



Faculdade de Informática e Administração Paulista

# **Domain Driving Using Java**

## **Sprint 4**

# INTEGRANTES

<b>RM (Somente Numeros)</b>	<b>Nome Completo (Sem abreviar)</b>
554874	João Gabriel Boaventura Marques e Silva
558791	Lucas de Melo Pinheiro Pinho
551124	Lucas Leal das Chagas

## Sumário

1. DESCRIÇÃO DO PROJETO.....	4
2. INTERAGINDO COM O SISTEMA.....	6
3. MODELO DO BANCO DE DADOS.....	10
4. DIAGRAMA DE CLASSES.....	14

## **1 – DESCRIÇÃO DO PROJETO**

O código funcionará para manter informações importantes do cliente para que possibilite o contato, e principalmente para que o assistente consiga ter acesso às informações tanto de automóveis quanto do plano, para assim trazer agilidade e a ajuda necessária para o devido problema do usuário. Para os planos, o código fornecerá informações sobre os diferentes tipos de carros que atendem, dependendo do estilo de vida, sobre os serviços oferecidos. O código funcionará para fornecer informações sobre as Oficinas disponíveis para o conserto de seu veículo a partir de seu plano, lhe informando sobre Avaliações, Endereço e Serviços. E também de Lojas Parceiras que lhe fornecera as peças de que necessita para a reparação de seu automóvel, disponibilizando Peças e diferentes serviços.

O Principal problema que o programa tem como função ajudar, é trazer soluções rápidas para aquelas pessoas que não possuem e nem querem possuir conhecimento mecânico, trazendo facilidade e praticabilidade para a vida do cliente.

-Cliente: Nome, CPF, CNH, Telefone, E-mail e Endereço.

-ClienteDAO: InserirCliente.

-Automóveis: Placa, Modelo, Marca.

-AutomóveisDAO: inserirAutomoveis.

-Oficina: Endereço, Tipos de Serviço, Especialização.

-OficinaDAO: InserirOficina.

-Lojas Parceira :Peças, Endereço, Avaliações.

-Loja ParceiraDAO: InserirLoja\_Parceira.

-Peca: Nome, Preço, Quantidade de peças.

-PecaDAO: InserirPeca.

