Faculdade de Informática e Administração Paulista

Domain Driving Using Java

Sprint 4

INTEGRANTES

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

Sumário

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.

- VO: Contém as classes que representam as entidades principais do sistema com seus atributos.

BO: Contém as classes que implementam a lógica de negócios. Elas são responsáveis por validar regras e aplicar lógica antes de interagir com o DAO.

DAO: Contém as classes que realizam operações de CRUD no banco de dados. Elas interagem diretamente com a base de dados usando ConnDAO.

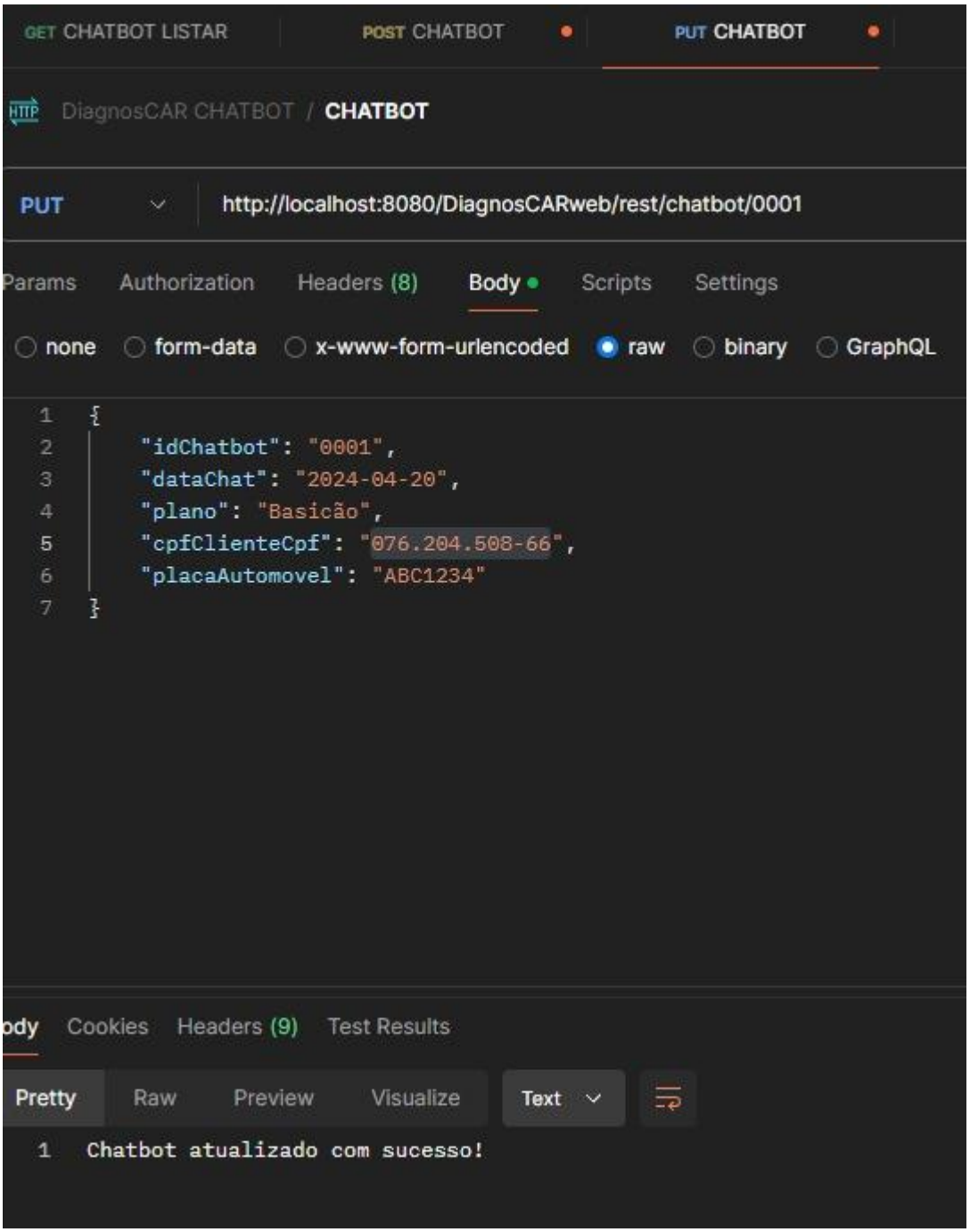
Resources: Contém arquivos de configuração e outros recursos necessários para o funcionamento do sistema

