**Projeto: App Imesa**

Equipe: Diego Terrengui Pasini – RM: 361228

**1. Introdução**

**Descrição do problema**

Um grupo de restaurantes se uniu para desenvolver um sistema de gestão único e compartilhado, visando reduzir custos com soluções individuais. O sistema permitirá que os restaurantes gerenciem suas operações e que os clientes consultem informações, avaliem e façam pedidos online.

A entrega será feita em fases, garantindo uma implementação gradual, eficaz e com espaço para melhorias contínuas com base no uso e feedback de usuários.

**Objetivo do projeto**

Desenvolver um backend robusto utilizando Spring Boot para gerenciar usuários e

atender aos requisitos definidos.

**2. Arquitetura do Sistema**

**Descrição da Arquitetura**

Este projeto foi desenvolvido utilizando a **Arquitetura Hexagonal**, também conhecida como **Ports and Adapters**, com o objetivo de separar de forma clara a **lógica de negócio** das **interfaces externas**, como banco de dados, web controllers, ou serviços externos.

Essa abordagem promove um design mais limpo e testável, onde:

* A **regra de negócio (casos de uso)** está no centro da aplicação.
* As **interfaces de entrada** (ports in) representam como os dados entram no sistema (por exemplo, controllers ou APIs).
* As **interfaces de saída** (ports out) representam os recursos externos que a aplicação utiliza (como repositórios, gateways, serviços de terceiros).
* Os **adapters** implementam essas interfaces de entrada e saída, conectando o núcleo da aplicação ao mundo externo.

**Diagrama da Arquitetura**

[Insira um diagrama explicando os componentes principais.]

**3. Descrição dos Endpoints da API**

**Tabela de Endpoints**

|  |  |
| --- | --- |
| **Http Method** | POST |
| **URL** | <http://localhost:9000/imesa/v1/users> |
| **Objetivo** | Criar um novo Usuário e Login |
| **Body** | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |

|  |  |
| --- | --- |
| **Http Method** | POST |
| **URL** | <http://localhost:9000/imesa/v1/users/auth/logins> |
| **Objetivo** | Validar Usuário e Senha |
| **Body** | {  "login": "pasini",  "password": "pasini123"  } |

|  |  |
| --- | --- |
| **Http Method** | PUT |
| **URL** | [http://localhost:9000/imesa/v1/users/{userId}/senha](http://localhost:9000/imesa/v1/users/%7buserId%7d/senha) |
| **Objetivo** | Atualizar a senha de um usuário |
| **Body** | {  "senhaAntiga":"pasini123",  "senhaNova":"pasini421",  "confirmacaoSenhaNova":"pasini421"  } |

|  |  |
| --- | --- |
| **Http Method** | PATCH |
| **URL** | [http://localhost:9000/imesa/v1/users/{userId}](http://localhost:9000/imesa/v1/users/%7buserId%7d) |
| **Objetivo** | Atualizar os dados de um usuário |
| **Body** | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "diego\_pasini",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |

|  |  |
| --- | --- |
| **Http Method** | GET |
| **URL** | [http://localhost:9000/imesa/v1/users/{userId}](http://localhost:9000/imesa/v1/users/%7buserId%7d) |
| **Objetivo** | Consultar Dados de um usuário por Id |
| **Body** | No Body |

|  |  |
| --- | --- |
| **Http Method** | GET |
| **URL** | [http://localhost:9000/imesa/v1/users?login={login}](http://localhost:9000/imesa/v1/users?login=%7blogin%7d) |
| **Objetivo** | Consultar Dados de um usuário com base em login |
| **Body** | No Body |

|  |  |
| --- | --- |
| **Http Method** | DELETE |
| **URL** | [http://localhost:9000/imesa/v1/users/{userId}](http://localhost:9000/imesa/v1/users/%7buserId%7d) |
| **Objetivo** | Deletar/Remover um usuário |
| **Body** | No Body |

