

Relatório do Projeto: Sistema de Reserva de Hotéis

Disciplina: Projeto e Modelagem de Banco de Dados

Professora: Gabrielle K. Canalle

Ano: 2025

Autores: Gabriel Antonio, Gustavo Laporte, Matheos Guerra, Pedro Dhalia

Objetivo do Projeto

Desenvolver um sistema de banco de dados relacional completo para um *Sistema de Reserva de Hotéis*, capaz de gerenciar:

- Hóspedes
- Funcionários
- Quartos
- Reservas
- Pagamentos
- Serviços adicionais
- Manutenções
- Avaliações

Incluindo modelagem conceitual, lógica, scripts SQL e uma interface funcional em Java (com JDBC puro, sem frameworks ORM).

Mini Mundo

O hotel disponibiliza quartos de diferentes tipos e preços. Hóspedes podem realizar reservas para um ou mais quartos, escolhendo serviços adicionais durante a estadia. Cada reserva

possui um ou mais pagamentos, que podem ser efetuados em dinheiro, cartão ou pix. Funcionários podem supervisionar outros funcionários e executar manutenções nos quartos. Após a estadia, hóspedes podem avaliar a experiência.

As informações registradas contemplam dados pessoais dos hóspedes e funcionários, dados dos quartos, histórico de reservas e serviços utilizados, avaliações e registros de manutenção.

Descrição das Etapas do Processo de Desenvolvimento

1. Mini Mundo

- Descrição textual do mini mundo.

2. Modelagem Conceitual

- Elaboração do Diagrama Entidade-Relacionamento (DER) utilizando o BRModelo.

3. Modelagem Lógica Relacional

- Transformação do modelo conceitual para o modelo lógico relacional.
- Definição de tabelas, atributos, tipos de dados e restrições.

4. Criação do Banco de Dados

- Escrita do script SQL de criação das tabelas no MySQL, incluindo chaves primárias, estrangeiras, constraints e integridade referencial.

5. Povoamento de Dados

- Desenvolvimento do script de inserção com dados fictícios.

6. Interface CRUD em Java

- Implementação da interface funcional usando Java com JDBC puro para manipulação dos dados.

