

# CHALLENGE

## DOMAIN DRIVEN DESIGN



**Gabriel Torres Fernandes – 553635**

**Julia Gonzalez Pugliese Ribeiro – 553427**

**Lucas Ponciani Marinelli - 553112**

**FIAP - 1TDSPC 2024**

# SUMÁRIO

<b>PROPOSTA DE SOLUÇÃO .....</b>	<b>3</b>
PROBLEMA .....	3
SOLUÇÃO .....	3
BENEFÍCIOS ESPERADOS .....	3
 <b>ESCOPO DO PROJETO .....</b>	 <b>4</b>
OBJETIVOS .....	4
 <b>FUNCIONALIDADES PRINCIPAIS .....</b>	 <b>5</b>
 <b>IMPLEMENTAÇÕES JAVA .....</b>	 <b>6</b>
TECNOLOGIAS UTILIZADAS .....	6
EXECUÇÃO DO PROGRAMA .....	6
REQUISITOS.....	6
CONFIGURAÇÃO DO PROJETO .....	6
EXECUÇÃO PELO INTELLIJ IDEA .....	6
TESTANDO A CONEXÃO WEB.....	7
 <b>DIAGRAMA DE CLASSES .....</b>	 <b>8</b>
 <b>DIAGRAMAS SQL .....</b>	 <b>9</b>
DIAGRAMA LÓGICO .....	9
DIAGRAMA RELACIONAL .....	10
 <b>EQUIPE .....</b>	 <b>11</b>

# PROPOSTA DE SOLUÇÃO

## PROBLEMA

Site que necessita atender com mais eficiência o grupo de pessoas com diversos tipos de habilidades, incluindo aquelas com deficiências visuais, auditivas, motoras, cognitivas ou outras limitações.

## SOLUÇÃO

OBJETIVO: criar um mais novo e acessível portal para a Salesforce, onde possa ser implementado em outros diversos dispositivos, juntamente com demais tecnologias assistivas.

## BENEFÍCIOS ESPERADOS

1. Melhorar a experiência de compra dos clientes, proporcionando uma plataforma de comércio eletrônico intuitiva e fácil de usar.
2. Aumentar as vendas e a satisfação do cliente através de um processo de compra simplificado e seguro.
3. Facilitar a gestão de produtos e pedidos para os administradores da empresa.
4. Estabelecer uma presença online forte e competitiva no mercado de venda de softwares.

## **ESCOPO DO PROJETO**

O projeto proposto é uma aplicação Java Maven para uma empresa de venda de softwares online. A aplicação será desenvolvida utilizando tecnologias web para oferecer uma plataforma de comércio eletrônico eficiente e segura.

### **OBJETIVOS**

1. Desenvolver uma plataforma de comércio eletrônico para venda de softwares.
2. Permitir que os clientes naveguem pelo catálogo de produtos e contratem serviços da empresa de forma remota.
3. Integrar um banco de dados para armazenar informações de produtos, clientes, pedidos e transações.
4. Implementar um sistema de gerenciamento de usuários para permitir que os clientes façam login, gerenciem seus perfis e acompanhem o histórico de pedidos.

## **FUNCIONALIDADES PRINCIPAIS**

1. Cadastro de Produtos: Permitir que os administradores cadastrem novos produtos, incluindo informações como nome, descrição, preço e categoria.
2. Navegação e Pesquisa: Os clientes podem navegar pelo catálogo de produtos, filtrar por categoria e realizar pesquisas por palavras-chave.
3. Autenticação de Usuário: Implementar um sistema de autenticação de usuário para permitir que os clientes façam login, gerenciem seus perfis e alterem suas informações pessoais.
4. Administração do Sistema: Painel de administração para gerenciamento de produtos, pedidos, clientes e configurações do sistema.

# IMPLEMENTAÇÕES JAVA

## TECNOLOGIAS UTILIZADAS

- Java: Linguagem de programação principal para o desenvolvimento da aplicação.
- Maven: Gerenciador de dependências e construtor de projetos para simplificar o processo de construção e manutenção do projeto.
- HTML, CSS e JavaScript: Para o desenvolvimento da interface do usuário (UI) e experiência do usuário (UX).
- Banco de Dados Relacional (por exemplo, MySQL, PostgreSQL): Para armazenar informações sobre produtos, clientes, pedidos e transações.

