

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SÃO PAULO**

**2º SEMESTRE ADS – CJOBDD1 - PROFº PAULO GIOVANI DE FARIA ZEFERINO**

**BRIAN GUSTAVO DE OLIVEIRA BRANDÃO**

**SISTEMA PARA GERENCIAMENTO DE HOTÉIS**

**CAMPOS DO JORDÃO**

**2025**

## RESUMO

Este trabalho aborda a criação de bancos de dados a serem usados em sistemas de gerenciamento de hotéis. Inicia com o levantamento dos requisitos para o funcionamento de tais sistemas, procede para o desenvolvimento de diagramas entidade-relacionamento do projeto e finaliza com seu enfoque, que é a implementação prática em um SGDB. Possui o objetivo de propor um sistema funcional, dada a proeminência da hotelaria em Campos do Jordão e constante informatização do setor.

**Palavras-Chave:** banco de dados; sistema de software; gerenciamento; hotel.

## 1 INTRODUÇÃO

O avanço tecnológico tem transformado diversos setores da economia, e a gestão de hotéis não é exceção. A administração eficiente de um hotel exige o controle de uma grande quantidade de dados, que envolvem reservas, *check-ins* e *check-outs*, disponibilidade de quartos, preços, entre outros. Nesse contexto, os sistemas de gerenciamento se tornam ferramentas essenciais para garantir a organização, agilidade e precisão na gestão dessas informações.

Contudo, para que tais sistemas possam sequer funcionar, é necessário um banco de dados robusto, já que a área da hotelaria trabalha com um grande tráfego de pessoas, e subsequentemente, de informações a serem guardadas e processadas. Tendo isto em mente, este trabalho visa propor um projeto a ser utilizado para tal finalidade.

No decorrer deste texto, serão abordados os principais conceitos e as melhores práticas na criação e implementação de um banco de dados para sistemas de gerenciamento de hotéis, considerando principalmente o aspecto da modelagem entidade-relacionamento que será usada para representar o funcionamento do projeto e seu subsequente código.

## **1.1 Objetivos**

Este trabalho tem por objetivo detalhar a criação de um banco de dados para sistemas de gerenciamento de hotéis, com enfoque na parte das modelagens conceitual, lógica e física deste banco.

Para a consecução deste objetivo, foram estabelecidos os objetivos específicos:

- Realizar uma investigação sobre as regras de negócio, além de demais requisitos, fundamentais para o funcionamento de um hotel;
- Elaborar um projeto de banco de dados que atenda as demandas aferidas no objetivo anterior;
- Propor o projeto criado para potenciais clientes e guiar sua futura implementação.

## **1.2 Justificativa**

A cidade de Campos do Jordão é renomada por seus hotéis, tanto em quantidade quanto em qualidade, e assim como quaisquer outros, estes buscam se modernizar a todo momento para melhor atender sua clientela. Portanto, este trabalho possui suma relevância para a comunidade na qual está inserido.

## **1.3 Aspectos Metodológicos**

O presente estudo fez uso das pesquisas de natureza bibliográfica para o que remete à parte escrita, ferramentas de modelagem de banco de dados para o que remete à elaboração de diagramas entidade-relacionamento e SGDBs (sistemas de gerenciamento de banco de dados) para a criação e teste do modelo físico.

## **1.4 Aporte Teórico**

As bases teóricas na qual esta pesquisa se fundamentou foram principalmente livros, tanto para a questão do funcionamento do setor hoteleiro quanto para a modelagem de bancos de dados (cujos conceitos também foram abordados ao longo da disciplina), mas também diretrizes legais e ademais.

## **2 PROJETO PROPOSTO**

Nesta seção será apresentada detalhadamente a metodologia utilizada neste trabalho. Para a modelagem dos diagramas entidade-relacionamento, utilizou-se a notação do professor Carlos Alberto Heuser, conforme descrito em sua obra "Projeto de banco de dados", através da ferramenta virtual brModelo em sua versão de aplicativo.

As regras de negócio foram coletadas através de um questionário, que propunha as seguintes questões:

- Quais são os componentes essenciais para o funcionamento de um hotel?
- Como tais componentes interagem entre si?
- Quais informações precisam ser armazenadas sobre esses componentes?
- Qual a legislação relevante para a área da hotelaria?
- O que um sistema precisaria ter para atender adequadamente à demanda tanto dos clientes quanto dos colaboradores e gerência?
- Há algum detalhe importante que não foi abordado nas questões anteriores? Se sim, poderia elaborar quanto a este?

