Data: 06/10/2022 Aluno: Pedro H. Rosário

Domínio Escolhido: Aplicativo de controle de gastos - Money.

# 1 a)

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.

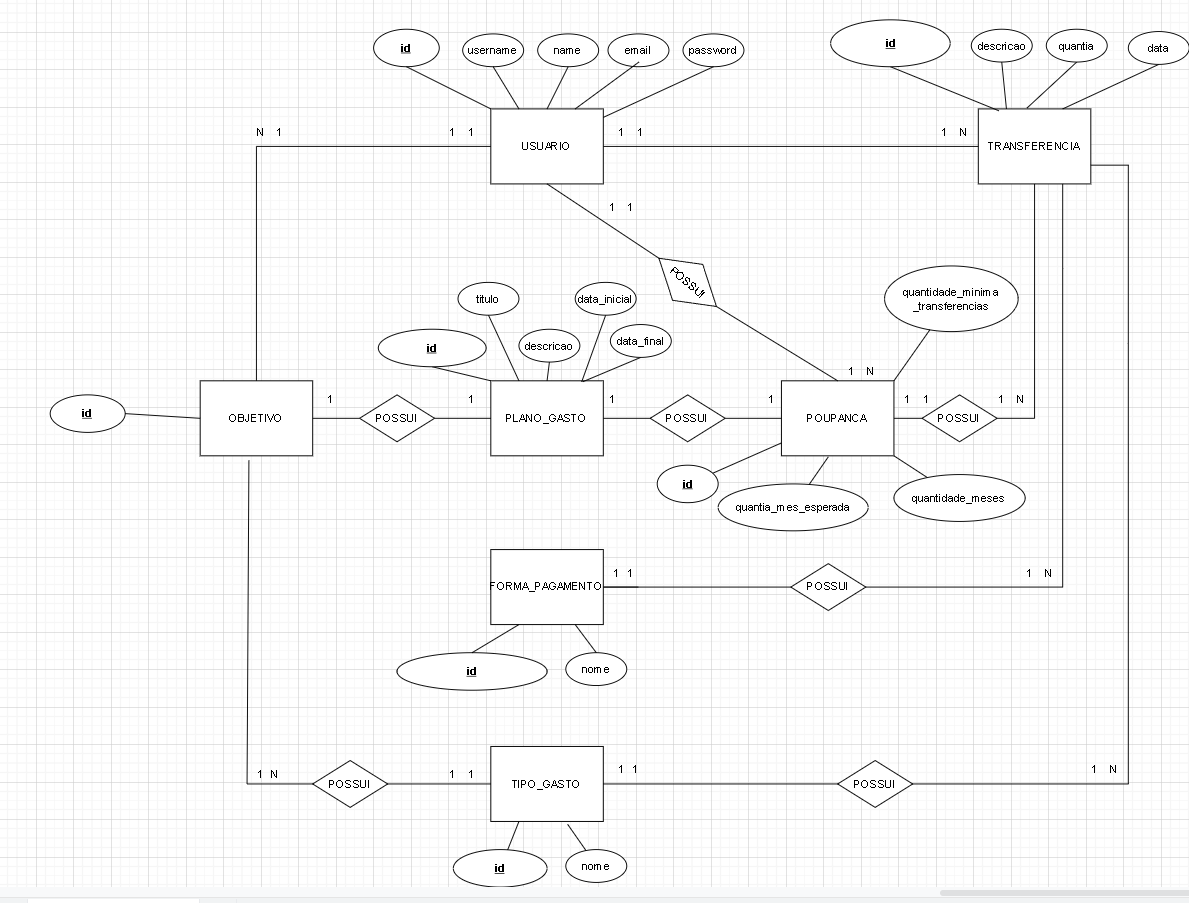
No APP, você pode cadastrar novos tipos de gastos como por exemplo: COMIDA, ACADEMIA, CONTA DE TELEFONE, etc. Semelhantemente será possível fazer isso para forma de pagamento, podendo o usuário inserir mais formas de pagamento. Entretanto, quando for passar o nome de uma forma de pagamento ou tipo de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

