

**UNIVERSIDADE DO OESTE DE SANTA CATARINA - UNOESC**  
**CAMPUS SÃO MIGUEL DO OESTE**  
**CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

FERNANDO CAMILO SCHNEIDER  
JEAN CARLO TORAL  
JOÃO ULISSES PORTO ALEGRE CIRIACO TEIXEIRA

**PARK'ALOT**

São Miguel do Oeste, SC  
2024

FERNANDO CAMILO SCHNEIDER  
JEAN CARLO TORAL  
JOÃO ULISSES PORTO ALEGRE CIRIACO TEIXEIRA

**PARK'ALOT**

Relatório de formação apresentado à  
unidade curricular Banco de dados do  
Curso de Graduação em Ciência da  
Computação da Universidade do Oeste  
de Santa Catarina - UNOESC, Campus  
São Miguel do Oeste.

Orientador: Roberson Junior Fernandes  
Alves.

São Miguel do Oeste – Santa Catarina  
2024

## SUMÁRIO

1 INTRODUÇÃO.....	5
2 DESENVOLVIMENTO.....	6
2.1 Requisitos.....	6
2.2 Modelo Relacional.....	6
2.2.1 Dicionário de dados.....	7
2.3 Banco de dados.....	11
2.3.1 Relatórios.....	11
3 CONCLUSÃO.....	13
4 REFERÊNCIAS.....	14

## LISTA DE IMAGENS

<b>Imagem 1</b> - Modelo Relacional.....	7
<b>Imagem 2</b> - Dicionário de Dados.....	8
<b>Imagem 3</b> - Dicionário de Dados.....	9
<b>Imagem 4</b> - Dicionário de Dados.....	10
<b>Imagem 5</b> - Scripts de Consulta.....	11

## **1 INTRODUÇÃO**

O objetivo deste trabalho é desenvolver um sistema de banco de dados para gerenciar e alugar vagas de estacionamento em grandes cidades onde a demanda por estacionamento é alta, aplicando todo o conhecimento adquirido sobre banco de dados. Para o desenvolvimento desse sistema foi utilizado diversas ferramentas como Dbeaver e Visual Paradigm, o PostgreSQL como SGBD, e a linguagem SQL.

## **2 DESENVOLVIMENTO**

### **2.1 Requisitos**

Para o desenvolvimento do banco de dados do sistema de estacionamento, foram considerados os seguintes requisitos:

- Cadastro de Clientes e Funcionários
- Cadastro de um ou vários carros por cliente
- Gerenciamento de Vagas
- Gerenciamento de Manutenção das Vagas
- Registro de Entrada e Saída de Veículos
- Sistema de Pagamentos Diversos
- Cadastro de Endereço do Cliente
- Registro de Reserva de Vagas
- Controle de Acesso
- Relatórios e Análises

### **2.2 Modelo Relacional**

Para o projeto, agora que temos as informações necessárias, é hora de criar o modelo relacional. Isso significa trabalhar os requisitos e transformá-los em um mapa estrutural do banco de dados.

A criação do modelo relacional envolve a definição visual da estrutura do banco de dados, incluindo tabelas, relações entre elas, colunas e os tipos de dados que cada coluna armazenará. Essa etapa é essencial para garantir que o banco de dados atenda aos requisitos do projeto e funcione de maneira eficiente e organizada.