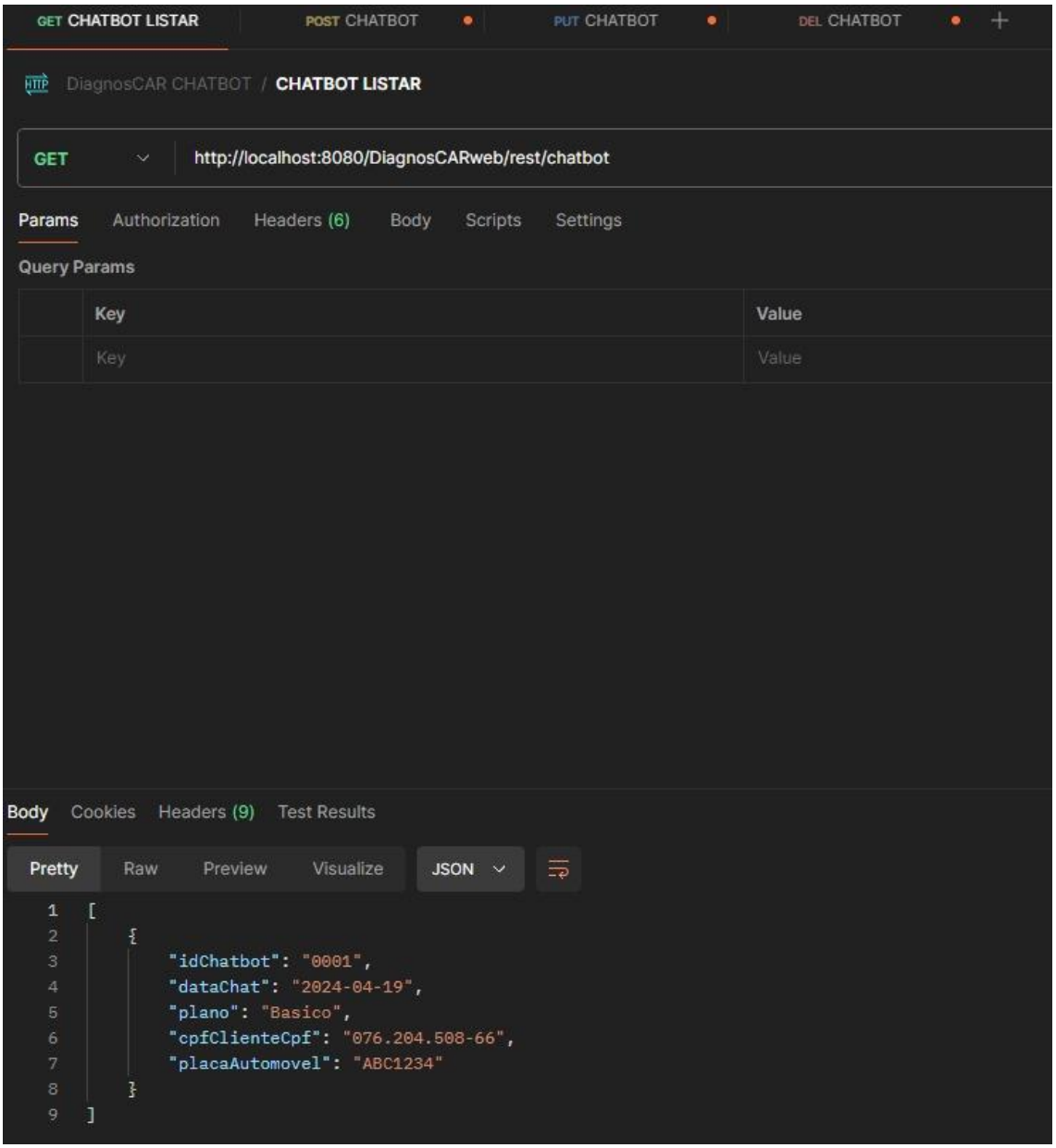
Connection: **ConnDAO**: Classe que gerencia a conexão com o banco de dados.