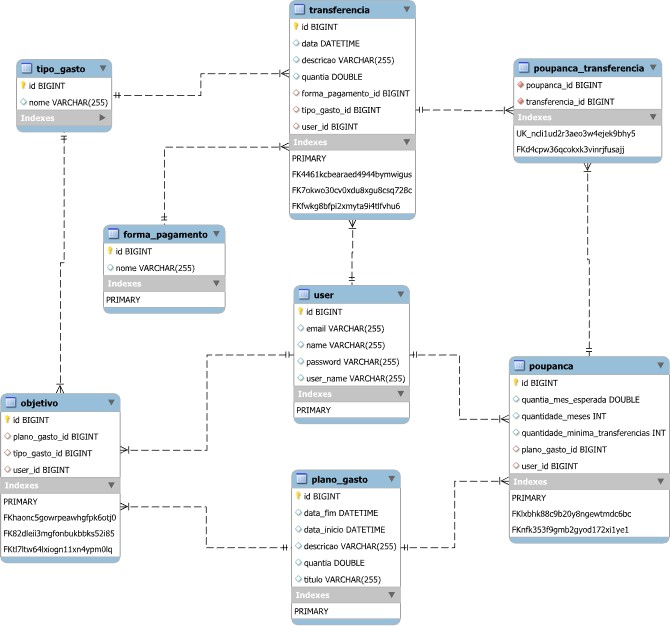
A tabela plano\_gasto foi criada para evitar a duplicidade de campo entre as tabelas de objetivo e poupança. Dessa forma, cada objetivo ou poupança criada terá um vinculo único com um plano de gasto.

Como requisito do trabalho era necessário desenvolver consultas que interagissem com as diferentes tabelas. Segue abaixo a descrição de cada um dos relatórios desenvolvido

* Listar Transferências Vinculadas a Poupanças: Com essa consulta o usuário poderá visualizar as transferências vinculadas a sua poupança ativa (Só pode haver uma poupança ativa por usuário).
* Listar Objetivos de Datas inferiores a Data Atual: Com essa consulta o usuário poderá ver seus objetivos passados, esses traram informações como data de inicio data, data fim, tipo de gasto e uma soma de todas as transferências efetuadas para aquele tipo de gasto dentro do período de datas do objetivo.
* Listar Status dos Objetivos Correntes: Essa funcionalidade segue a mesma ideia da listagem anterior, porém demostrando seus objetivos correntes e com a cotação atual dos seus gastos.

# 1 b)





**1 c)**

**DICIONÁRIO DE DADOS:**

Forma de Pagamento:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Domínio | Tamanho | RI | Descrição |
| id | Inteiro |  | Chave Primária | Código do tipo de gasto |
| nome | Texto | 255 |  | Nome do tipo de gasto |

Tipo de Gasto:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Domínio | Tamanho | RI | Descrição |
| id | Inteiro |  | Chave Primária | Código do tipo de gasto |
| nome | Texto | 255 |  | Nome do tipo de gasto |

Objetivo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Domínio | Tamanho | RI | Descrição |
| id | Inteiro |  | Chave Primária | Código do objetivo |
| plano\_gasto\_id | Inteiro |  | Chave Estrangeira | Código do plano de gasto vinculado ao objetivo. |
| tipo\_gasto\_id | Inteiro |  | Chave Estrangeira | Código do tipo de gasto vinculado ao objetivo. |
| user\_id | Inteiro |  | Chave Estrangeira | Código do usuário vinculo ao objetivo |

Usuário:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Domínio | Tamanho | RI | Descrição |
| id | Inteiro |  | Chave Primária | Código do usuário |
| email | Texto | 255 |  | Email do usuário |
| nome | Texto | 255 |  | Nome completo do usuário |
| password | Texto | 255 |  | Senha do usuário |
| user\_nmae | Texto | 255 |  | Username(apelido) do usuário |

Transferência:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Domínio | Tamanho | RI | Descrição |
| id | Inteiro |  | Chave Primária | Código da transferência |
| data | Data |  |  | Data da transferência |
| descricao | Texto | 255 |  | Descricao de transferência |
| quatia | Decimal |  |  | Quatia transferência |
| forma\_pagamento\_id | Inteiro |  | Chave Estrangeira | Código da forma de pagamento vinculada |
| tipo\_gasto\_id | Inteiro |  | Chave Estrangeira | Código do tipo de gasto vinculado |
| user\_id | Inteiro |  | Chave Estrangeira | Código do usuário vinculado |

Poupança:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Domínio | Tamanho | RI | Descrição |
| id | Initeiro |  | Chave Primárioa | Código da poupança |
| quantia\_mes\_esperada | Decimal |  |  | Quantidade de dinheiro esperado para o mês |
| quantidade\_meses | Inteiro |  |  | Quantidade de meses de desposito na poupança |
| quantidade\_minima\_transferencias | Inteiro |  |  | Quantidade mínima de transferências que seram feitas pelo usuário |
| plano\_gasto\_id | Inteiro |  | Chave Estrangeira | Código do plano de gasto vinculado |
| user\_id | Inteiro |  | Chave Estrangeira | Código do usuário vinculado |

# 2 a)

Link Repositório GitHub: https://github.com/pdrohrosario/money/tree/v1.0.0.

