Avant

Projeto SustenAI

IA para Previsão Personalizada de Demanda em E-commerce:
Explorando o Mercado de Produtos Sustentáveis

Integrantes:

Clara Barreto Cerqueira – RM98175
Guilherme Magalhães de Souza – RM551805
Ming Nut Tan – RM99150
Pedro Batista de Araújo – RM550334
Rafaela Rodrigues Luz – RM551857

Sumário

Instrução

1. Importe o projeto com o código fonte do GitHub;
2. Descompacte a pasta e o projeto java e importe para a IDE de sua preferência
3. Rode a classe de aplicação
4. Utilize o Postman ou Insominia para realizar os testes

Proposta

O projeto visa desenvolver um sistema de previsão de vendas para produtos sustentáveis em e-commerce no Brasil, utilizando inteligência artificial e deep analytics. A aplicação será capaz de prever a demanda por produtos sustentáveis específicos e identificar tendências de mercado com base em dados e históricos de vendas.
Diante do crescente interesse por produtos sustentáveis e da necessidade de práticas mais eficientes no comércio, o projeto visa proporcionar uma ferramenta que auxilie na antecipação de demandas e no alinhamento das ofertas com as tendências de consumo. A utilização de inteligência artificial e técnicas avançadas de análise de dados permitirá uma previsão mais precisa, beneficiando tanto os consumidores quanto as empresas que adotam práticas sustentáveis. O sistema conta também com um ranking de vendas onde o usuário poderá ver a quantidade de vendas de cada produto, sabendo assim qual está vendendo mais ou menos.
O sistema será desenvolvido utilizando técnicas de inteligência artificial, como aprendizado de máquina, e deep analytics para analisar dados de vendas e identificar padrões que possam indicar tendências de consumo futuro. O processo incluirá a coleta de dados relevantes, a modelagem preditiva, e a validação dos resultados obtidos.
Espera-se que a aplicação seja capaz de prever com precisão a demanda por produtos sustentáveis específicos no e-commerce brasileiro e identificar as principais tendências de mercado, contribuindo para uma melhor tomada de decisão por parte das empresas que atuam nesse setor.

Funcionalidades

1. Segmentação por Consciência Ambiental:
Em vez de prever vendas de uma forma genérica, o sistema se concentra em prever a demanda para produtos sustentáveis. Isso incluiria itens como cosméticos naturais, alimentos orgânicos, roupas de materiais sustentáveis (que entra em moda sustentável), e outros produtos ecologicamente corretos.
2. Recomendação de Produtos com Base em Perfil Ecológico:
Sistema de recomendação que sugira produtos sustentáveis aos consumidores com base no seu perfil ecológico, histórico de compras e comportamentos de navegação.

Tecnologias – arquitetura de solução

FrontEnd

O Android Studio será utilizado como ferramenta para o desenvolvimento do FrontEnd. Utilização do Kotlin como linguagem para criação de aplicativos móveis para Android e iOS. Armazenamento de Dados: Integração com o Firebase para armazenamento de dados em tempo real, autenticação de usuários e hospedagem de arquivos.

BackEnd

Desenvolvimento BackEnd em Java, utilizando IntelliJ como IDE. Implementação de padrões de projeto de Domain-Driven Design (DDD) para estruturar a aplicação de forma modular.
Implementação de uma API RESTful com templates dinâmicos com Thymeleaf para comunicação entre o FrontEnd e o BackEnd, permitindo o acesso aos dados e operações CRUD (Create, Read, Update, Delete). Integração com banco de dados em nuvem.

Banco de Dados

Utilização do Banco Oracle para armazenamento dos dados. Serão utilizadas consultas SQL e procedures PL/SQL para interação com o banco de dados pelas outras tecnologias.

IA

Utilização de Python como linguagem de programação principal para desenvolvimento de soluções de inteligência artificial. Biblioteca Beautiful Soup para extrair dados de páginas web, analisando o código HTML e XML dos sites. Biblioteca Sklearn para aprendizado de máquina.
Conexão com Banco de Dados: Utilização da biblioteca cx_Oracle/sql para a conexão entre a aplicação e o banco de dados Oracle.