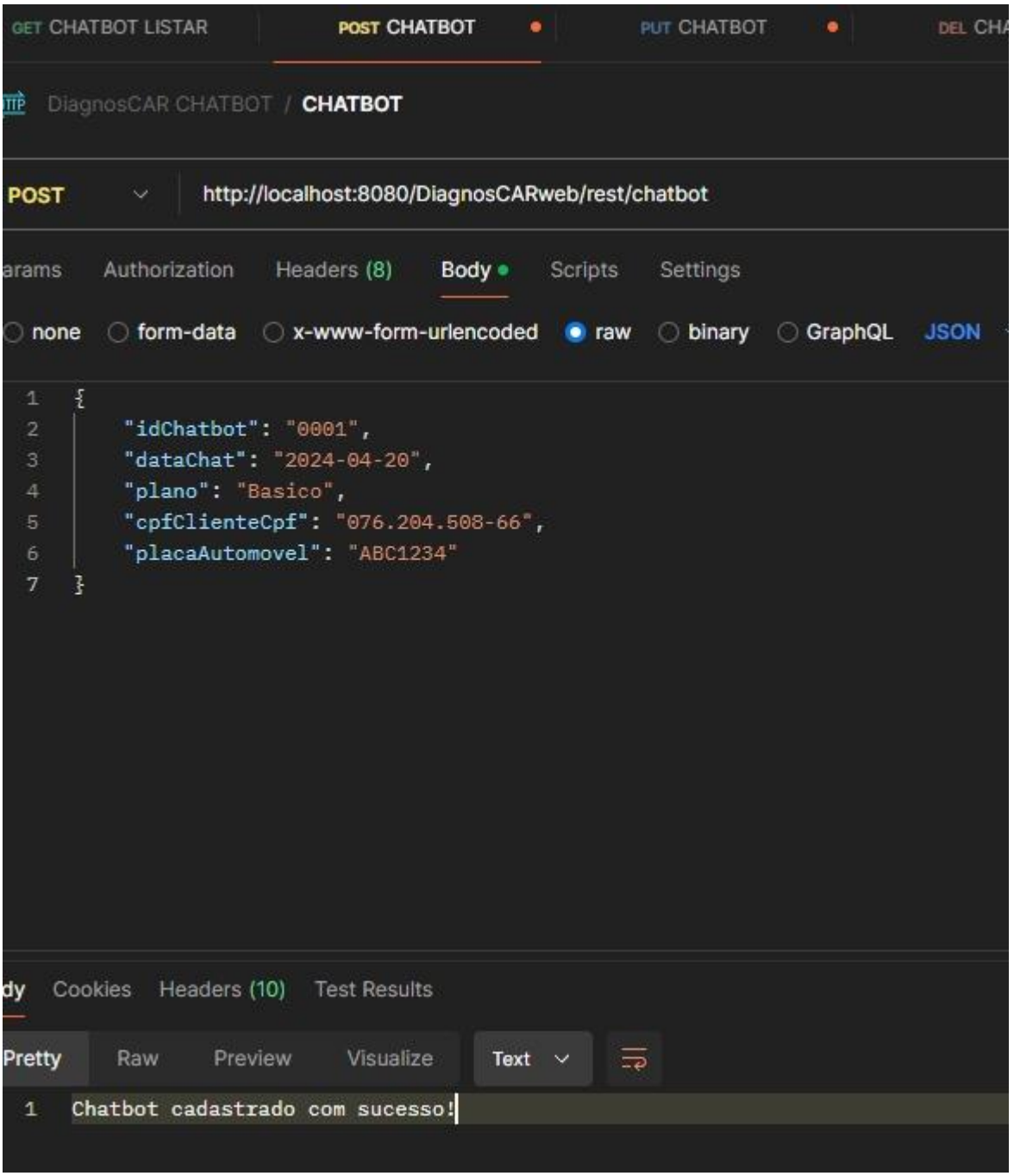
Breve Descrição das Principais Funcionalidades do Projeto

