

# Relatório Final



## Departamento de Engenharia Informática e de Sistemas

*[nome do curso]*

**Base de Dados**

2024/2025

### Tema

**Sistema de gestão de serviços e quartos de um hotel**

### Identificação dos Autores

Autores:

Nº Aluno	Prática	Nome	Email
2023141564P5		Ana Luísa Silva Guedes	a2023141564@isec.pt
2023133076P5		Gabriel Matos Dias	a2023133076@isec.pt
2023141751P7		Guilherme Lourenço Rosa Antunes Graça	a2023141751@isec.pt
2023134919P5		Rodrigo Ferreira Neves	a2023134919@isec.pt

Distribuição do esforço:

Tarefa	Ana Guedes	Gabriel Dias	Guilherme Graça	Rodrigo Neves
Análise da situação atual	25%	25%	25%	25%
Identificação do problema/melhorias	25%	25%	25%	25%
Determinar Solução	25%	25%	25%	25%
Modelo Conceptual e Físico	25%	25%	25%	25%
Relatório	25%	25%	25%	25%

## Cap 1 - Introdução

O objetivo deste projeto é desenvolver um sistema automatizado e eficiente para a gestão de um hotel, integrando todas as informações essenciais relacionadas às principais áreas operacionais, como quartos, hóspedes, reservas, funcionários e serviços de restaurante.

Este sistema não só armazena e organiza dados cruciais para o funcionamento diário do hotel, como também oferece uma visão abrangente e integrada das atividades. Isto facilita a tomada de decisões em tempo real, assegurando que as operações globais decorram de forma fluida e eficiente.

A base de dados deste sistema é relacional, composta por tabelas interligadas através de chaves primárias e estrangeiras, o que garante que todas as operações e interações entre os diferentes componentes estejam interligadas. Esta estrutura permite a automatização de muitos processos, reduzindo a necessidade de intervenção manual e minimizando o risco de erros operacionais.

No capítulo 2, faz-se um enquadramento da relevância da proposta no negócio de venda de livros da livraria, descreve-se a situação actual, os principais problemas encontrados e, por fim, descreve-se a proposta que irá permitir solucionar esses problemas e atingir o objectivo desejado. No capítulo 3 apresenta-se a análise de dados necessária para a solução proposta, o modelo conceptual da base de dados, descrevendo-se todas as entidades e os relacionamentos existentes entre elas. No capítulo 4, apresenta-se o modelo físico (ou diagrama de tabelas) da base de dados que vai servir de suporte ao sistema de informação proposto, assim como o respectivo *script* de criação da base de dados. Por fim, no capítulo 5 são apresentadas conclusões do trabalho desenvolvido.

## Cap 2 - Enquadramento da Proposta

Neste capítulo faz-se um enquadramento da proposta de trabalho para o sistema automatizado e eficiente para a gestão de um hotel. Na secção 2.1 descreve-se o funcionamento actual do hotel, apresentando-se na secção 2.2 o conjunto de problemas existentes actualmente e que se pretendem solucionar. A secção 2.3 descreve a proposta de solução que permitirá solucionar os problemas existentes e satisfazer as necessidades encontradas.

## Sec 2.1 - Diagnóstico da Situação Atual

Esta base de dados serve para gerir um hotel que fornece serviços de alta qualidade aos clientes que o visitam.

O hotel dispõe de uma bela recepção para os hóspedes, onde estes terão de fazer o check-in que é concebido com todas as informações dos hóspedes.

No check-in os hóspedes poderão escolher, a seu gosto, o tipo de quarto que desejam de acordo com a disponibilidade. A partir desta informação cabe aos funcionários gerirem o bom funcionamento do quarto, tendo sempre a limpeza em dia.

Apenas os hóspedes podem ter acesso ao restaurante do hotel.

## Sec 2.2 - Problemas encontrados

Um dos principais problemas é a grande procura existente no hotel na época alta e escassez de procura na época baixa, no que se manifesta na procura imensa para os quartos deste hotel apenas em alguns meses. Este é um dos principais problemas porque poderá não existir disponibilidade para alguns quartos que são mais procurados pelas pessoas. Na época baixa devido á escassez da procura, os recursos humanos e materiais podem ser geridos de forma a otimizar os gastos e a organização do hotel, tal como dispensar alguns funcionários e diminuir o stock de alimentos.