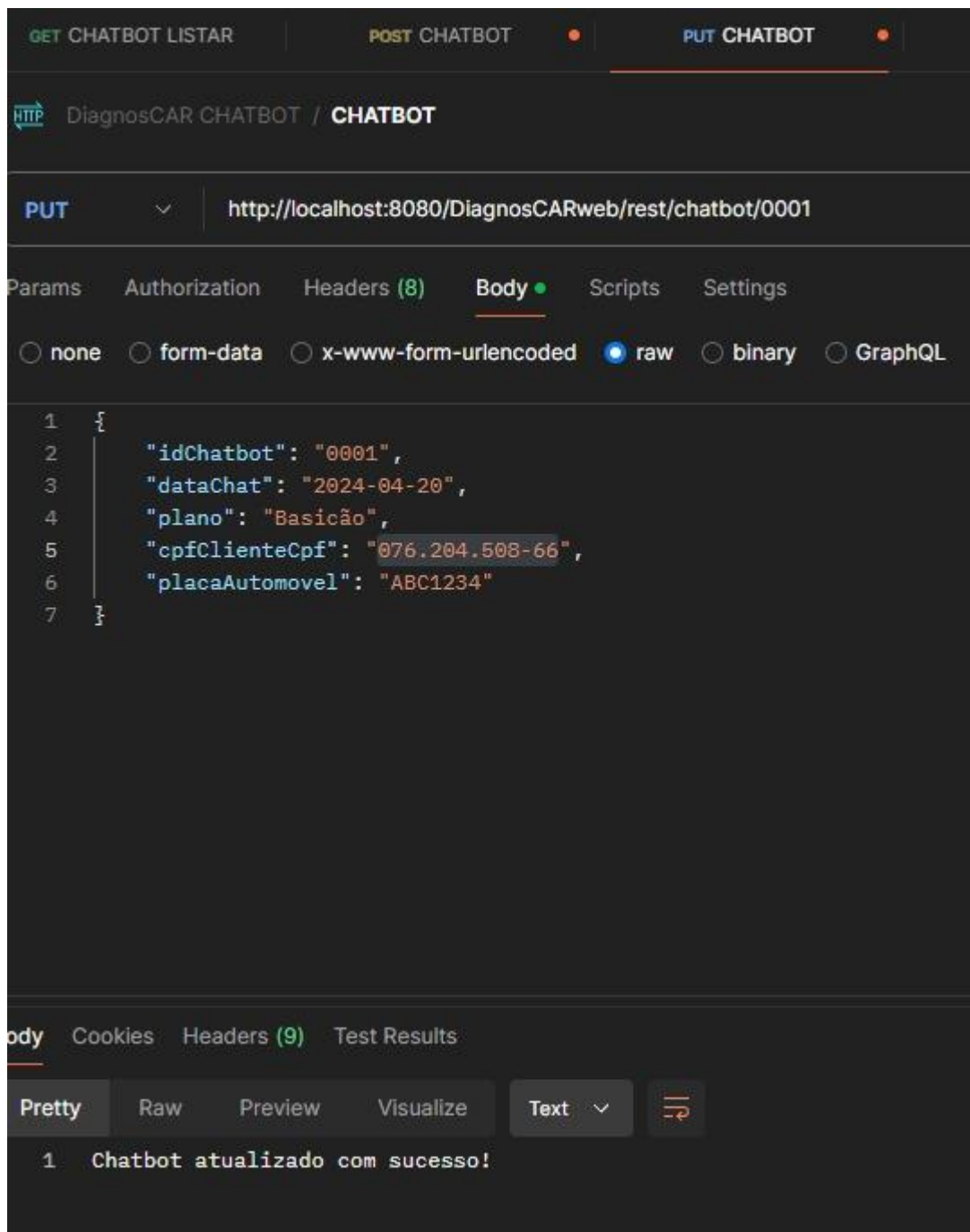
-TesteDAO: Classe Main.

-ConnDAO: conectar e desconectar.

## Breve Descrição das Principais Funcionalidades do Projeto

- **Gerenciamento de Clientes:** Cadastro, edição, exclusão e busca de clientes, armazenando informações como CPF, CNH, e dados de contato.
- **Gerenciamento de Automóveis:** Vinculação de veículos aos clientes, incluindo informações como placa, marca, modelo e ano.
- **Gerenciamento de Lojas Parceiras e Oficinas:** Cadastro e listagem de lojas e oficinas, com informações de CNPJ, nome e especialização.
- **Gerenciamento de Peças:** Registro de peças disponíveis nas lojas parceiras, permitindo consulta e inserção de peças no sistema.

