

J.A.R.V.I.S

DOMAIN DRIVE DESIGN

AutoCar

Challenge Fiap – Porto Seguro
SPRINT 4

Nome:

Turma: 1DSPV

Raphael Pabst rm98525

Silvio Junior rm550821

Pedro Braga rm551061

Leonardo Ribeiro Barral RM551472

Eduardo Reis Braga rm551987

Sumário

1. Objetivo e escopo do projeto.....	3
2. Descrição das funcionalidades.....	4
3. Tabela dos EndPoints.....	5
4. Modelo do banco de dados.....	6
5. Diagrama de Classe.....	7

Objetivo e escopo do projeto

A proposta do nosso projeto é o desenvolvimento e aperfeiçoamento de um software capaz de aumentar o nível de acertos na escolha de modal durante o atendimento ao cliente com a entrada de dados disponibilizados pelo cliente na aberta Ordem de Serviço (OS).

Nosso Projeto de modelagem de classe possui início na Classe Cliente. Nesta classe, identificamos o perfil do cliente e realizamos uma comparação de dados sensíveis como CPF e NOME para averiguar as informações pessoais, da apólice e na sequência realizar a OS. Neste momento, na classe Tipo de Sinistro, o cliente deve inserir informações sobre suas necessidades e relações do sinistro ocorrido a Inteligência artificial (IA) será capaz de analisar a entrada de dados e realizar testes de validação para escolha do Modal adequado de acordo com os dados fornecidos pelo cliente e pela base de dados.

A classe Veículo do Cliente possui atributos de identificação do veículo como tipo do veículo, número do Chassis, peso, altura e entre outros dados já coletados posteriormente. Por este fato, podemos observar uma associação de classes entre Veículo do Cliente e Tipo de Sinistro. Desta forma, podemos dar sequência ao próximo passo.

A escolha do modal ocorre de duas formas, pela localização do sinistro atribuída pela classe Endereço de Sinistro e qual modal está mais próximo do local pelas classes Veículos Modal e sua composição Endereço Modal. Ao fazer essa associação, será possível identificar e enviar o mais rápido possível o modal adequado para suporte ao cliente, essa atribuição irá fornecer a entrada dos métodos (tempo Chegada e modal Disponível), que será responsável por validar se aquele ponto de assistência possui ou não o modal necessário para atender o sinistro.

Feito este processo, o cliente será informado do tempo de chegada do modal no endereço informado na OS e deve aguardar a chegada do mesmo no local. Após a realização do atendimento ao cliente, o modal será responsável em encaminhar o veículo do cliente no endereço informado e assim encerrar o atendimento e retornar para base. Com a constante entrada de dados neste sistema, podemos cada vez mais aumentar a assertividade de nossa IA e oferecer um serviço de excelência para nossos clientes.

PROTÓTIPO

METODOS GET

Aqui tem alguns exemplos dos métodos get para Cliente ,Email, Telefone

REST client interface showing a GET request to `http://localhost:8080/api/cliente`. The response is a JSON array of two client objects.

```
1 {
2   "dt_cadastro": "2018-03-18",
3   "dt_data_nascimento": "1995-07-20",
4   "id_cliente": 1,
5   "nm_cliente": "Ronaldo da silva ",
6   "nr_cnh": "47968336842",
7   "nr_cpf": "58189933889",
8   "nr_rg": "111631221"
9 }
10
11 {
12   "dt_cadastro": "2015-02-05",
13   "dt_data_nascimento": "1999-06-15",
14   "id_cliente": 2,
15   "nm_cliente": "Maria",
16   "nr_cnh": "82784968087",
17   "nr_cpf": "6804436884",
18   "nr_rg": "461484837"
```

REST client interface showing a GET request to `http://localhost:8080/api/cliente/1`. The response is a JSON object representing a single client.

```
1 {
2   "dt_cadastro": "2018-03-18",
3   "dt_data_nascimento": "1995-07-20",
4   "id_cliente": 1,
5   "nm_cliente": "Ronaldo da silva ",
6   "nr_cnh": "47968336842",
7   "nr_cpf": "58189933889",
8   "nr_rg": "111631221"
9 }
```