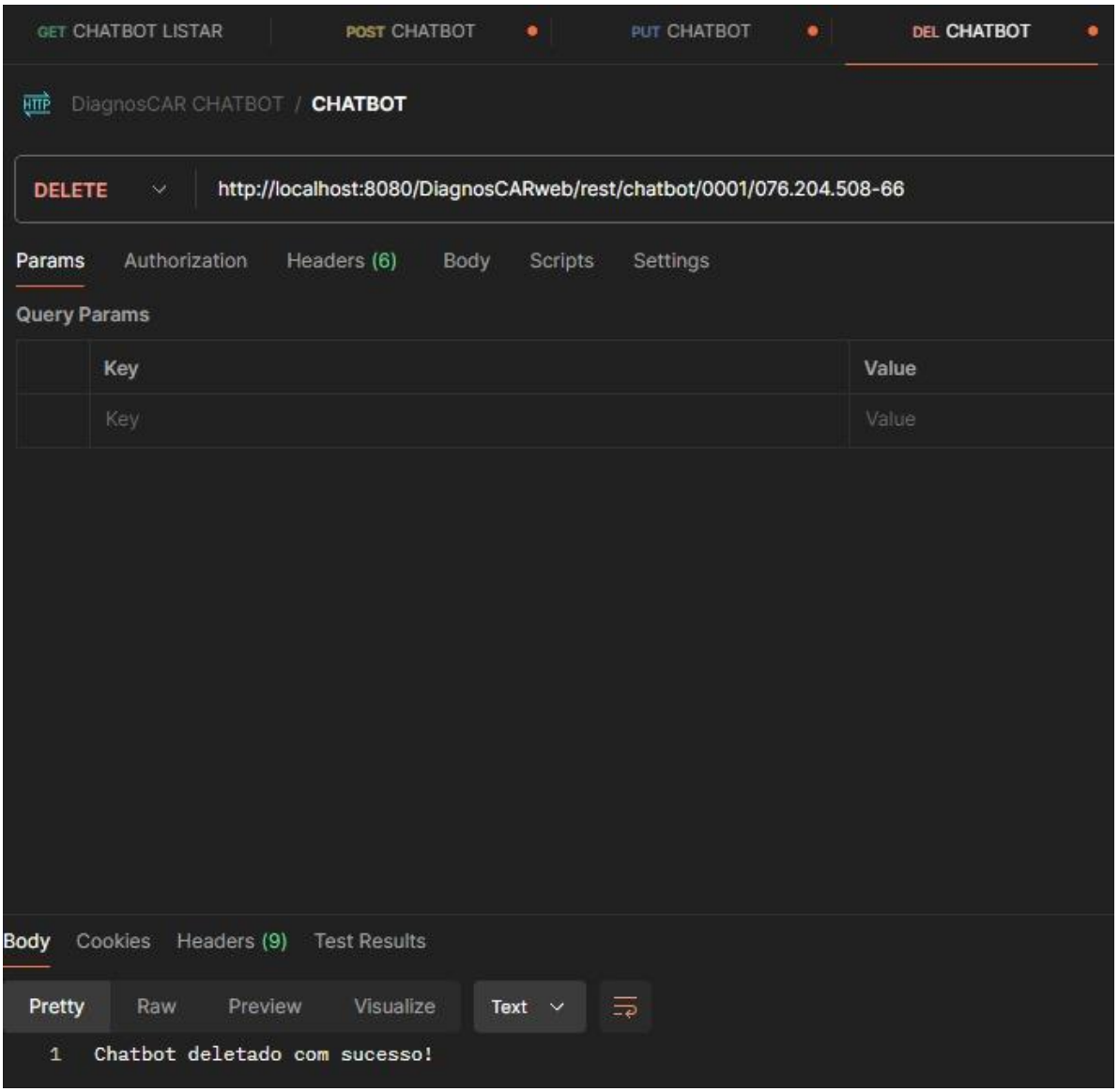
- **Gerenciamento de Clientes:** Cadastro, edição, exclusão e busca de clientes, armazenando informações como CPF, CNH, e dados de contato.
- **Gerenciamento de Automóveis:** Vinculação de veículos aos clientes, incluindo informações como placa, marca, modelo e ano.
- **Gerenciamento de Lojas Parceiras 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
Params
Authorization
Headers (6)
Body
Scripts
Settings
Query Params
Key
Value
Key
Value
Body
Cookies
Headers (9)
Test Results
Pretty
Raw
Preview
Visualize
JSON
[
{
"idChatbot": "0001",
"dataChat": "2024-04-19",
"plano": "Basico",
"cpfClienteCpf": "076.204.508-66",
"placaAutomovel": "ABC1234"
}
]


GET CHATBOT LISTAR
POST CHATBOT
PUT CHATBOT
DiagnosCAR CHATBOT / CHATBOT
POST
http://localhost:8080/DiagnosCARweb/rest/chatbot
Authorization
Headers (8)
Body
Scripts
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"idChatbot": "0001",
"dataChat": "2024-04-20",
"plano": "Basico",
"cpfClienteCpf": "076.204.508-66",
"placaAutomovel": "ABC1234"
}
Cookies
Headers (10)
Test Results
Pretty
Raw
Preview
Visualize
Text
Chatbot cadastrado com sucesso!