DevOps

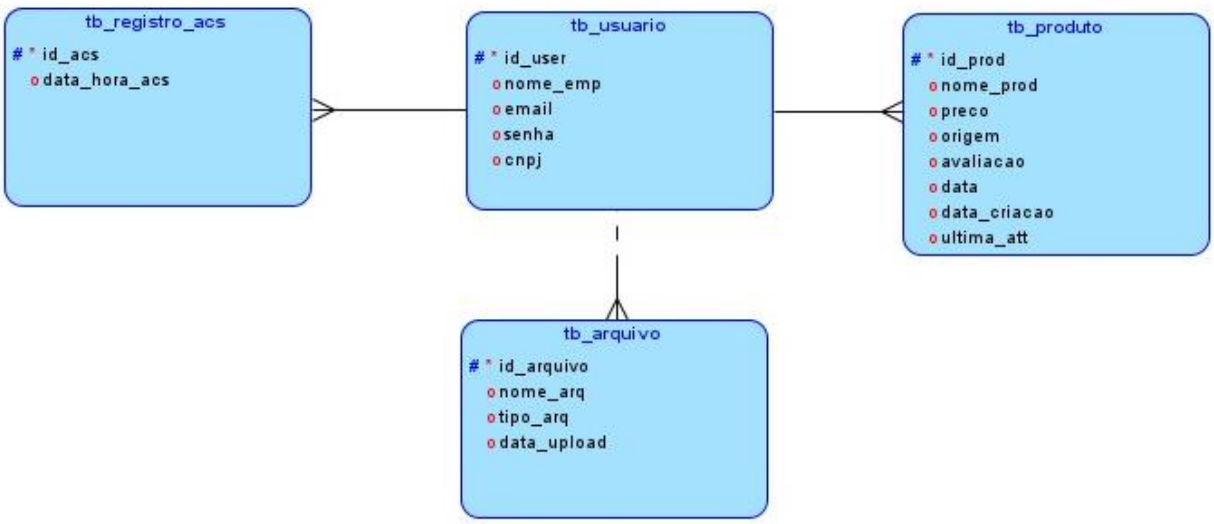
Plataforma em Nuvem: Utilização do Microsoft Azure como ferramenta para hospedagem da aplicação backend em nuvem e serviços relacionados.

Quality Assurance

Construção do Togaf para organização e controle do projeto através do Azure DevOps.

Diagramas

Modelo Lógico-Relacional:



Modelo Entidade-Relacional:

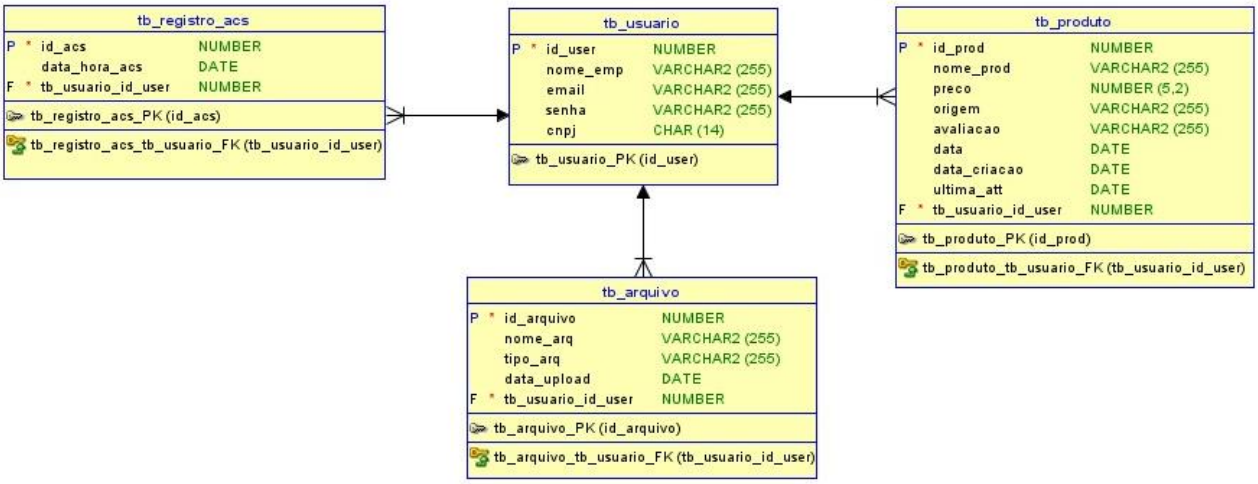


Diagrama de Classe das Entidades (UML):

