

**FIAP**  
**Faculdade de Informática e Administração**  
**Paulista**

**Chatcar**

Henrique Garcia RM: 558062

Larissa Muniz RM: 557197

Laura Cintra RM: 558843

**Challenge – Porto Seguro**

DDD Using Java

4ªSprint

**São Paulo**

**2024**

## **SUMÁRIO**

<b>1.Objetivo do Projeto:</b>	<b>1</b>
<b>2.Cadastro do Usuário e Veículo:</b>	<b>1</b>
<b>3.Gestão da Conta do Cliente:</b>	<b>1</b>
<b>4.Gestão dos dados de Veículo</b>	<b>2</b>
<b>5.Tabela dos Endpoints, URIs verboHTTP:</b>	<b>3</b>
<b>6.Diagrama de Classes:</b>	<b>4</b>
<b>7. Modelo de Banco de dados:</b>	<b>4</b>
<b>8. Protótipo da tela de cadastro:</b>	<b>5</b>
<b>9. Protótipo da tela de acesso / login:</b>	<b>5</b>
<b>10. Protótipo da tela de interação com a IA:</b>	<b>6</b>
<b>11. Protótipo da tela sobre o grupo:</b>	<b>6</b>
<b>12. Cadastro de carro:</b>	<b>7</b>
<b>13. Excluindo carro:</b>	<b>7</b>
<b>14. Listagem de carros:</b>	<b>8</b>
<b>15. Editando dados de carro:</b>	<b>8</b>
<b>16. Procedimentos para rodar a aplicação:</b>	<b>9</b>

### **1.Objetivo do Projeto:**

**Nosso projeto tem como objetivo primordial trazer uma experiência inigualável de reparo e cuidados com seu veículo de maneira prática e eficiente, A missão é facilitar e esclarecer os problemas automotivos que muitas pessoas enfrentam atualmente, os problemas serão apresentados pelo usuário para o nosso chatbot. Com a nossa Inteligência Artificial via chatbot, vamos trazer diagnósticos precisos e juntamente com eles pré-orçamentos dos pontos apresentados e identificados pelo cliente relacionados aos seus veículos.**

### **2.Cadastro do Usuário e Veículo:**

**O usuário acessará nossa página web onde será guiado para realizar seu cadastro, tendo como requisitos algumas informações obrigatórias, sendo elas: nome, e-mail, telefone, endereço, CPF. Ao finalizar, o cliente já poderá efetuar o diagnóstico, onde vai indicar o veículo que precisa da vistoria, especificando se ele é um carro, moto, carreta entre outros, também será necessário declarar algumas informações sobre ele como: cor, modelo, ano de fabricação, marca, placa, essas informações adicionais contribuem significativamente com a precisão, assim trazendo maior assertividade do diagnóstico.**

### **3.Gestão da Conta do Cliente:**

**O processo para diagnóstico será feito acessando a aba da interação com a IA, o chatbot conduzirá o cliente através de perguntas e apresentará alguns passos para o cliente verificar o**

possível problema, visando chegar o mais próximo de um bom diagnóstico. Após toda a apresentação do diagnóstico, será concedido a opção de realizar um pré-orçamento gerado pela IA, para o reparo das possíveis peças que estejam danificadas. Em seguida, será sugerido ao cliente uma oficina parceira mais próxima do local presente, para que ele já resolva o problema o quanto antes e ajude a manter o veículo em boas condições. Na sequência, será possível o cliente vincular o automotivo apresentado a sua conta cadastrada, podendo assim, acessar os históricos e os pré-orçamentos, tanto anteriores como as mais recentes, que já foram feitos nas suas consultas quando desejar. 4.Armazenamento de Dados: O histórico, de toda interação feita, ficará salvo em servidores neles estará contido dados como as perguntas e as respostas entregue ao usuário, por meio dele, o cliente só tem acesso aos seus respectivos históricos que estarão vinculados aos seus veículos. Os dados serão analisados para facilitar o acesso e resolução dos próximos problemas, por meio de Perguntas Frequentes que serão apresentadas para todos os usuários, a fim de agilizar o processo de encontrar o problema com eficácia.

