

Sistema de Locação de Veículos: Modelagem e Implementação do Banco de Dados

**Ravi K. Santos Vieria, Renan O. Moraes, Michel S. Félix,
Francisco Henrique S. Silva, Lucas Balduino B. Araújo**

Sistemas de Informação – Centro Universitário Paraíso (UNIFAP)
Juazeiro do Norte – CE – Brasil

Abstract. This article presents the modeling and implementation of a database system designed for managing a car rental company. The proposed system seeks to automate processes such as customer registration, vehicle control, reservation recording, and payment management. The modeling was developed using the Entity-Relationship (ER) model, followed by conversion to the relational model and implementation of the tables in SQL. Finally, example queries are presented that demonstrate the business value of the system and its applicability in a real rental environment.

Resumo. Este artigo apresenta a modelagem e implementação de um sistema de banco de dados voltado para o gerenciamento de uma locadora de veículos. O sistema proposto busca automatizar processos como cadastro de clientes, controle de veículos, registro de reservas e administração de pagamentos. A modelagem foi desenvolvida utilizando o modelo Entidade-Relacionamento (ER), seguido da conversão para o modelo relacional e implementação das tabelas em SQL. Por fim, são apresentadas consultas de exemplo que demonstram o valor de negócios do sistema e sua aplicabilidade em um ambiente real de locação.

1. Descrição do Sistema

O sistema proposto tem como objetivo gerenciar o processo de locação de veículos de forma eficiente, segura e integrada. Ele foi desenvolvido para atender às necessidades de uma empresa de aluguel de carros, permitindo o controle completo de **clientes**, **veículos**, **reservas** e **pagamentos**.

A aplicação possibilita o cadastro de clientes e suas respectivas informações pessoais, incluindo endereço e dados da CNH, que é requisito obrigatório para a locação de veículos. O sistema aceita apenas clientes com CNH de categoria **B**, **C** ou **D**, pois a locadora trabalha exclusivamente com veículos de passeio e utilitários. Também é realizado o registro detalhado dos veículos disponíveis, suas **categorias**, status e características como marca, cor, placa, ano de fabricação e capacidade de passageiros.

O módulo de reservas permite que o cliente realize a locação de um veículo por um determinado período, registrando datas de retirada e devolução, além do status da reserva. Cada reserva está associada a um pagamento, que pode ser efetuado por diferentes meios, como **cartão (crédito ou débito)** ou **pix**, garantindo flexibilidade nas transações.

Por meio do gerenciamento de **pagamentos**, o sistema é capaz de controlar valores, status e datas de quitação, detalhando as informações de acordo com o tipo de transação. Essa estrutura garante rastreabilidade e organização nas operações financeiras.

Em resumo, o sistema visa otimizar o processo de locação, reduzindo erros manuais, centralizando informações e fornecendo suporte à tomada de decisão através de consultas e relatórios personalizados.

2. Requisitos do Sistema

Tabela 1. Requisitos Funcionais do Sistema

Nº	Requisito Funcional	Descrição
1	Cadastrar cliente	O Sistema deve permitir cadastro, atualização e exclusão de cliente.
2	Cadastrar veículo	Deve permitir o registro de cada veículo com dados como categoria, placa, modelo, ano e status. Além do valor da diária.
3	Controlar disponibilidade	Deve atualizar automaticamente o status de um veículo conforme ele é reservado ou devolvido.
4	Registrar pagamento	Registrar pagamentos de reservas incluindo o valor, método (crédito, débito ou pix)

Tabela 2. Requisitos Não Funcionais do Sistema

Nº	Requisito Não Funcional	Descrição
1	Integridade	Deve manter a consistência e integridade referencial entre as tabelas.
2	Manutenabilidade	A estrutura deve permitir futuras alterações sem impacto significativo do Sistema.
3	Banco de dados relacional	O Sistema deve utilizar um SGBD relacional como Postgree por exemplo.