Requisições

PRODUTO:

Post:

http://localhost:8080/api/produto

```
{
  "nomeProd": "placa solar",
  "descricao": "produto que não agride o meio ambiente",
  "preco": 920.56,
  "categoria": "TECNOLOGIA_VERDE"
}
```

Get All:

http://localhost:8080/api/produto

Get Id:

http://localhost:8080/api/produto/id

Put:

http://localhost:8080/api/produto/id

```
{
  "nomeProd": "placa solar",
  "descricao": "produto que não agride o meio ambiente",
  "preco": 1549.99,
  "categoria": "TECNOLOGIA_VERDE"
}
```

Delete:

http://localhost:8080/api/produto/id

USUARIO:

Post:

http://localhost:8080/api/usuario

```
{
  "nomeEmpresa": "Empresa Ficticia",
  "email": "empresaficticia@gmail.com",
  "senha": "123456",
  "cnpj": "11111111111"
}
```

Get All:

http://localhost:8080/api/usuario

Get Id:

http://localhost:8080/api/usuario/id

Put:

http://localhost:8080/api/usuario/id

```
{
  "nomeEmpresa": "Empresa Ficticia teste update",
  "email": "empresaficticia@gmail.com",
  "senha": "123456",
  "cnpj": "11111111111"
}
```

Delete:

http://localhost:8080/api/usuario/id

ARQUIVO:

Post:

http://localhost:8080/api/arquivos

```
{
  "nomeArquivo": "Relatorio mensal de gastos",
  "tipo": "arquivo de dados",
  "usuario": {"id": 14}
}
```

Get All:

http://localhost:8080/api/arquivos

Get Id:

http://localhost:8080/api/arquivos/id

Put:

http://localhost:8080/api/arquivos/id

```
{
  "nomeArquivo": "Relatorio semanal de gastos",
  "tipo": "arquivo de dados",
  "usuario": {"id": 14}
}
```

Delete:

http://localhost:8080/api/arquivos/id

REGISTROACESSO:

Post:

http://localhost:8080/api/registroAcesso

```
{
  "usuario": {"id": 14}
}
```

Get All:

http://localhost:8080/api/registroAcesso

Get Id:

http://localhost:8080/api/registroAcesso/id

Put:

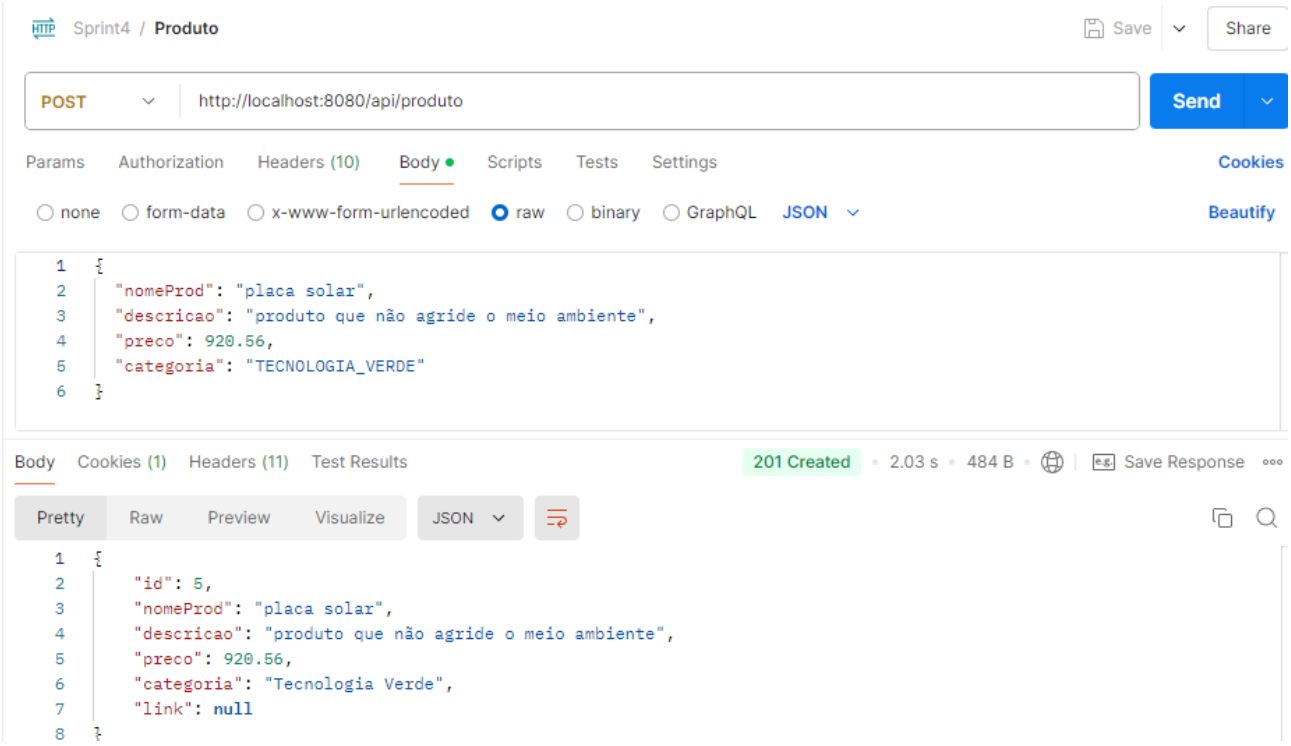
http://localhost:8080/api/registroAcesso/id

```
{
  "usuario": {"id": 10}
}
```