#### **4.Gestão dos dados de Veículo.**

Após o login o cliente terá acesso ao cadastro do seu veículo, possibilitando edição e exclusão do mesmo caso seja necessário. Será disponibilizado junto a função para verificar seus veículos, caso o cliente tenha mais um registrado.

## **5.Tabela dos Endpoints, URIs verboHTTP.**

**Caminho HTTP: http://localhost:8080/**

### **5.1 Path: chatcarcliente**

#### **5.1.1 POST: cadastrocliente**

**Cadastra o cliente no banco de dados.**

#### **5.1.2 POST: logincliente**

**Acessa a conta do cliente verificando os dados com o banco de dados**

#### **5.1.3 PUT: alteracliente**

**Altera informações da conta do cliente**

#### **5.1.4 GET: lendocliente/{email}**

**Exibe dados do cliente de acordo com o email**

#### **5.1.5 POST: autenticcliente**

**Verifica os dados de acordo com email e senha**

#### **5.1.6 DELETE: deletecliente/{email}**

**Apaga os dados do cliente de acordo com email**

### **5.2 Path: chatcarveiculo**

#### **5.2.1 POST: cadastraveiculo**

**Cadastra o veículo no banco de dados**

#### **5.2.2 PUT: alteraveiculo**

**Altera dados do veículo no banco de dados**

#### **5.2.3 GET: buscaveiculo/{idveiculo}**

**Apresenta todos os dados do veículo escolhido pelo IDveiculo.**

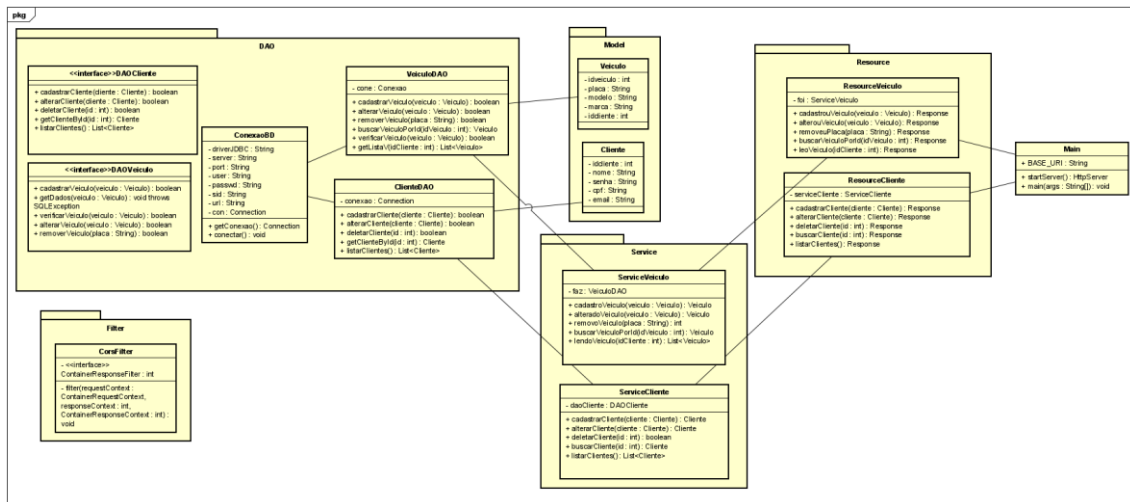
#### **5.2.4 GET: lendoveiculo/{idcliente}**

Apresenta uma lista dos veículos de acordo com o IDcliente.