3. Modelagem Conceitual

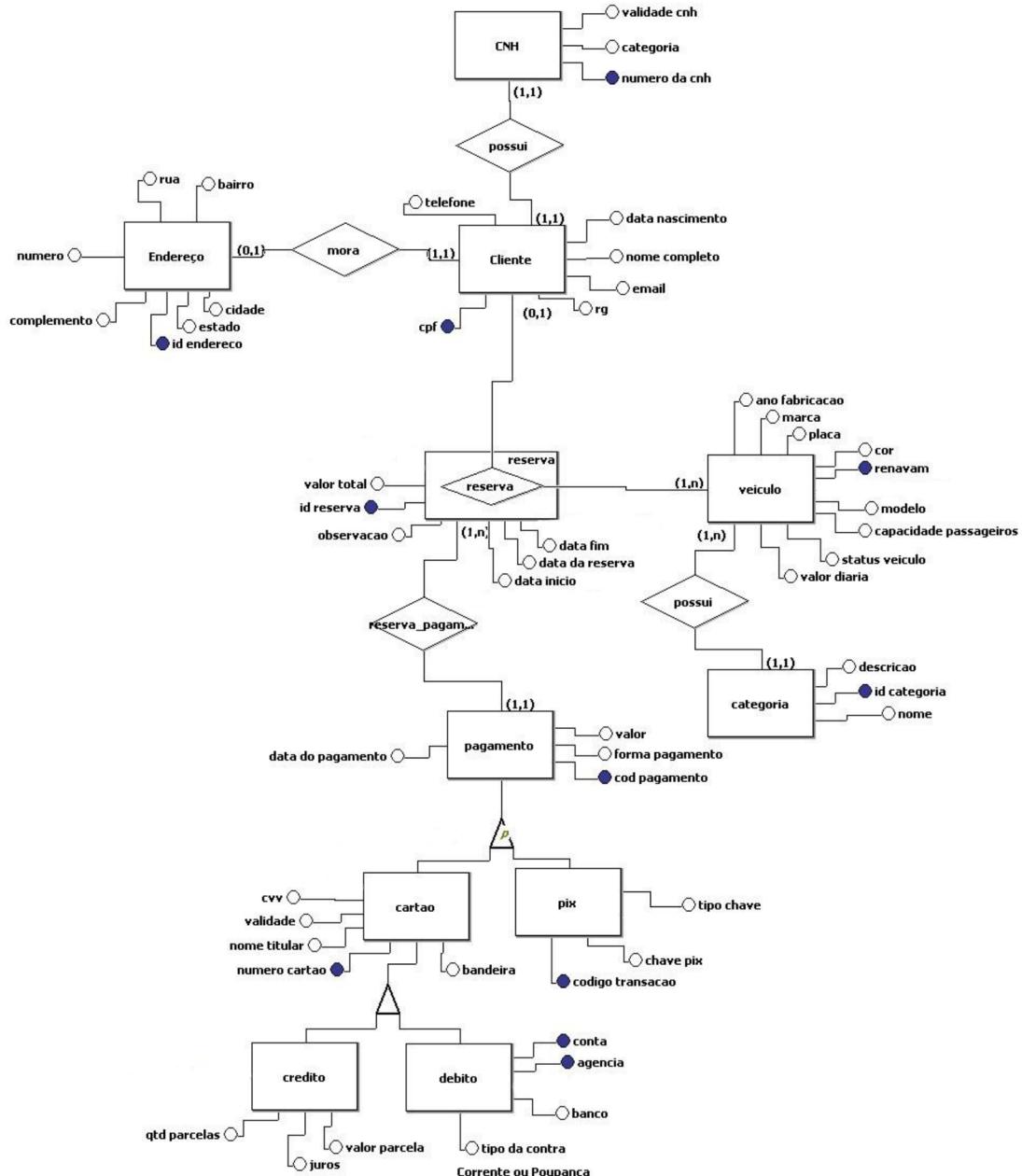


Figura 1. Modelo Entidade Relacionamento

O modelo conceitual acima representa o sistema de **locação de veículos**, descrevendo as principais entidades envolvidas e seus relacionamentos.

