



Faculdade de Informática e Administração Paulista

# **Building Relational Database**

**PortoAutoTech**

**Sprint 2**

# INTEGRANTES

RM (SOMENTE NÚMEROS)	NOME COMPLEMENTO (SEM ABREVIAR)
558981	Giovanna Revito Roz
558986	Kaian Gustavo de Oliveira Nascimento
554424	Lucas Kenji Kikuchi



**SUMÁRIO**

1 – DESCRIÇÃO DO PROJETO .....	5
2 – Descrição dos dados/atributos e Regras de Negócio.....	7
3 – Descrição das Tabelas.....	11

## 1 – Descrição do Projeto

Somos o sistema PortoAutoTech, uma aplicação web, que visa a implementação de um sistema para auxiliar clientes que possuam pouco ou nenhum conhecimento sobre automotivos, que não tem tempo para procurar bom mecânico com preço acessível para resolver seu problema. Um sistema tecnológico e inovador, que resolve de maneira mais fácil. Com mais precisão. Mais rapidez. Mais facilidade. Tanto para os assegurados como os mais novos usuários/clientes da Porto Seguro. Através do nosso sistema é possível que o usuário, ao inicializar o serviço selecione a marca do veículo e com a filtragem de perguntas, o sistema identifique o carro. Iniciando o diagnóstico são feitas perguntas para que a IA realize um diagnóstico e mostre a probabilidade de ser  $n$  problema. Com isso, o orçamento é exibido, mostrando a peça que precisa para o veículo, a unidade mais próxima para o conserto, o tempo até o local e estimativa de finalização.

Nosso objetivo é a criação de que até o fim do projeto, possamos ter um sistema que permita ao cliente ter:

- Um autodiagnóstico do problema apresentado pelo veículo através de uma I.A treinada através de Machine Learning em Python, incorporando o Chatbot para o diagnóstico.
- A possibilidade de se cadastrar no sistema com suas informações pessoais (CPF, nome, telefone, Email...etc.) e informações de seu veículo (como modelo, ano, marca) através de nosso website;
- Informações sobre o orçamento e a estimativa de prazo de término do serviço definidos com antecedência;
- Um pré-orçamento, com um cálculo baseado no valor do serviço (mão-de-obra + custos adicionais) + valor das peças, e a estimativa de prazo de término do serviço definido com antecedência;
- Através da sua localidade identificar e fazer o direcionamento de oficinas mais próximas da Porto ou de empresas parceiras, informando a disponibilidade de peças e agendamento de determinados serviços com a integração de um banco de dados da oficina;
- Notificação automática de manutenção preventiva, baseada na última vez que o cliente agendou um serviço e estimativas de futuras manutenções usando a quilometragem;
- notificação automática da disponibilidade de uma determinada peça, que será acionada quando a peça for registrada como disponível no banco de dados.
- interface que informa os pontos mais próximos de carregamento para carros elétricos;
- Um site responsivo e integrado com o banco de dados e a IA compondo um sistema.

- Progresso da manutenção, visualizada através de uma barra de progresso, indicando cada mudança importante sobre o status da manutenção, manualmente alterado pelo mecânico através de um sistema;
- Ligação em chamada com mecânico (opcional), explicando o progresso da manutenção.
- Processo de verificação das credenciais do cadastro que um usuário fornece com a autenticação
- Comunicação através de voz (speech-to-text) com o Chatbot, permitindo descrever o problema oralmente ou enviar ruídos emitidos pelo carro, que serão analisados pela I.A;

## 2 – Descrição dos dados/atributos e Regras de Negócio apuradas e Relacionadas as Entidades.

Logo abaixo, estão descritos os atributos de cada entidade, e suas possíveis soluções:

### Entidade: diagnostico

- **id\_diagnostico**: atributo identificador do diagnóstico realizado. Através dele, é possível ter os dados do diagnóstico;
- **descricao\_sintoma**: descrição dada pelo usuário sobre os sintomas no carro;
- **categoria\_problema**: identifica a categoria do problema. Ex: problema no câmbio, problema na bateria etc.
- **solucao**: armazena a solução para o problema identificado;
- **status\_diagnostico**: indica a gravidade do problema.

### Entidade: agendamento

- **id\_agendamento**: chave primaria única para identificar o agendamento;
- **data\_agendamento**: armazena a data de agendamento do serviço;
- **horário\_agendamento**: guarda o horário de agendamento do serviço. Ex: 08:00;
- **descricao\_agendamento**: armazena uma breve descrição sobre o agendamento.