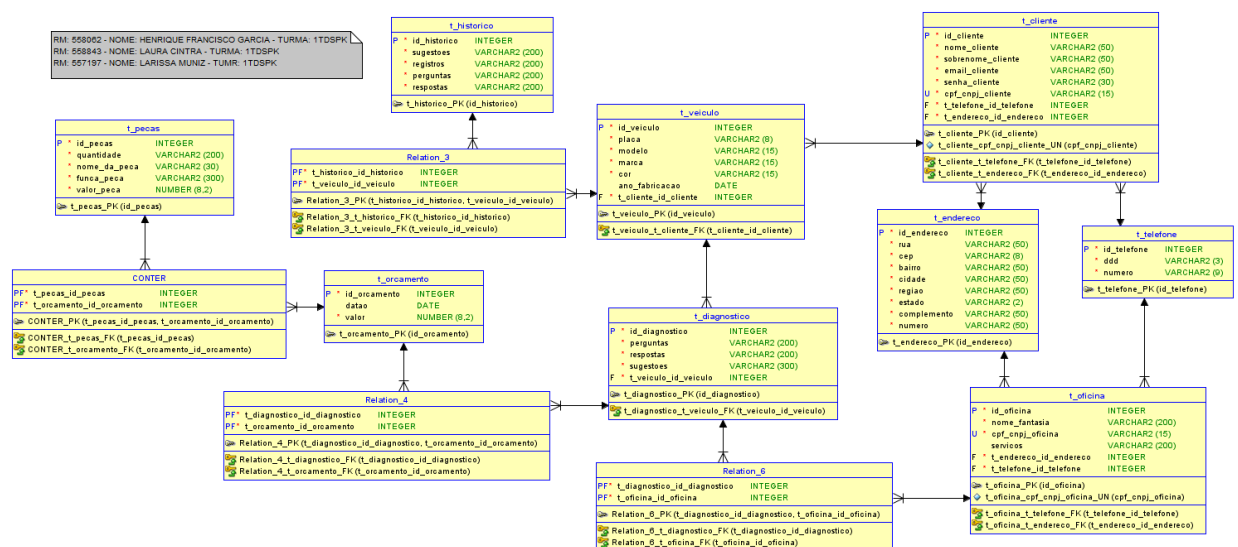
### 5.2.5 DELETE: deleteveiculo/{placa}

Apaga os dados de veículo de acordo com a placa

## 6. Diagrama de Classes



## 7. Modelo de Banco de dados



## 8. Protótipo da tela de cadastro.

### Cadastrar Conta

Nome

CPF

E-mail

Senha

Confirmar Senha

[Cadastrar](#)

[Limpar](#)

Já possui uma conta cadastrada?  
[Entrar em conta existente](#)

© 2024 ChatCar. Todos os direitos reservados.

Na tela de cadastro o cliente / oficina irá realizar seu cadastro no sistema preenchendo os dados solicitados na tela.

## 9. Protótipo da tela de acesso / login.

### Acessar Conta

E-mail

Senha

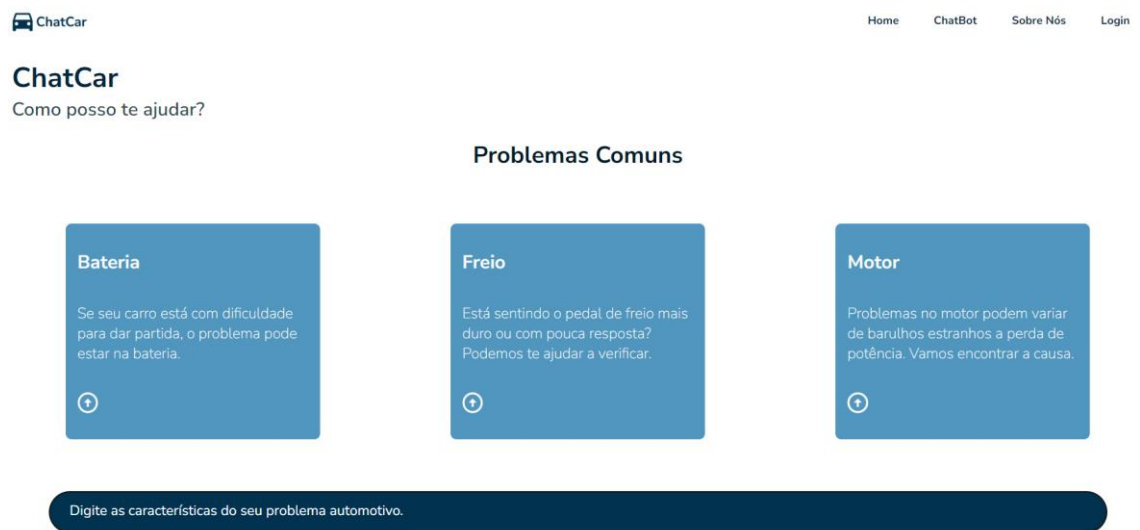
[Entrar](#)