**Observações: Para melhores exemplos de entrada e saída, após a execução da aplicação é possível acessar a url a partir de seu navegador:**  
<http://localhost:9000/imesa/swagger-ui/index.html#/>

**4. Configuração do Projeto**

**Configuração do Docker Compose**

O arquivo Dockerfile:

|  |
| --- |
| FROM maven:3.9.6-eclipse-temurin-17 AS *build* WORKDIR /app  # Copia somente os arquivos de configuração primeiro para aproveitar o cache do Docker COPY pom.xml . COPY .mvn .mvn COPY mvnw . RUN ./mvnw dependency:go-offline  # Copia o restante do código COPY 'src' './src'  # Compila o projeto RUN ./mvnw clean package -DskipTests  # Segunda etapa: imagem mais leve apenas com o JAR FROM eclipse-temurin:17-jdk WORKDIR /app COPY --from=*build* '/app/target/\*.jar' 'app.jar'  EXPOSE 9000 ENTRYPOINT ["java", "-jar", "app.jar"] |

No arquivo dockerfile, foi utilizado o multipart build, onde primeiro copiamos os arquivos de configuração, posteriormente o restante do código, então foi configurado para realizar o build do projeto via maven poara que pudesse baixar as dependências do projeto. Então em um segundo momento, após o projeto estar compilado, copiarmos a jar gerado para dentro da imagem da aplicação.

O arquivo Docker-compose é este:

|  |
| --- |
| services:  postgres:  image: postgres:16  container\_name: container-postgres-imesa  restart: always  environment:  POSTGRES\_USER: ${POSTGRES\_USER}  POSTGRES\_PASSWORD: ${POSTGRES\_PASSWORD}  POSTGRES\_DB: ${POSTGRES\_DB}  ports:  - "5432:5432" *# volumes: # - postgres\_data:/var/lib/postgresql/data* app:  build:   context: .  dockerfile: Dockerfile  container\_name: container-app-imesa  restart: always  environment:  SPRING\_DATASOURCE\_URL: ${SPRING\_DATASOURCE\_URL}  SPRING\_DATASOURCE\_USERNAME: ${SPRING\_DATASOURCE\_USERNAME}  SPRING\_DATASOURCE\_PASSWORD: ${SPRING\_DATASOURCE\_PASSWORD}  ports:  - "9000:9000"  depends\_on:  - postgres |

**Explicação:**   
No primeiro serviço: **postgres**

**image: postgres:16**

Usa a imagem oficial do PostgreSQL na versão 16 para criar o container do banco de dados.

**container\_name: container-postgres-imesa**

Dá um nome amigável para o container, útil para referência direta.

**restart: always**

Garante que o container reinicie automaticamente se cair ou se o Docker reiniciar.

**environment**

Variáveis de ambiente que utilizamos no banco de dados.

**ports**

Mapeia a porta 5432 do container para a porta 5432 da sua máquina local (host), para que você consiga acessar o banco de fora do container. Utilizando algum gerenciador de banco de dados de sua preferência.

**volumes**

O volume postgres\_data é montado para a pasta onde o PostgreSQL guarda os dados, garantindo que seus dados persistam mesmo se o container for destruído. Em nosso caso deixamos comentado apenas para quando for testar não ficar consumindo espaço da maquina quando o container for destruído.

Já o segundo serviço:

**build**

Indica que o container da aplicação será construído a partir de um Dockerfile localizado no diretório atual (context: .).

**container\_name: container-app-imesa**

Nome amigável do container da sua aplicação.

**restart: always**

Também reinicia automaticamente caso caia.

**environment**

Passa as variáveis de ambiente para configuração do datasource.

**ports**

Mapeia a porta 9000 do container para a porta 9000 do host (sua máquina local), para que você consiga acessar a aplicação via navegador ou cliente HTTP.

**depends\_on**

Garante que o serviço app só será iniciado depois que o serviço postgres estiver iniciado.