### Entidade: usuario

- **cpf\_usuario**: atributo identificador de cada usuário. O CPF único ajuda a rastrear todas as ações, informações, diagnósticos e veículos. Conectando suas “pegadas”, atribuindo a um só;

- **nome\_usuario:** nome escolhido pelo cliente, usado para login e identificação.

Deverão ser únicos;

- **email:** um endereço de e-mail válido associado com a conta do usuário, usado para comunicação, identificação e para propósitos de autenticação;

- **telefone:** número de telefone associado a conta do usuário, onde poderá ser usado para comunicação do Centro Automotivo, mecânico e do Usuário quando necessário;

- **senha:** usada para acessar a conta do usuário.

### Entidade: veículo

- **placa:** atributo identificador de cada veículo, usado para rastrear, gerenciar os veículos e seu histórico.

- **marca:** usado para guardar a do veículo;

- **modelo:** indica o modelo do veículo;

- **ano:** salva o ano de produção do veículo. Ajuda a identificar o veículo, se relacionando com o front-end para a seleção dele;

- **quilometragem:** indica a quilometragem do veículo. Pode ajudar no diagnóstico e para diagnósticos de manutenção preventiva.

### Entidade: orcamento

- **id\_orcamento:** atributo identificador para cada orçamento, usado para rastrear e gerenciar histórico de orçamentos relacionados a cada usuário, diagnóstico etc. Também usado para organizar os gastos baseado na sua descrição, natureza e propósito.

- **descricao\_orcamento:** uma descrição do orçamento, contendo o contexto, e informações relacionadas ao diagnóstico, problema, solução, e gastos esperados.

- **valor\_total:** o total do valor do orçamento que precisará ser alocado no reparo. Podendo ajudar a rastrear os gastos;



- **status\_orcamento:** data do status do reparo, se foi iniciado, alterado, completado, permitindo o ajuste do orçamento (se precisará alteração do custo, dependendo de mais, ou menos problemas identificados).

#### Entidade: **peca**

- **id\_peca:** atributo identificador usado para identificar cada peça de reposição.
- **nome\_peca:** nome descritivo da peça, podendo ajudar a identificar seu propósito.
- **preco\_peca:** o preço associado a peça de reposição, o que pode ajudar a calcular o orçamento, baseado no diagnóstico e mão de obra.
- **disponibilidade\_peca:** identifica a disponibilidade da peça, se ela se encontra em estoque ou em falta;

#### Entidade: **servico**

- **tipo\_servico:** descreve a qual tipo o serviço pertence, seja freio, suspensão etc.;
- **id\_servico:** chave primaria do serviço;
- **descricao\_servico:** descrição detalhada do serviço;
- **preco\_servico:** quanto custará o serviço;
- **duracao:** indica a duração estimada para o serviço ser concluído (em minutos).

#### Entidade: **centro\_automotivo**

- **id\_centro:** chave primária do centro automotivo;
- **nome\_centro:** indica o nome da unidade do centro automotivo;
- **endereco\_centro:** armazena o endereço do centro automotivo;
- **telefone\_centro:** armazena o telefone de contato do centro automotivo;

- **horário\_funcionamento**: horário de funcionamento do centro automotivo. Ex: Segunda a sábado, das 9h às 17h etc.

#### Entidade: cargo

- **id\_cargo**: chave primária do cargo;
- **nome\_cargo**: contém o nome do cargo, por exemplo auxiliar, gerente, mecânico etc.;
- **area\_cargo**: indica a área do trabalho, por exemplo gerência, auxiliar, administrativo etc.;
- **descricao\_cargo**: descrição das funções associado ao cargo;
- **salario\_cargo**: armazena o salário relacionado ao cargo.

#### Entidade: funcionario

- **matricula\_func**: chave primária do funcionário;
- **nome\_func**: indica o nome do funcionário. Pode conter um ou mais nomes, dependendo da convenção adotada pela organização;
- **horário\_trabalho**: indica o horário em que o funcionário trabalha. Pode ser representado de diferentes formas, como um intervalo de horas ou um conjunto de dias da semana;
- **disponibilidade\_func**: indica a disponibilidade do funcionário para o serviço no Centro Automotivo, se está disponível ou não.