Relativamente aos funcionários existentes no hotel, pode acontecer de faltar a limpeza de um quarto, ou troca de lençóis e não haver disponibilidade por parte dos funcionários de efetuar esse serviço, ou não existir funcionários suficientes para o número de serviços necessários. Uma possível melhoria para este problema seria a correta distribuição dos funcionários pelos quartos, recepção e restaurante.

Na área da restauração, teremos que ser inclusivos na comida existente. Teremos que aumentar a variedade de comida, porque se existir uma pessoa vegetariana ou com uma alimentação mais restrita, teremos a solução ideal.

## Sec 2.3 - Descrição da Solução Proposta

Este sistema de base de dados organiza e controla as operações de um hotel, abrangendo reservas, gestão de hóspedes, quartos, refeições e funcionários.

**Reservas:** A base de dados guarda as reservas feitas pelos clientes, armazenando informações como o estado da reserva, número de noites e o número de hóspedes. Cada reserva está associada a um ou mais quartos e é feita exclusivamente por um cliente.

**Quartos:** Cada quarto tem características específicas, como o tipo (simples ou duplo), capacidade de ocupação, custo por noite, disponibilidade e estado da limpeza. Um quarto pode estar associado a apenas uma reserva durante a ocupação dos quartos, mas cada reserva pode incluir múltiplos quartos.

**Hóspedes:** Os hóspedes têm informações como nome, contacto, NIF, e-mail e data de nascimento. Eles podem fazer múltiplas reservas e, durante a estadia, podem ocupar vários quartos e pedir refeições específicas.

**Alimentação:** O hotel tem como serviço de alimentação apenas um buffet. Vários funcionários do hotel estão envolvidos no preparo e no serviço, tal como o chef (cozinheiro), e os empregados de mesa.

**Funcionários:** Os funcionários desempenham funções específicas:

- **Rececionistas:** São responsáveis por processar as entradas.
- **Empregados de mesa:** Apresentam o menu de bebidas e entregam as mesmas.

- **Empregados de limpeza:** Garantem a manutenção e a higienização dos quartos.
- **Cozinheiros:** Preparam a comida servida aos hóspedes.

#### Interações e Relações:

- Cada reserva é vinculada a um cliente específico e pode incluir múltiplos quartos.
- Hóspedes e quartos têm uma relação de muitos-para-muitos, já que um hóspede pode ocupar vários quartos e cada quarto pode ser compartilhado entre hóspedes diferentes.
- A comida é preparada pelos cozinheiros, servida pelos empregados de mesa e solicitadas pelos hóspedes.
- Os quartos são limpos por empregados, que podem realizar essa tarefa em vários quartos, compartilhando a responsabilidade com outros colegas.

Esse sistema garante a integração e a eficiência no gerenciamento do hotel, permitindo que todas as operações sejam realizadas de forma coordenada e bem organizada.

## Cap 3 - Análise de Dados

Neste capítulo pretende-se apresentar o modelo conceptual da base de dados e descrever detalhadamente todas as entidades e relacionamentos envolvidos na solução proposta.

### Sec 3.1 - Entidades

Nesta secção vão ser descritas todas as Entidades relevantes para o o nosso hotel. Após uma análise aprofundada do modelo, constatou-se a necessidade das seguintes Entidades:

- Hospede
- Reserva
- Quarto
- tipo\_quarto
- Metodo\_pagamento
- Entrada
- Funcionarios
- Buffet
- Bebida

#### Sec 3.1.1 - Hospede

A entidade Hospede representa a informação relativa aos hóspedes que irão frequentar o hotel. É inserido um novo registo nesta entidade sempre que uma nova pessoa der check-in no hotel.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
id_hospede	Numérico (4 dígitos)	Código interno único a cada hóspede. É um número de 4 dígitos gerado sequencialmente. Ex. 1111
NIF	Numérico (9 dígitos)	Número de identificação fiscal de cada cliente. Ex. 123456789
Nome	30 Caracteres	Nome do hóspede. Ex. Gabriel Dias
Email	30 Caracteres	Email do hóspede. Ex. gabriel@isec.pt
Data_nascimento	Data	Data de nascimento do hóspede. Ex. 02/01/2000
Num_telemovei	Numérico (9 dígitos)	Contacto telefónico do Hóspede. Ex. 912258852

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
------------------	---------------	-----------------	-------------

id_hospede	N	S	Identificador (chave primária), não admite nulos. Não existem dois hóspedes com o mesmo código.
NIF	N	S	Não admite nulos. Não existem dois hóspedes com o mesmo NIF.
Nome	N	N	Não admite nulos. Dois hóspedes podem ter o mesmo nome.
Email	N	N	Não admite valor nulos.
Data_nascimento	N	N	Data de nascimento. Não admite nulos Pode haver várias datas de nascimento iguais.
Num_telemovei	N	N	Valor não nulo positivo.

## Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
responsavel	1:N	Reserva	Hospede
contem	N:M	Reserva	

alojado	N:M	Quarto	
pede	1:N		

### Sec 3.1.2 - Entidade Reserva

A entidade Reserva representa a informação relativa às reservas feitas no hotel. Esta entidade guarda todos os detalhes, tal como o número de noites, estado da reserva e número de pessoas em cada reserva.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
id_reserva	Numérico (4 dígitos)	Código interno único atribuído a cada reserva. É um número de 4 dígitos gerado sequencialmente. Ex. 1111
estado_reserva	10 Caracteres	Pago, por pagar, meio pago, cancelado
num_noites	Numérico (3 dígitos)	Número de noites frequentadas no hotel. Ex. 12
n_pessoas	Numérico (2 dígitos)	Número de pessoas presentes na reserva. Ex. 10

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
id_reserva	N	S	Identificador (chave primária), não admite nulos. Não existem duas reservas com o mesmo código.

estado_reserva	N	N	Não admite valor nulos.
num_noites	N	N	Não admite valor nulos.
n_pessoas	N	N	Não admite valor nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
paga	1:N	metodo_pagamento	metodo_pagamento
contem	N:M	hospede	
responsavel	1:N	hospede	hospede
tem	N:M	quarto	
exige	N:M	entrada	

### Sec 3.1.3 - Entidade Quarto

A entidade Quarto representa a informação relativa aos quarto presentes na reserva. Esta entidade guarda todos os detalhes sobre o quarto, tal como a disponibilidade, capacidade e preço.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
------------------	---------------	-----------



id_quarto	Numérico (4 dígitos)	Código interno único atribuído a cada quarto. É um número de 4 dígitos gerado sequencialmente. Ex. 1111
Disponibilidade	10 Caracteres	Disponibilidade do quarto. ex: Livre, Ocupado
Capacidade	Numérico (2 dígitos)	Número máximo de pessoas que o quarto aloja. Ex. 12
Preco_quarto	Float (6 dígitos)	Preço do quarto. Ex. 50.56

## Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
id_quarto	N	S	Identificador (chave primária), não admite nulos. Não existem dois quartos com o mesmo código.
Disponibilidade	N	N	Não admite valor nulos.
Capacidade	N	N	Não admite valor nulos.
Preco_quarto	N	N	Não admite valor nulos.

## Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
tem	N:M	reserva	
tipo_quarto	1:N	tipo_quarto	tipo_quarto

relationship_14		limpa	
alojado	N:M	hospede	

### Sec 3.1.4 - Entidade tipo\_quarto

A entidade tipo\_quarto representa a informação relativa aos tipos de cada quarto. Esta entidade guarda todos os detalhes sobre as características do quarto.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
id_tipo	Numérico (4 dígitos)	Código interno único atributo a cada tipo quarto. É um número de 4 dígitos gerado sequencialmente. Ex. 1111
designacao	30 Caracteres	Designação do quarto. Ex: Quarto duplo com vista mar.

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
id_tipo	N	N	Identificador (chave primária), não admite nulos. Podem existir dois quartos com o mesmo código de tipo de quarto.
designacao	N	N	Não admite valor nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
tipo_quarto	1:N	quarto	

### Sec 3.1.5 - Entidade Metodo\_pagamento

A entidade Metodo\_pagamento representa a informação relativa ao pagamento da reserva.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
id_pagamento	Numérico (4 dígitos)	Código interno único atributo ao pagamento. É um número de 4 dígitos gerado sequencialmente. Ex. 1111
tipo	20 Caracteres	Descrição do tipo de pagamento. Ex: Pagamento total em dinheiro.

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
id_pagamento	N	S	Identificador (chave primária), não admite nulos. Não podem existir dois pagamentos com o mesmo código.
tipo	N	N	Não admite valor nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
paga	1:N	reserva	metodo_pagamento

### Sec 3.1.6 - Entidade Entradas

A entidade Entradas representa a informação relativa á entrada dos hóspedes no hotel. Esta entidade guarda todos os detalhes sobre nome dos hóspedes, preço total, check-in e check-out.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
id_entrada	Numérico (4 dígitos)	Código interno único atributo a cada entrada. É um número de 4 dígitos gerado sequencialmente. Ex. 1111
nome_hospedes	30 Caracteres	Nome dos hóspedes. Ex: Guilherme Graça.
preco_total	Float (6 dígitos)	Preço total da reserva dos quartos Ex. 925.36
checkin	Date	Data do check-in Ex: 01/03/2025
checkout	Date	Data do check-out Ex: 09/03/2025