## 2- Interagindo com o sistema



GET CHATBOT LISTAR

POST CHATBOT

PUT CHATBOT

DEL CHATBOT

+

DiagnosCAR CHATBOT / CHATBOT LISTAR

GET

▼

http://localhost:8080/DiagnosCARweb/rest/chatbot

ParamsAuthorizationHeaders (6)BodyScriptsSettings

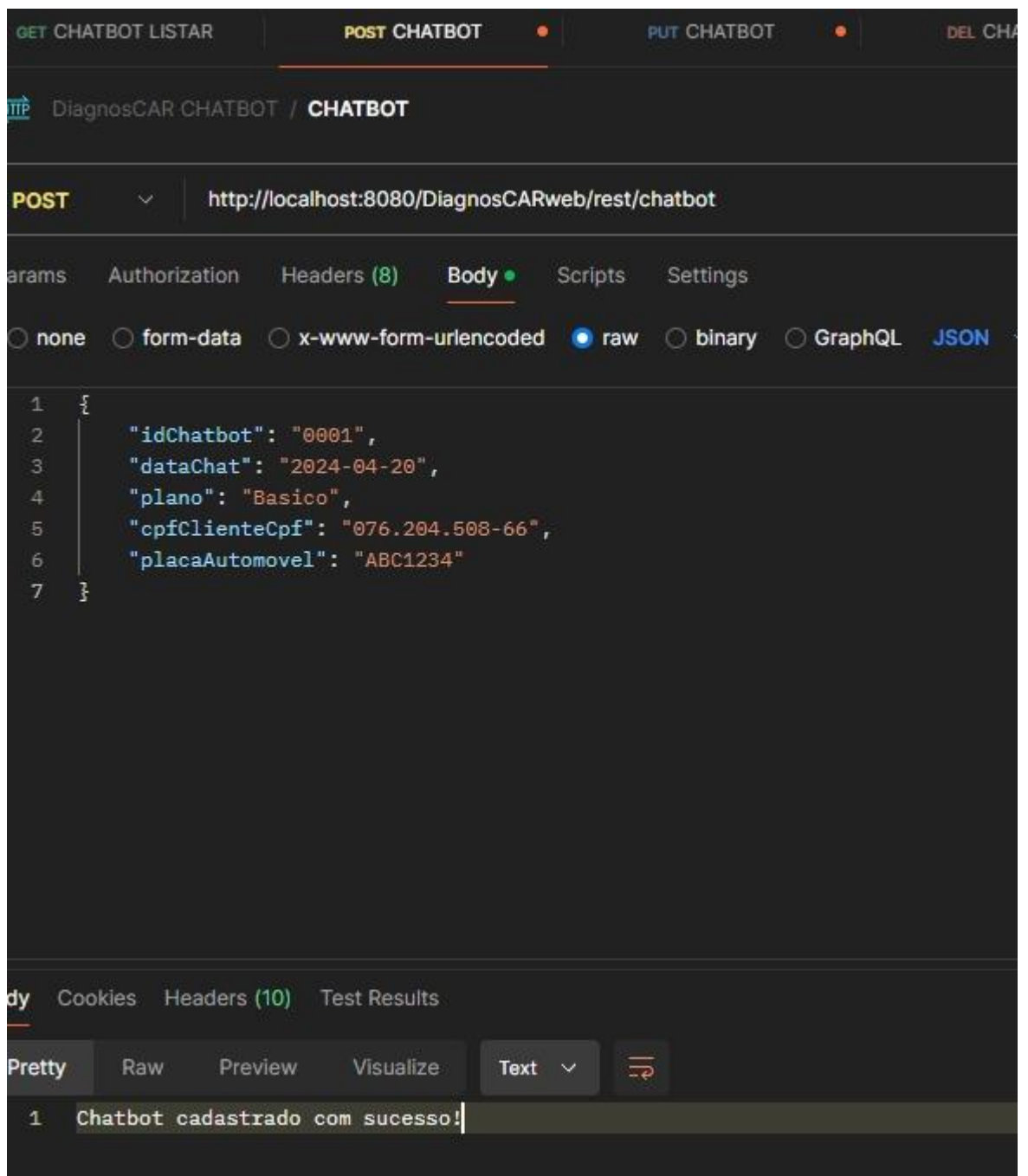
Query Params

	Key	Value
	Key	Value

BodyCookiesHeaders (9)Test Results

PrettyRawPreviewVisualizeJSON▼

```
1  [
2    {
3      "idChatbot": "0001",
4      "dataChat": "2024-04-19",
5      "plano": "Basico",
6      "cpfClienteCpf": "076.204.508-66",
7      "placaAutomovel": "ABC1234"
8    }
9  ]
```





GET CHATBOT LISTAR

POST CHATBOT

PUT CHATBOT

DEL CHATBOT

DiagnosCAR CHATBOT / CHATBOT

DELETE

http://localhost:8080/DiagnosCARweb/rest/chatbot/0001/076.204.508-66

Params

Authorization

Headers (6)

Body

Scripts

Settings

Query Params

	Key	Value
	Key	Value

Body

Cookies

Headers (9)