## EXECUÇÃO DO PROGRAMA

### REQUISITOS

Antes de começar, certifique-se de ter os seguintes requisitos instalados em seu sistema:

1. Java Development Kit (JDK)
2. Apache Maven
3. IntelliJ IDEA (opcional)

### CONFIGURAÇÃO DO PROJETO

1. Clone o repositório do projeto para o seu ambiente local:

```
bash git clone https://github.com/juliapugliese/Challenge\_Sprint3\_Java\_ATT.git
```

2. Navegue até o diretório raiz do projeto:

```
bash cd Challenge_Sprint3_Java_ATT
```

3. Verifique se o arquivo pom.xml está presente no diretório raiz. Este arquivo contém as configurações do projeto Maven, incluindo dependências e plugins.

### EXECUÇÃO PELO INTELIJ IDEA

1. Abra o IntelliJ IDEA e selecione "Open" no menu principal.
2. Navegue até o diretório onde o projeto foi clonado e selecione-o.
3. Aguarde o IntelliJ importar e configurar o projeto.
4. Após a importação, localize o arquivo principal do aplicativo Java.
5. Clique com o botão direito no arquivo principal e selecione "Run" para executar o aplicativo.

## **TESTANDO A CONEXÃO WEB**

Após a execução bem-sucedida do aplicativo, abra um navegador da web e acesse a URL fornecida no código do projeto para testar a conexão web.

## TABELA DE END-POINTS

### EXPLICAÇÃO DOS CÓDIGOS DE STATUS

- *200 OK*: A solicitação foi bem-sucedida.
- *201 Created*: A solicitação foi bem-sucedida e um novo recurso foi criado.
- *204 No Content*: A solicitação foi bem-sucedida, mas não há conteúdo para enviar no corpo da resposta.
- *400 Bad Request*: A solicitação é inválida ou malformada.
- *404 Not Found*: O recurso solicitado não foi encontrado.

### EXEMPLOS DE USO

#### 1. *GET /api/cliente*

- *Descrição*: Retorna uma lista de todos os usuários.
- *Exemplo de Resposta*: 200 OK
- **JSON**:

```
[{  
  "id": 2,  
  "email": "rm554585@fiap.com.br",  
  "nomeCompleto": "Henrique J. Buarque",  
  "nomeUsuario": "Henrique",  
  "senha": "henrique123",  
  "cargo": "Estagiario",  
  "cpf": 14451445751,  
  "empresa": {  
    "id": 2,  
    "cnpj": 1454256481475,  
    "nomeEmpresa": "sabesp",  
    "pais": "Brasil",  
    "segmento": "Administrativo",  
    "tamanhoEmpresa": "Grande"  
  },  
  "telefone": "11456699887"
```



```

},
{
  "id": 3,
  "email": "rm545585@fiap.com.br",
  "nomeCompleto": "Fabiana Montero de Melo",
  "nomeUsuario": "Fabiana",
  "senha": "bibi02",
  "cargo": "Estagiario",
  "cpf": 11585457451,
  "empresa": {
    "id": 3,
    "cnpj": 1454242581475,
    "nomeEmpresa": "danone",
    "pais": "Brasil",
    "segmento": "Alimenticio",
    "tamanhoEmpresa": "Grande"
  },
  "telefone": "11893699887"
}}

```

## 2. GET /api/cliente/{id}

- *Descrição:* Retorna os dados de um usuário específico.
- *Exemplo de Resposta:* 200 OK
- **JSON:**

```

{
  "id": 2,
  "email": "rm554585@fiap.com.br",
  "nomeCompleto": "Henrique J. Buarque",
  "nomeUsuario": "Henrique",
  "senha": "henrique123",
  "cargo": "Estagiario",
  "cpf": 14451445751,
  "empresa": {
    "id": 2,

```

```

        "cnpj": 1454256481475,
        "nomeEmpresa": "sabesp",
        "pais": "Brasil",
        "segmento": "Administrativo",
        "tamanhoEmpresa": "Grande"
    },
    "telefone": "11456699887"
}