As entidades **Cliente**, **Veículo**, **Categoria**, **Reserva**, **Pagamento**, **Cartão**, **Crédito**, **Débito** e **Pix** foram modeladas de forma a garantir a integridade dos dados e refletir as regras de negócio da locadora.

O relacionamento entre **Cliente** e **Reserva** é de um-para-muitos, pois um cliente pode realizar várias reservas. Cada **Reserva** está associada a um **Veículo** e a um **Pagamento**, que pode ser especializado em **Crédito**, **Débito** ou **Pix**.

A especialização da entidade **Pagamento** foi utilizada para representar as diferentes formas de pagamento, mantendo as particularidades de cada tipo.

A entidade **Categoria** agrupa os tipos de veículos disponíveis para locação (como compactos, sedans, SUVs e picapes), possibilitando a classificação e organização dos automóveis cadastrados.

Essa modelagem busca garantir clareza na estrutura dos dados, facilitando a implementação física e a manutenção do sistema.

4. Modelagem Lógica

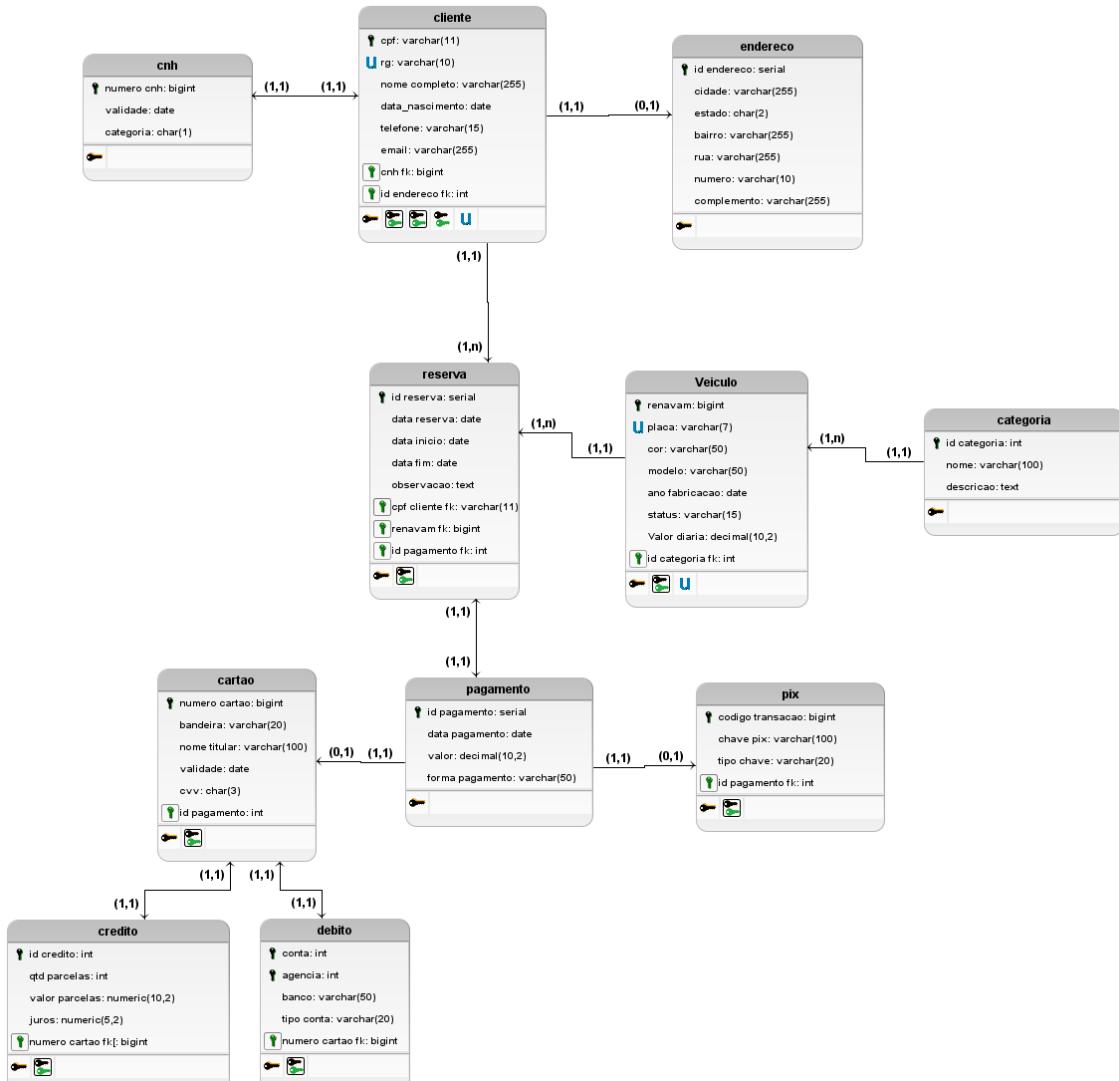


Figura 2. Modelo Relacional

A modelagem lógica acima representa o sistema em termos de tabelas relacionais, especificando seus atributos, tipos de dados e relacionamentos. Nessa etapa, o modelo conceitual foi traduzido para o formato relacional, respeitando as regras de integridade e os vínculos entre entidades.

5. Criação do Banco de Dados

O banco de dados foi implementado no SGBD PostgreSQL, utilizando comandos DDL para definir as estruturas das tabelas, chaves primárias, estrangeiras e restrições de integridade. A modelagem reflete o sistema de locação de veículos, garantindo consistência e relacionamento entre as entidades.

5.1 Estrutura da Tabela (DDL)

```
create database bd_tde3;

create table cliente (
    cpf varchar(11) primary key,
    nome_completo varchar(255) not null,
    rg varchar(10) not null unique,
    data_nascimento date not null,
    telefone varchar(15) not null,
    email varchar(255) not null,
