DESAFIO MAIDA - BANK SYSTEM

Requisitos que foram feitos com sucesso:

- Adicionar Usuário
- Autenticação com JWT
- Criação de conta de banco
- Transferência entre contas
- Checar saldo da conta

Requisitos que ficaram pendentes:

- Criar testes unitários e suas suítes

-> INFORMAÇÕES IMPORTANTES

- Para iniciar a API, basta rodar a classe Application como um Spring Boot App.
- Foi se utilizado a database persistente JPA para guardar as entidades USUÁRIO, CONTA DE BANCO E USUÁRIOS LOGADOS em um repositório. Logo, tudo ficará salvo enquanto o Spring Boot não tiver sido fechado/reiniciado.
- Todos os testes foram feitos usando as rotas especificadas via postman, a seguir irá ser listado os body em JSON de modo que a requisição seja feita com sucesso para cada requisito (e também para dar trigger em cada exceção que foi criada).

ATENÇÃO - FORAM CRIADAS ROTAS GET DE MODO QUE POSSA SE CHECAR OS REPOSITÓRIOS DE CADA ENTIDADE (USUÁRIO, CONTAS DE BANCO E USUÁRIOS LOGADOS E SEUS TOKENS) SEGUE SUAS ROTAS PARA PUXAR ESSAS INFORMAÇÕES

- CHECAR USUÁRIOS REGISTRADOS - ROTA - GET localhost:8080/users
- CHECAR CONTAS DE BANCO REGISTRADAS - ROTA - GET localhost:8080/accountlist
- CHECAR USUÁRIOS LOGADOS E SEUS TOKENS - ROTA - GET localhost:8080/logged

CRIAÇÃO DE USUÁRIO - ROTA - POST localhost:8080/users

- **ADICIONAR USUÁRIO COM SUCESSO** (para dar trigger numa conta que já foi criada, apenas requisitar o mesmo post com estas mesmas informações)

  { "email" : "default@email.com", "name" : "John Doe", "password" : "12345678" }

- **EXCEÇÃO: CAMPO FALTANDO** (pode retirar qualquer dos 3 campos)

  { "name" : "John Doe", "password" : "12345678" }

- **EXCEÇÃO: EMAIL NO FORMATO ERRADO**

  { "email" : "defaultemail.com", "name" : "John Doe", "password" : "12345678" }

- **EXCEÇÃO: SENHA COM TAMANHO FORA DO RECOMENDADO (8 CARACTERES)**

  { "email" : "default@email.com", "name" : "John Doe", "password" : "123" }

AUTENTICAÇÃO - ROTA - POST
localhost:8080/auth

- **AUTENTICAÇÃO COM SUCESSO** (basta utilizar esta rota com o email e password do usuário que já foi criado na database)

  { "email" : "default@email.com", "password" : "12345678" }

- **EXCEÇÃO: CREDENCIAIS NÃO SENDO CORRESPONDENTES** (basta usar um email ou senha diferente do usuário que foi criado na database, ou retirar um dos campos)

  { "email" : "default@email.com", "password" : "123456789" }

ATENÇÃO! AS PRÓXIMAS REQUISIÇÃO SÃO NECESSÁRIAS QUE O TOKEN RETORNADO NA AUTENTICAÇÃO COM SUCESSO ESTEJA NO HEADER "Authorization", CASO CONTRÁRIO, O ACESSO SERÁ NEGADO E RETORNARÁ UMA EXCEÇÃO/ERRO!

CRIAÇÃO DE CONTA DE BANCO - ROTA - POST localhost:8080/accounts

- **CRIAÇÃO DE CONTA COM SUCESSO** (dependendo do token utilizado, a conta irá ser registrada no token correspondente a conta que foi autenticada com ela)

  { "number" : "123-4", "balance" : 20 }

- **EXCEÇÃO: CAMPO FALTANTE** (se não botar balance a conta é salva com 0 de saldo)

  { "balance" : 20 }

- **EXCEÇÃO: SALDO NEGATIVO** (não permitido)

  { "number" : "123-4", "balance" : -20 }

- **EXCEÇÃO: NÚMERO DA CONTA JÁ EXISTENTE** (para dar trigger numa conta que ja foi criada, apenas requisitar o mesmo POST com estas mesmas informações)

  { "number" : "123-4", "balance" : 20 }

TRANSFERENCIA ENTRE CONTAS - ROTA - POST localhost:8080/accounts/transfer

- **Lembrar de criar mais de uma conta para algum usuário, recomenda-se criar uma conta com número "134-4" de modo a se adequar aos exemplos.**

- **TRANSFERÊNCIA COM SUCESSO** (se a conta de destino não estiver com o token da header atrelado a conta de usuário que a criou, uma exceção irá ser retornada falando que a conta não existe para o usuário em questão)

  { "source_account" : "123-4", "destiny_account" : "134-4", "ammount" : 20 }

- **EXCEÇÃO: CONTA DE ORIGEM NÃO EXISTENTE** (basta botar um número diferente da conta que foi criada)

  { "source_account" : "123-5", "destiny_account" : "134-4", "ammount" : 20 }

- **EXCEÇÃO: CONTA DE DESTINO NÃO EXISTENTE** (basta botar um número diferente da conta que foi criada)

  { "source_account" : "123-4", "destiny_account" : "134-5", "ammount" : 20 }

- **EXCEÇÃO: VALOR A SER TRANSFERIDO NÃO PODE SER NEGATIVO**

  { "source_account" : "123-4", "destiny_account" : "134-4", "ammount" : -20 }

- **EXCEÇÃO: SALDO INSUFICIENTE** (basta esgotar o saldo de alguma conta fazendo algumas transferências, ou registrá-la já com 0 de saldo)

  { "source_account" : "123-4", "destiny_account" : "134-4", "ammount" : 1000000 }

CHECAR SALDO - ROTA - POST
localhost:8080/accounts/balance

- **CHECAGEM COM SUCESSO** (se a conta de checagem não estiver com o token da header atrelado a conta de usuário que a criou, uma exceção irá ser retornada falando que a conta não existe para o usuário em questão)

  { "number" : "123-4" }

- **EXCEÇÃO: CONTA PARA CHECAGEM NÃO EXISTENTE** (basta inserir um número de conta que não se encontra na database)

  { "number" : "123-5" }