**Instruções para execução local**

**1. Clone o repositório**  
  
git clone https://github.com/inisap/pos-fiap-imesa  
  
**2. Suba os containers com Docker Compose**  
  
acesse a pasta do projeto e execute o comando abaixo:  
  
docker-compose up --build

**5. Qualidade do Código**

**Boas Práticas Utilizadas**

Este projeto segue os seguintes princípios do **SOLID**:

**SRP – Single Responsibility Principle (Princípio da Responsabilidade Única)**

Cada classe possui uma única responsabilidade bem definida:

* Os **casos de uso** orquestram o fluxo de regras de negócio.
* Os **mappers** cuidam exclusivamente da transformação entre objetos (DTO ↔️ Domain).
* Os **adapters** fazem apenas a conexão com recursos externos (como banco de dados).

Essa separação favorece **manutenção**, **clareza** e **testabilidade**.

**DIP – Dependency Inversion Principle (Princípio da Inversão de Dependência)**

A camada de negócio **não depende de implementações concretas**, mas sim de **interfaces (ports)**. Isso é possível graças à arquitetura hexagonal:

* A lógica do caso de uso depende apenas de **interfaces definidas no núcleo**.
* As implementações concretas dessas interfaces são injetadas (via Spring, por exemplo), mantendo o núcleo **desacoplado** de tecnologias e frameworks.

Isso permite:

* Substituição fácil de tecnologias (ex: mudar banco de dados ou serviço externo).
* Testes unitários sem dependência externa (mockando as portas de saída).

**6. Collections para Teste**

**Link para a Collection do Postman**

Abaixo os curl das chamadas.

|  |
| --- |
| **[1] Endpoint responsavel por Criar um novo Usuario** |
| curl --request POST \  --url http://localhost:9000/imesa/v1/users \  --header 'Content-Type: application/json' \  --header 'User-Agent: insomnia/11.1.0' \  --data '{  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela"  }  }  ' |

|  |
| --- |
| **[2] Endpoint que valida o Login e senha de um Usuario** |
| curl --request POST \  --url http://localhost:9000/imesa/v1/users/auth/logins \  --header 'Content-Type: application/json' \  --header 'User-Agent: insomnia/11.1.0' \  --data '{  "login": "pasini",  "password": "pasini123"  }' |

|  |
| --- |
| **[3] Endpoint que atualiza a senha de um Usuario** |
| curl --request PUT \  --url http://localhost:9000/imesa/v1/users/1/senha \  --header 'Content-Type: application/json' \  --header 'User-Agent: insomnia/11.1.0' \  --data '{  "senhaAntiga":"pasini421",  "senhaNova":"pasini421",  "confirmacaoSenhaNova":"pasini421"  }' |

|  |
| --- |
| **[4] Endpoint responsavel por atualizar os dados de um Usuario** |
| curl --request PATCH \  --url http://localhost:9000/imesa/v1/users/23 \  --header 'Content-Type: application/json' \  --header 'User-Agent: insomnia/11.1.0' \  --data '{  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "diego\_pasini",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  }' |

|  |
| --- |
| **[5] Endpoint responsavel por Consultar os dados de um Usuarios baseado em seu Id** |
| curl --request GET \  --url http://localhost:9000/imesa/v1/users/1 \  --header 'User-Agent: insomnia/11.1.0' |

|  |
| --- |
| **[6] Endpoint responsavel por Consultar o Id do usuario baseado em seu login** |
| curl --request GET \  --url 'http://localhost:9000/imesa/v1/users?login=pasini' \  --header 'User-Agent: insomnia/11.1.0' |

|  |
| --- |
| **[7] Endpoint responsavel por Deletar um usuario** |
| curl --request DELETE \  --url http://localhost:9000/imesa/v1/users/1 \  --header 'User-Agent: insomnia/11.1.0' |