```

### 3. POST /api/cliente

- *Descrição:* Cria um novo usuário.
- *Exemplo de Requisição:*
- **JSON:**

```

{
    "email": "rm553536@fiap.com.br",
    "nomeCompleto": "Giuliana R. Silva",
    "nomeUsuario": "Giuliana",
    "senha": "giu789",
    "cargo": "Estagiario",
    "cpf": 14451445751,
    "empresa": {
        "cnpj": 1454256481475,
        "nomeEmpresa": "sabesp",
        "pais": "Brasil",
        "segmento": "Administrativo",
        "tamanhoEmpresa": "Grande"
    },
    "telefone": "11456699887"
}

```

- *Exemplo de Resposta:* 201 Created
- **JSON:**

```

{
    "id": 4,
    "email": "rm553536@fiap.com.br",

```

```

"nomeCompleto": "Giuliana R. Silva",
"nomeUsuario": "Giuliana",
"senha": "giu789",
"cargo": "Estagiario",
"cpf": 14451445751,
"empresa": {
  "id": 2,
  "cnpj": 1454256481475,
  "nomeEmpresa": "sabesp",
  "pais": "Brasil",
  "segmento": "Administrativo",
  "tamanhoEmpresa": "Grande"
},
"telefone": "11456699887"
}

```

#### 4. PUT /api/cliente/{id}

- *Descrição:* Atualiza os dados de um usuário específico.
- *Exemplo de Requisição:*
- **JSON:**

```

{
  "email": "rm553536@fiap.com.br",
  "nomeCompleto": "Giuliana R. Silva",
  "nomeUsuario": "Giuliana",
  "senha": "giu789",
  "cargo": "Administrado",
  "cpf": 14451445751,
  "empresa": {
    "cnpj": 1454256481475,
    "nomeEmpresa": "santander",
    "pais": "Brasil",
    "segmento": "comércio",
    "tamanhoEmpresa": "Grande"
  },
}

```

```
"telefone": "11895519887"
}
```

- *Exemplo de Resposta: 200 OK*

- **JSON:**

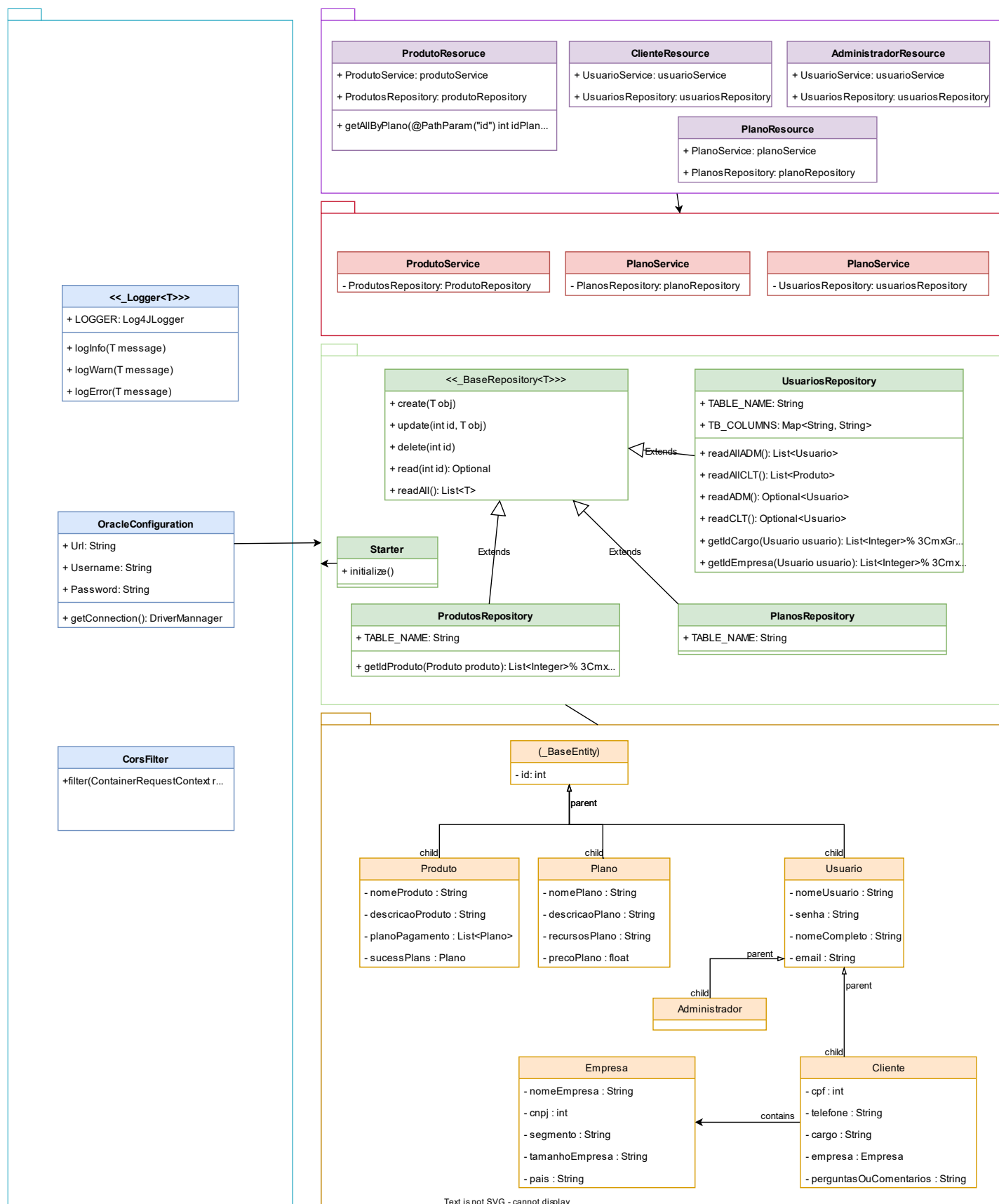
```
{
  "id": 4,
  "email": "rm553536@fiap.com.br",
  "nomeCompleto": "Giuliana R. Silva",
  "nomeUsuario": "Giuliana",
  "senha": "giu789",
  "cargo": "Administrador",
  "cpf": 14451445751,
  "empresa": {
    "id": 4,
    "cnpj": 1454256481475,
    "nomeEmpresa": "santander",
    "pais": "Brasil",
    "segmento": "comércio",
    "tamanhoEmpresa": "Grande"
  },
  "telefone": "11895519887"
}
```