Após receber com sucesso o devido retorno e consultar a bibliografia para preencher lacunas de informação, iniciou-se o trabalho no projeto. Com a finalização do conceito, a parte prática foi trabalhada através do SQL Server 2022 (versão Express) e sua interface, o Management Studio 20, para fins de simplicidade. De acordo com a documentação da Microsoft, os requisitos mínimos para instalação são 6GB de espaço de disco rígido disponível, monitor com resolução 800x600, acesso à Internet, 512MB de memória e processador x64 1,4 GHz.

### **3 AVALIAÇÃO**

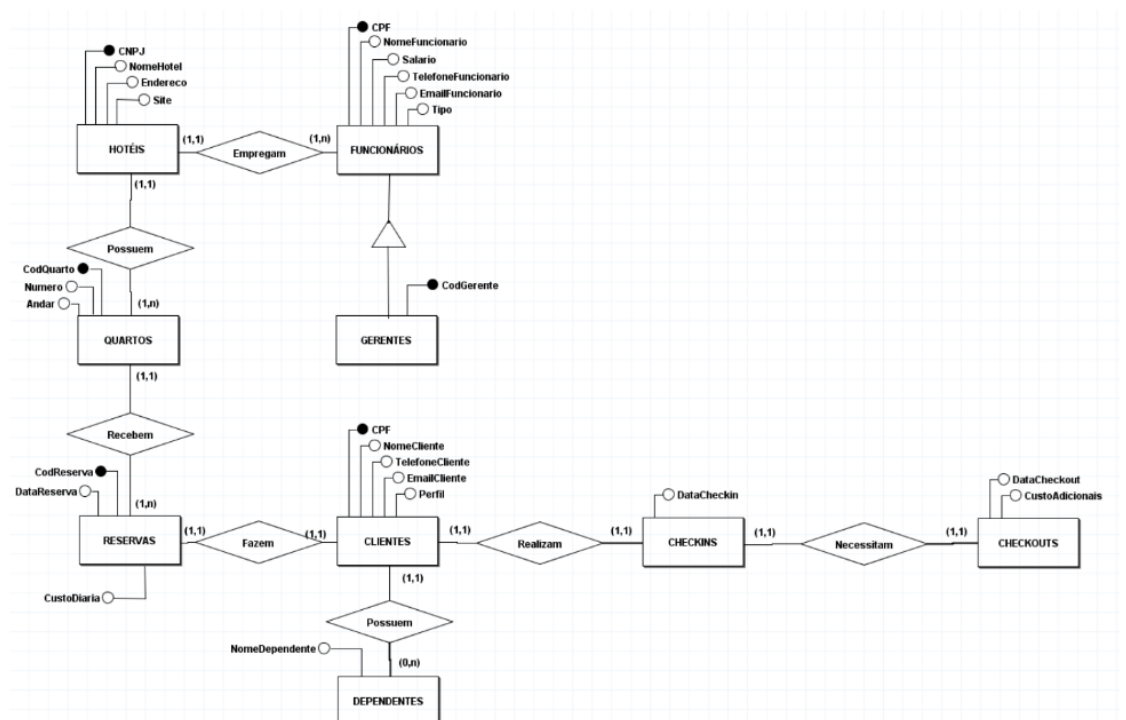
Nesta seção serão apresentados os resultados deste trabalho e uma discussão sobre eles.

### 3.1 Condução

O questionário criado para levantamento das regras de negócio foi distribuído para alguns especialistas locais da área, tendo estes uma semana para responder as perguntas da forma mais detalhada e descritiva possível. Durante o íterim entre a entrega e o retorno, alguns livros encontrados sobre o tópico foram consultados para garantir a veracidade e preencher eventuais lacunas de qualquer informação fornecida.

### 3.2 Resultados

Tendo os dados necessários, o próximo passo foi montar um diagrama entidade-relacionamento baseado nestes para que fosse possível visualizar a ideia do projeto. O subsequente resultado pode ser visto na figura 1, localizada na página a seguir.



**Figura 1 – Diagrama do projeto.**

Cada hotel emprega um ou vários funcionários (que, por sua vez, só podem trabalhar para um hotel), podendo ser estes gerentes ou não. Além disso, possuem diversos quartos, que são reservados por vários clientes em ocasiões diferentes. Os ditos clientes podem ou não ter dependentes, mas todos farão o *check-in* ao entrar, que obrigatoriamente será seguido por um *check-out*. Para maiores detalhes sobre os atributos, a tabela a seguir contém um dicionário de dados.