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!

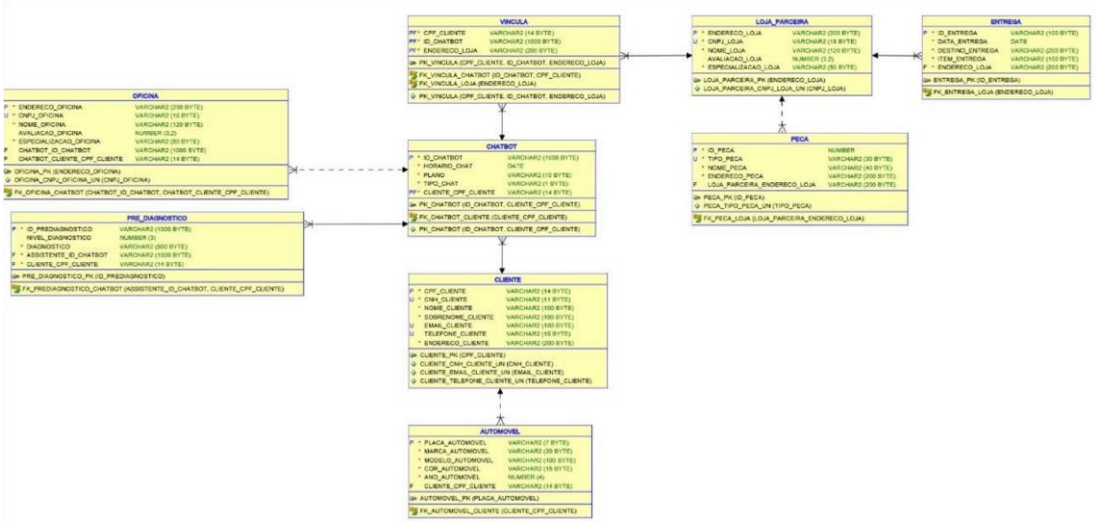
3- Modelo do Banco de Dados

```
CREATE TABLE Cliente (

   CPF_Cliente VARCHAR2(14) PRIMARY KEY NOT NULL,

   CNH_Cliente VARCHAR2(11) UNIQUE NOT NULL,

   Nome_Cliente VARCHAR2(100) NOT NULL,

   Sobrenome_Cliente VARCHAR2(100) NOT NULL,

   Email_Cliente VARCHAR2(100) UNIQUE,

   Telefone_Cliente VARCHAR2(15) UNIQUE,

   Endereco_Cliente VARCHAR2(200) NOT NULL,

   CONSTRAINT CHK_Cliente_CPF CHECK ( REGEXP_LIKE(CPF_Cliente,
'^[0-9]{3}\.[0-9]{3}\.[0-9]{3}-[0-9]{2}$')),

   CONSTRAINT CHK_Cliente_Telefone CHECK (REGEXP_LIKE(Telefone_Cliente,
'^\(\d{2}\) \d{4,5}-\d{4}$')),

   CONSTRAINT CHK_Cliente_Email CHECK ( REGEXP_LIKE(Email_Cliente, '^[A-
Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'))

) ;



CREATE TABLE Automovel (

   Placa_Automovel VARCHAR2(7) PRIMARY KEY NOT NULL,

   Marca_Automovel VARCHAR2(20) NOT NULL,

   Modelo_Automovel VARCHAR2(100) NOT NULL,

   Cor_Automovel VARCHAR2(15) NOT NULL,

   Ano_Automovel NUMBER(4) NOT NULL,

   Cliente_CPF_Cliente VARCHAR2(14),

   CONSTRAINT FK_Automovel_Cliente FOREIGN KEY ( Cliente_CPF_Cliente )
REFERENCES Cliente(CPF_Cliente)
) ;
```