#### Regras de negócio:

- Cada usuário pode ter um ou mais veículos vinculados a ele;
- O diagnóstico só existe caso um veículo, um serviço e um orçamento estejam presentes;
- O orçamento é calculado com base no diagnóstico;

- O agendamento só é feito caso o centro automotivo, o serviço e o veículo estejam presentes;
- Um veículo pode ter múltiplos agendamentos vinculados a ele;
- Um veículo pode ter vários diagnósticos;
- Um funcionário só pode pertencer a um centro automotivo, mas um centro automotivo pode ter vários funcionários;
- Um funcionário pertence a apenas um cargo, mas um cargo tem vários funcionários;
- Um centro automotivo pode oferecer vários serviços, e um serviço pode ser oferecido em vários centros;
- Uma peça é fornecida em vários serviços, e um serviço pode usar uma ou várias peças;

### 3 – Descrição das Tabelas.

#### 1. Tabela usuário

```
CREATE TABLE usuario (  
  cpf_usuario CHAR(11) CONSTRAINT usuario_cpf_pk PRIMARY KEY,  
  nome_usuario VARCHAR(80) CONSTRAINT usuario_nm_nn NOT NULL CONSTRAINT usuario_nm_unique UNIQUE,  
  email VARCHAR(255) CONSTRAINT usuario_mail_nn NOT NULL,  
  telefone CHAR(11) CONSTRAINT usuario_tel_nn NOT NULL,  
  senha VARCHAR(30) CONSTRAINT usuario_sen_nn NOT NULL,  
  CONSTRAINT chk_senha_usuario CHECK (LENGTH(senha) > 6)  
);
```

- **Descrição:** Armazena os dados dos usuários cadastrados no sistema, que podem ser clientes assegurados ou pessoas que utilizam o sistema.
- **Colunas:**
  - cpf\_usuario: CPF do usuário (Primary Key).
  - nome\_usuario: Nome completo do usuário (único e obrigatório).
  - email: E-mail do usuário (obrigatório).

- telefone: Telefone do usuário (obrigatório).
- senha: Senha do usuário, com uma verificação para ter mais de 6 caracteres.
- **Chave Primária:** cpf\_usuario.

## 2. Tabela veiculo

```
CREATE TABLE veiculo (  
  placa CHAR(7) CONSTRAINT veic_pl_pk PRIMARY KEY,  
  marca VARCHAR(50) CONSTRAINT veic_mrc_nn NOT NULL,  
  modelo VARCHAR(50) CONSTRAINT veic_mdl_nn NOT NULL,  
  ano NUMBER(4) CONSTRAINT veic_ano_nn NOT NULL,  
  quilometragem NUMBER(10,2) CONSTRAINT veic_quil_nn NOT NULL,  
  CONSTRAINT chk_quil_veiculo CHECK (quilometragem > 0),  
  usuario_cpf_usuario CHAR(11) CONSTRAINT veic_cpf_fk REFERENCES usuario(cpf_usuario) ON DELETE CASCADE  
);
```

**Descrição:** Armazena informações sobre os veículos cadastrados no sistema, vinculados aos seus respectivos usuários.

- **Colunas:**
  - placa: Placa do veículo (Primary Key).
  - marca: Marca do veículo.
  - modelo: Modelo do veículo.
  - ano: Ano de fabricação do veículo.
  - quilometragem: Quilometragem atual do veículo (deve ser maior que 0).
  - usuario\_cpf\_usuario: CPF do usuário proprietário do veículo (Foreign Key, referenciando usuario).
- **Chave Primária:** placa.
- **Chave Estrangeira:** usuario\_cpf\_usuario referenciando usuario(cpf\_usuario).

### 3. Tabela orçamento

```
CREATE TABLE orcamento (  
  id_orcamento CHAR(36) CONSTRAINT orc_id_pk PRIMARY KEY,  
  descricao_orcamento VARCHAR(255) CONSTRAINT orc_desc_nn NOT NULL,  
  valor_total NUMBER(9,2) CONSTRAINT orc_vt_nn NOT NULL,  
  status_orcamento VARCHAR(50) CONSTRAINT orc_stts_nn NOT NULL,  
  CONSTRAINT chk_status_orcamento CHECK (status_orcamento IN ('PENDENTE', 'APROVADO', 'REJEITADO'))  
);
```

- **Descrição:** Armazena os orçamentos gerados para os serviços solicitados pelos usuários.
- **Colunas:**
  - id\_orcamento: Identificador único do orçamento (Primary Key).
  - descricao\_orcamento: Descrição detalhada do orçamento.
  - valor\_total: Valor total do orçamento.
  - status\_orcamento: Status do orçamento (PENDENTE, APROVADO, REJEITADO).
- **Chave Primária:** id\_orcamento.
- **Check Constraint:** Verifica que o status do orçamento está entre as opções válidas ('PENDENTE', 'APROVADO', 'REJEITADO').