Nome	Tipo	Descrição?	Nulo
CNPJ	numérico	Cadastro de pessoa jurídica do hotel	Não
CPF	numérico	Cadastro de pessoa física, para funcionários e clientes	Não
Cod	numérico	Código genérico, para identificação	Não
Nome	categórico	Nome da pessoa/hotel	Não
Endereço	categórico	Localização física do hotel	Não
Site	categórico	Endereço eletrônico do hotel	Sim

Salario	numérico	Salário dos funcionários	Não
Telefone	numérico	Telefone do cliente/funcionário, para contato	Sim
Email	categórico	Email do cliente/funcionário, para contato	Sim
Tipo	categórico	Usado para permitir a especialização no diagrama	Não
Numero	numérico	Número do quarto	Não
Andar	categórico	Andar em que o quarto se localiza	Não
Data	tempo	Dia/mês/ano	Não
Custo	numérico	Preço a ser pago pelo cliente, com os adicionais cobrados no <i>check-out</i>	Não
Perfil	categórico	Caracterização do cliente em local ou	Não



		estrangeiro, de acordo com a Regulação Geral dos Meios de Hospedagem	
--	--	--	--

Tendo isto concluído, o próximo passo foi converter o modelo para código em linguagem SQL, através do SQL Server. O resultado foi o seguinte:

```
CREATE DATABASE ProjetoBDD2;

GO

USE ProjetoBDD2;

GO

CREATE TABLE HOTEIS (

        CNPJ VARCHAR(18) PRIMARY KEY NOT NULL,

        NomeHotel VARCHAR(100) NOT NULL,

        Endereco VARCHAR(255) NOT NULL,

        Site VARCHAR(100)

);

CREATE TABLE FUNCIONARIOS (

        CPF VARCHAR(14) PRIMARY KEY NOT NULL,

        NomeFuncionario VARCHAR(100) NOT NULL,

        Salario DECIMAL(10, 2) NOT NULL,

        TelefoneFuncionario VARCHAR(20),

        EmailFuncionario VARCHAR(100),
```

```

        Tipo VARCHAR(50) NOT NULL,

        CNPJ_Hotel VARCHAR(18) NOT NULL,

        CONSTRAINT FK_Funcionarios_Hoteis FOREIGN KEY (CNPJ_Hotel) REFERENCES
HOTEIS (CNPJ)

);

CREATE TABLE GERENTES (

        CPF_Gerente VARCHAR(14) PRIMARY KEY NOT NULL,

        CONSTRAINT FK_Gerentes_Funcionarios FOREIGN KEY (CPF_Gerente)
REFERENCES FUNCIONARIOS (CPF)

);

CREATE TABLE QUARTOS (

        CodQuarto INT PRIMARY KEY NOT NULL,

        Numero VARCHAR(10) NOT NULL,

        Andar INT NOT NULL,

        CNPJ_Hotel VARCHAR(18) NOT NULL,

        CONSTRAINT FK_Quartos_Hoteis FOREIGN KEY (CNPJ_Hotel) REFERENCES
HOTEIS (CNPJ)

);

CREATE TABLE CLIENTES (

        CPF VARCHAR(14) PRIMARY KEY NOT NULL,

        NomeCliente VARCHAR(100) NOT NULL,

        TelefoneCliente VARCHAR(20),

        EmailCliente VARCHAR(100),

        Perfil VARCHAR(50) NOT NULL

);

```

```

CREATE TABLE DEPENDENTES (

    CPF_Cliente VARCHAR(14) NOT NULL,

    NomeDependente VARCHAR(100) NOT NULL,

    PRIMARY KEY (CPF_Cliente, NomeDependente),

    CONSTRAINT FK_Dependentes_Clientes FOREIGN KEY (CPF_Cliente)
REFERENCES CLIENTES(CPF)

);

CREATE TABLE RESERVAS (

    CodReserva INT PRIMARY KEY NOT NULL,

    DataReserva DATE NOT NULL,

    ContaDiaria DECIMAL(10, 2),

    CodQuarto INT NOT NULL,

    CPF_Cliente VARCHAR(14) NOT NULL,

    CONSTRAINT FK_Reservas Quartos FOREIGN KEY (CodQuarto) REFERENCES
QUARTOS(CodQuarto),

    CONSTRAINT FK_Reservas_Clientes FOREIGN KEY (CPF_Cliente) REFERENCES
CLIENTES(CPF)

);

CREATE TABLE CHECKINS (

    CodCheckin INT PRIMARY KEY NOT NULL,

    DataCheckin DATE NOT NULL,

    CodReserva INT NOT NULL UNIQUE,

    CPF_Cliente VARCHAR(14) NOT NULL,

    CONSTRAINT FK_Checkins_Reservas FOREIGN KEY (CodReserva) REFERENCES
RESERVAS(CodReserva),

    CONSTRAINT FK_Checkins_Clientes FOREIGN KEY (CPF_Cliente) REFERENCES
CLIENTES(CPF)

);

```

```

CREATE TABLE CHECKOUTS (

    CodCheckout INT PRIMARY KEY NOT NULL,

    DataCheckout DATE NOT NULL,

    CustoAdicionais DECIMAL(10, 2),

    CodCheckin INT NOT NULL UNIQUE,

    CONSTRAINT FK_Checkouts_Checkins FOREIGN KEY (CodCheckin) REFERENCES
CHECKINS (CodCheckin)

);

```

### **Código 1 – Criação do modelo físico do projeto no SQL Server.**

O código apresentado é uma conversão praticamente 1:1 da Figura 1, com algumas pequenas diferenças inseridas quando necessário devido às particularidades do SGDB escolhido, como IDs próprios para tabelas como CHECKINS e CHECKOUTS.

