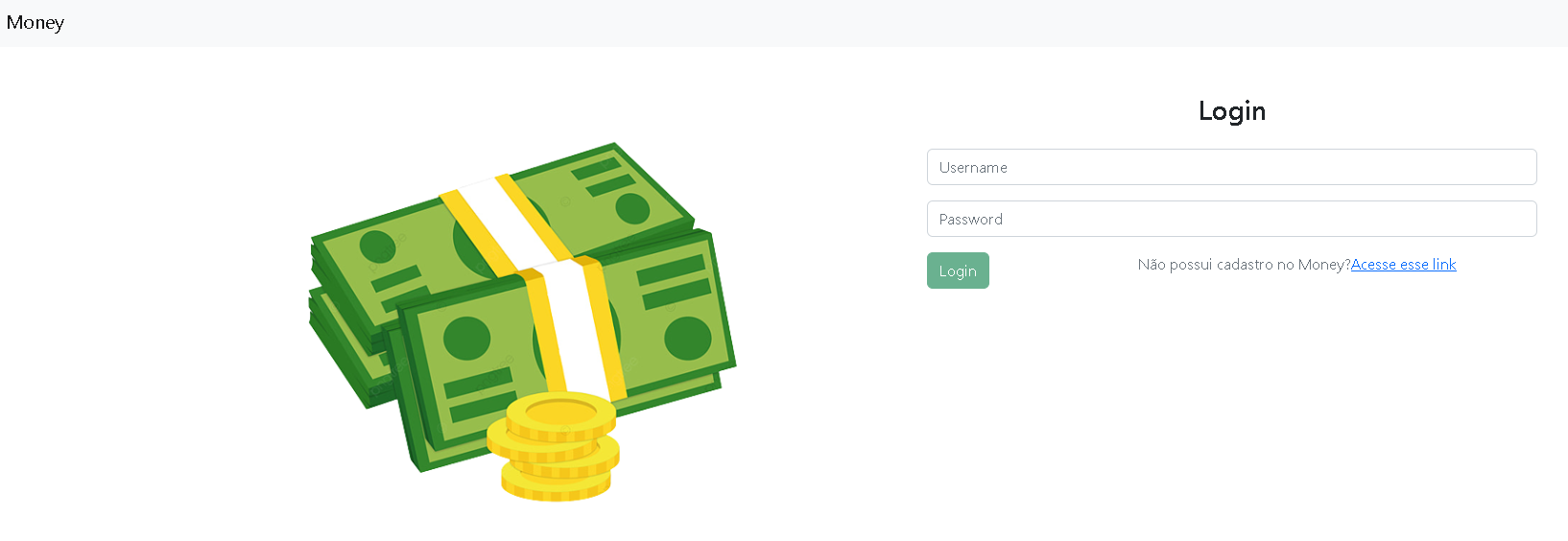
****

1. Abra um navegador de sua preferência e acesse a URL abaixo:

**http://localhost:4200/home**

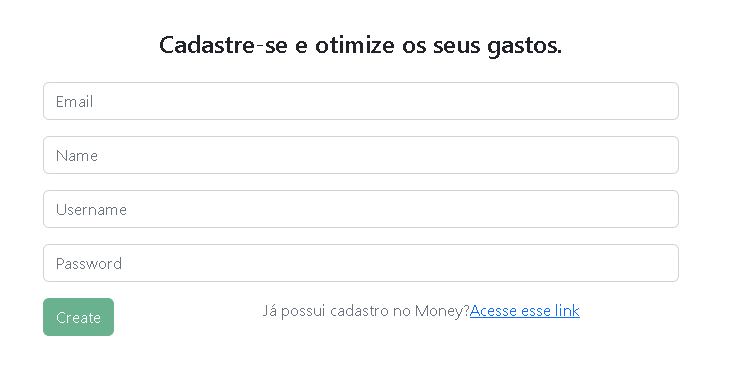
**Instruções para Usar a Aplicação:**

Tela inicial:



1. Crie um usuário:

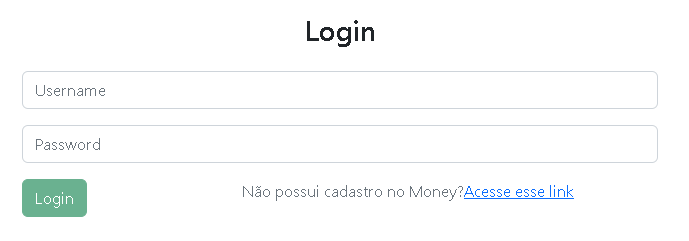
Acesse o link da tela inicial e você será redirecionado ao cadastrado abaixo.



Quando terminar o cadastro clique em criar e você será redirecionado a tela inicial.

1. Login

Digite suas informações de acesso de usuário para entrar na aplicação.