### 4. Tabela serviço

```
CREATE TABLE servico (  
  id_servico CHAR(6) CONSTRAINT serv_id_pk PRIMARY KEY,  
  tipo_servico VARCHAR(50) CONSTRAINT serv_tp_nn NOT NULL,  
  descricao_servico VARCHAR(255) CONSTRAINT serv_desc_nn NOT NULL,  
  preco_servico NUMBER(9,2) CONSTRAINT serv_prec_nn NOT NULL,  
  duracao NUMBER(4) CONSTRAINT serv_dur_nn NOT NULL  
);
```

- **Descrição:** Armazena informações sobre os serviços disponíveis no sistema.
- **Colunas:**
  - id\_servico: Identificador único do serviço (Primary Key).
  - tipo\_servico: Tipo de serviço (ex.: troca de óleo, revisão).
  - descricao\_servico: Descrição do serviço oferecido.

- preco\_servico: Preço do serviço.
- duracao: Duração estimada do serviço em minutos.
- **Chave Primária:** id\_servico.

## 5. Tabela peca

```
CREATE TABLE peca (  
  id_pecas CHAR(6) CONSTRAINT peca_id_pk PRIMARY KEY,  
  disponibilidade_pecas NUMBER(8) CONSTRAINT peca_disp_nn NOT NULL,  
  CONSTRAINT chk_disponibilidade_pecas CHECK (disponibilidade_pecas > 0),  
  nome_pecas VARCHAR(255) CONSTRAINT peca_nm_nn NOT NULL CONSTRAINT peca_nm_unique UNIQUE,  
  preco_pecas NUMBER(9,2) CONSTRAINT peca_pc_nn NOT NULL,  
  CONSTRAINT chk_preco_pecas CHECK (preco_pecas > 0)  
);
```

- **Descrição:** Armazena dados sobre as peças disponíveis para os serviços.
- **Colunas:**
  - id\_pecas: Identificador único da peça (Primary Key).
  - disponibilidade\_pecas: Quantidade disponível da peça (deve ser maior que 0).
  - nome\_pecas: Nome da peça (único e obrigatório).
  - preco\_pecas: Preço da peça (deve ser maior que 0).
- **Chave Primária:** id\_pecas.
- **Check Constraints:** Verifica se a quantidade de peças e o preço são maiores que 0.

## 6. Tabela centro\_automotivo

```
CREATE TABLE centro_automotivo (  
  id_centro CHAR(4) CONSTRAINT centro_id_pk PRIMARY KEY,  
  nome_centro VARCHAR(155) CONSTRAINT centro_nm_nn NOT NULL,  
  endereco_centro VARCHAR(255) CONSTRAINT centro_end_nn NOT NULL,  
  CONSTRAINT chk_end_centro CHECK (LENGTH(endereco_centro) > 20),  
  telefone_centro CHAR(11) CONSTRAINT centro_tel_nn NOT NULL,  
  horario_funcionamento VARCHAR(70) CONSTRAINT centro_hr_func_nn NOT NULL  
);
```

- **Descrição:** Contém as informações dos centros automotivos parceiros.

- **Colunas:**

- id\_centro: Identificador único do centro automotivo (Primary Key).
- nome\_centro: Nome do centro automotivo (obrigatório).
- endereco\_centro: Endereço completo do centro automotivo (com um mínimo de 20 caracteres).
- telefone\_centro: Telefone de contato do centro.
- horario\_funcionamento: Horário de funcionamento do centro automotivo.

- **Chave Primária:** id\_centro.

## 7. Tabela cargo

```
CREATE TABLE cargo (  
    id_cargo CHAR(4) CONSTRAINT cargo_id_pk PRIMARY KEY,  
    nome_cargo VARCHAR(50) CONSTRAINT cargo_nm_nn NOT NULL,  
    area_cargo VARCHAR(50) CONSTRAINT cargo_area_nn NOT NULL,  
    descricao_cargo VARCHAR(255) CONSTRAINT cargo_desc_nn NOT NULL,  
    salario_cargo NUMBER(9,2) CONSTRAINT cargo_sal_nn NOT NULL CONSTRAINT cargo_sal_unique UNIQUE,  
    CONSTRAINT chk_salario_cargo CHECK (salario_cargo > 0)  
);
```

- **Descrição:** Armazena as informações sobre os cargos dos funcionários nos centros automotivos.

