



Black MAN {Coding}

If black, then code.



Equipe

- Diego Moura dos Santos
- Luis Henrique Calixto de Souza
- Rodrigo Nunes da Silva
- Rosemberg Vieira Araújo Filho



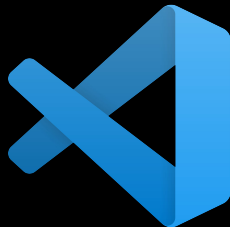
Desafio Final - API HelloBank

- Criar uma API para gerenciar as transações utilizando java, banco de dados e AWS.
- O sistema deve permitir o cadastro de novos clientes, incluindo dados pessoais e dados para contato.
- O cliente deve ser atrelado a uma conta bancária e registrar histórico de transações entre as contas.

Como encaramos esse desafio ?



Tecnologias e ferramentas utilizadas



Metodologia Ágil

Trello (Método Kanban)

BackLog

- Tarefas
- 1ª Criação do Repositório.
- 2ª Back-end e Banco de Dados
- 3ª Documentação
- 4ª Infra
- 5ª Teste automatizados

TO DO

- Fazer
- + Adicionar um cartão

DOING BACK END

- Fazendo
- Apresentação em slide
- 20 de set de 2021 - 21 de set de 2021
- + Adicionar um cartão

DONE INFRA

- Feito
- 1 - Fazer a Criação do repositório
- 4 de set - 5 de set
- 2ª Back-end:
- 4 de set - 9 de set
- 4ª Infra
- 11 de set - 19 de set
- 3ª Documentação
- + Adicionar um cartão

Fase de teste

- Teste
- 5ª Teste automatizados & Apresentação
- 19 de set - 20 de set
- + Adicionar um cartão

Concluído

- Trabalho Concluído
- [Tarefa concluída]
- 22 de set de 2021
- + Adicionar um cartão

Footer: GAMA ACADEMY IBM if black, then code

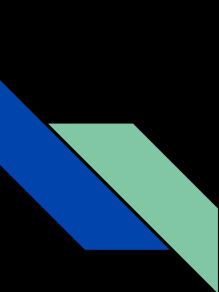
<https://trello.com/b/narX2YOV/banco-fict%C3%ADcio-hellobank-desafio-final-ibm>

Back-end

```
1 package ibm.ibtc.Hello_bank_G6.DTO;
2
3 import lombok.Data;
4
5 @Data
6 public class ClienteCriarDTO {
7     private String nome;
8     private String email;
9     private String senha;
10    private String cpf;
11    private String tel;
12    private String endereco;
13 }
```

- Cadastro de clientes.
- Listagem de clientes.
- Atualização de clientes.
- Deletar clientes.
- Histórico de transações entre contas.

```
1 package ibm.ibtc.Hello_bank_G6.DTO;
2
3 import lombok.Data;
4
5 import java.util.UUID;
6
7 @Data
8 public class TransacaoCriarDTO {
9     private String tipoTransacao;
10    private Double valor;
11    private String clienteId;
12 }
```

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help Hello_bank_G6 - JWSValidatorFilter.java
Hello_bank_G6 · src · main · java · ibm · ibtc · Hello_bank_G6 · Security · JWSValidatorFilter · getAuthenticationToken

Project JWSValidatorFilter.java application.properties

Run: HelloBankG6Application x
Console Actuator
2022-09-21 10:04:10.938 INFO 7100 — [ restartedMain] org.hibernate.dialect.Dialect
Hibernate:

create table cliente_t (
  c_cliente_id uuid not null,
  c_cpf varchar(255) not null,
  c_created_at timestamp not null,
  c_email varchar(255) not null,
  c_endereco varchar(255),
  c_nome varchar(255) not null,
  c_senha varchar(255) not null,
  c_tel varchar(255),
  c_updated_at timestamp,
  primary key (c_cliente_id)
)
Hibernate:

create table conta_corrente_t (
  c_conta_corrente_id uuid not null,
  c_created_at timestamp not null,
  c_limite_negativo float8,
  c_saldo float8 not null,
  c_updated_at timestamp,
  c_cliente_id uuid,
  primary key (c_conta_corrente_id)
)
Hibernate:

create table transacao_t (
  c_transacao_id uuid not null,
  c_created_at timestamp not null,
  c_tipo_transacao varchar(255) not null,
  c_updated_at timestamp,
  c_valor float8 not null,
  c_cliente_id uuid not null,
  primary key (c_transacao_id)
)
```

Hibernate - BD

O Hibernate é um framework ORM (Mapeamento objeto-relacional), ou seja, a implementação física do que será usado para persistir, remover, atualizar ou buscar dados no SGBD (Sistema de gerenciamento de banco de dados).

Imagem do Hibernate gerando o SQL

Repositório no Github

Hello_Bank_G6

Public

★ Starred

Projeto Hello Banck Grupo 6



Java



2



3



Updated 11 hours ago

📁 .mvn/wrapper	Init Projctct	16 days ago
📁 src	Default route	2 days ago
📄 .env	Auth + REST cliente	8 days ago
📄 .gitignore	Default route	2 days ago
📄 Dockerfile	Add production files	2 days ago
📄 README.md	Versão 2	13 hours ago
📄 build.sh	Adicionando scripts	16 days ago
📄 elasticbeanstalk.json	Add production files	2 days ago
📄 git_docker_push.sh	Add production files	2 days ago
📄 mvnw	Init Projctct	16 days ago
📄 mvnw.cmd	Init Projctct	16 days ago
📄 pom.xml	Default route	2 days ago
📄 run.ps1	Commit inicial	15 days ago
📄 run.sh	Default route	2 days ago

https://github.com/Luiz-lab/Hello_Bank_G6

HELLO_BANK_DOC

Introduction

- ▼  Auth (NO TOKEN)
 - POST** Login
 - POST** Criar Cliente
- ▼  Cliente
 - PUT** Update Cliente por ID Copy
 - POST** Get Cliente por ID Copy
 - DEL** Delete Cliente Copy
- ▼  Conta Corrente
 - PUT** Update Conta Corrente
 - POST** Get Conta Corrente por ID
- ▼  Transferência
 - POST** Criar Transferência
 - POST** Get Transferencias ENVIADAS ...
 - POST** Get Transferencias RECEBIDAS...
- ▼  Transação
 - POST** Criar Transação SAQUE
 - POST** Criar Transação DEPOSITO
 - POST** Get Transações por ID do Clie...

Endpoints

A maioria das rotas necessitam de um token de autenticação que pode ser obtido ao fazer login ou criação de cliente.

Rotas

- Autenticação (no token).
- Cliente.
- Conta Corrente .
- Transferência .
- Transação .

<https://documenter.getpostman.com/view/13139925/2s7Z12H4hf>

Documentação



Endpoints

<https://documenter.getpostman.com/view/13139925/2s7Z12H4h>

Documentação PDF

<https://drive.google.com/file/d/1W3iM5lwryciPVExCJUzRzmQ6VZjsikuM/view?usp=sharing>



" Qualquer pessoa de sucesso sabe que é uma peça importante,
mas que não conseguirá nada **sozinha**".

Agradecimentos



Bruna [Yellow Belt]