Delete:

http://localhost:8080/api/registroAcesso/id

Algumas evidências dos testes feitos:



Sprint4 / **Produto All**

Save | Share

GET http://localhost:8080/api/produto **Send**

Params Authorization Headers (8) Body Scripts Tests Settings Cookies

Body Cookies (1) Headers (11) Test Results 200 OK 2.04 s 896 B Save Response

Pretty Raw Preview Visualize JSON


```
        "nomeProd": "produto2",
        "descricao": "sustentabilidade",
        "preco": 12.56,
        "categoria": "Beleza e cuidados pessoais",
        "link": {
            "rel": "self",
            "href": "http://localhost:8080/api/produto/2"
        }
    },
    {
        "id": 5,
        "nomeProd": "placa solar",
```


Sprint4 / **Produto Id**

Save | Share

GET http://localhost:8080/api/produto/5 **Send**

Params Authorization Headers (8) Body Scripts Tests Settings Cookies

Body Cookies (1) Headers (11) Test Results 200 OK 758 ms 545 B Save Response

Pretty Raw Preview Visualize JSON


```
{
    "id": 5,
    "nomeProd": "placa solar",
    "descricao": "produto que não agride o meio ambiente",
    "preco": 920.56,
    "categoria": "Tecnologia Verde",
    "link": {
        "rel": "Lista de produtos",
        "href": "http://localhost:8080/api/produto"
    }
}
```


Sprint4 / **Produto**

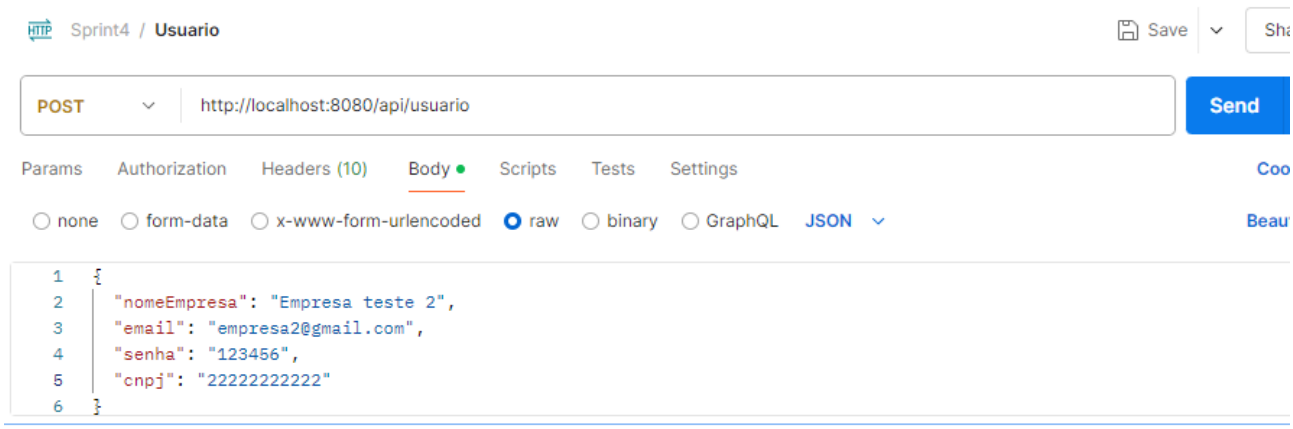
DELETE http://localhost:8080/api/produto/5