- **Colunas:**

- id\_cargo: Identificador único do cargo (Primary Key).
- nome\_cargo: Nome do cargo (ex.: mecânico, supervisor).
- area\_cargo: Área de atuação do cargo.
- descricao\_cargo: Descrição detalhada do cargo.
- salario\_cargo: Salário associado ao cargo (deve ser maior que 0).

- **Chave Primária:** id\_cargo.

- **Check Constraint:** Verifica se o salário é maior que 0.

## 8. Tabela funcionário

```
CREATE TABLE funcionario (
    matricula_func CHAR(6) CONSTRAINT func_matric_pk PRIMARY KEY,
    nome_func VARCHAR(70) CONSTRAINT func_nm_nn NOT NULL,
    horario_trabalho VARCHAR(70) CONSTRAINT func_hr_nn NOT NULL,
    disponibilidade_func CHAR(1) CONSTRAINT func_disp_nn NOT NULL CHECK (disponibilidade_func IN ('S', 'N')),
    centro_automotivo_id_centro CHAR(4) CONSTRAINT func_id_centro_fk REFERENCES centro_automotivo(id_centro) ON DELETE SET NULL,
    cargo_id_cargo CHAR(4) CONSTRAINT func_id_cargo_fk REFERENCES cargo(id_cargo) ON DELETE SET NULL
);
```

- **Descrição:** Armazena os dados dos funcionários dos centros automotivos.
- **Colunas:**
  - **matricula\_func:** Matrícula única do funcionário (Primary Key).
  - **nome\_func:** Nome do funcionário.
  - **horario\_trabalho:** Horário de trabalho do funcionário.
  - **disponibilidade\_func:** Disponibilidade do funcionário ('S' para sim, 'N' para não).
  - **centro\_automotivo\_id\_centro:** Identificador do centro automotivo onde o funcionário trabalha (Foreign Key referenciando centro\_automotivo).
  - **cargo\_id\_cargo:** Identificador do cargo do funcionário (Foreign Key referenciando cargo).
- **Chave Primária:** matricula\_func.
- **Chaves Estrangeiras:**
  - **centro\_automotivo\_id\_centro** referenciando centro\_automotivo(id\_centro).
  - **cargo\_id\_cargo** referenciando cargo(id\_cargo).

## 9. Tabela diagnostico

```
CREATE TABLE diagnostico (
    id_diagnostico CHAR(36) CONSTRAINT diag_id_pk PRIMARY KEY,
    descricao_sintomas VARCHAR(255) CONSTRAINT diag_desc_nn NOT NULL,
    categoria_problema VARCHAR(50) CONSTRAINT diag_cat_nn NOT NULL,
    solucao VARCHAR(255) CONSTRAINT diag_sol_nn NOT NULL CONSTRAINT diag_sol_unique UNIQUE,
    status_diagnostico VARCHAR(50) CONSTRAINT diag_stt_nn NOT NULL,
    CONSTRAINT chk_status_diagnostico CHECK (status_diagnostico IN ('EM ANÁLISE', 'ANALISADO')),
    veiculo_placa CHAR(7) CONSTRAINT diag_veiculo_fk REFERENCES veiculo(placa) ON DELETE CASCADE,
    orcamento_id_orcamento CHAR(36) CONSTRAINT diag_orcamento_fk REFERENCES orcamento(id_orcamento) ON DELETE CASCADE,
    servico_id_servico CHAR(6) CONSTRAINT diag_servico_fk REFERENCES servico(id_servico) ON DELETE CASCADE
);
```



- **Descrição:** Armazena os diagnósticos realizados nos veículos.
- **Colunas:**
  - id\_diagnostico: Identificador único do diagnóstico (Primary Key).
  - descricao\_sintomas: Descrição dos sintomas observados no veículo.
  - categoria\_problema: Categoria do problema diagnosticado.
  - solucao: Solução proposta para o problema.
  - status\_diagnostico: Status do diagnóstico (EM ANÁLISE, ANALISADO).
  - veiculo\_placa: Placa do veículo relacionado ao diagnóstico (Foreign Key referenciando veiculo).
  - orcamento\_id\_orcamento: Identificador do orçamento associado (Foreign Key referenciando orcamento).
  - servico\_id\_servico: Identificador do serviço associado (Foreign Key referenciando servico).
- **Chave Primária:** id\_diagnostico.
- **Chaves Estrangeiras:**
  - veiculo\_placa referenciando veiculo(placa).
  - orcamento\_id\_orcamento referenciando orcamento(id\_orcamento).
  - servico\_id\_servico referenciando servico(id\_servico).