[Limpar](#)

Ainda não tem uma conta?  
[Cadastre-se agora!](#)

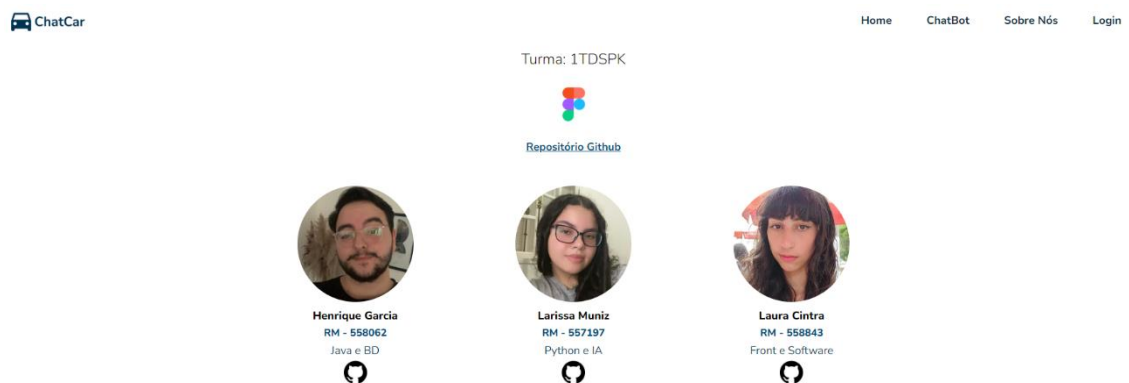
A parte do acesso será apresentada logo após o cadastro ser efetuado, com a opção de cadastrar caso não tenha feito.

## 10. Protótipo da tela de interação com a IA.



Nesta aba o usuário fará sua interação com o a IA, onde o cliente perguntará e será respondido e conduzido pela IA para a checagem do possível problema.

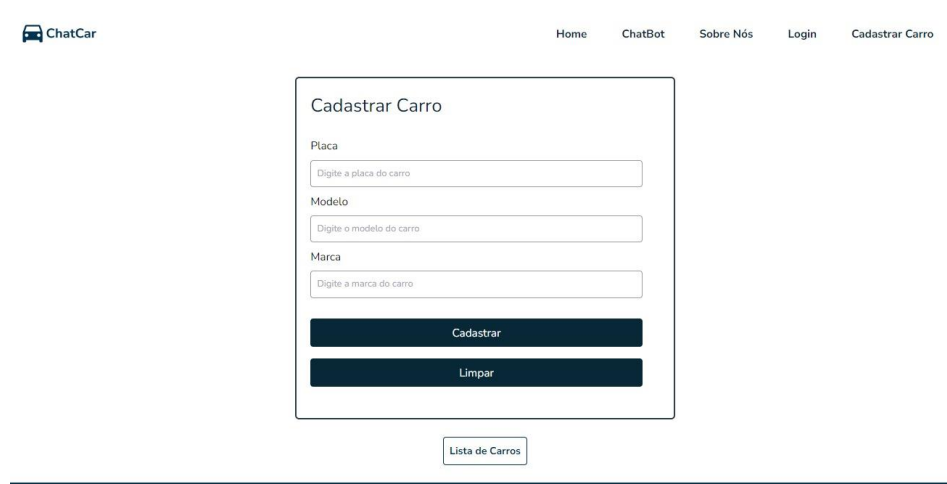
## 11. Protótipo da tela sobre o grupo.



Tela de acesso aos dados dos integrantes do grupo, junto o link do github de cada um para fácil acesso.