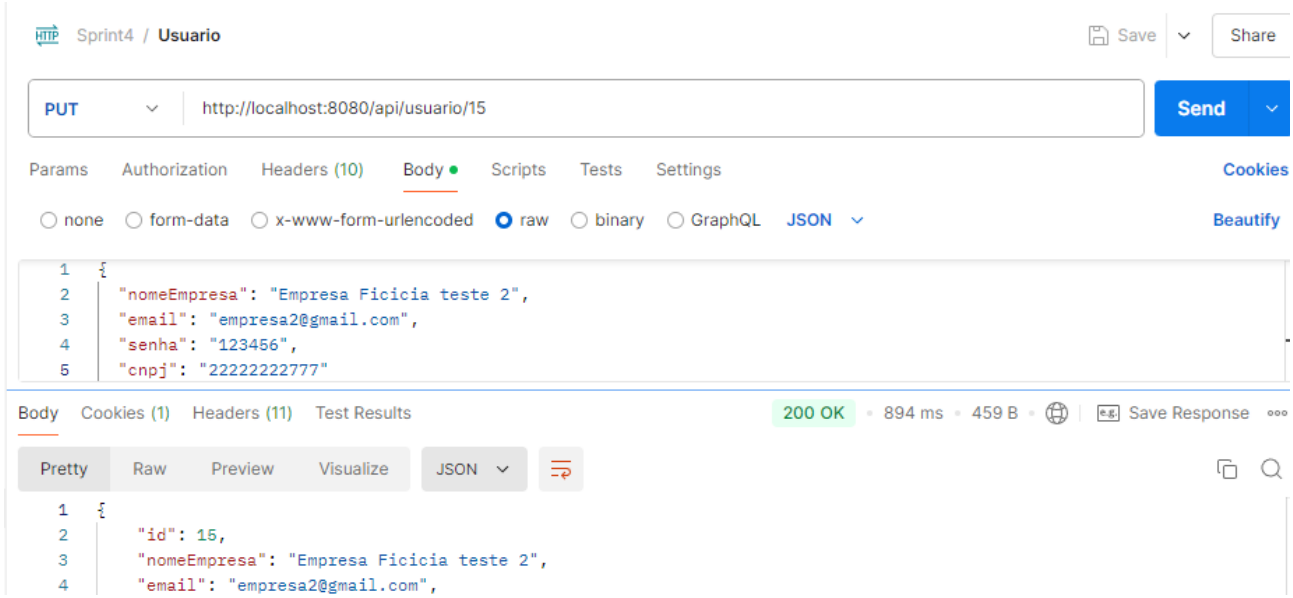
Params Authorization Headers (8) Body Scripts Tests Settings

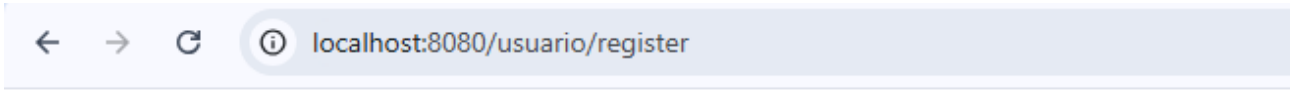
Body Cookies (1) Headers (10) Test Results 200 OK 216 ms 293 B

Pretty Raw Preview Visualize Text

```
1
```


Sprint4 / Usuario
Save
POST
http://localhost:8080/api/usuario
Send
Params
Authorization
Headers (10)
Body
Scripts
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"nomeEmpresa": "Empresa teste 2",
"email": "empresa2@gmail.com",
"senha": "123456",
"cnpj": "22222222222"
}
Body
Cookies (1)
Headers (11)
Test Results
201 Created
924 ms
456 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"id": 15,
"nomeEmpresa": "Empresa teste 2",