## 10. Tabela agendamento

```
CREATE TABLE agendamento (
  id_agendamento CHAR(36) CONSTRAINT agend_id_pk PRIMARY KEY,
  data_agendamento DATE CONSTRAINT agend_dt_nn NOT NULL,
  horario_agendamento CHAR(5) CONSTRAINT agend_hr_nn NOT NULL,
  descricao_agendamento VARCHAR(255) CONSTRAINT agend_desc_nn NOT NULL,
  servico_id_servico CHAR(6) CONSTRAINT agend_id_serv_fk REFERENCES servico(id_servico) ON DELETE CASCADE,
  centro_automotivo_id_centro CHAR(4) CONSTRAINT agend_id_centro_fk REFERENCES centro_automotivo(id_centro) ON DELETE CASCADE,
  veiculo_placa CHAR(7) CONSTRAINT agend_veiculo_fk REFERENCES veiculo(placa) ON DELETE CASCADE
);
```

- **Descrição:** Armazena os agendamentos feitos pelos usuários para serviços nos centros automotivos.
- **Colunas:**

- id\_agendamento: Identificador único do agendamento (Primary Key).
  - data\_agendamento: Data do agendamento.
  - horario\_agendamento: Horário do agendamento.
  - descricao\_agendamento: Descrição do serviço agendado.
  - servico\_id\_servico: Identificador do serviço agendado (Foreign Key referenciando servico).
  - centro\_automotivo\_id\_centro: Identificador do centro automotivo onde o serviço será realizado (Foreign Key referenciando centro\_automotivo).
  - veiculo\_placa: Placa do veículo relacionado ao agendamento (Foreign Key referenciando veiculo).
- **Chave Primária:** id\_agendamento.
  - **Chaves Estrangeiras:**
    - servico\_id\_servico referenciando servico(id\_servico).
    - centro\_automotivo\_id\_centro referenciando centro\_automotivo(id\_centro).
    - veiculo\_placa referenciando veiculo(placa).

## 11. Tabela oferece

```
CREATE TABLE oferece (
  servico_id_servico CHAR(6),
  centro_automotivo_id_centro CHAR(4),
  CONSTRAINT oferece_pk PRIMARY KEY (servico_id_servico, centro_automotivo_id_centro),
  CONSTRAINT oferece_servico_fk FOREIGN KEY (servico_id_servico) REFERENCES servico(id_servico) ON DELETE CASCADE,
  CONSTRAINT oferece_id_centro_fk FOREIGN KEY (centro_automotivo_id_centro) REFERENCES centro_automotivo(id_centro) ON DELETE CASCADE
);
```

- **Descrição:** Relaciona os serviços oferecidos pelos centros automotivos.
- **Colunas:**
  - servico\_id\_servico: Identificador do serviço (Primary Key, Foreign Key referenciando servico).
  - centro\_automotivo\_id\_centro: Identificador do centro automotivo (Primary Key, Foreign Key referenciando centro\_automotivo).
- **Chave Primária Composta:** (servico\_id\_servico, centro\_automotivo\_id\_centro).

## 12. Tabela fornece

```
CREATE TABLE fornece (
  peca_id_pecas CHAR(6),
  servico_id_servico CHAR(6),
  CONSTRAINT fornece_pk PRIMARY KEY (peca_id_pecas, servico_id_servico),
  CONSTRAINT fornece_pecas_fk FOREIGN KEY (peca_id_pecas) REFERENCES pecas(id_pecas) ON DELETE CASCADE,
  CONSTRAINT fornece_servico_fk FOREIGN KEY (servico_id_servico) REFERENCES servicos(id_servico) ON DELETE CASCADE
);
```

- **Descrição:** Relaciona as peças fornecidas para os serviços automotivos.
- **Colunas:**
  - peca\_id\_pecas: Identificador da peça (Primary Key, Foreign Key referenciando pecas).
  - servico\_id\_servico: Identificador do serviço (Primary Key, Foreign Key referenciando servicos).
- **Chave Primária Composta:** (peca\_id\_pecas, servico\_id\_servico).