# 2 b)

**OBS:** Quando estava montando o trabalho não consegui corrigir um erro do Spring Boot JPA de gerar algumas tabelas de relacionamento. Essas tabelas pelo que eu pesquisei, aparecem por que eu adicionei o relacionamento das tabelas pelas classes. Porém essas tabelas extras não são utilizadas para nenhuma operação. Se aparecerem no banco de dados, elas estarão vazias pois eu só utilizo para o app as tabelas abaixo.

# BANCO DE DADOS NORMALIZADO:

-USUARIO (#usuario\_id, nome, email, username, senha)

-TRANSFERENCIA (#transferencia\_id, descricao, data, &forma\_pagamento\_id, &tipo\_gasto\_id)

-POUPANCA (#poupanca\_id, quantia\_mes\_esperada, quantidade\_meses, quantidade\_minima\_transferencias, &plano\_gasto\_id, &usuario\_id)

-TIPO\_GASTO (#tipo\_gasto\_id, nome)

-FORMA\_PAGAMENTO (#forma\_pagamento\_id, nome)

-POUPANCA\_TRANFERENCIA (#& poupanca\_id, #& transferencia\_id)

-OBJETIVO (#objetivo\_id, &tipo\_gasto\_id,&usuario\_id)

-PLANO\_GASTO(#plano\_gasto\_id, data\_fim, data\_inicio, descricao, quantia, titulo)

Arquivo dump disponibilizado no zip que está no moodle.

# 2 c)

O aplicativo foi desenvolvido na linguagem Java e utilizando o Spring Boot Framework e para banco de dados foi usado Mysql. Para visualizar o resultado das consultas e demais funções do CRUD de cada tabela será necessário usar o APP Postman. O Postman simula chamadas de uma aplicação web que rodaria no navegador.

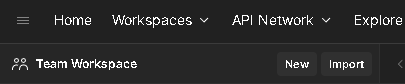
# 1° OBS: Caso no Postman não apresente um retorno no console e apareça erro 400 Bad Request(exceto situações de deleção de algum item) você deve conferir novamente as informações preenchidas.

Para rodar ele basta clonar o repositório do GitHub, importar o projeto em uma IDE e ao iniciar aplicação, executar as chamadas pelo Postman.

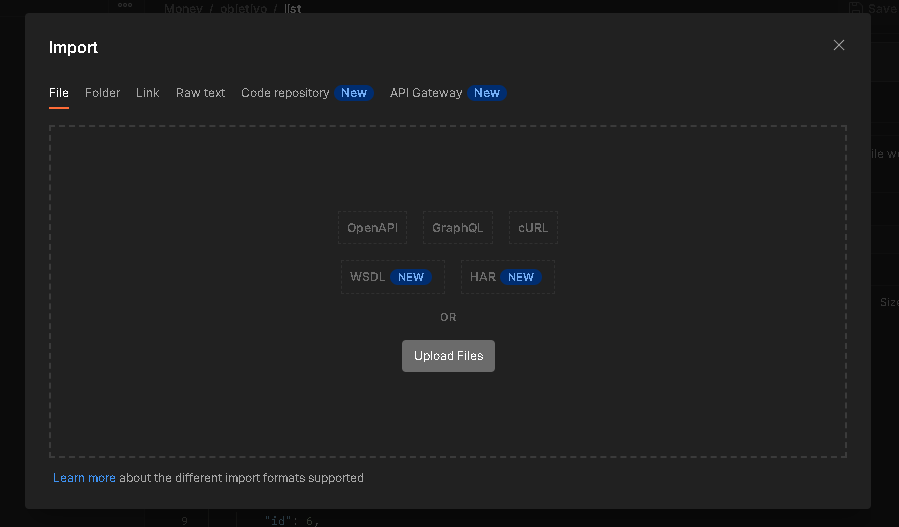
Será disponibilizado uma coleção de chamadas para que possa ser usado pelo professor.

Para usar o Postman basta baixar no site oficial: <https://www.postman.com/>, criar uma conta e baixar o software.

Após isso, ao abrir o postman importar o arquivo. json no menu demonstrado na imagem abaixo. 1