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
id_entrada	N	S	Identificador (chave primária), não admite nulos. Não existem duas entradas com o mesmo código.
nome_hospedes	N	N	Não admite valor nulos.
preco_total	N	N	Não admite valor nulos.
checkin	N	N	Não admite valor nulos.
checkout	N	N	Não admite valor nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
exige	N:M	reserva	
processa	1:N	recepcionista	recepcionista

### Sec 3.1.7 - Entidade Funcionários

A entidade Funcionários representa a informação relativa aos funcionários do hotel. Esta entidade guarda informações como o id do funcionario, o turno e o nome do mesmo.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
------------------	---------------	-----------

id_funcionarios	Numérico (4 dígitos)	Código interno único atributo a cada Funcionário. É um número de 4 dígitos gerado sequencialmente. Ex. 1111
Turno	10 Caracteres	Descrição do turno. Ex: Manhã, Tarde, Noite
Nome	30 Caracteres	Nome do funcionário. Ex: Ana Guedes

## Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
id_funcionarios	N	S	Identificador (chave primária), não admite nulos. Não existem dois funcionários com o mesmo código.
Turno	N	N	Não admite valor nulos.
Nome	N	N	Não admite valor nulos.

## Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
------------------------	---------------	----------------------	-----------------------------

cargo	1:1	Empregado_Limpeza, Recessionista, Cozinheiro, Empregado_Mesa	
-------	-----	--	--

Sec 3.1.8- Entidade Buffet

A entidade Buffet representa a informação relativa ao buffet disponível. Esta entidade guarda todos os detalhes sobre nome a comida.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
id_buffet	Numérico (2 dígitos)	Código interno único atributo a cada buffet. É um número de 2 dígitos gerado sequencialmente. Ex. 11
descricao	20 Caracteres	Descrição do buffet. Ex: Buffet de comida mediterrânea.
data	Date	Data em que o buffet esteve disponível. Ex: 01/12/2024

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
id_buffet	N	S	Identificador (chave primária), não admite nulos. Podem existir dois buffets com o mesmo código.

descricao	N	N	Não admite valor nulos.
data	N	N	Não admite valor nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
faz	N:M	cozinheiro	
pede	1:N	hospede, bebida	

### Sec 3.1.8- Entidade Bebida

A entidade Bebida representa a informação relativa às bebidas disponíveis. Esta entidade guarda todos os detalhes sobre as bebidas pedidas pelo hospede.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
id_bebida	Numérico (2 dígitos)	Código interno único atributo a cada bebida. É um número de 2 dígitos gerado sequencialmente. Ex. 11
nome_bebida	20 Caracteres	Descrição da bebida. Ex: Lata de sumol laranja



preco	Float (4 dígitos)	Preço da bebida. Ex: 1.20
-------	----------------------	------------------------------

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
id_bebida	N	S	Identificador (chave primária), não admite nulos. Não podem existir duas bebidas com o mesmo código.
nome_bebida	N	N	Não admite valor nulos.
preco	N	N	Não admite valor nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
serve	N:M	empregado_mesa	
nome_bebida	1:N	buffet,hospede	

## Sec 3.2 - Relacionamentos

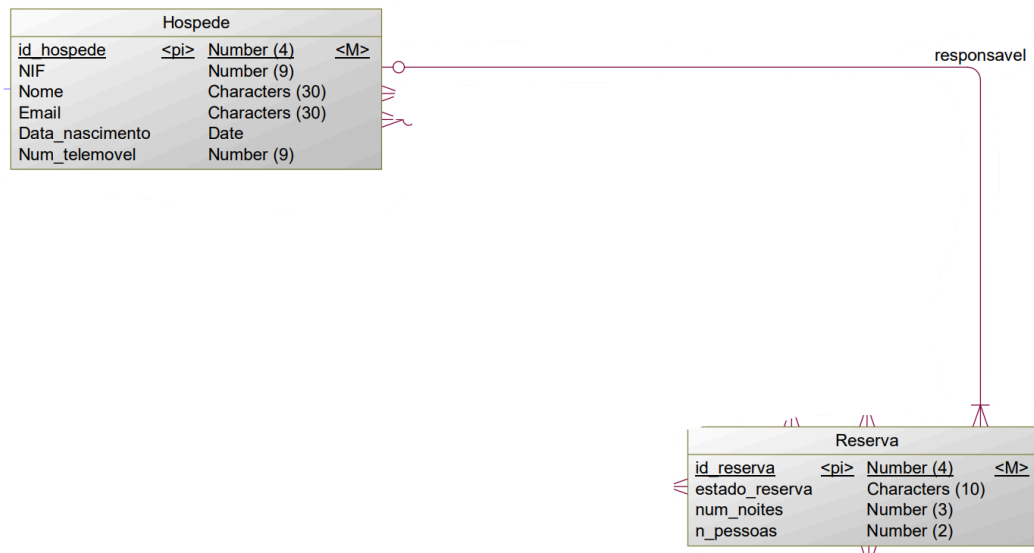
Nesta secção são descritos todos os relacionamentos existentes entre as várias entidades. Após uma análise aprofundada do sistema de gestão de serviços e quartos de um hotel, constatou-se a necessidade dos seguintes relacionamentos:

- Responsável
- Contem
- Paga
- Alojado
- Tem
- Exige
- Tipo\_Quarto
- Processa
- Relationship\_14
- Relationship\_13
- Pedes
- Serve
- Faz

(Nota: Nas imagens as participações obrigatórias estão do lado oposto ao considerado quando usamos o modelo de *Chen*, ou quando fazemos a ficha sobre *Power Designer*. Estão no lado usado por defeito pelo *Power Designer*. Devem usar como fazemos na aula)

### Sec 3.2.1 - Relacionamento: Responsável

Este relacionamento pretende expressar o relacionamento existente entre as Entidades HOSPEDE e RESERVA, no que concerne ao responsável da reserva. O objectivo é expressar qual o HOSPEDE responsável pela RESERVA.



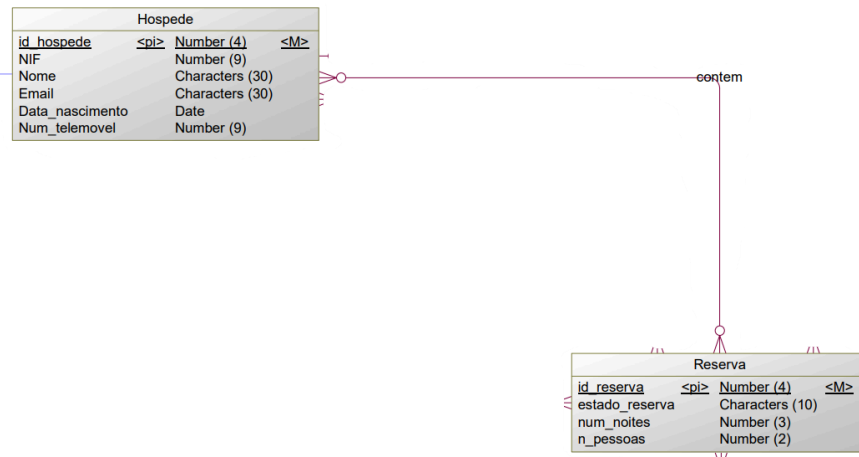
Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
HOSPEDE	NÃO	1 : N	SIM	RESERVA
Observações				
<ul style="list-style-type: none"> <li>Um hospede pode ser responsável por várias reservas.</li> <li>As reservas são obrigatoriamente reservadas apenas por um hospede.</li> </ul>				

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_HOSPEDE	N	S	PRIMARY KEY, identifica o hospede responsável
ID_RESERVA	N	S	PRIMARY KEY, identifica a reserva

### Sec 3.2.2 - Relacionamento: Contem

Este relacionamento pretende expressar o relacionamento existente entre as Entidades HOSPEDE e RESERVA, no que concerne aos hospedes que a reserva contem.



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
----------	-------------	---------------	-------------	----------

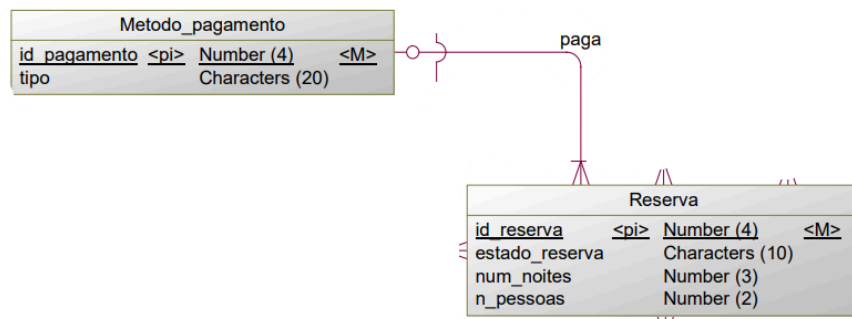
HOSPEDE	NÃO	N : M	NÃO	RESERVA
Observações				
<ul style="list-style-type: none"> <li>Um hospede pode conter a várias reservas.</li> <li>A reserva pode conter vários hospedes.</li> </ul>				