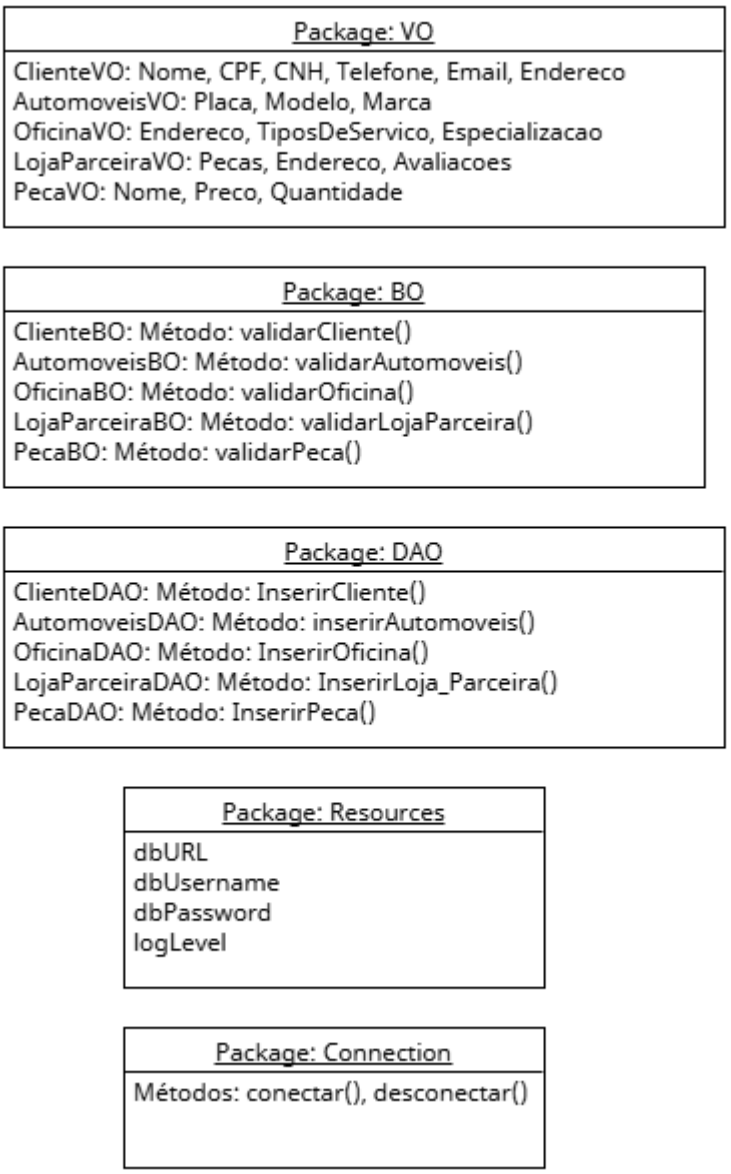
```
CREATE TABLE Chatbot(
   ID_Chatbot VARCHAR2(1000) NOT NULL,

   Horario_Chat DATE NOT NULL,

   Plano VARCHAR2(10) NOT NULL,

   Tipo_Chat VARCHAR2(1) NOT NULL,

   Cliente_CPF_Cliente VARCHAR2(14),

   CONSTRAINT PK_Chatbot PRIMARY KEY (ID_Chatbot, Cliente_CPF_Cliente),

   CONSTRAINT FK_Chatbot_Cliente FOREIGN KEY ( Cliente_CPF_Cliente )
REFERENCES Cliente(CPF_Cliente)

) ;



CREATE TABLE Pre_Diagnostico (

   ID_PreDiagnostico VARCHAR2(1000) PRIMARY KEY NOT NULL,

   Nivel_Diagnostico NUMBER(3) CHECK (Nivel_Diagnostico >= 1 AND
Nivel_Diagnostico <= 100) ,

   Diagnostico VARCHAR2(500) NOT NULL,

   Assistente_ID_Chatbot VARCHAR2(1000) NOT NULL,

   Cliente_CPF_Cliente VARCHAR2(14) NOT NULL,

   CONSTRAINT FK_PreDiagnostico_Chatbot FOREIGN KEY
(Assistente_ID_Chatbot, Cliente_CPF_Cliente)

      REFERENCES Chatbot(ID_Chatbot, Cliente_CPF_Cliente)

) ;



CREATE TABLE Loja_Parceira (

   Endereco_Loja VARCHAR2(200) PRIMARY KEY NOT NULL,
   Cnpj_Loja VARCHAR2(18) NOT NULL UNIQUE,

   Nome_Loja VARCHAR2(120) NOT NULL,

   Avaliacao_Loja NUMBER(3, 2) ,
```

```
   Especializacao_Loja VARCHAR2(50) NOT NULL,
   CONSTRAINT CHK_Loja_Cnpj CHECK ( REGEXP_LIKE(Cnpj_Loja,
'^[0-9]{2}\.[0-9]{3}\.[0-9]{3}/[0-9]{4}-[0-9]{2}$')),

   CONSTRAINT CHK_Loja_Avaliacao CHECK (Avaliacao_Loja BETWEEN 0 AND
10)

) ;



CREATE TABLE Oficina (