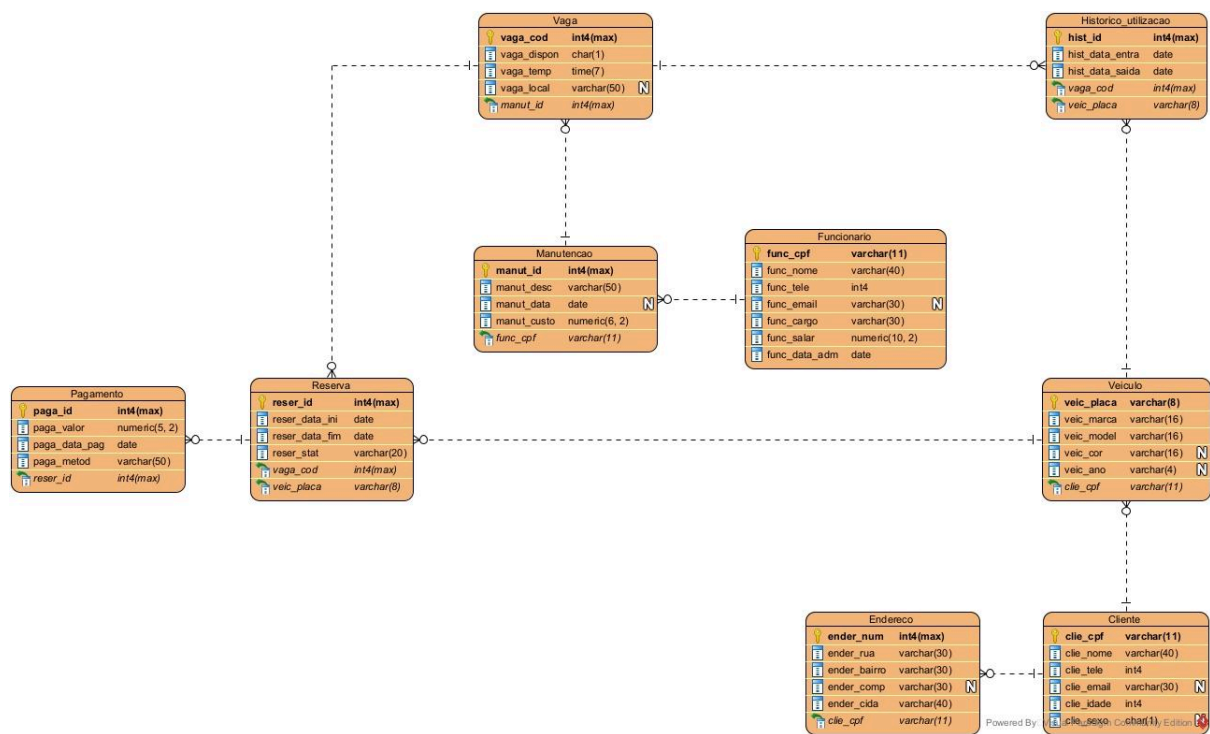


Imagem 1 - Modelo Relacional

## 2.2.1 Dicionário de Dados

As figuras abaixo mostram o dicionário de dados com as informações sobre as tabelas e seus atributos.

## 1. Data Dictionary



















Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
 Cliente	Tabela de Gerenciamento de Clientes	 clie_cpf	Cpf do Cliente	varchar	11	true	false	false
		 clie_email	Email do Cliente	varchar	30	false	true	false
		 clie_idade	Idade do Cliente	int4	0	false	false	false
		 clie_nome	Nome do Cliente	varchar	40	false	false	false
		 clie_sexo	Sexo do Cliente	char	1	false	true	false
			M - Masculino F - Feminino					
 Endereco	Tabela de gerenciamento de endereço dos clientes	 clie_cpf		varchar	11	false	false	false
		 ender_bairro	Bairro do endereço do cliente	varchar	30	false	false	false
		 ender_cidade	Cidade do endereço do cliente	varchar	40	false	false	false
		 ender_complemento	Complemento do endereço do cliente	varchar	30	false	true	false
		 ender_numero	Número da casa do endereço	int4	-1	true	false	false
		 ender_rua	Rua do endereço do cliente	varchar	30	false	false	false
 Funcionario	Tabela de gerenciamento de funcionários	 func_cargo	Cargo do funcionário	varchar	30	false	false	false
		 func_cpf	Cpf do funcionário	varchar	11	true	false	false
		 func_data_admissao	Data de admissão do funcionário	date	0	false	false	false
		 func_email	Email do funcionário	varchar	30	false	true	false