Para fins de teste, foram feitas 20 consultas com o banco de dados criado. As consultas, e seus resultados, serão exibidas em sequência.

### **Consulta 1: Inserção de dados de hotéis dentro das tabelas.**

Foram inseridos os seguintes dados, baseados em 5 hotéis reais de Campos do Jordão:

```

INSERT INTO HOTEIS VALUES

    ('46.745.006/0001-99', 'Toriba', 'Av. Ernesto Diederichsen, 2962',
'www.toriba.com.br'),

    ('28.653.131/0001-97', 'Quebra-Noz', 'Av. Emílio Lang Júnior, 323',
'quebranoz.com.br'),

    ('51.538.380/0001-53', 'Bendito Cacao', 'R. Dr. José Mestres, 2145',
'www.benditocacaoresort.com.br'),

    ('21.277.360/0001-05', 'Golden Park', 'Rod. Floriano Rodrigues Pinheiro,
2000', NULL),

    ('08.329.726/0002-65', 'Garnier', 'R. Joaquim Pinto Seabra, 420',
'www.hotelgarnier.com.br');

```

GO

Os dados foram inseridos com sucesso, e salvos na tabela HOTEIS. Note o site nulo na quarta inserção, conforme possibilitado.

### **Consulta 2:** Inserção de dados de clientes dentro das tabelas.

Foram inseridos 10 clientes (fictícios) na tabela CLIENTES, da seguinte forma:

```
INSERT INTO CLIENTES VALUES

    ('507.502.600-17', 'Brian Oliveira', '12345-6789', NULL, 'Local'),

    ('706.171.960-52', 'Paulo Giovani', '00000-0001',
'professor@yahoo.com', 'Local'),

    ('770.772.070-99', 'Hebe Camargo', '98765-4321',
'hebecamargo@hotmail.com', 'Local'),

    ('653.475.800-89', 'Lewis Hamilton', '12121-3434', 'fl@gmail.com',
'Estrangeiro'),

    ('867.260.320-64', 'Fernanda Torres', '56565-7878',
'oscar@yahoo.com', 'Local'),

    ('831.747.050-45', 'Virgínia Fonseca', NULL, 'tigrinho@gmail.com',
'Local'),

    ('722.266.950-79', 'Renato Russo', '90909-1234',
'legiaourbana@hotmail.com', 'Local'),

    ('203.927.250-02', 'Machado de Assis', NULL, NULL, 'Local'),

    ('468.557.940-25', 'Lionel Messi', '00666-0000',
'barcelona@gmail.com', 'Estrangeiro'),

    ('596.310.600-12', 'Yudi Tamashiro', '04002-8922',
'playstation2@uol.com.br', 'Local');
```

GO

A operação ocorreu com sucesso. Os CPFs são falsos, gerados através de ferramentas online para desenvolvedores, assim como as outras informações.

### **Consulta 3:** Inserção de dados de funcionários dentro das tabelas.

Foram inseridos dados de 5 funcionários (fictícios) na tabela FUNCIONARIOS, um para cada hotel anteriormente citado:

```
INSERT INTO FUNCIONARIOS VALUES
```

```
    ('182.927.670-02', 'Cristiano Ronaldo', 5000.00, NULL, NULL,  
'Funcionário', '46.745.006/0001-99'),  
  
    ('725.228.750-51', 'Neymar Júnior', 3500.00, NULL,  
'caicai@uol.com.br', 'Funcionário', '28.653.131/0001-97'),  
  
    ('190.987.150-87', 'Ronaldo Gaúcho', 4700.00, '01100-6975', NULL,  
'Funcionário', '51.538.380/0001-53'),  
  
    ('692.576.710-20', 'Sílvio Santos', 2200.50, '10000-5555',  
'maoe@gmail.com', 'Gerente', '21.277.360/0001-05'),  
  
    ('055.304.380-32', 'Ivete Sangalo', 6100.00, '99998-7771',  
'ivete@hotmail.com', 'Funcionário', '08.329.726/0002-65');
```

```
GO
```