GET

http://localhost:8080/api/cliente/email

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

BodyCookiesHeaders (2)Test Results

Status: 200 OKTime: 141 msSize: 1.66 KBSave as example

PrettyRawPreviewVisualizeJSON

```
1  {
2    {
3      "ds_email": "RedVelvet@gmail.com",
4      "id_cliente": 21,
5      "id_email": 21,
6      "st_email": "A "
7    },
8    {
9      "ds_email": "RedVelvet@gmail.com",
10     "id_cliente": 21,
11     "id_email": 1,
12     "st_email": "I "
13   },
14   {
15     "ds_email": "Maria@gmail.com",
16     "id_cliente": 2,
17     "id_email": 2,
18     "st_email": "A "
```

GET

http://localhost:8080/api/cliente/telefone

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

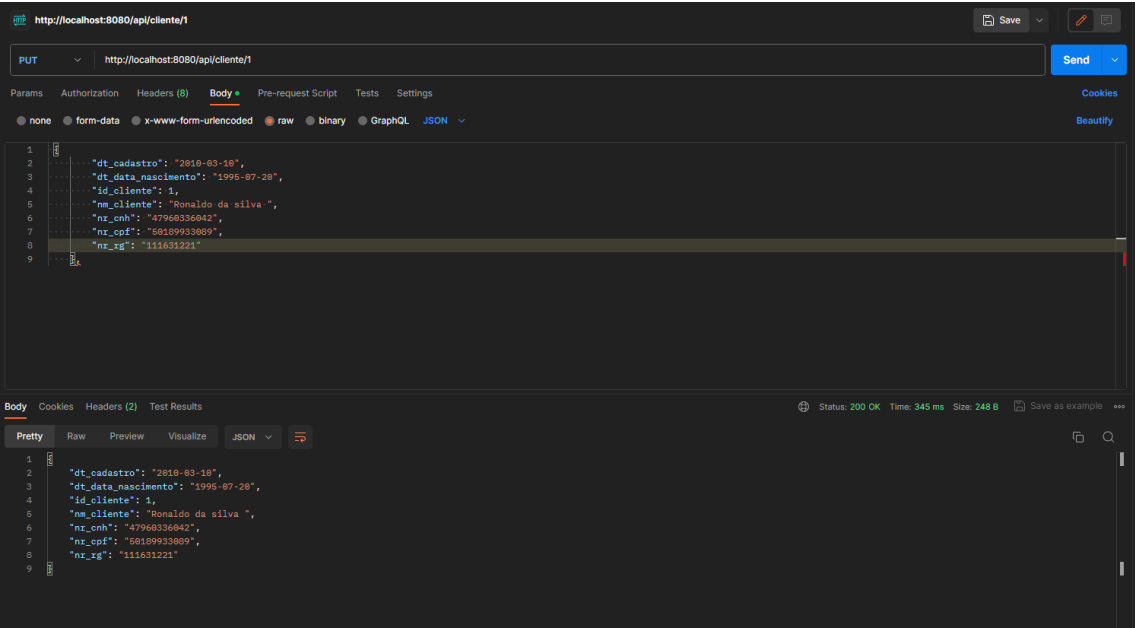