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_HOSPEDE	N	S	PRIMARY KEY, identifica os hospedes
ID_RESERVA	N	S	PRIMARY KEY, identifica a reserva

### Sec 3.2.3 - Relacionamento: Paga

Este relacionamento pretende expressar o relacionamento existente entre as Entidades METODO\_PAGAMENTO e RESERVA.



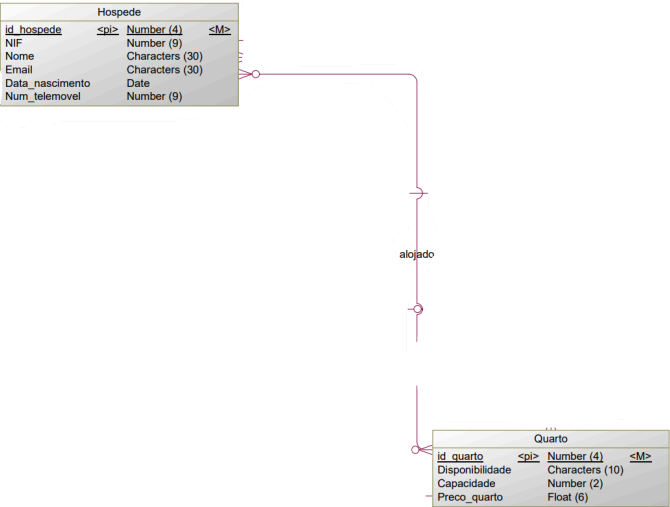
Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
METODO_PAGAMENTO	Sim	1 : N	Não	RESERVA
Observações				
<ul style="list-style-type: none"> <li>Uma reserva é processada por um método de pagamento.</li> <li>Um método de pagamento pode pagar varias reservas.</li> </ul>				

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_PAGAMENTO	N	S	PRIMARY KEY, identifica o pagamento
ID_RESERVA	N	S	PRIMARY KEY, identifica a reserva

Sec 3.2.4 - Relacionamento: Alojado

Este relacionamento pretende expressar o relacionamento existente entre as Entidades HOSPEDE e QUARTO, no que concerne à estadia de um hospede num quarto. O objetivo é saber que hóspedes ficam em que quartos.



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
HOSPEDE	Não	M : N	Não	QUARTO
Observações				

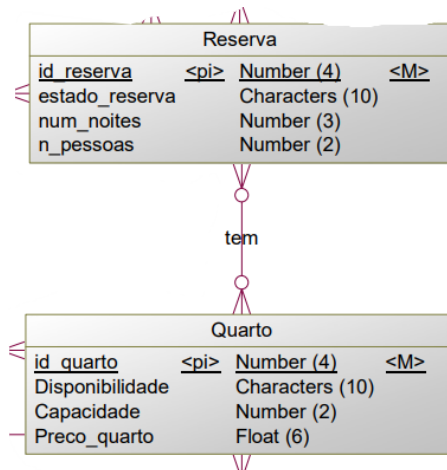
- Um quarto pode ter vários hóspedes.
- Vários hóspedes podem ter vários quartos.

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_QUARTO	N	S	PRIMARY KEY, identifica o quarto
ID_HOSPEDE	N	S	PRIMARY KEY, identifica o hóspede

### Sec 3.2.5 - Relacionamento: Tem

Este relacionamento pretende expressar o relacionamento existente entre as Entidades RESERVA e QUARTO, no que concerne aos quartos de cada reserva. O objetivo é expressar quais os quartos reservados na mesma reserva.