## 5. DELETE /api/cliente/{id}

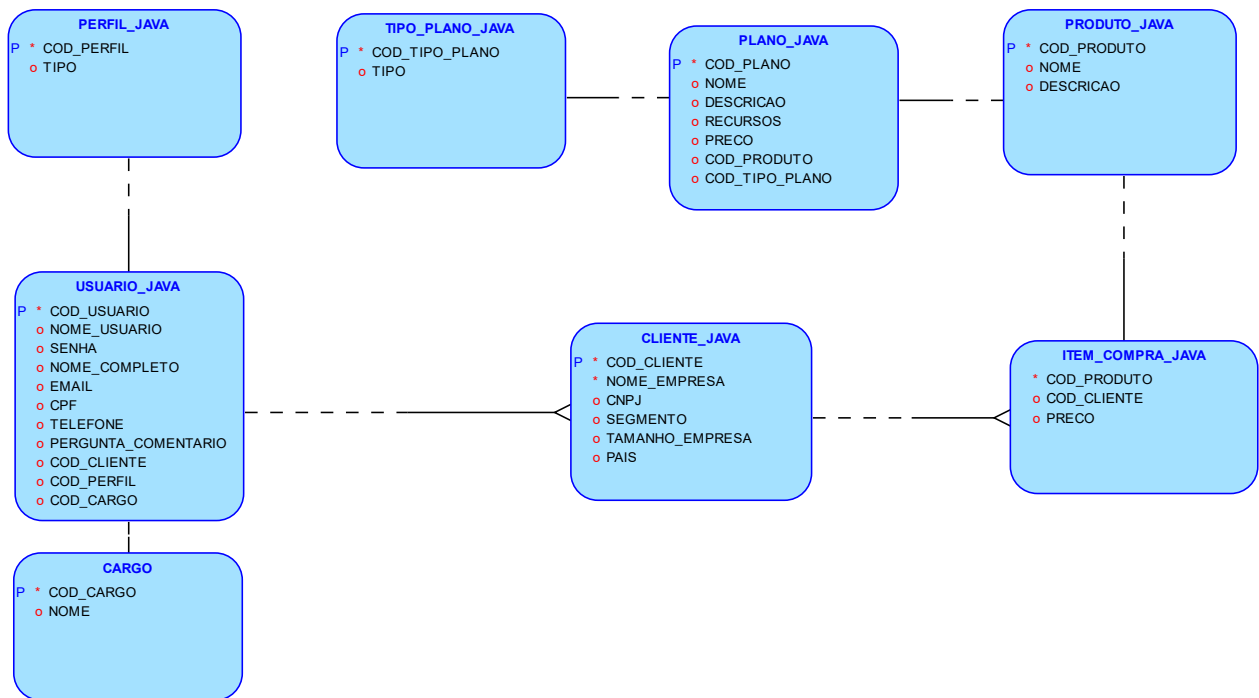
- *Descrição:* Remove um usuário específico.
- *Exemplo de Resposta: 204 No Content*