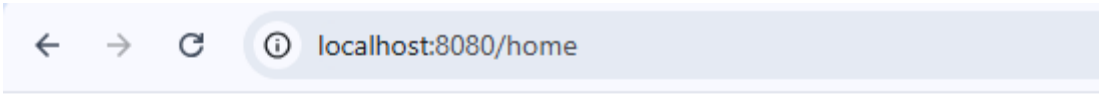

Sprint4 / Usuario
Save
Share
PUT
http://localhost:8080/api/usuario/15
Send
Params
Authorization
Headers (10)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"nomeEmpresa": "Empresa Ficicia teste 2",
"email": "empresa2@gmail.com",
"senha": "123456",
"cnpj": "22222222777"
}
Body
Cookies (1)
Headers (11)
Test Results
200 OK
894 ms
459 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"id": 15,
"nomeEmpresa": "Empresa Ficicia teste 2",
"email": "empresa2@gmail.com",

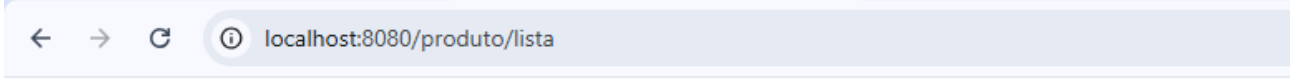


Registro

Nome da Empresa: Teste4
Email: teste4@gmail.com
Senha: ••••••
CNPJ: 44444444444444
Registrar

Já tem uma conta? Faça login

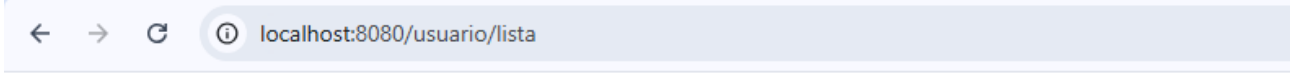




Lista de Produtos

Adicionar novo produto

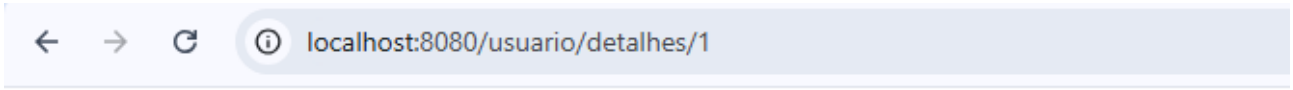
ID	Nome	Descrição	Preço	Categoria	Ações
1	produtoo1	produto teste	10.15	ALIMENTOS_ORGANICOS	Detalhes | Editar | Deletar
2	produto2	sustentabilidade	12.56	BELEZA_E_CUIDADOS_PESSOAIS	Detalhes | Editar | Deletar



Lista de Usuários

Adicionar novo usuário

ID	Nome da Empresa	Email	CNPJ	Ações
1	Natura1	natura@gmail.com	11111111111111	Detalhes | Editar | Deletar
2	Teste	teste@gmail.com	22222222222222	Detalhes | Editar | Deletar
3	Teste2	teste2@gmail.com	33333333333333	Detalhes | Editar | Deletar
4	Teste4	teste4@gmail.com	44444444444444	Detalhes | Editar | Deletar