**Descrição dos Testes Manuais**

|  |  |
| --- | --- |
| **Cenário 1** | **Criação de um Novo Usuário** |
| Chamar o Endpoint | [1] Endpoint responsável por Criar um novo Usuário |
| Descrição | Informar um usuário que ainda não foi criado na aplicação |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| HttpStatus Esperado | 201 Created |

|  |  |
| --- | --- |
| **Cenário 2** | **Tentativa de Criar um Usuario com o mesmo email** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir 2x o mesmo e-mail |
| Payload | {  "nome": "Diego Pasini",  "email": "**diego\_t\_pasini@hotmail.com**",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "Já existe um usuário com este e-mail."  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 3** | **Tentativa de Criar um Usuario com o mesmo login** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir 2x o mesmo login |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini2@hotmail.com",  "login": "**pasini**",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "Já existe um usuário com este login."  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 4** | **Tentativa de Criar um Usuario sem nome** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem nome |
| Payload | {  "email": "diego\_t\_pasini2@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "nome: é obrigatório; nome: não pode ser vazio;"  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 5** | **Tentativa de Criar um Usuario sem email** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem email |
| Payload | {  "nome": "Diego Pasini",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "email: é obrigatório; email: não pode ser vazio;"  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 6** | **Tentativa de Criar um Usuario sem login** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem login |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "login: é obrigatório; login: não pode ser vazio;"  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 7** | **Tentativa de Criar um Usuario sem password** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem password |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "password: é obrigatória; password: não pode ser vazio;"  } |
| HttpStatus Esperado | 400 |
|  |  |
| **Cenário 8** | **Tentativa de Criar um Usuario sem tipoUsuario** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem tipoUsuario |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "tipoUsuario: é obrigatório; tipoUsuario: precisa ser 'RESTAURANTE' ou 'CLIENTE';"  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 9** | **Tentativa de Criar um Usuario sem endereco** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem endereco |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE"  } |
| Retorno esperado | {  "status": 400,  "message": "endereco: é obrigatório;"  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 10** | **Tentativa de Criar um Usuario sem logradouro** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem logradouro |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "endereco.logradouro: é obrigatório; endereco.logradouro: não pode ser vazio;"  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 11** | **Tentativa de Criar um Usuario sem numero** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem numero |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "endereco.numero: não pode ser vazio; endereco.numero: é obrigatório;"  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 12** | **Tentativa de Criar um Usuario sem cep** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem cep |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "endereco.cep: não pode ser vazio; endereco.cep: é obrigatório;"  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 13** | **Tentativa de Criar um Usuario sem cidade** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem cidade |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "status": 400,  "message": "endereco.cidade: não pode ser vazio; endereco.cidade: é obrigatório;"  } |
| HttpStatus Esperado | 400 |

|  |  |
| --- | --- |
| **Cenário 14** | **Tentativa de Criar um Usuario sem estado** |
| Chamar o Endpoint | [1] Endpoint responsavel por Criar um novo Usuario |
| Descrição | Tentar inserir um usuario sem estado |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "password": "pasini123",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela"  }  } |
| Retorno esperado | {  "status": 400,  "message": "endereco.estado: é obrigatório; endereco.estado: não pode ser vazio;"  } |
| HttpStatus Esperado | 400 Bad Request |

|  |  |
| --- | --- |
| **Cenário 15** | **Tentativa de login com usuario e senha correto** |
| Chamar o Endpoint | [2] Endpoint que valida o Login e senha de um Usuario |
| Descrição | Realziar login com usuario e senha valido |
| Payload | {  "login": "pasini",  "password": "pasini123"  } |
| Retorno esperado | {  "nome": "Diego Pasini",  "login": "pasini",  "tipo": "CLIENTE"  } |
| HttpStatus Esperado | 200 OK |

|  |  |
| --- | --- |
| **Cenário 16** | **Tentativa de login com usuario incorreto** |
| Chamar o Endpoint | [2] Endpoint que valida o Login e senha de um Usuario |
| Descrição | Realziar login com usuario incorreto |
| Payload | {  "login": "pasini111",  "password": "pasini123"  } |
| Retorno esperado | {  "status": 401,  "message": "Usuário ou senha inválidos"  } |
| HttpStatus Esperado | 401 Unauthorized |