Esta tabela cobre os principais métodos e códigos de status esperados para uma API RESTful básica voltada para gerenciamento de usuários.

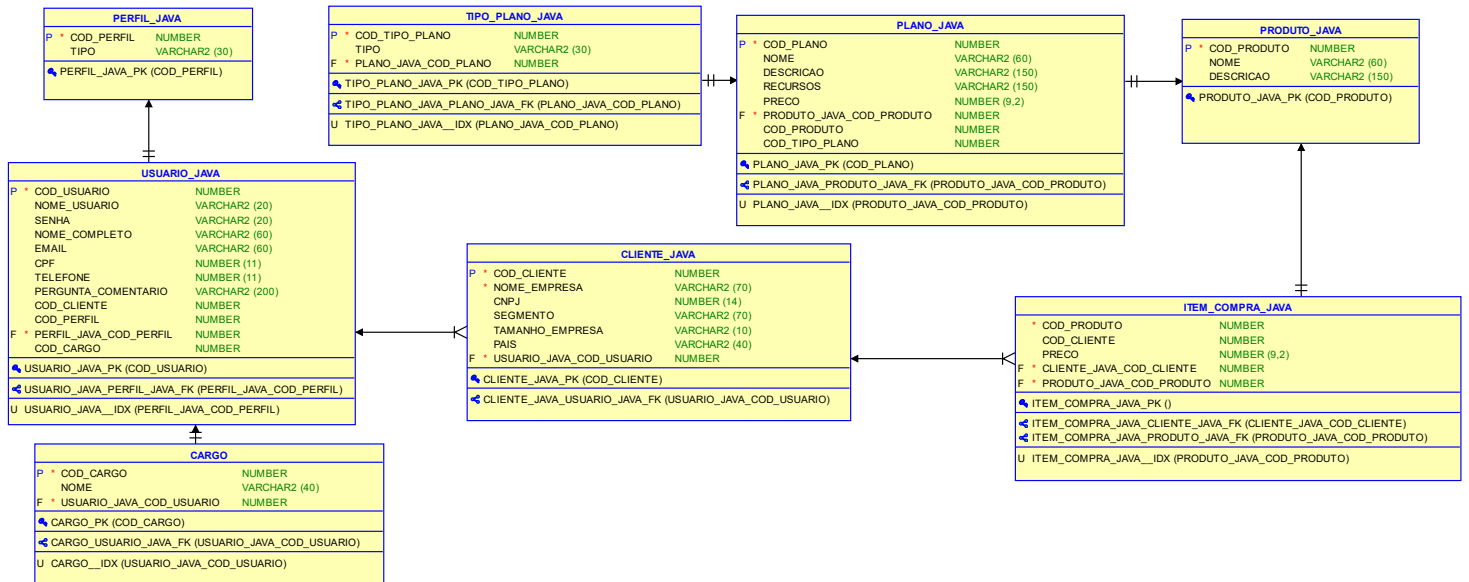
## DIAGRAMA DE CLASSES JAVA



## DIAGRAMAS SQL – LÓGICO



# DIAGRAMAS SQL – RELACIONAL



## **EQUIPE**

Gabriel Torres Fernandes - 553635

Julia Gonzalez Pugliese Ribeiro - 553427

Lucas Ponciani Marinelli – 553112

## **ARQUIVO**

[https://github.com/juliapugliese/Challenge\\_Sprint3\\_Java\\_ATT](https://github.com/juliapugliese/Challenge_Sprint3_Java_ATT)