Test Results

Pretty

Raw

Preview

Visualize

Text

1 Chatbot deletado com sucesso!

### 3- Modelo do Banco de Dados

```
CREATE TABLE Cliente (  
    CPF_Cliente VARCHAR2(14) CONSTRAINT PK_Cliente PRIMARY KEY,  
    CNH_Cliente VARCHAR2(11) CONSTRAINT CNH_Cliente UNIQUE,  
    RG_Cliente VARCHAR2(12) CONSTRAINT RG_Cliente UNIQUE,  
    Nome_Cliente VARCHAR2(100) CONSTRAINT Nome_Cliente NOT NULL,  
    Sobrenome_Cliente VARCHAR2(100) CONSTRAINT Sobrenome_Cliente NOT NULL,  
    DataNasc_Cliente DATE CONSTRAINT DataNasc_Cliente NOT NULL,  
    Email_Cliente VARCHAR2(100) CONSTRAINT Email_Cliente UNIQUE,  
    Senha_Cliente VARCHAR2(50) CONSTRAINT Senha_Cliente NOT NULL,  
    Telefone_Cliente VARCHAR2(15) CONSTRAINT Telefone_Cliente UNIQUE,  
    Endereco_Cliente VARCHAR2(200) CONSTRAINT Endereco_Cliente NOT NULL,  
  
    CONSTRAINT CPF_Cliente CHECK (REGEXP_LIKE(CPF_Cliente, '[0-9]{3}\.[0-9]{3}\.[0-9]{3}-[0-9]{2}$')),  
    CONSTRAINT Telefone_Cliente_CHK CHECK (REGEXP_LIKE(Telefone_Cliente, '^\(\d{2}\)\d{4,5}-\d{4}$')),  
    CONSTRAINT Email_Cliente_CHK CHECK (REGEXP_LIKE(Email_Cliente, '[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'))  
);
```

```
CREATE TABLE Automovel (  
    Placa_Automovel VARCHAR2(7) CONSTRAINT PK_Automovel PRIMARY KEY,  
    Marca_Automovel VARCHAR2(20) CONSTRAINT Marca_Automovel NOT NULL,  
    Modelo_Automovel VARCHAR2(100) CONSTRAINT Modelo_Automovel NOT NULL,  
    Ano_Automovel NUMBER(4) CONSTRAINT Ano_Automovel NOT NULL,  
    Cliente_CPF_Cliente VARCHAR2(14),  
  
    CONSTRAINT FK_Automovel_Cliente FOREIGN KEY (Cliente_CPF_Cliente) REFERENCES  
    Cliente(CPF_Cliente)  
);
```

```
CREATE TABLE Chatbot (  
    ID_Chatbot VARCHAR2(1000) CONSTRAINT ID_Chatbot NOT NULL,  
    Data_Chat DATE CONSTRAINT Horario_Chatbot NOT NULL,  
    Plano VARCHAR2(10) CONSTRAINT Plano_Chatbot NOT NULL,  
    Cliente_CPF_Cliente VARCHAR2(14),  
    Placa_Automovel VARCHAR2(7),  
  
    CONSTRAINT PK_Chatbot PRIMARY KEY (ID_Chatbot, Cliente_CPF_Cliente),  
    CONSTRAINT FK_Chatbot_Cliente FOREIGN KEY (Cliente_CPF_Cliente) REFERENCES  
    Cliente(CPF_Cliente),  
    CONSTRAINT FK_Chatbot_Automovel FOREIGN KEY (Placa_Automovel) REFERENCES
```