   Endereco_Oficina VARCHAR2(200) PRIMARY KEY NOT NULL,

   Cnpj_Oficina VARCHAR2(18) NOT NULL UNIQUE,

   Nome_Oficina VARCHAR2(120) NOT NULL,

   Avaliacao_Oficina NUMBER(3, 2) ,

   Especializacao_Oficina VARCHAR2(50) 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 CHK_Oficina_Cnpj CHECK ( REGEXP_LIKE(Cnpj_Oficina,
'^[0-9]{2}\.[0-9]{3}\.[0-9]{3}/[0-9]{4}-[0-9]{2}$')),

   CONSTRAINT CHK_Oficina_Avaliacao CHECK (Avaliacao_Oficina BETWEEN 0
AND 10)

) ;



CREATE TABLE Peca (
   ID_Peca NUMBER PRIMARY KEY NOT NULL,

   Tipo_Peca VARCHAR2(30) NOT NULL UNIQUE,

   Nome_Peca VARCHAR2(40) NOT NULL,

   Endereco_Peca VARCHAR2(200) 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) PRIMARY KEY NOT NULL,

    Data_Entrega DATE NOT NULL,

    Destino_Entrega VARCHAR2(200) NOT NULL,

    Item_Entrega VARCHAR2(100) NOT NULL,

    Endereco_Loja VARCHAR2(200) NOT NULL,

    CONSTRAINT FK_Entrega_Loja FOREIGN KEY (Endereco_Loja) REFERENCES
Loja_Parceira(Endereco_Loja)

) ;



CREATE TABLE Vincula (

    CPF_Cliente VARCHAR2(14) NOT NULL,

    ID_Chatbot VARCHAR2(1000) NOT NULL,

    Endereco_Loja VARCHAR2(200),

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

    CONSTRAINT FK_Vincula_Chatbot FOREIGN KEY (CPF_Cliente, ID_Chatbot)
REFERENCES Chatbot(Cliente_CPF_Cliente, ID_Chatbot),
    CONSTRAINT FK_Vincula_Loja FOREIGN KEY (Endereco_Loja) REFERENCES
Loja_Parceira(Endereco_Loja)

) ;
```