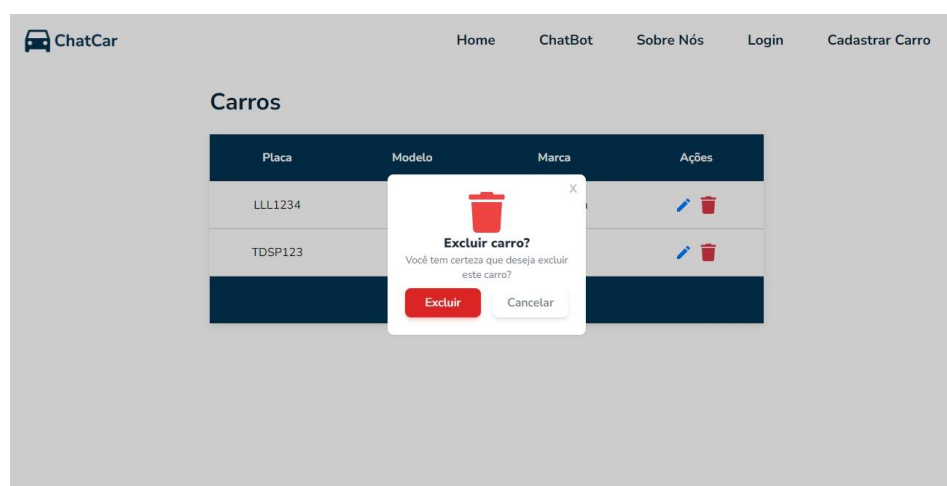
## 12. Cadastro de carro







The screenshot shows the 'Cadastrar Carro' (Register Car) form in the ChatCar application. The form is titled 'Cadastrar Carro' and contains three input fields: 'Placa' (Plate) with the placeholder 'Digite a placa do carro', 'Modelo' (Model) with the placeholder 'Digite o modelo do carro', and 'Marca' (Brand) with the placeholder 'Digite a marca do carro'. Below the input fields are two buttons: 'Cadastrar' (Register) and 'Limpar' (Clear). At the bottom of the form, there is a link labeled 'Lista de Carros' (List of Cars).

Tele onde o usuário irá realizar o cadastro de seus veículos após login.

## 13. Excluindo carro







The screenshot shows the 'Carros' (Cars) table in the ChatCar application. The table has four columns: 'Placa' (Plate), 'Modelo' (Model), 'Marca' (Brand), and 'Ações' (Actions). The table contains two rows of data: one with 'LLL1234' and another with 'TDSP123'. Each row has a pencil icon for editing and a trash can icon for deleting. A confirmation dialog is open over the table, asking 'Excluir carro?' (Delete car?) with the text 'Você tem certeza que deseja excluir este carro?' (Are you sure you want to delete this car?). The dialog has two buttons: 'Excluir' (Delete) and 'Cancelar' (Cancel).

Placa	Modelo	Marca	Ações
LLL1234			 
TDSP123			 

Nesta tela o usuário poderá excluir seu carro cadastrado no seu login.

## 14. Listagem de carros.

### Carros

Placa	Modelo	Marca	Ações
LLL1234	Civic	Volkswagem	 
TDSP123	Uno	Fiat	 
Total de Carros: 2			

Tela o qual será apresentado os carros registrados na conta do cliente

## 15. Editando dados de carro.

### Editar Carro

Placa

Modelo

Marca

[Editar Carro](#)

Opção de editar os dados do carro caso o usuário precise realizar alguma alteração.

## **16. Procedimentos para rodar a aplicação.**

1º Passo – Faça o login no SQL Developer utilizando os dados encontrados na classe ConexaoBD no código de java.

2º Passo - Crie no SQL Developer as tabelas, usando o código já apresentado e ordenados.

3 º Passo – Realize no SQL Developer os inserts já disponíveis para a Tabela Cliente e a Tabela Veículo, para que o ID seja gerado.

4 º Passo – Abra o código de java no IntelliJ e execute, para que ele retorna o HTTP

5 º Passo – Abra o Postman ou Insomnia para utilizar o HTTP os endpoints junto com as URIs para executar os comandos

6 º Passo – Alguns serão PathParams então será necessário verificar os ID gerados no SQL para o teste, outros serão json de acordo com o que é solicitado no código.