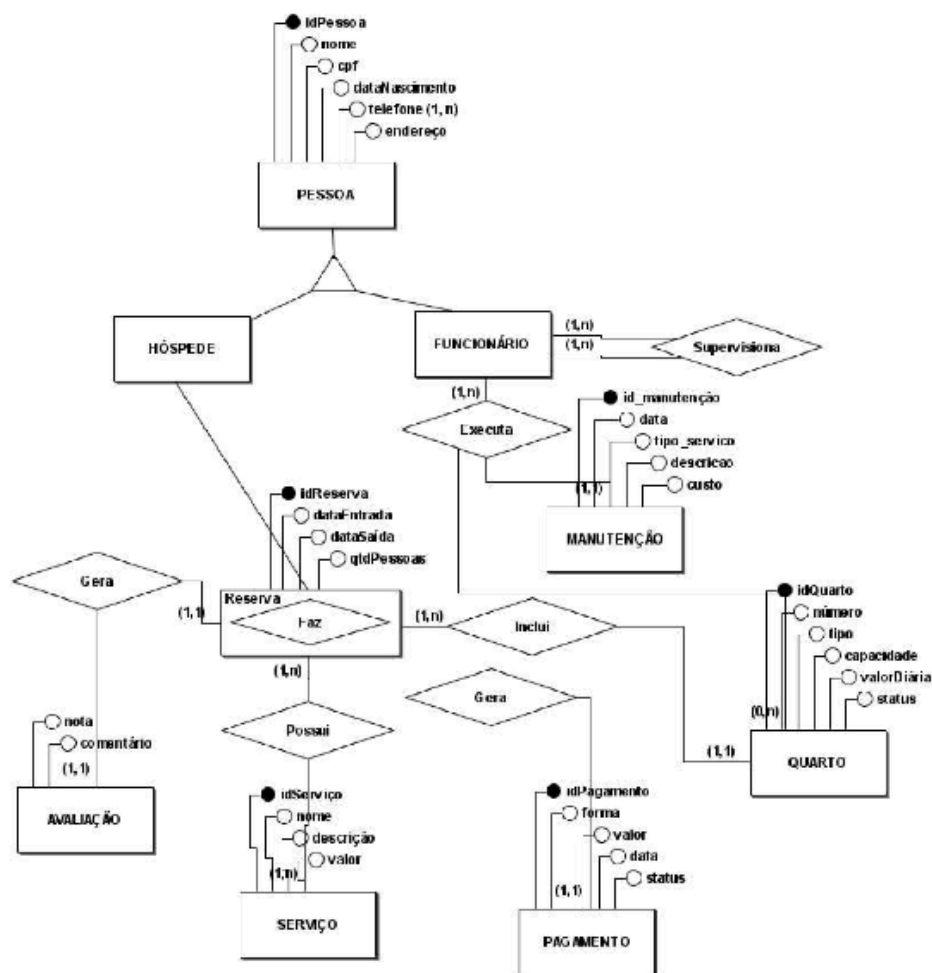
7. Consultas SQL

- Criação de consultas para extração de relatórios e informações úteis para gestão do hotel.

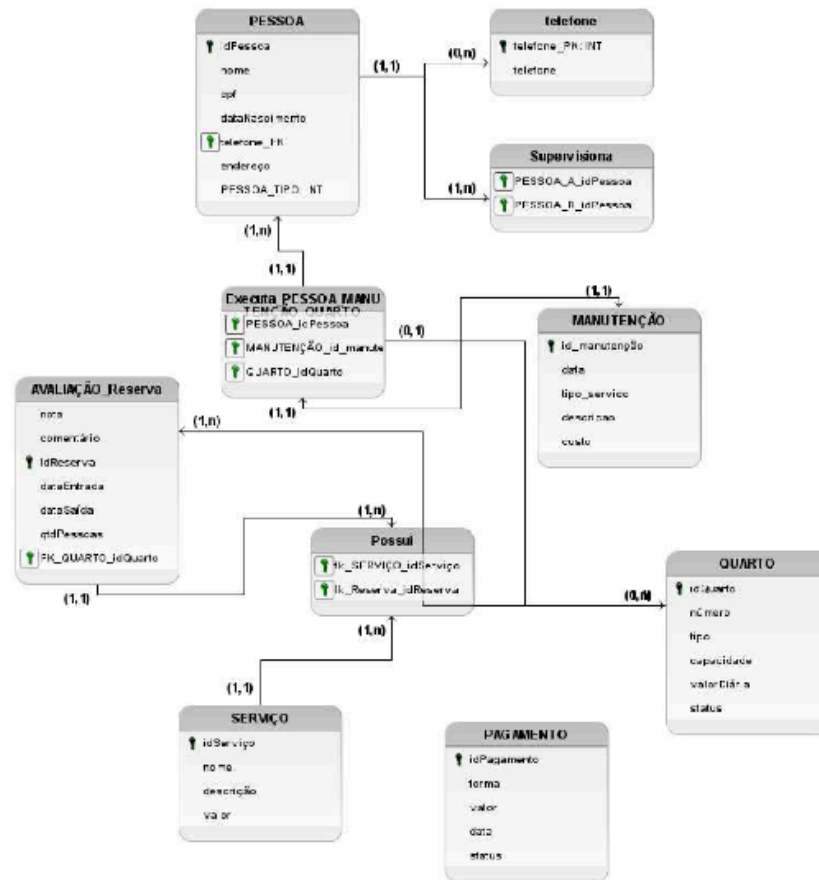
8. Elaboração do Relatório Final

- Compilação das etapas, modelos, scripts e resultados para documentação.