    cnh_fk varchar(11),
    id_endereco int,
    foreign key (cnh_fk) references cnh (numero_cnh),
    foreign key (id_endereco) references endereco (id)
);

create table cnh (
    numero_cnh varchar(11) primary key,
    validade date not null,
    categoria char(1) constraint cat_chk check (
        categoria in ('B', 'C', 'D')
    ) not null
);

create table endereco (
    id serial primary key,
    cidade varchar(255) not null,
    estado char(2) not null,
    bairro varchar(255) not null,
    rua varchar(255) not null,
    numero varchar(10) not null,
    complemento varchar(255)
);
```

```

create table categoria (
    id serial primary key,
    nome varchar(100) not null,
    descricao text not null
);

create table veiculo (
    renavam bigint primary key,
    placa varchar(7) not null unique,
    cor varchar(50) not null,
    modelo varchar(150) not null,
    marca varchar(50) not null,
    ano_fabricacao int not null,
    valor_diaria decimal(10, 2) not null,
    status varchar(15) constraint status_chk check (
        status in ('DISPONIVEL', 'OCUPADO')
    ) not null,
    id_categoria_fk int not null,
    foreign key (id_categoria_fk) references categoria (id)
);

-- Tabelas de baixo
create table pagamento (
    id_pagamento serial primary key,
    data_pagamento date not null,
    valor decimal(10, 2) not null,
    forma_pagamento varchar(50) not null
);

create table cartao (
    numero_cartao int not null primary key,
    bandeira varchar(20) not null,
    nome_titular varchar(100) not null,
    validade date not null,
    cvv char(3) not null,
    id_pagamento int,
    foreign key (id_pagamento) references pagamento (id_pagamento)
);

create table credito (
    id_credito serial primary key,
    qtd_parcelas int not null,
    valor_parcela numeric(10, 2) not null,
    juros numeric(5, 2),
    numero_cartao_fk int,
    foreign key (numero_cartao_fk) references cartao (numero_cartao)
);

