

FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

EQUIPE: JOVINLU TECH

VINICIUS RIBEIRO NERY COSTA – RM: 559165

JOÃO PEDRO SANSON MIGUEL BENITO – RM: 556802

LUCAS PERSON SERVOLLO – RM: 559186

CHALLENGE PORTO

SOLUÇÃO: PORTO PARA TODOS

SÃO PAULO/SP

NOVEMBRO/2024

Sumário

| | |
|---|----------|
| Objetivo e escopo do projeto | 3 |
| Tabela dos endpoints | 3 |
| Protótipo | 3 |
| Procedimentos para rodar a aplicação | 5 |
| Diagrama de classes atualizada | 6 |
| Modelo do banco de dados..... | 8 |

Objetivo e escopo do projeto

Objetivos do Challenger

O objetivo do Projeto Challenger Porto é melhorar a experiência do cliente na manutenção do seu veículo utilizando métodos de autodiagnóstico, criar um banco de dados e um auto-orçamento criando uma interação muito mais ágil e focado em colocar o cliente no centro do atendimento durante todo o projeto.

Ideias Challenger

A criação do web site terá a formatação que a Porto utiliza em seus sites mantendo assim um padrão, o cliente poderá ter acesso ao chatbot que será integrado na página juntamente com uma breve explicação de como ele funciona tirando possíveis dúvidas dos usuários.

O banco de dados irá servir para armazenar dados sobre os clientes e parceiros porto (oficinas parceiras), o cliente irá criar um cadastro inicial, cadastro veículo e cadastro do possível problema do veículo.

Tabela dos endpoints

/api/clientes | POST | 200 (OK), 500 (Erro Interno)

/api/clientes/{cpf} | GET | 200 (OK), 404 (Não Encontrado), 500 (Erro Interno)

/api/clientes/{cpf} | DELETE | 200 (OK), 500 (Erro Interno)

/api/clientes/consulta/{placa} | GET | 200 (OK), 500 (Erro Interno)

/api/clientes/veiculo/{placa} | GET | 200 (OK), 500 (Erro Interno)

/api/clientes/download/{cpf} | GET | 200 (OK), 404 (Não Encontrado), 500 (Erro Interno)

Protótipo

```
Opções:  
1. Fazer login  
2. Não tenho login  
3. Sair  
2
```

Se você já possui login, digite "1". Caso não tenha login, digite "2" para criar uma conta rapidamente.

```
Opções:
1. Fazer login
2. Não tenho login
3. Sair
2
Digite o CPF: 41483164888
Digite o nome: Vinicius
Digite a data de nascimento (DDMMYYYY): 19032006
Digite o telefone: 11993640084
Digite o email: vininery2006@gmail.com
```

A imagem acima apresenta as perguntas que o cliente precisará responder caso não tenha um login.

```
Opções:
1. Fazer login
2. Não tenho login
3. Sair
1
Digite seu CPF: 41483164888
Login bem-sucedido!
```

Para realizar o login basta o cliente inserir o cpf já cadastrado.

```
Menu do Cliente:
1. Cadastrar Problema
2. Exportar Dados para JSON
3. Alterar meus dados
4. Deletar meu cadastro
5. Sair
1
Digite a placa do veículo: flm2688
```

A imagem acima exibe as perguntas que o cliente precisará responder sobre o carro, informando apenas a placa do veículo a API que foi implementada traz as informações restantes do carro.

```
Selecione o problema:
1. Superaquecimento - R$ 60,99
2. Pane elétrica - R$ 120,00
3. Problemas no câmbio - R$ 150,00
4. Bateria ruim - R$ 99,99
5. Falta de combustível - R$ 70,00
6. Carro trepidando - R$ 100,00
7. Pneus furados - R$ 80,00
8. Correia dentada com muito uso - R$ 400,00
9. Pneus carecas - R$ 150,00
10. Falta de revisão - R$ 200,00
1
```

Depois de preencher o campo com a placa do veículo são exibidos todos os possíveis problemas que o cliente pode estar enfrentando. Basta selecionar o número da opção desejada.

Procedimentos para rodar a aplicação

- Java Development Kit (JDK) 17: Verificar que o JDK 17 está instalado na máquina
- Apache Maven: Instale o Maven para gerenciar as dependências do projeto e compilar o código.
- Banco de Dados Oracle: A aplicação usa um banco de dados Oracle. Você precisará das credenciais e da URL de conexão.
- 4. Configuração do Banco de Dados: No arquivo `ConnectionFactory.java`, ajuste a URL de conexão, o login e a senha para corresponder ao seu ambiente de banco de dados.

Diagrama de classes atualizada



Modelo do banco de dados

```
DROP TABLE CLIENTES CASCADE CONSTRAINTS;  
DROP TABLE VEICULOS CASCADE CONSTRAINTS;  
DROP TABLE DIAGNOSTICOS CASCADE CONSTRAINTS;
```

-- Criação da tabela de clientes

```
CREATE TABLE clientes (  
    cpf VARCHAR2(11) PRIMARY KEY,  
    nome VARCHAR2(100) NOT NULL,  
    data_de_nascimento DATE NOT NULL,  
    telefone VARCHAR2(15),  
    email VARCHAR2(100),  
    senha VARCHAR2(255) -- Adicionado campo para senha  
);
```

-- Criação da tabela de veículos

```
CREATE TABLE veiculos (  
    placa VARCHAR2(7) PRIMARY KEY,  
    marca VARCHAR2(50) NOT NULL,  
    modelo VARCHAR2(50) NOT NULL,  
    ano_fabricacao NUMBER(4) CHECK (ano_fabricacao >=  
1886 AND ano_fabricacao <= 2025) -- Defina um ano  
máximo fixo  
);
```

-- Criação da tabela de diagnósticos

```
CREATE TABLE diagnosticos (  
    id NUMBER GENERATED BY DEFAULT AS IDENTITY  
PRIMARY KEY,  
    placa VARCHAR2(7) NOT NULL,  
    problema VARCHAR2(255) NOT NULL,  
    valor VARCHAR2(255), -- Mantido como valor  
    data_diagnostico TIMESTAMP DEFAULT  
CURRENT_TIMESTAMP, -- Alterado para TIMESTAMP  
    FOREIGN KEY (placa) REFERENCES veiculos(placa)  
);
```