|  |  |
| --- | --- |
| **Cenário 17** | **Tentativa de login com senha incorreta** |
| Chamar o Endpoint | [2] Endpoint que valida o Login e senha de um Usuario |
| Descrição | Realziar login com senha invalida |
| Payload | {  "login": "pasini",  "password": "asdasd"  } |
| Retorno esperado | {  "status": 401,  "message": "Usuário ou senha inválidos"  } |
| HttpStatus Esperado | 401 Unauthorized |

|  |  |
| --- | --- |
| **Cenário 18** | **Atualizando senha com sucesso** |
| Chamar o Endpoint | [3] Endpoint que atualiza a senha de um Usuario |
| Descrição | Alterar senha com sucesso |
| Payload | {  "senhaAntiga":"pasini123",  "senhaNova":"pasini421",  "confirmacaoSenhaNova":"pasini421"  } |
| Retorno esperado | Senha alterada com sucesso |
| HttpStatus Esperado | 200 OK |

|  |  |
| --- | --- |
| **Cenário 19** | **Tentativa de atualizar a senha com senha antiga invalida** |
| Chamar o Endpoint | [3] Endpoint que atualiza a senha de um Usuario |
| Descrição | Tentar atualizar a senha, porem passando a senha antiga invalida |
| Payload | {  "senhaAntiga":"5756756",  "senhaNova":"pasini421",  "confirmacaoSenhaNova":"pasini421"  } |
| Retorno esperado | {  "status": 400,  "message": "A senha antiga não confere"  } |
| HttpStatus Esperado | 400 Bad Request |

|  |  |
| --- | --- |
| **Cenário 20** | **Tentativa de atualizar a senha com senha com divergencia** |
| Chamar o Endpoint | [3] Endpoint que atualiza a senha de um Usuario |
| Descrição | Tentativa de alterar a senha colocando a senha nova divergente da confirmacaoSenhaNova |
| Payload | {  "senhaAntiga":"pasini421",  "senhaNova":"pasini4213",  "confirmacaoSenhaNova":"pasini421"  } |
| Retorno esperado | {  "status": 400,  "message": "As senha nova não coincide com a confirmação de senha"  } |
| HttpStatus Esperado | 400 Bad Request |

|  |  |
| --- | --- |
| **Cenário 21** | **Tentativa de atualizar a senha passando id de usuario inexistente** |
| Chamar o Endpoint | [3] Endpoint que atualiza a senha de um Usuario |
| Descrição | Tentar inserir um usuario sem login |
| Parametro na url | Passa um id de usuario que não existe na base  http://localhost:9000/imesa/v1/users/**99999**/senha |
| Payload | {  "senhaAntiga":"pasini421",  "senhaNova":"pasini421",  "confirmacaoSenhaNova":"pasini421"  } |
| Retorno esperado | {  "status": 404,  "message": "Id de usuario não encontrado"  } |
| HttpStatus Esperado | 404 Not Found |

|  |  |
| --- | --- |
| **Cenário 22** | **Atualizando dados de um usuario com sucesso** |
| Chamar o Endpoint | [4] Endpoint responsavel por atualizar os dados de um Usuario |
| Descrição | Tentar atualizar dados de um usuario com sucesso |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "diego\_pasini",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "diego\_pasini",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "cidade": "Carapicuiba",  "estado": "São Paulo",  "cep": "06364550",  "complemento": "casa 2",  "bairro": "Jardim Ana Estela"  }  } |
| HttpStatus Esperado | 200 OK |

|  |  |
| --- | --- |
| **Cenário 23** | **Atualizando dados de um usuario passando id de usuario inexistente** |
| Chamar o Endpoint | [4] Endpoint responsavel por atualizar os dados de um Usuario |
| Descrição | Tentar atualizar dados de um usuario com sucesso |
| Parametro na url | Passa um id de usuario que não existe na base  http://localhost:9000/imesa/v1/users/**98765** |
| Payload | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "diego\_pasini",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "complemento": "casa 2",  "cep": "06364550",  "cidade": "Carapicuiba",  "bairro":"Jardim Ana Estela",  "estado": "São Paulo"  }  } |
| Retorno esperado | {  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "diego\_pasini",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "cidade": "Carapicuiba",  "estado": "São Paulo",  "cep": "06364550",  "complemento": "casa 2",  "bairro": "Jardim Ana Estela"  }  } |
| HttpStatus Esperado | 200 OK |