```

```
create table debito (
    conta int generated always as identity,
    agencia int generated always as identity,
    banco varchar(50) not null,
    tipo_conta varchar(20) constraint tipo_conta_chk check
        (tipo_conta in ('Corrente', 'Poupanca')) not null,
    primary key (conta, agencia),
    numero_cartao_fk int,
    foreign key (numero_cartao_fk) references cartao (numero_cartao)
);

create table pix (
    codigo_transacao int generated always as identity primary key,
    chave_pix varchar(100) not null,
    tipo_chave varchar(20) constraint tipo_chave_chk check (
        tipo_chave in ('CPF', 'CNPJ', 'EMAIL', 'TELEFONE')
    ) not null,
    id_pagamento_fk int not null,
    foreign key (id_pagamento_fk) references pagamento (id_pagamento)
);

create table reserva (
    id_reserva serial primary key,
    data_reserva date not null,
    data_inicio date not null,
    data_fim date not null,
    observacao text,
    cpf_cliente_fk varchar(11) not null,
    renavam_fk bigint not null,
    id_pagamento_fk int not null,
    foreign key (cpf_cliente_fk) references cliente (cpf),
    foreign key (renavam_fk) references veiculo (renavam),
    foreign key (id_pagamento_fk) references pagamento (id_pagamento)
);
```

6. Consultas e Valor de Negócio

A seguir, são apresentadas algumas consultas SQL desenvolvidas sobre o banco de dados proposto, juntamente com a justificativa de seu **valor de negócio**, ou seja, como cada consulta pode auxiliar a gestão e operação da locadora de veículos.

6.1 Listar todos os veículos disponíveis para locação

The screenshot shows a SQL query interface with the following details:

Query History: The tab is labeled "Query".

SQL Query:

```
1
2 select modelo, marca, placa, valor_diaria, status from veiculo
3 where status = 'DISPONIVEL';
4
5
```

Data Output: The tab is labeled "Data Output".

Table Headers: modelo (character varying (150)), marca (character varying (50)), placa (character varying (7)), valor_diaria (numeric (10,2)), status (character varying (15)).

Table Data:

	modelo character varying (150)	marca character varying (50)	placa character varying (7)	valor_diaria numeric (10,2)	status character varying (15)
1	MOBI WAY 1.0 Fire Flex 5p.	Fiat	HWN5533	80.00	DISPONIVEL
2	Grand Siena ESSENCE ITALIA 1.6 Flex 16V	Fiat	HXI7768	90.00	DISPONIVEL
3	Ranger XLT 4.0 4x4 CD	Ford	HXI2674	150.00	DISPONIVEL
4	Civic Sedan LXS 1.8/1.8 Flex 16V Mec. 4p	Honda	HKK5536	120.00	DISPONIVEL
5	Renegade Trailhawk 2.0 4x4 TB Diesel Aut	Jeep	HWG8457	160.00	DISPONIVEL

Permite que os atendentes ou o sistema mostrem rapidamente quais veículos estão disponíveis no momento, otimizando o processo de reserva e evitando tentativas de locação de veículos indisponíveis.

6.2 Consultar os clientes com reservas ativas

The screenshot shows a SQL query interface with the following details:

Query History: The tab is labeled "Query".

SQL Query:

```
1
2 select c.nome_completo, r.data_inicio, r.data_fim, v.modelo
3   from cliente c
4 inner join reserva r on c.cpf = r.cpf_cliente_fk
5 inner join veiculo v on r.renavam_fk = v.renavam
6 where CURRENT_DATE between r.data_inicio and r.data_fim;
7
```

Data Output: The tab is labeled "Data Output".