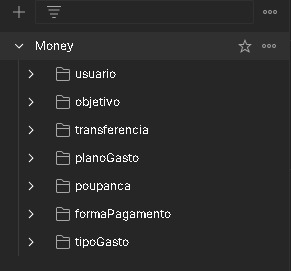


2



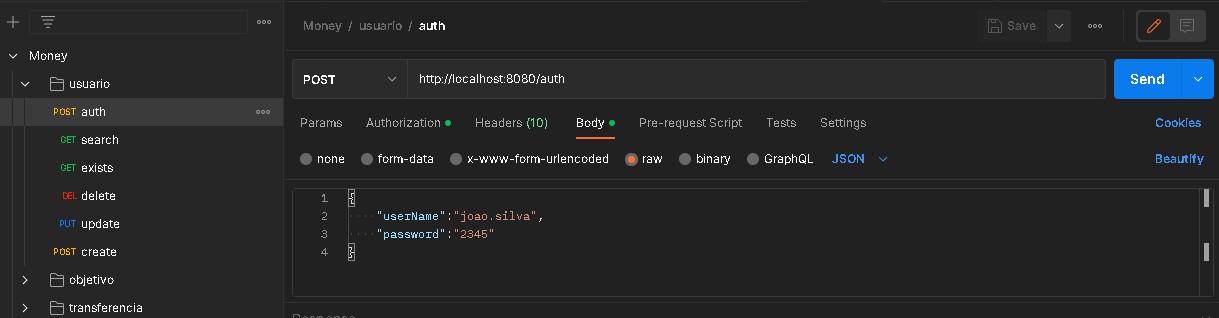
3

Aparecerá a coleção importada:

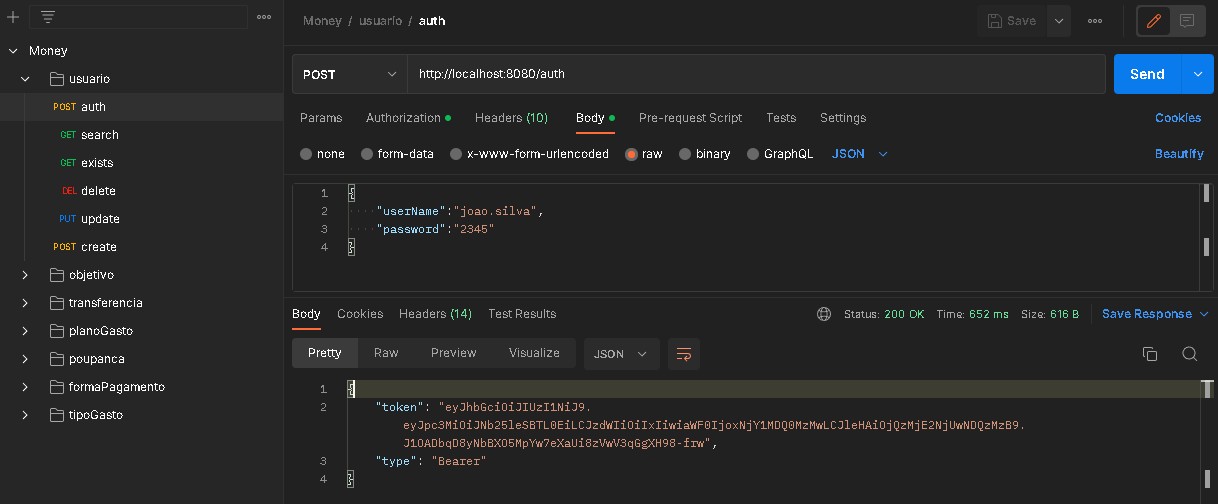


4

Clique na pasta usuario, e com e aplicação rodando execute o POST auth clicando em Send(Botão Azul) para gerar o token de autorização e depois copie ele.

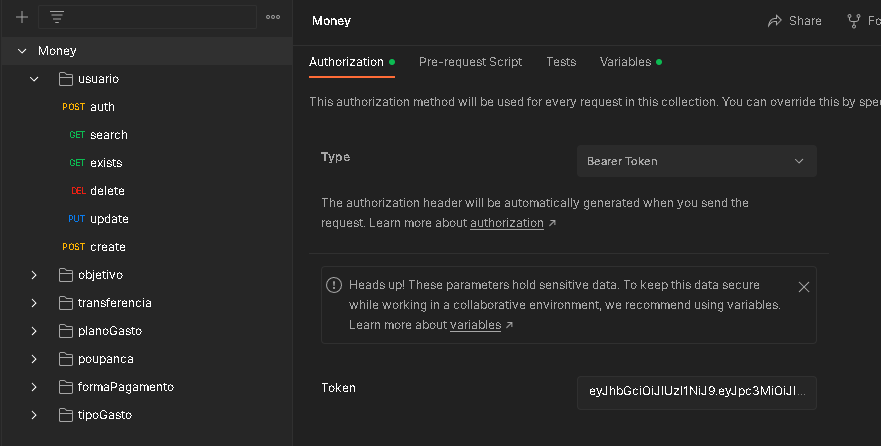


Resultado



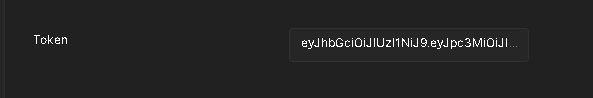
5

Clique no topo da coleção onde está escrito Money, você verá está tela:



6

Coloque o token que você copiou nesse campo



7

Agora basta escolher qualquer outra pasta para executar uma ação.