Imagem 2 - Dicionário de Dados
























Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
 Historico_utilizacao	Tabela de gerenciamento do histórico de utilização	 func_nome	Nome do funcionário	varchar	40	false	false	false
		 func_salario	Salário do funcionário	numeric	10.2	false	false	false
		 func_telefone	Telefone do funcionário	int4	0	false	false	false
		 hist_data_entrada	Histórico de entradas da vaga	date	0	false	false	false
		 hist_data_saida	Histórico de saída da vaga	date	0	false	false	false
		 hist_id	Id do histórico	int4	-1	true	false	false
 Manutencao	Tabela de gerenciamento da manutenção	 vaga_cod		int4	-1	false	false	false
		 veic_placa		varchar	8	false	false	false
		 func_cpf		varchar	11	false	false	false
		 manut_custo	Custo da manutenção da vaga	numeric	6.2	false	false	false
		 manut_data	Data da manutenção	date	0	false	true	false
		 manut_desc	Descrição da manutenção da vaga	varchar	50	false	false	false
 Pagamento	Tabela de gerenciamento de pagamento	 manut_id	Id da manutenção da vaga	int4	-1	true	false	false
		 paga_data_pag	Data de pagamento	date	0	false	false	false
		 paga_id	Id de pagamento da vaga	int4	-1	true	false	false
		 paga_metodo	Método de pagamento	varchar	50	false	false	false
		 paga_valor	Valor do pagamento da vaga	numeric	5.2	false	false	false
		 reser_id		int4	-1	false	false	false

Imagem 3 - Dicionário de Dados




















Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
 Reserva	Tabela de gerenciamento de reserva	 reser_data_fim	Data de encerramento da reserva da vaga	date	0	false	false	false
		 reser_data_ini	Início da data da reserva	date	0	false	false	false
		 reser_id	Id da reserva de vaga	int4	-1	true	false	false
		 reser_stat	Status da reserva	varchar	20	false	false	false
		 vaga_cod		int4	-1	false	false	false
		 veic_placa		varchar	8	false	false	false
 Vaga	Tabela de Gerenciamento das vagas	 manut_id		int4	-1	false	false	false
		 vaga_cod	Código da vaga	int4	-1	true	false	false
		 vaga_dispon	Disponibilidade da vaga	char	1	false	false	false
			D - Disponível I - Indisponível					
		 vaga_local	Localização das vagas	varchar	50	false	true	false
 Veiculo	Tabela de cadastro dos veículos	 clie_cpf		varchar	11	false	false	false
		 veic_ano	Ano do modelo do carro	varchar	4	false	true	false
		 veic_cor	Cor do veículo cadastrado	varchar	16	false	true	false
		 veic_marca	Marca dos veículos cadastrados	varchar	16	false	false	false
		 veic_model	Modelo do veículo cadastrado	varchar	16	false	false	false
		 veic_placa	Placa dos veículos cadastrados	varchar	8	true	false	false

Imagem 4 - Dicionário de Dados

## 2.3 Banco de dados

Depois da construção e normalização do modelo relacional com o uso da ferramenta Visual Paradigm, foi dada sequência no projeto com a criação dos scripts. Para fazer a criação dos scripts foi utilizada a ferramenta Dbeaver, a linguagem SQL, e o PostgreSQL. Previamente foi gerado o script de criação da base de dados, após isso foi feito um script para a inserção de dados na base.