BodyCookiesHeaders (2)Test Results

Status: 200 OKTime: 133 msSize: 2.11 KBSave as example

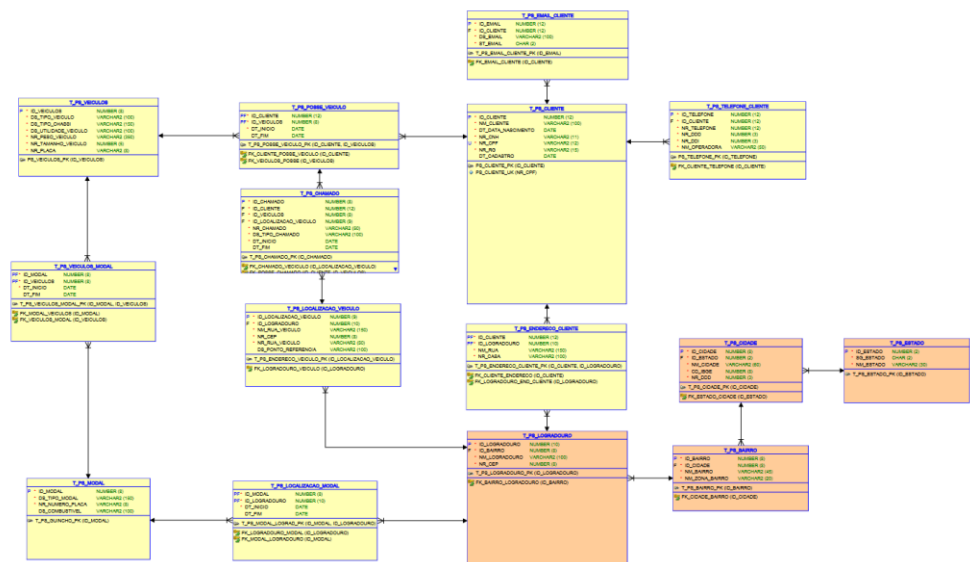
PrettyRawPreviewVisualizeJSON

```
1  {
2    {
3      "id_cliente": 21,
4      "id_telefone": 21,
5      "nm_operadora": "Vivo",
6      "nr_ddd": 81,
7      "nr_ddd": 85,
8      "nr_telefone": 81998887777
9    },
10   {
11     "id_cliente": 1,
12     "id_telefone": 1,
13     "nm_operadora": "Vivo",
14     "nr_ddd": 11,
15     "nr_ddd": 85,
16     "nr_telefone": 11942398787
17   },
18   {
```

Aqui tem um exemplo de métodos PUT para Cliente ,Email, Telefone que pegar a ID para identificar o cliente pelo ID



Modelo do Banco de Dados



Modelo do banco de Dados(enxuto)

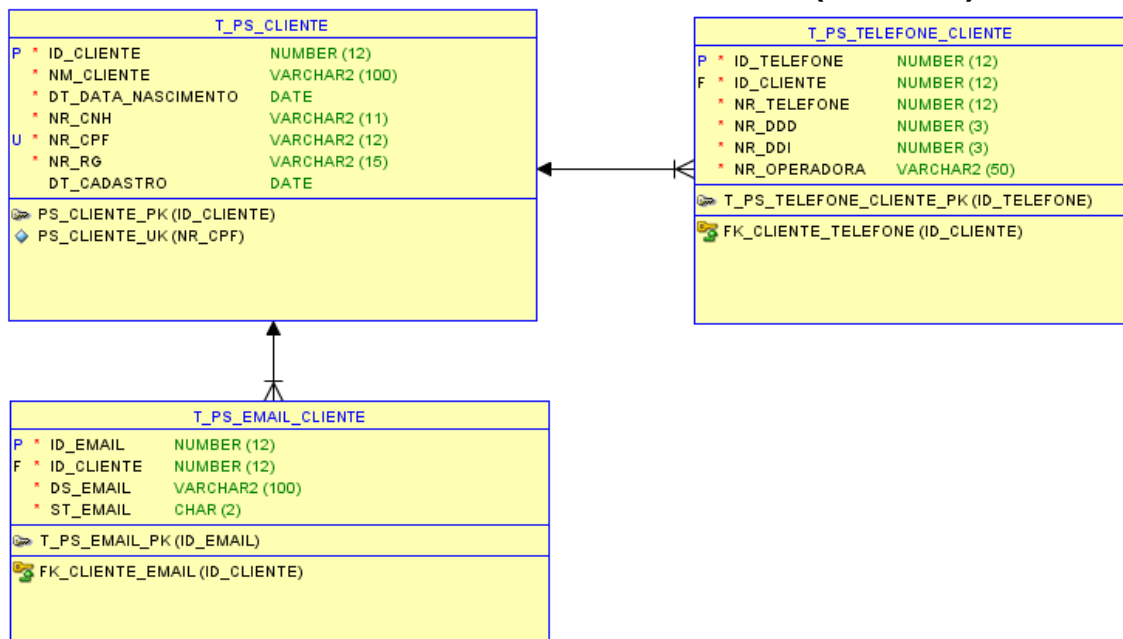


Diagrama de Classe