|  |  |
| --- | --- |
| **Cenário 24** | **Consultando dados de um usuario com base em id** |
| Chamar o Endpoint | [5] Endpoint responsavel por Consultar os dados de um Usuarios baseado em seu Id |
| Descrição | Tentar consultar dados de um usuario com base no userId com sucesso |
| Payload | No Body |
| Retorno esperado | {  "id": 1,  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "dataUltimaAlteracao": "2025-05-23T02:34:07.345237",  "tipoUsuario": "CLIENTE",  "endereco": {  "logradouro": "Rua Paraguacu Paulista",  "numero": "993",  "cidade": "Carapicuiba",  "estado": "São Paulo",  "cep": "06364550",  "complemento": "casa 2",  "bairro": "Jardim Ana Estela"  }  } |
| HttpStatus Esperado | 200 OK |

|  |  |
| --- | --- |
| **Cenário 25** | **Consultando dados de um usuario com base em id inexistente** |
| Chamar o Endpoint | [5] Endpoint responsavel por Consultar os dados de um Usuarios baseado em seu Id |
| Descrição | Tentar consultar dados de um usuario com base no userId com id inexistente |
| url parametro | http://localhost:9000/imesa/v1/users/**99876** |
| Payload | No Body |
| Retorno esperado | {  "status": 404,  "message": "Usuário não encontrado"  } |
| HttpStatus Esperado | 404 Not Found |

|  |  |
| --- | --- |
| **Cenário 26** | **Consultando dados de um usuario com base em login** |
| Chamar o Endpoint | [6] Endpoint responsavel por Consultar o Id do usuario baseado em seu login |
| Descrição | Tentar consultar dados de um usuario com base em um login |
| Exemplo url | http://localhost:9000/imesa/v1/users?login=pasini |
| Payload | No Body |
| Retorno esperado | {  "id": 1,  "nome": "Diego Pasini",  "email": "diego\_t\_pasini@hotmail.com",  "login": "pasini",  "tipoUsuario": "CLIENTE"  } |
| HttpStatus Esperado | 200 OK |

|  |  |
| --- | --- |
| **Cenário 27** | **Consultando dados de um usuario com base em login que não existe** |
| Chamar o Endpoint | [6] Endpoint responsavel por Consultar o Id do usuario baseado em seu login |
| Descrição | Tentar consultar dados de um usuario com base em um login que não existe |
| Exemplo url | http://localhost:9000/imesa/v1/users?login=usuario132 |
| Payload | No Body |
| Retorno esperado | {  "status": 404,  "message": "Usuário não encontrado"  } |
| HttpStatus Esperado | 404 Not Found |

|  |  |
| --- | --- |
| **Cenário 28** | **Deletando um usuario com sucesso** |
| Chamar o Endpoint | [7] Endpoint responsavel por Deletar um usuario |
| Descrição | Tentar deletar um usuario com sucesso |
| Exemplo url | http://localhost:9000/imesa/v1/users/1 |
| Payload | No Body |
| Retorno esperado | No Body |
| HttpStatus Esperado | 204 No Content |

|  |  |
| --- | --- |
| **Cenário 29** | **Deletando um usuario com id inexistente** |
| Chamar o Endpoint | [7] Endpoint responsavel por Deletar um usuario |
| Descrição | Tentar deletar um usuario com sucesso |
| Exemplo url | http://localhost:9000/imesa/v1/users/987321 |
| Payload | No Body |
| Retorno esperado | {  "status": 404,  "message": "Usuário não encontrado"  } |
| HttpStatus Esperado | 404 Not Found |

**7. Repositório do Código**

**URL do Repositório**

<https://github.com/inisap/pos-fiap-imesa>