**Sucesso.** Perceba que o CNPJ do hotel no qual o funcionário trabalha é uma chave estrangeira da tabela HOTEIS.

#### **Consulta 4:** Inserção de dados de quartos dentro das tabelas.

Foram inseridos na tabela QUARTOS três quartos para cada hotel, totalizando 15 entradas:

```
INSERT INTO QUARTOS VALUES
```

```
    (1, '1A', 1, '46.745.006/0001-99'),  
    (2, '2A', 1, '46.745.006/0001-99'),  
    (3, '3A', 1, '46.745.006/0001-99'),  
    (4, '1A', 1, '28.653.131/0001-97'),  
    (5, '1B', 2, '28.653.131/0001-97'),  
    (6, '1C', 3, '28.653.131/0001-97'),  
    (7, '1A', 1, '51.538.380/0001-53'),  
    (8, '2A', 1, '51.538.380/0001-53'),  
    (9, '1B', 2, '51.538.380/0001-53'),  
    (10, '1A', 1, '21.277.360/0001-05'),  
    (11, '1B', 2, '21.277.360/0001-05'),  
    (12, '2B', 2, '21.277.360/0001-05'),
```

```
(13, '1A', 1, '08.329.726/0002-65'),  
(14, '1C', 3, '08.329.726/0002-65'),  
(15, '1D', 4, '08.329.726/0002-65');
```

GO

A operação foi realizada com sucesso.

**Consulta 5:** Inserção de dados de dependentes dentro das tabelas.

Com a tabela CLIENTES já possuindo entradas, era possível fazer a inserção em DEPENDENTES, cuja chave primária é o CPF do cliente associado ao dependente. Foram inseridos 3 dependentes:

```
INSERT INTO DEPENDENTES VALUES  
  
( '831.747.050-45', 'Fortune Tiger'),  
( '596.310.600-12', 'Playstation Dois'),  
( '203.927.250-02', 'Dom Casmurro');
```

GO

As instruções foram executadas com êxito.

**Consulta 6:** Inserção de dados de gerentes dentro das tabelas.

A tabela FUNCIONARIOS permite especialização para a tabela GERENTES, e um dos funcionários inseridos necessita de tal especialização, que foi feita do seguinte modo:

```
INSERT INTO GERENTES VALUES ( '692.576.710-20' );
```

GO

Novamente, um sucesso.

**Consulta 7:** Verificação dos vínculos entre funcionários e seus respectivos hotéis.

Antes de fazer inserções em RESERVAS, CHECKINS e CHECKOUTS, era necessário verificar que a base estava funcionando como deveria. Portanto, primeiro

foi feito uma junção entre as tabelas HOTEIS e FUNCIONARIOS para ver se os funcionários estavam vinculados aos seus respectivos locais de trabalho:

```
SELECT NomeHotel AS 'Nome do Hotel',  
  
       NomeFuncionario AS 'Nome do Funcionário'  
  
FROM FUNCIONARIOS JOIN HOTEIS  
  
     ON FUNCIONARIOS.CNPJ_Hotel = HOTEIS.CNPJ;  
  
GO
```

Obteu-se o seguinte retorno:

	Nome do Hotel	Nome do Funcionário
1	Gamier	Ivete Sangalo
2	Toriba	Cristiano Ronaldo
3	Bendito Cacao	Ronaldinho Gaúcho
4	Golden Park	Sílvia Santos
5	Quebra-Noz	Neymar Júnior

Note o tratamento dado ao nome dos atributos com o comando “AS”.

**Consulta 8:** Verificação dos vínculos entre clientes e seus respectivos dependentes.

Similar ao que foi feito na Consulta 7, era necessário fazer a junção dos dependentes registrados com os clientes aos quais eram vinculados:

```
SELECT NomeCliente AS 'Nome do Cliente',  
  
       NomeDependente AS 'Nome do Dependente'  
  
FROM CLIENTES JOIN DEPENDENTES  
  
     ON CLIENTES.CPF = DEPENDENTES.CPF_Cliente;  
  
GO
```

O sistema retornou o seguinte:

	Nome do Cliente	Nome do Dependente
1	Machado de Assis	Dom Casmurro
2	Yudi Tamashiro	Playstation Dois
3	Virgínia Fonseca	Fortune Tiger



**Consulta 9:** Verificação dos vínculos entre funcionários e suas respectivas especializações.

Como última verificação antes da criação das reservas, verificou-se que o funcionário com a especialização de gerência estava registrado como tal no banco de dados:

```
SELECT NomeFuncionario AS 'Nome do Gerente',  
  
       Salario AS 'Salário',  
  
       TelefoneFuncionario AS 'Telefone do Gerente',  
  
       EmailFuncionario AS 'Email do Gerente'  
  
FROM FUNCIONARIOS JOIN GERENTES  
  
ON FUNCIONARIOS.CPF = GERENTES.CPF_Gerente;  
  
GO
```

Houve o retorno a seguir:

	Nome do Gerente	Salário	Telefone do Gerente	Email do Gerente
1	Silvio Santos	2200.50	10000-5555	maoe@gmail.com

**Consulta 10:** Inserção de dados de reservas dentro das tabelas.

Com a integridade do banco de dados assegurada, foi possível criar as reservas. Foram inseridas dez, uma para cada cliente:

```
INSERT INTO RESERVAS VALUES  
  
  (1, '25/04/2025', 250.00, 7, '507.502.600-17'),  
  
  (2, '24/04/2025', 110.25, 11, '706.171.960-52'),  
  
  (3, '26/04/2025', 125.75, 2, '770.772.070-99'),  
  
  (4, '11/03/2025', 220.00, 4, '653.475.800-89'),  
  
  (5, '30/01/2025', 150.00, 1, '867.260.320-64'),  
  
  (6, '28/02/2025', 300.00, 15, '831.747.050-45'),  
  
  (7, '31/12/2024', 100.00, 13, '722.266.950-79'),
```

```
(8, '02/05/2025', 105.50, 9, '203.927.250-02'),  
(9, '09/04/2025', 295.05, 5, '468.557.940-25'),  
(10, '15/02/2025', 305.75, 6, '596.310.600-12');
```

GO

O comando teve sucesso, e as reservas foram registradas, possibilitando inserções em CHECKINS.

**Consulta 11:** Inserção de dados de *check-ins* dentro das tabelas.

Com as reservas inseridas, posteriormente deveria se colocar seus equivalentes dentro da tabela CHECKINS quando aplicável, o que foi feito para 7 dos casos:

```
INSERT INTO CHECKINS VALUES  
  
(1, '20/05/2025', 1, '507.502.600-17'),  
(2, '19/05/2025', 2, '706.171.960-52'),  
(3, '21/05/2025', 4, '653.475.800-89'),  
(4, '20/05/2025', 5, '867.260.320-64'),  
(5, '24/05/2025', 7, '722.266.950-79'),  
(6, '22/05/2025', 9, '468.557.940-25'),  
(7, '17/05/2025', 10, '596.310.600-12');
```

GO

Com isso, se torna possível a inserção em CHECKOUTS.

**Consulta 12:** Inserção de dados de *check-outs* dentro das tabelas.

Para os sete valores inseridos em CHECKINS, foram inseridos quatro equivalentes em CHECKOUTS:

```
INSERT INTO CHECKOUTS VALUES  
  
(1, '23/05/2025', 10.00, 2),  
(2, '24/05/2025', 25.00, 4),  
(3, '25/05/2025', 0.00, 6),
```

```
(4, '24/05/2025', 38.00, 7);
```

GO

A partir de agora, se torna possível conduzir praticamente qualquer outro tipo de consulta, já que todas as tabelas estão preenchidas.

**Consulta 13:** Verificação dos vínculos entre clientes e suas respectivas reservas.

Foi feita uma junção entre as tabelas CLIENTES e RESERVAS para constatar que a integridade do banco foi mantida com as novas adições:

```
SELECT NomeCliente AS 'Nome do Cliente',  
  
       DataReserva AS 'Data da Reserva'  
  
FROM CLIENTES JOIN RESERVAS  
  
ON CLIENTES.CPF = RESERVAS.CPF_Cliente;
```

GO

A seguinte operação foi realizada:

	Nome do Cliente	Data da Reserva
1	Brian Oliveira	2025-04-25
2	Paulo Giovani	2025-04-24
3	Hebe Camargo	2025-04-26
4	Lewis Hamilton	2025-03-11
5	Fernanda Torres	2025-01-30
6	Virgínia Fonseca	2025-02-28
7	Renato Russo	2024-12-31
8	Machado de Assis	2025-05-02
9	Lionel Messi	2025-04-09
10	Yudi Tamashiro	2025-02-15

**Consulta 14:** Verificação dos vínculos entre reservas e *check-ins* já realizados.

Realizou-se uma nova junção envolvendo os valores de RESERVAS que possuíam correspondentes em CHECKINS:

```
SELECT RESERVAS.CPF_Cliente AS 'CPF do Cliente',  
  
       DataReserva AS 'Data da Reserva',
```

```

DataCheckin AS 'Data do Check-in'

FROM RESERVAS JOIN CHECKINS

ON RESERVAS.CodReserva = CHECKINS.CodReserva;

GO

```

A seguir, a resposta da seleção:

	CPF do Cliente	Data da Reserva	Data do Check-in
1	507.502.600-17	2025-04-25	2025-05-20
2	706.171.960-52	2025-04-24	2025-05-19
3	653.475.800-89	2025-03-11	2025-05-21
4	867.260.320-64	2025-01-30	2025-05-20
5	722.266.950-79	2024-12-31	2025-05-24
6	468.557.940-25	2025-04-09	2025-05-22
7	596.310.600-12	2025-02-15	2025-05-17