1. Tela de Transferências

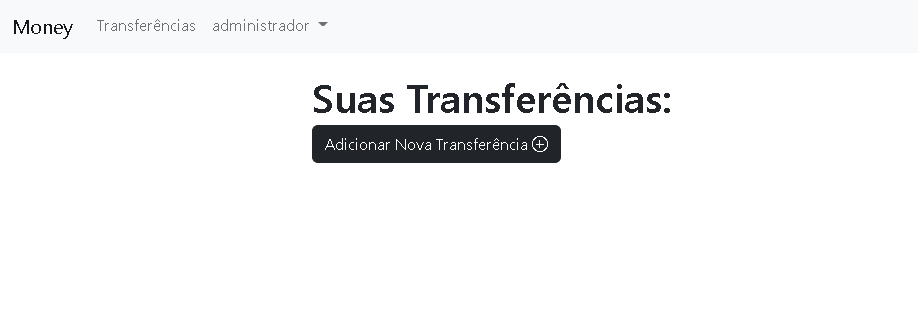
Após o usuário fazer login ele será redirecionado a tela de transferências, onde ele pode efetuar as seguintes operações sobre os gastos dele:

- cadastrar

- listar

- atualizar

- deletar



Para cadastrar uma transferência clique no botão “Adicionar uma nova transferência” e preencha as informações do formulário.

**OBS:** Os formulários da aplicação possuem validações, fica atento a isso, pois os botões para concluir cadastro só serão habilitados se não houver warning como este:



Após cadastrar uma transferência o usuário pode executar as demais operações:

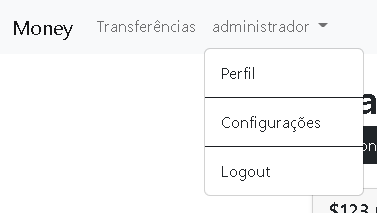


- Consultar(Lupa)

- Deletar(Lixeira)

- Editar(Lápis)

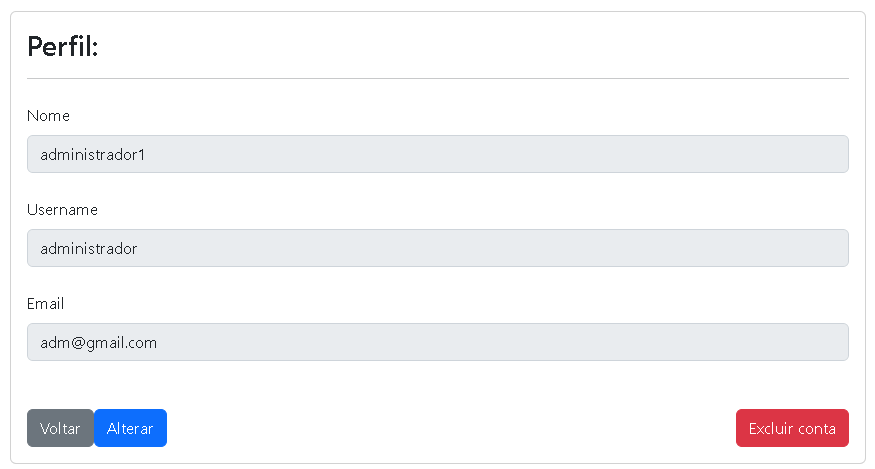
1. Barra de Navegação



Na barra de navegação aparace o userName do usuário e todos as funcionalidades para ele acessar.

**OBS**: A opção Logout serve para o usuário deslogar da aplicação.

1. Perfil

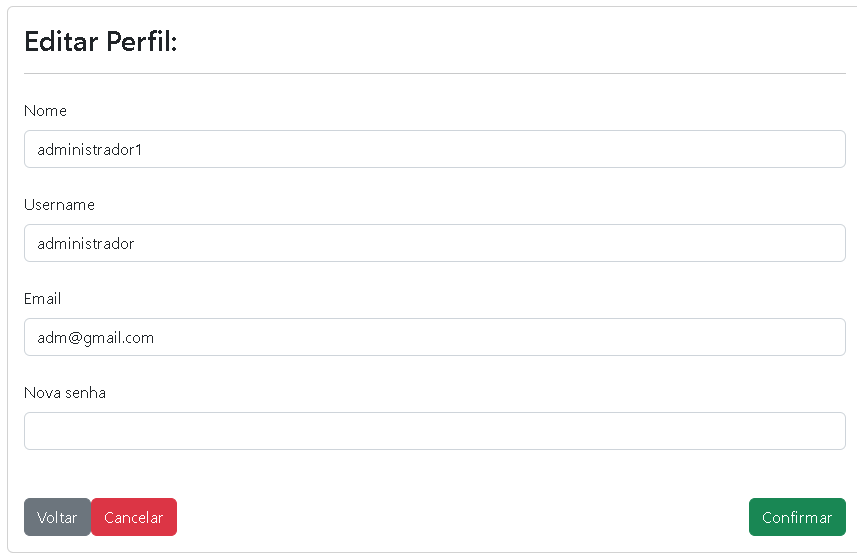


No perfil aparecem as informações cadastradas do usuário e também as funcionalidade do CRUD de usuário.

O usuário pode clicar no botão de alterar para modificar seu nome, userName, email ou a sua senha.



Se o usuário quiser também alterar a sua senha durante o processo basta clicar no botão Alterar Senha.



O usuário também pode excluir a sua conta, caso ele escolha isso, todas as suas transferências serão apagadas também.