Modelo Conceitual



Modelo Lógico Relacional



Script de Criação

```
CREATE DATABASE IF NOT EXISTS Hotel;
```

```
USE Hotel;
```

```
CREATE TABLE Pessoa (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    nome VARCHAR(100) NOT NULL,
```

```
    cpf CHAR(11) UNIQUE NOT NULL,
```

```
    data_nascimento DATE NOT NULL,
```

```
    rua VARCHAR(100),

    numero VARCHAR(10),

    cidade VARCHAR(50),

    estado CHAR(2),

    cep CHAR(8),

    telefone1 VARCHAR(255) NULL,

    telefone2 VARCHAR(15)

);

CREATE TABLE Hospede (

    id_pessoa INT PRIMARY KEY,

    FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id)

);

CREATE TABLE Funcionario (

    id_pessoa INT PRIMARY KEY,

    supervisor_id INT,

    FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id),

    FOREIGN KEY (supervisor_id) REFERENCES Funcionario(id_pessoa)

);

CREATE TABLE Quarto (

    numero VARCHAR(5) PRIMARY KEY,

    tipo ENUM('simples', 'duplo', 'suíte'),

    capacidade INT NOT NULL,

    valor_diaria DECIMAL(10,2) NOT NULL,
```

```
        status ENUM('livre', 'reservado', 'ocupado') NOT NULL
    );

CREATE TABLE Reserva (

    id INT AUTO_INCREMENT PRIMARY KEY,

    id_hospede INT,

    data_entrada DATE NOT NULL,

    data_saida DATE NOT NULL,

    qtd_hospedes INT NOT NULL,

    quartos TEXT,

    servicos TEXT,

    FOREIGN KEY (id_hospede) REFERENCES Hospede(id_pessoa)

);

CREATE TABLE Servico (

    id INT AUTO_INCREMENT PRIMARY KEY,

    nome VARCHAR(50) NOT NULL,

    descricao TEXT,

    preco DECIMAL(10,2) NOT NULL

);

CREATE TABLE Pagamento (

    id INT AUTO_INCREMENT PRIMARY KEY,

    id_reserva INT,

    valor_total DECIMAL(10,2) NOT NULL,

    data_pagamento DATE,
```

```
        forma_pagamento ENUM('dinheiro', 'cartao', 'pix'),
        status_pagamento ENUM('pago', 'pendente'),
        FOREIGN KEY (id_reserva) REFERENCES Reserva(id)
    );

CREATE TABLE Avaliacao (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_reserva INT,
    nota INT CHECK (nota BETWEEN 1 AND 5),
    comentario TEXT,
    FOREIGN KEY (id_reserva) REFERENCES Reserva(id)
);

CREATE TABLE Manutencao (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_quarto VARCHAR(5),
    id_funcionario INT,
    descricao TEXT NOT NULL,
    data DATE NOT NULL,
    custo DECIMAL(10,2),
    status ENUM('pendente', 'concluída'),
    FOREIGN KEY (id_quarto) REFERENCES Quarto(numero),
    FOREIGN KEY (id_funcionario) REFERENCES Funcionario(id_pessoa)
);
```

Script de Povoamento

```
USE Hotel;
```

```
INSERT INTO Pessoa (nome, cpf, data_nascimento, rua, numero, cidade, estado, cep, telefone1, telefone2)
```

```
VALUES
```

```
('João Silva', '12345678901', '1985-04-20', 'Rua A', '123', 'Recife', 'PE', '50000000', '81999999999', NULL),
```

```
('Maria Souza', '98765432100', '1990-08-15', 'Av. B', '456', 'Olinda', 'PE', '53000000', '81988888888', NULL),
```

```
('Carlos Lima', '11122233344', '1982-12-05', 'Rua C', '789', 'Recife', 'PE', '51000000', '81977777777', NULL),
```

```
('Ana Pereira', '22233344455', '1995-11-30', 'Rua D', '101', 'Jaboatão', 'PE', '54000000', '81966666666', NULL),
```

```
('Lucas Menezes', '33344455566', '1979-03-11', 'Av. E', '202', 'Paulista', 'PE', '53400000', '81955555555', NULL),
```

```
('Fernanda Costa', '44455566677', '1988-07-25', 'Rua F', '303', 'Caruaru', 'PE', '55000000', '81944444444', NULL);
```

```
INSERT INTO Hospede (id_pessoa) VALUES (1), (4);
```

```
INSERT INTO Funcionario (id_pessoa, supervisor_id)
```

```
VALUES
```

```
(2, NULL),
```

```
(3, 2),
```

```
(5, NULL),
```

```
(6, 5);
```

```
INSERT INTO Quarto (numero, tipo, capacidade, valor_diaria, status)
```


VALUES

```
('101', 'simples', 1, 150.00, 'livre'),  
( '102', 'duplo', 2, 250.00, 'ocupado'),  
( '201', 'suíte', 3, 400.00, 'reservado'),  
( '202', 'simples', 1, 140.00, 'livre'),  
( '203', 'duplo', 2, 260.00, 'livre'),  
( '301', 'suíte', 4, 500.00, 'ocupado');
```

INSERT INTO Servico (nome, descricao, preco)

VALUES

```
('Café da manhã', 'Buffet completo servido no salão principal',  
30.00),  
( 'Lavanderia', 'Serviço de lavagem e passagem de roupas', 50.00),  
( 'Translado', 'Transporte do aeroporto ao hotel', 100.00),  
( 'Spa', 'Sessão de relaxamento e massagem', 150.00),  
( 'Room Service', 'Entrega de alimentos e bebidas no quarto', 40.00);
```

INSERT INTO Reserva (id_hospede, data_entrada, data_saida,
qtd_hospedes, quartos, servicos)