**Consulta 15:** Verificação dos vínculos entre *check-ins* e *check-outs* já realizados.

O processo feito foi similar ao da Consulta 14, mas entre os valores das tabelas CHECKINS e CHECKOUTS:

```

SELECT CPF_Cliente AS 'CPF do Cliente',

DataCheckin AS 'Data do Check-in',

DataCheckout AS 'Data do Check-out'

FROM CHECKINS JOIN CHECKOUTS

ON CHECKINS.CodCheckin = CHECKOUTS.CodCheckin;

GO

```

Resposta a seguir:

	CPF do Cliente	Data do Check-in	Data do Check-out
1	706.171.960-52	2025-05-19	2025-05-23
2	867.260.320-64	2025-05-20	2025-05-24
3	468.557.940-25	2025-05-22	2025-05-25
4	596.310.600-12	2025-05-17	2025-05-24

**Consulta 16:** Verificação dos vínculos entre quartos e seus respectivos hotéis.

Como um passo adicional para constatação da integridade, se realizou uma junção entre QUARTOS e HOTEIS:

```
SELECT NomeHotel AS 'Nome do Hotel',  
  
       CodQuarto AS 'Código do Quarto',  
  
       Numero AS 'Número do Quarto',  
  
       Andar AS 'Andar do Quarto'  
  
FROM QUARTOS JOIN HOTEIS  
  
     ON QUARTOS.CNPJ_Hotel = HOTEIS.CNPJ;  
  
GO
```

Junção esta que obteve o retorno:

	Nome do Hotel	Código do Quarto	Número do Quarto	Andar do Quarto
1	Toriba	1	1A	1
2	Toriba	2	2A	1
3	Toriba	3	3A	1
4	Quebra-Noz	4	1A	1
5	Quebra-Noz	5	1B	2
6	Quebra-Noz	6	1C	3
7	Bendito Cacao	7	1A	1
8	Bendito Cacao	8	2A	1
9	Bendito Cacao	9	1B	2
10	Golden Park	10	1A	1
11	Golden Park	11	1B	2
12	Golden Park	12	2B	2
13	Gamier	13	1A	1
14	Gamier	14	1C	3
15	Gamier	15	1D	4

**Consulta 17:** Verificação dos vínculos entre reservas e seus respectivos quartos.

RESERVAS, além do vínculo com CLIENTES, também possuía vínculo com QUARTOS que deveria ser verificado:

```
SELECT CPF_Cliente AS 'CPF do Cliente',  
  
       DataReserva AS 'Data da Reserva',  
  
       RESERVAS.CodQuarto AS 'Código do Quarto',
```

```

        Numero AS 'Número do Quarto',

        Andar AS 'Andar do Quarto'

FROM RESERVAS JOIN QUARTOS

        ON RESERVAS.CodQuarto = QUARTOS.CodQuarto;

GO

```

A verificação se provou um sucesso:

	CPF do Cliente	Data da Reserva	Código do Quarto	Número do Quarto	Andar do Quarto
1	507.502.600-17	2025-04-25	7	1A	1
2	706.171.960-52	2025-04-24	11	1B	2
3	770.772.070-99	2025-04-26	2	2A	1
4	653.475.800-89	2025-03-11	4	1A	1
5	867.260.320-64	2025-01-30	1	1A	1
6	831.747.050-45	2025-02-28	15	1D	4
7	722.266.950-79	2024-12-31	13	1A	1
8	203.927.250-02	2025-05-02	9	1B	2
9	468.557.940-25	2025-04-09	5	1B	2
10	596.310.600-12	2025-02-15	6	1C	3

**Consulta 18:** Verificação dos vínculos entre clientes e seus respectivos *check-ins* quando realizados.

Como última verificação, CHECKINS também possuía vínculo com CLIENTES:

```

SELECT NomeCliente AS 'Nome do Cliente',

        DataCheckin AS 'Data do Check-in'

FROM CLIENTES JOIN CHECKINS

        ON CLIENTES.CPF = CHECKINS.CPF_Cliente;

GO

```

E conforme exibido, o retorno:

	Nome do Cliente	Data do Check-in
1	Brian Oliveira	2025-05-20
2	Paulo Giovanni	2025-05-19
3	Lewis Hamilton	2025-05-21
4	Fernanda Torres	2025-05-20
5	Renato Russo	2025-05-24
6	Lionel Messi	2025-05-22
7	Yudi Tamashiro	2025-05-17