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
RESERVA	Não	M : N	Não	QUARTO
Observações				

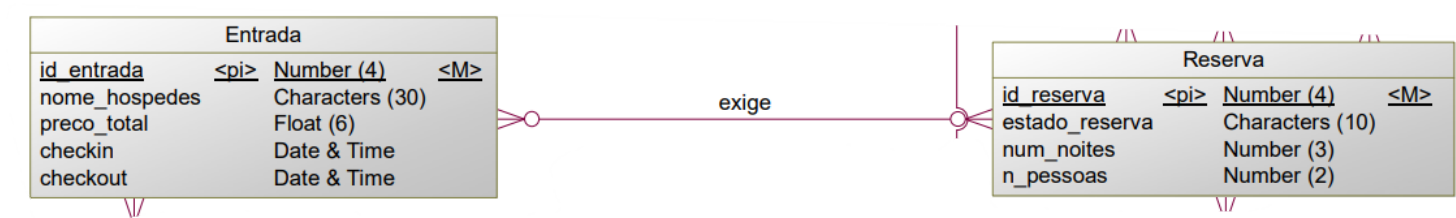
- Uma reserva pode ter vários quartos.

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_RESERVA	N	S	PRIMARY KEY, identifica a reserva
ID_QUARTO	N	S	PRIMARY KEY, identifica o quarto

Sec 3.2.6 - Relacionamento: Exige

Este relacionamento pretende expressar o relacionamento existente entre as Entidades ENTRADA e RESERVA.



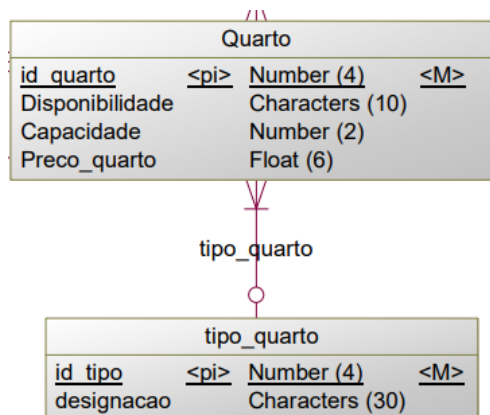
Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
ENTRADA	SIM	N : M	Não	RESERVA
Observações				
• Várias reservas podem ter várias entradas.				

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_ENTRADA	N	S	PRIMARY KEY, identifica a entrada
ID_RESERVA	N	S	PRIMARY KEY, identifica a reserva

### Sec 3.2.7 - Relacionamento: Tipo\_Quarto

Este relacionamento pretende expressar o relacionamento existente entre as Entidades QUARTO e TIPO\_QUARTO, no que concerne ao tipo do quarto. O objetivo é expressar qual o tipo do quarto.



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
QUARTO	Não	N : 1	Sim	TIPO_QUARTO
Observações				
<ul style="list-style-type: none"> <li>Vários quartos podem ter o mesmo tipo</li> <li>Um quarto só pode ter um tipo de quarto</li> </ul>				

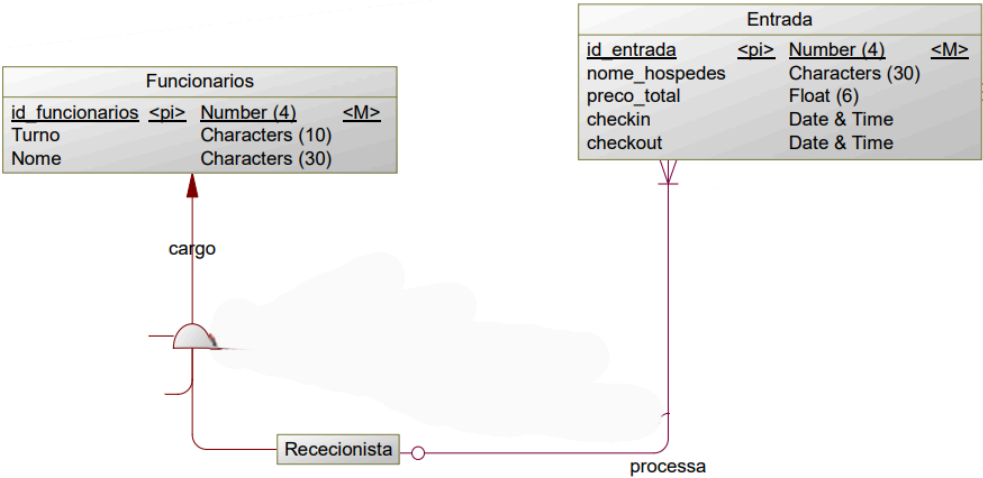
Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_QUARTO	N	S	PRIMARY KEY, identifica o quarto
ID_TIPO	N	S	PRIMARY KEY, identifica o tipo de quarto

### Sec 3.2.8 - Relacionamento: Processa

Este relacionamento pretende expressar o relacionamento existente entre as Entidades FUNCIONARIOS e ENTRADA, no que concerne à autoria dos livros. O objetivo é expressar quais os Funcionário que processa a entrada dos hóspedes