Automovel(Placa\_Automovel)  
);

```
CREATE TABLE Pre_Diagnostico (  
    ID_PreDiagnostico VARCHAR2(1000) CONSTRAINT PK_PreDiagnostico PRIMARY KEY,  
    Nivel_Diagnostico NUMBER(3) CONSTRAINT Nivel_PreDiagnostico NOT NULL,  
    Diagnostico VARCHAR2(500) CONSTRAINT Diagnostico_PreDiagnostico NOT NULL,  
    Assistente_ID_Chatbot VARCHAR2(1000) CONSTRAINT Assistente_PreDiagnostico NOT  
NULL,  
    Cliente_CPF_Cliente VARCHAR2(14) CONSTRAINT Cliente_PreDiagnostico NOT NULL,  
    Placa_Automovel VARCHAR2(7),  
  
    CONSTRAINT Nivel_PreDiagnostico_CHK CHECK (Nivel_Diagnostico >= 1 AND  
Nivel_Diagnostico <= 100),  
    CONSTRAINT FK_PreDiagnostico_Chatbot FOREIGN KEY (Assistente_ID_Chatbot,  
Cliente_CPF_Cliente) REFERENCES Chatbot(ID_Chatbot, Cliente_CPF_Cliente),  
    CONSTRAINT FK_PreDiagnostico_Automovel FOREIGN KEY (Placa_Automovel)  
REFERENCES Automovel(Placa_Automovel)  
);
```

```
CREATE TABLE Oficina (  
    Endereco_Oficina VARCHAR2(200) CONSTRAINT PK_Oficina PRIMARY KEY,  
    Cnpj_Oficina VARCHAR2(18) CONSTRAINT Cnpj_Oficina UNIQUE,  
    Nome_Oficina VARCHAR2(120) CONSTRAINT Nome_Oficina NOT NULL,  
    Avaliacao_Oficina NUMBER(3, 2),  
    Especializacao_Oficina VARCHAR2(50) CONSTRAINT Especializacao_Oficina NOT NULL,  
    Chatbot_ID_Chatbot VARCHAR2(1000),  
  
    Chatbot_Cliente_CPF_Cliente VARCHAR2(14),  
    CONSTRAINT FK_Oficina_Chatbot FOREIGN KEY (Chatbot_ID_Chatbot,  
Chatbot_Cliente_CPF_Cliente)REFERENCES Chatbot(ID_Chatbot, Cliente_CPF_Cliente),  
    CONSTRAINT Cnpj_Oficina_CHK CHECK (REGEXP_LIKE(Cnpj_Oficina, '[0-9]{2}\.[0-  
9]{3}\.[0-9]{3}/[0-9]{4}-[0-9]{2}$')),  
    CONSTRAINT Avaliacao_Oficina_CHK CHECK (Avaliacao_Oficina BETWEEN 0 AND 10)  
);
```

```
CREATE TABLE Loja_Parceira (  
    Endereco_Loja VARCHAR2(200) CONSTRAINT PK_Loja_Parceira PRIMARY KEY,  
    Cnpj_Loja VARCHAR2(18) CONSTRAINT Cnpj_Loja_Parceira UNIQUE,  
    Nome_Loja VARCHAR2(120) CONSTRAINT Nome_Loja_Parceira NOT NULL,  
    Avaliacao_Loja NUMBER(3, 2),
```

```

Especializacao_Loja VARCHAR2(50) CONSTRAINT Especializacao_Loja_Parceira NOT
NULL,
CONSTRAINT Cnpj_Loja_Parceira_CHK CHECK (REGEXP_LIKE(Cnpj_Loja, '^[0-9]{2}\.[0-
9]{3}\.[0-9]{3}/[0-9]{4}-[0-9]{2}$')),
CONSTRAINT Avaliacao_Loja_Parceira_CHK CHECK (Avaliacao_Loja BETWEEN 0 AND
10)
);

```

```

CREATE TABLE Peca (
ID_Peca VARCHAR(10) CONSTRAINT PK_Peca PRIMARY KEY,
Tipo_Peca VARCHAR2(30) CONSTRAINT Tipo_Peca NOT NULL,
Nome_Peca VARCHAR2(40) CONSTRAINT Nome_Peca NOT NULL,
Descricao_pecas VARCHAR2(200) CONSTRAINT Descricao_Peca NOT NULL,
Loja_Parceira_Endereco_Loja VARCHAR2(200),

CONSTRAINT FK_Peca_Loja FOREIGN KEY (Loja_Parceira_Endereco_Loja) REFERENCES
Loja_Parceira(Endereco_Loja)
);