VINCULA
CPF_CLIENTE VARCHAR2 (14 BYTE)
ID_CHATBOT VARCHAR2 (1000 BYTE)
ENDERECO_LOJA VARCHAR2 (200 BYTE)
PK_VINCULA (CPF_CLIENTE, ID_CHATBOT, ENDERECO_LOJA)
FK_VINCULA_CHATBOT (ID_CHATBOT, CPF_CLIENTE)
FK_VINCULA_LOJA (ENDERECO_LOJA)
LOJA_PARCEIRA
ENDERECO_LOJA VARCHAR2 (200 BYTE)
CNPJ_LOJA VARCHAR2 (18 BYTE)
NOME_LOJA VARCHAR2 (120 BYTE)
AVALIACAO_LOJA NUMBER (3,2)
ESPECIALIZACAO_LOJA VARCHAR2 (50 BYTE)
LOJA_PARCEIRA_PK (ENDERECO_LOJA)
LOJA_PARCEIRA_CNPJ_LOJA_UN (CNPJ_LOJA)
ENTREGA
ID_ENTREGA VARCHAR2 (100 BYTE)
DATA_ENTREGA DATE
DESTINO_ENTREGA VARCHAR2 (200 BYTE)
ITEM_ENTREGA VARCHAR2 (100 BYTE)
ENDERECO_LOJA VARCHAR2 (200 BYTE)
ENTREGA_PK (ID_ENTREGA)
FK_ENTREGA_LOJA (ENDERECO_LOJA)
OFICINA
ENDERECO_OFICINA VARCHAR2 (200 BYTE)
CNPJ_OFICINA VARCHAR2 (18 BYTE)
NOME_OFICINA VARCHAR2 (120 BYTE)
AVALIACAO_OFICINA NUMBER (3,2)
ESPECIALIDADE_OFICINA VARCHAR2 (50 BYTE)
CHATBOT_ID_CHATBOT VARCHAR2 (1000 BYTE)
CHATBOT_CLIENTE_CPF_CLIENTE VARCHAR2 (14 BYTE)
OFICINA_PK (ENDERECO_OFICINA)
OFICINA_CNPJ_OFICINA_UN (CNPJ_OFICINA)
FK_OFICINA_CHATBOT (CHATBOT_ID_CHATBOT, CHATBOT_CLIENTE_CPF_CLIENTE)
CHATBOT
ID_CHATBOT VARCHAR2 (1000 BYTE)
HORARIO_CHAT DATE
PLANO VARCHAR2 (10 BYTE)
TIPO_CHAT VARCHAR2 (1 BYTE)
CLIENTE_CPF_CLIENTE VARCHAR2 (14 BYTE)
PK_CHATBOT (ID_CHATBOT, CLIENTE_CPF_CLIENTE)
FK_CHATBOT_CLIENTE (CLIENTE_CPF_CLIENTE)
PK_CHATBOT (ID_CHATBOT, CLIENTE_CPF_CLIENTE)
PECA
ID_PECA NUMBER
TIPO_PECA VARCHAR2 (30 BYTE)
NOME_PECA VARCHAR2 (40 BYTE)
ENDERECO_PECA VARCHAR2 (200 BYTE)
LOJA_PARCEIRA_ENDERECO_LOJA VARCHAR2 (200 BYTE)
PECA_PK (ID_PECA)
PECA_TIPO_PECA_UN (TIPO_PECA)
FK_PECA_LOJA (LOJA_PARCEIRA_ENDERECO_LOJA)
PRE_DIAGNOSTICO
ID_PREDIAGNOSTICO VARCHAR2 (1000 BYTE)
NIVEL_DIAGNOSTICO NUMBER (3)
DIAGNOSTICO VARCHAR2 (500 BYTE)
ASSISTENTE_ID_CHATBOT VARCHAR2 (1000 BYTE)
CLIENTE_CPF_CLIENTE VARCHAR2 (11 BYTE)
PRE_DIAGNOSTICO_PK (ID_PREDIAGNOSTICO)
FK_PREDIAGNOSTICO_CHATBOT (ASSISTENTE_ID_CHATBOT, CLIENTE_CPF_CLIENTE)
CLIENTE
CPF_CLIENTE VARCHAR2 (14 BYTE)
CNH_CLIENTE VARCHAR2 (11 BYTE)
NOME_CLIENTE VARCHAR2 (100 BYTE)
SOBRENOME_CLIENTE VARCHAR2 (100 BYTE)
EMAIL_CLIENTE VARCHAR2 (150 BYTE)
TELEFONE_CLIENTE VARCHAR2 (15 BYTE)
ENDERECO_CLIENTE VARCHAR2 (200 BYTE)
CLIENTE_PK (CPF_CLIENTE)
CLIENTE_CNH_CLIENTE_UN (CNH_CLIENTE)
CLIENTE_EMAIL_CLIENTE_UN (EMAIL_CLIENTE)
CLIENTE_TELEFONE_CLIENTE_UN (TELEFONE_CLIENTE)
AUTOMOVEL
PLACA_AUTOMOVEL VARCHAR2 (7 BYTE)
MARCA_AUTOMOVEL VARCHAR2 (30 BYTE)
MODELO_AUTOMOVEL VARCHAR2 (100 BYTE)
COR_AUTOMOVEL VARCHAR2 (15 BYTE)
ANO_AUTOMOVEL NUMBER (4)
CLIENTE_CPF_CLIENTE VARCHAR2 (14 BYTE)
AUTOMOVEL_PK (PLACA_AUTOMOVEL)
FK_AUTOMOVEL_CLIENTE (CLIENTE_CPF_CLIENTE)

4 – Diagrama de classes



***Feito no UMLetino**