### 2.3.1 Relatórios

Os relatórios foram criados na base de dados baseados nos quatro requisitos a seguir:

1) Relacionar todos os veículos da marca FIAT fabricados depois de 2008. Ordenar o relatório pela placa dos veículos em ordem decrescente;

2) Relacionar as reservas registradas em meses pares e de clientes dos bairros Centro e Agostini. Ordene o relatório pelo nome do cliente de forma decrescente;

3) Relacionar todos as reservas de clientes das cidades de Maravilha, Descanso, Itapiranga e Guaraciaba que tem mais de 40 anos e do sexo masculino. Ordene o relatório das cidades com mais reservas para cidades com menos reservas;

4) Relacionar o mês, o total de veículos estacionados e o valor total em valores realizados em 2024. Relacionar do mês com o maior valor para o mês com o menor valor

```
-- 1) Relacionar todos os veículos da marca FIAT fabricados depois de 2008. Ordenar o relatório pela placa dos veículos em ordem decrescente:
SELECT *
FROM Veiculo
WHERE veic_marca = 'FIAT' AND veic_ano > '2008'
ORDER BY veic_placa DESC;
```

```
-- 2) Relacionar as reservas registradas em meses pares e de clientes dos bairros Centro e Agostini. Ordene o relatório pelo nome do cliente de forma decrescente:
SELECT R.*, C.clie_nome
FROM Reserva R
inner JOIN Veiculo V ON R.veic_placa = V.veic_placa
inner JOIN Cliente C ON V.clie_cpf = C.clie_cpf
inner JOIN Endereco E ON C.clie_cpf = E.clie_cpf
WHERE CAST(EXTRACT(MONTH FROM R.reser_data_ini) AS INTEGER) % 2 = 0
AND (E.ender_bairro = 'Centro' OR E.ender_bairro = 'Agostini')
ORDER BY C.clie_nome DESC;
```

```

● -- 3) Relacionar todas as reservas de clientes das cidades de Maravilha, Descanso, Itapiranga e Guaraciaba que têm mais de 40 anos e do sexo masculino.
-- Ordene o relatório das cidades com mais reservas para cidades com menos reservas:

● SELECT E.ender_cida, COUNT(*) AS num_reservas
FROM Reserva R
inner JOIN Veiculo V ON R.veic_placa = V.veic_placa
inner JOIN Cliente C ON V.clie_cpf = C.clie_cpf
left JOIN Endereco E ON C.clie_cpf = E.clie_cpf
WHERE E.ender_cida IN ('Maravilha', 'Descanso', 'Itapiranga', 'Guaraciaba')
AND C.clie_idade > 40 AND C.clie_sexo = 'M'
GROUP BY E.ender_cida
ORDER BY num_reservas DESC;

```

```

● -- 4) Relacionar o mês, o total de veículos estacionados e o valor total em pagamentos realizados em 2024.
-- Relacionar do mês com o maior valor para o mês com o menor valor:

● SELECT EXTRACT(MONTH FROM P.paga_data_pag) AS mes,
        COUNT(DISTINCT H.veic_placa) AS total_veiculos,
        SUM(P.paga_valor) AS valor_total
FROM Pagamento P
full JOIN Reserva R ON P.reser_id = R.reser_id
full JOIN Historico_utilizacao H ON R.veic_placa = H.veic_placa
WHERE EXTRACT(YEAR FROM P.paga_data_pag) = 2024
GROUP BY mes
ORDER BY valor_total DESC;

```

Imagem 5 - Scripts de Consulta

### **3. CONCLUSÃO**

A criação de um banco de dados robusto e bem estruturado é fundamental para o sucesso do projeto. O uso do modelo relacional foi crucial para transformar os requisitos iniciais em uma estrutura organizada e eficiente. Integrar o desenvolvimento do modelo conceitual e relacional permitiu acelerar o processo, garantindo que as tabelas, relações, colunas e tipos de dados fossem definidos com precisão desde o início. Esse enfoque garante que o banco de dados não apenas atenda aos requisitos do projeto, mas também suporte o desenvolvimento contínuo de maneira ágil e eficaz.

#### 4. REFERÊNCIAS

ALVES, Roberson J. F. **Apostila de Banco de Dados**. São Miguel do Oeste: Unoesc, 2023. Material didático.