```

```

CREATE TABLE Entrega (
ID_Entrega VARCHAR2(100) CONSTRAINT PK_Entrega PRIMARY KEY,
Data_Entrega DATE CONSTRAINT Data_Entrega NOT NULL,
Destino_Entrega VARCHAR2(200) CONSTRAINT Destino_Entrega NOT NULL,
Item_Entrega VARCHAR2(1000) CONSTRAINT Item_Entrega NOT NULL,
Endereco_Loja VARCHAR2(200) CONSTRAINT Endereco_Entrega NOT NULL,

```

```

CONSTRAINT FK_Entrega_Cliente FOREIGN KEY (Destino_Entrega) REFERENCES
Cliente(CPF_Cliente),
CONSTRAINT FK_Entrega_Peca FOREIGN KEY (Item_Entrega) REFERENCES
Peca(ID_Peca),
CONSTRAINT FK_Entrega_Loja FOREIGN KEY (Endereco_Loja) REFERENCES
Loja_Parceira(Endereco_Loja)
);

```

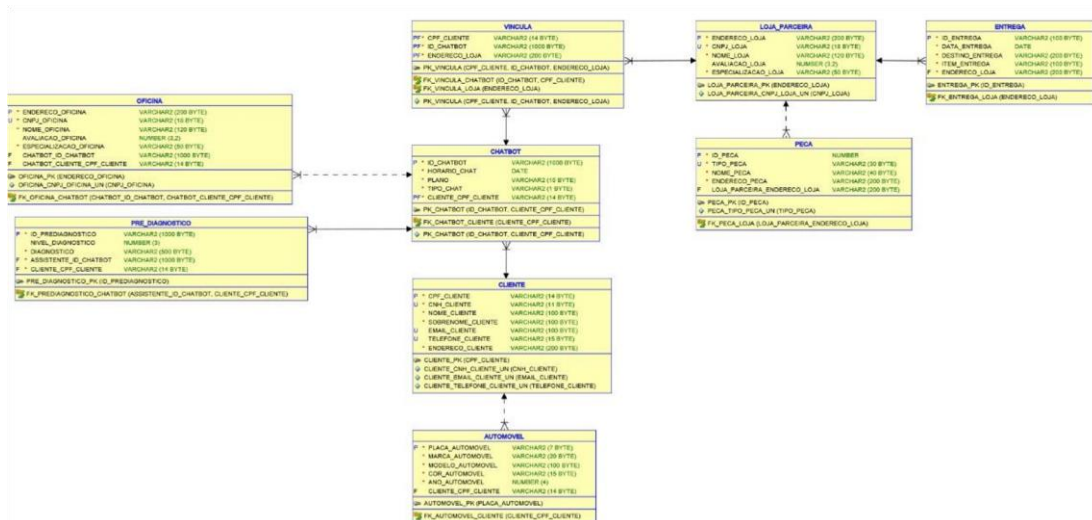