Table Headers: nome_completo (character varying (255)), data_inicio (date), data_fim (date), modelo (character varying (150)).

Table Data:

	nome_completo character varying (255)	data_inicio date	data_fim date	modelo character varying (150)
1	Geraldo Iago Novaes	2025-10-27	2025-10-29	UNO SPORTING 1.4 B.Edit. Flex 8V 5P
2	Kauê Eduardo da Mota	2025-10-26	2025-10-28	CITY Sedan LX 1.5 Flex 16V 4p Mec.
3	Otávio Osvaldo Daniel da Rocha	2025-10-27	2025-10-29	Frontier SL CD 4X4 2.5TB Diesel Aut
4	Benedito Giovanni Ferreira	2025-10-25	2025-10-27	Hilux CD SRV D-4D 4x2 3.0 163cv TDI Dies

Ajuda a equipe de gestão a identificar quais clientes estão com veículos atualmente alugados, permitindo melhor controle da frota e acompanhamento das devoluções.

6.3 Veículos mais alugados

Query Query History

```
1
2 select v.modelo, v.marca, count(r.id_reserva) as total_reserva
3   from veiculo v
4 inner join reserva r on v.renavam = r.renavam_fk
5 group by v.modelo, v.marca
6 order by total_reserva desc;
```

Data Output Messages Notifications

SQL

	modelo character varying (150)	marca character varying (50)	total_reserva bigint
1	Strada WORKING Hard 1.4 Fire Flex 8V CE	Fiat	1
2	Frontier SL CD 4X4 2.5TB Diesel Aut	Nissan	1
3	UNO WAY XINGU 1.4 EVO FFlex 8V	Fiat	1
4	CITY Sedan LX 1.5 Flex 16V 4p Mec.	Honda	1
5	EcoSport SE 1.6 16V Flex 5p Mec.	Ford	1
6	Hilux CD SRV D-4D 4x2 3.0 168cv TDI Dies	Toyota	1
7	UNO SPORTING 1.4 B.Edit. Flex 8V 5P	Fiat	1

Ajuda a identificar quais veículos têm maior demanda, orientando futuras aquisições e estratégias de especificação da locadora.

6.4 Reservas futuras agendadas

Query Query History

```
1 SELECT c.nome_completo, v.modelo, r.data_inicio, r.data_fim
2 FROM reserva r
3 JOIN cliente c ON r.cpf_cliente_fk = c.cpf
4 JOIN veiculo v ON r.renavam_fk = v.renavam
5 WHERE r.data_inicio > CURRENT_DATE;
```

Data Output Messages Notifications

SQL

	nome_completo character varying (255)	modelo character varying (150)	data_inicio date	data_fim date
1	Mario Gustavo Thiago Oliveira	Strada WORKING Hard 1.4 Fire Flex 8V CE	2025-10-28	2025-10-30
2	Matheus Fernando Aparicio	UNO WAY XINGU 1.4 EVO FFlex 8V	2025-10-29	2025-10-31
3	Renato Raimundo Assis	EcoSport SE 1.6 16V Flex 5p Mec.	2025-10-28	2025-10-30

Facilita o planejamento operacional, permitindo que a locadora organize a limpeza, manutenção e disponibilidade dos veículos para as próximas locações.

Referencias

- BOSON TREINAMENTOS, Curso de Modelagem de Banco de Dados, YouTube, playlist. Disponível em:
https://www.youtube.com/playlist?list=PLucm8g_ezqNoNHU8tjVeHmRGBFnjDlx_D. Acesso em: 20 out. 2025.
- HEUSER, Carlos Alberto. Projeto de Banco de Dados. 7.^a ed. Porto Alegre: Bookman, 2024.
- BOSON TREINAMENTOS, Curso de Modelagem de Banco de Dados, YouTube, playlist. Disponível em:
https://www.youtube.com/playlist?list=PLucm8g_ezqNoAkYKXN_zWupyH6hQCA_wxY. Acesso em: 18 out. 2025.