VALUES

```
(1, '2025-04-25', '2025-04-30', 2, '102,201', '1,3'),  
(4, '2025-05-05', '2025-05-10', 1, '202', '2,4'),  
(1, '2025-06-01', '2025-06-07', 3, '203,301', '1,5');
```

INSERT INTO Pagamento (id_reserva, valor_total, data_pagamento,
forma_pagamento, status_pagamento)

VALUES

```
(1, 1150.00, '2025-04-22', 'pix', 'pago'),  
(2, 650.00, '2025-05-01', 'cartao', 'pago'),  
(3, 1800.00, NULL, 'dinheiro', 'pendente');
```

INSERT INTO Avaliacao (id_reserva, nota, comentario)

VALUES

```
(1, 4, 'Atendimento excelente, mas o ar-condicionado estava com  
defeito.'),  
(2, 5, 'Muito confortável e serviço excelente!'),  
(3, 3, 'Hotel bom, mas houve atraso no room service.');
```

INSERT INTO Manutencao (id_quarto, id_funcionario, descricao, data,
custo, status)

VALUES

```
('102', 3, 'Reparo no ar-condicionado', '2025-04-21', 200.00,  
'concluída'),  
( '301', 6, 'Troca do chuveiro e revisão elétrica', '2025-04-28',  
320.00, 'pendente'),  
( '202', 5, 'Verificação do ar-condicionado e limpeza do filtro',  
'2025-04-20', 80.00, 'concluída');
```

Consultas SQL

Exemplos de consultas implementadas:

```
SELECT p.nome, r.data_entrada, r.data_saida  
FROM Pessoa p  
JOIN Hospede h ON p.id = h.id_pessoa  
JOIN Reserva r ON h.id_pessoa = r.id_hospede;
```

```
SELECT SUM(valor_total) AS total_recebido  
FROM Pagamento  
WHERE status_pagamento = 'pago';
```

```
SELECT numero, tipo, status  
FROM Quarto  
WHERE status = 'ocupado';
```

Interface Funcional com Dashboard

A interface CRUD foi desenvolvida utilizando Java com JDBC puro. Ela permite:

- Cadastro, alteração e exclusão de hóspedes, funcionários e reservas.
- Listagem de quartos e serviços disponíveis.
- Registro e consulta de pagamentos.
- Emissão de relatórios simples via terminal.

Tela Inicial

Adicionar Pessoa

Fazer Reserva

Gerenciar Quartos

Fazer Avaliação

CRUD - Avaliação de Reserva

ID Reserva:

Nota (1-5):

Comentário:

ID Avaliação (p/ editar/deletar):

Inserir

Alterar

Deletar

Visualizar

Fazer Reserva

ID Hóspede:

Data Entrada (yyyy-mm-dd):

Data Saída (yyyy-mm-dd):

Quantidade de Hóspedes:

Quartos (separados por vírgula):

Serviços (separados por vírgula):

Fazer Reserva

Deletar Reserva

Alterar Reserva

Visualizar Reserva

- Nome
- CPF
- Data de nascimento
- Rua
- Número
- Cidade
- Estado
- CEP
- Telefone 1
- Telefone 2

Na parte inferior, uma tabela exibe os registros cadastrados, permitindo visualizar informações de maneira organizada.

Tela de Reserva

Funcionalidade para criação, alteração e exclusão de reservas. Campos disponíveis:

- ID do hóspede
- Data de entrada
- Data de saída
- Quantidade de hóspedes
- Quartos (separados por vírgula)
- Serviços (separados por vírgula)

Botões para:

- Fazer Reserva

- Deletar Reserva
- Alterar Reserva
- Visualizar Reserva

Tela de Avaliação

Permite a avaliação de reservas realizadas anteriormente. Campos disponíveis:

- ID da reserva
- Nota (de 1 a 5)
- Comentário
- ID da avaliação (usado para editar ou deletar avaliações)

Botões para:

- Inserir
- Alterar
- Deletar
- Visualizar

Fluxo de Navegação

O usuário inicia na **Tela Inicial**, onde escolhe a funcionalidade desejada entre as opções de cadastro de pessoa, fazer reserva, gerenciar quartos ou fazer avaliação. Cada uma dessas opções direciona para sua respectiva tela CRUD, onde é possível gerenciar os dados de forma simples e eficiente, com campos bem definidos e botões de ação claros.

Observações

- Nenhum framework ORM foi utilizado.
- Todas as constraints e chaves foram implementadas manualmente.
- Sistema 100% compatível com MySQL e DBeaver.
- Documentação organizada no diretório `docs/` e scripts no diretório `sql/`.