```

CREATE TABLE Tabela_de_Associacao (
CPF_Cliente VARCHAR2(14) CONSTRAINT CPF_Associacao NOT NULL,
ID_Chatbot VARCHAR2(1000) CONSTRAINT ID_Associacao NOT NULL,
Endereco_Loja VARCHAR2(200) CONSTRAINT EnderecoLoja_Associacao NOT NULL,

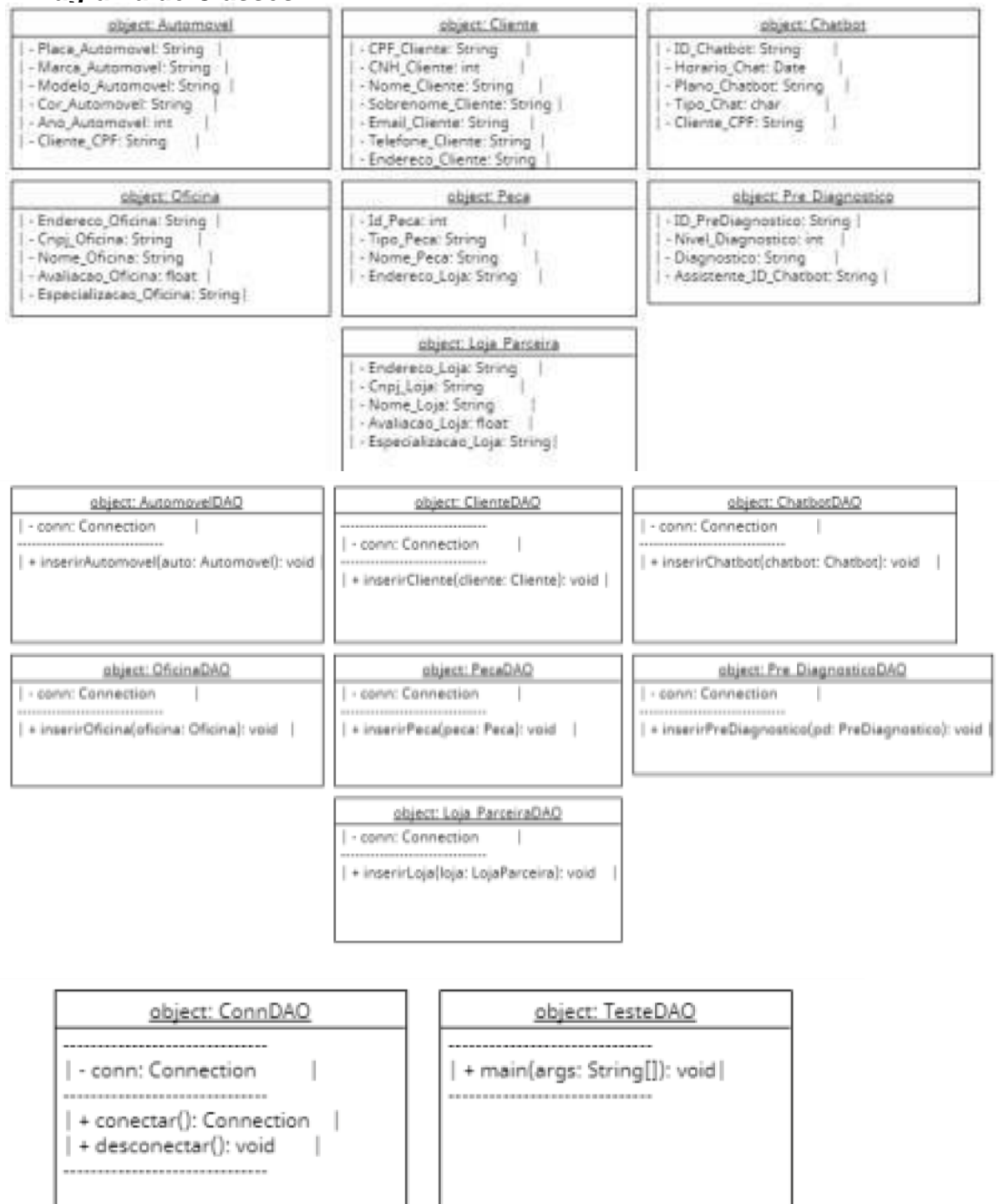
CONSTRAINT PK_Associacao PRIMARY KEY (CPF_Cliente, ID_Chatbot, Endereco_Loja),

```

CONSTRAINT FK\_Associacao\_Chatbot FOREIGN KEY (ID\_Chatbot, CPF\_Cliente)  
REFERENCES Chatbot(ID\_Chatbot, Cliente\_CPF\_Cliente),  
CONSTRAINT FK\_Associacao\_Loja FOREIGN KEY (Endereco\_Loja) REFERENCES  
Loja\_Parceira(Endereco\_Loja)  
);



#### 4- Diagrama de Classes



\*Feitos no UMLetino