### Consulta 19: Atualização de dados das tabelas.

Com a integridade completamente garantida, fez-se necessário verificar se atualizações aos dados das tabelas através do comando “UPDATE” funcionariam. Como exemplo, foi feita uma atualização de 15% para o salário de todos os funcionários que recebiam menos de R\$4000,00:

```
UPDATE FUNCIONARIOS
```

```
SET Salario = Salario * 1.15
```

```
WHERE Salario < 4000;
```

```
GO
```

Usando uma seleção posteriormente para confirmar o resultado, constatou-se que os funcionários das linhas 4 e 5 receberam um reajuste conforme desejado, enquanto os outros valores permaneceram inalterados:

	CPF	NomeFuncionario	Salario	TelefoneFuncionario	EmailFuncionario	Tipo	CNPJ_Hotel
1	055.304.380-32	Ivete Sangalo	6100.00	99998-7771	ivete@hotmail.com	Funcionário	08.329.726/0002-65
2	182.927.670-02	Cristiano Ronaldo	5000.00	NULL	NULL	Funcionário	46.745.006/0001-99
3	190.987.150-87	Ronaldinho Gaúcho	4700.00	01100-6975	NULL	Funcionário	51.538.380/0001-53
4	692.576.710-20	Silvio Santos	2530.58	10000-5555	maoe@gmail.com	Gerente	21.277.360/0001-05
5	725.228.750-51	Neymar Júnior	4025.00	NULL	caicai@uol.com.br	Funcionário	28.653.131/0001-97

### Consulta 20: Exibição de valores de maneira ordenada conforme especificações.

A última consulta foi um teste do comando “ORDER BY”, usando as tabelas CLIENTES e RESERVAS para ordenar todos os clientes baseado em quem pagou mais pela sua diária:

```
SELECT NomeCliente AS 'Nome do Cliente',
```

```

DataReserva AS 'Data da Reserva',

ContaDiaria AS 'Diária'

FROM CLIENTES JOIN RESERVAS

ON CLIENTES.CPF = RESERVAS.CPF_Cliente

ORDER BY ContaDiaria DESC;

GO

```

A ordenação funcionou como solicitada:

	Nome do Cliente	Data da Reserva	Diária
1	Yudi Tamashiro	2025-02-15	305.75
2	Virgínia Fonseca	2025-02-28	300.00
3	Lionel Messi	2025-04-09	295.05
4	Brian Oliveira	2025-04-25	250.00
5	Lewis Hamilton	2025-03-11	220.00
6	Fernanda Torres	2025-01-30	150.00
7	Hebe Camargo	2025-04-26	125.75
8	Paulo Giovanni	2025-04-24	110.25
9	Machado de Assis	2025-05-02	105.50
10	Renato Russo	2024-12-31	100.00

### 3. Discussão

Ao montar o diagrama, já notou-se que o caráter generalista do projeto não permitiu maior especialização, dado que diferentes hotéis podem fornecer uma vasta gama de serviços, como restaurantes, spas, recreações, dentre outros. Logo, ele atende as necessidades mais básicas, mas cada usuário do sistema iria precisar realizar suas próprias modificações a fim de atender adequadamente suas necessidades.

Tendo isto dito, as operações essenciais estão todas funcionais e a conversão para o modelo físico foi realizada com sucesso necessitando apenas de mínimos ajustes por questões de *software*, portanto o projeto ainda é uma excelente base, por mais que talvez careça de maiores detalhes em áreas específicas.



## **4 CONCLUSÃO**

Retomando os objetivos inicialmente propostos, foi possível investigar as regras de negócio de um hotel e levantar os requisitos necessários para seu funcionamento com sucesso, além de elaborar um projeto em cima dos dados obtidos no processo. Contudo, propor o resultado para clientes de forma que seja feito uso profissional é inviável no momento, dada a enorme variedade dos hotéis atualmente em operação.

Pesquisas futuras podem usar o modelo fornecido de base para montar sistemas mais específicos, com operações de maior complexidade, e assim potencialmente obter capacidade de adesão comercial.

## REFERÊNCIAS

CASTELLI, Geraldo. **Gestão hoteleira**. 2. ed. São Paulo: Saraiva, 2016. 512 p.

VENISON, Peter. **Managing hotels**. Londres: Butterworth-Heinemann, 1983. 148 p.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009. 282 p.

INSTITUTO BRASILEIRO DE TURISMO — EMBRATUR. **Regulação Geral dos Meios de Hospedagem**. 2002.

ELMASRI, Ramez; NAVATHE, Shamkant. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Education do Brasil, 2011. 1152 p.

ROB, Peter; CORONEL, Carlos. **Sistemas de banco de dados: projeto, implementação e gerenciamento**. São Paulo: Cengage Learning, 2011. 711 p.

NETO, Dilson Jatahy Fonseca. **Turismo e hotelaria: comentários à legislação**. São Paulo: Liber Ars, 2016. 152 p.