Detalhes do Usuário

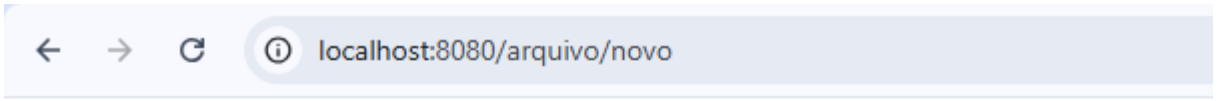
ID: 1

Nome da Empresa: Natura1

Email: natura@gmail.com

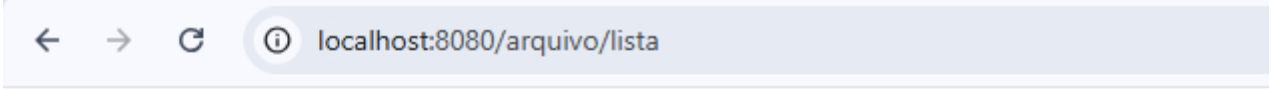
CNPJ: 11111111111111

Editar



Novo Arquivo

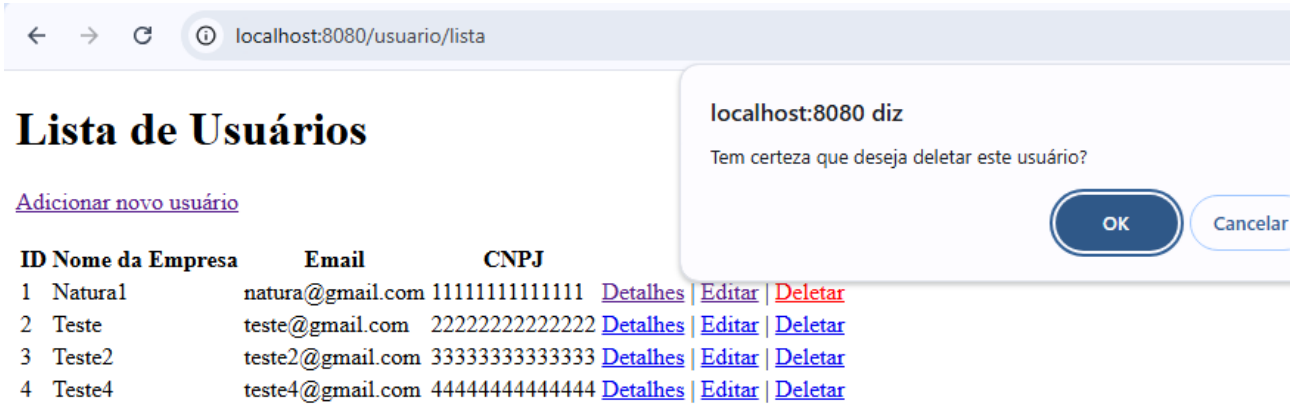
Nome Arquivo: relatorio
Tipo: pdf
Usuário: Teste
Salvar



Lista de Arquivos

Adicionar novo arquivo

ID	Nome Arquivo	Tipo	Ações
1	relatorio	pdf	Detalhes | Editar | Deletar



Lista Endpoints

EndPoints API:

Classe ProdutoController

Post: /api/produto
Get All: /api/produto
Get Id: /api/produto/{id}
Put: /api/produto/{id}
Delete: /api/produto/{id}

Classe ArquivoController

Post: /api/arquivos
Get All: /api/arquivos
Get Id: /api/arquivos/{id}
Put: /api/arquivos/{id}
Delete: /api/arquivos/{id}

Classe RegistroAcessoController

Post: /api/registroAcesso
Get All: /api/registroAcesso
Get Id: /api/registroAcesso/{id}
Put: /api/registroAcesso/{id}
Delete: /api/registroAcesso/{id}

Classe UsuarioController

Post: /api/usuario
Get All: /api/usuario
Get Id: /api/usuario/{id}
Put: /api/usuario/{id}
Delete: /api/usuario/{id}

EndPoints MVC:

HomeController

/home

LoginController

/login

ProdutoViewController

Get All: /produto/lista
Adicionar: /produto/novo
Editar: /produto/editar/{id}
Detalhes: /produto/detalhes/{id}
Post: /produto/salvar
Atualizar: /produto/atualizar/{id}
Deletar: /produto/deletar/{id}

ArquivoViewController

Get All: /arquivo/lista
Adicionar: /arquivo/novo
Editar: /arquivo/editar/{id}
Detalhes: /arquivo/detalhes/{id}
Post: /arquivo/salvar
Atualizar: /arquivo/atualizar/{id}
Deletar: /arquivo/deletar/{id}

RegistroAcessoViewController

Get All: /acesso/listaAcessos
Adicionar: acesso/novoAcesso
Post: /acesso/salvarAcesso

UsuarioViewController

Get All: /usuario/lista
Adicionar: /usuario/register
Editar: /usuario/editar/{id}
Detalhes: /usuario/detalhes/{id}
Post: /usuario/register
Atualizar: /usuario/atualizar/{id}
Deletar: /usuario/deletar/{id}

Grupo

Responsabilidades:
Clara Barreto Cerqueira – Java
Guilherme Magalhães – Quality Assurance
Ming Nut Tan – Inteligência Artificial e DevOps
Pedro Batista – Mobile Development