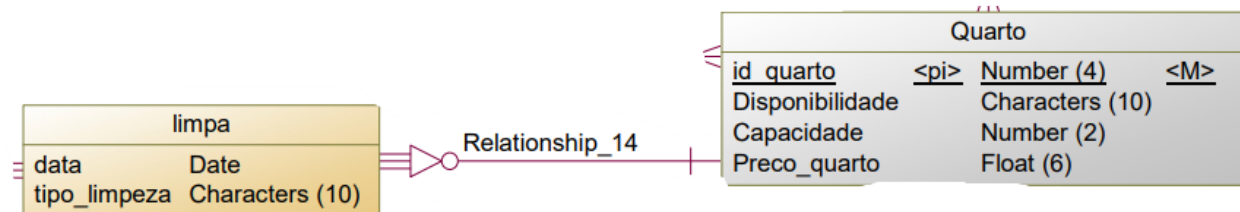
Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
FUNCIONARIOS	SIM	1 : N	Não	ENTRADA
Observações				
<ul style="list-style-type: none"><li>Um funcionário pode processar varias reservas</li><li>Uma reserva só pode ser processada por um rececionista</li></ul>				

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_FUNCIONARIO	N	S	PRIMARY KEY, identifica o funcionário
ID_ENTRADA	N	S	PRIMARY KEY, identifica a entrada

Sec 3.2.9 - Relacionamento: Relationship\_14

Este relacionamento pretende expressar o relacionamento existente entre as Entidades EMPREGADO\_LIMPEZA e LIMPA, no que concerne quem vai fazer as limpezas. O objectivo é expressar qual o EMPREGADO\_LIMPEZA que vai fazer a limpeza.



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
LIMPA	NÃO	N : 1	SIM	QUARTO
Observações				
<ul style="list-style-type: none"> <li>Um quarto é limpo várias vezes.</li> <li>As limpezas são obrigatoriamente feitas por quarto.</li> </ul>				

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
TIPO_LIMPEZA	N	N	Identifica o tipo de limpeza que vai ser feita no quarto
ID_QUARTO	N	S	PRIMARY KEY, identifica o quarto

### Sec 3.2.10 - Relacionamento: Relationship\_13

Este relacionamento pretende expressar o relacionamento existente entre as Entidades LIMPA e QUARTO, no que concerne a limpeza de cada quarto. O objectivo é saber qual o QUARTO que será limpo.



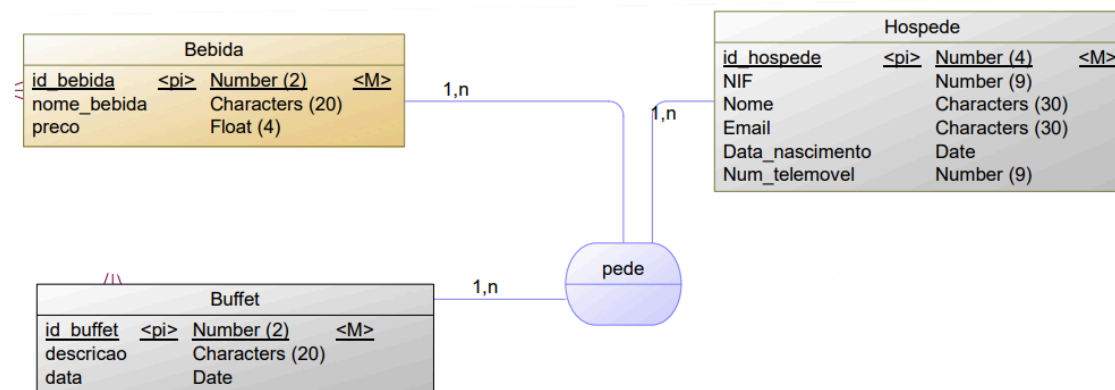
Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
EMPREGADO_LIMPEZA	SIM	1 : N	Não	LIMPA
Observações				
<ul style="list-style-type: none"> <li>Um empregado de limpeza faz obrigatoriamente várias limpezas.</li> <li>As limpezas são feitas por um empregado de limpeza.</li> </ul>				

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
TIPO_LIMPEZA	N	N	Identifica o tipo de limpeza que vai ser feita no quarto
DATA	N	N	Data em que cada limpeza foi efetuada

### Sec 3.2.11 - Relacionamento: Pede

Este relacionamento pretende expressar o relacionamento existente entre as Entidades FUNCIONARIOS e ENTRADA, no que concerne à autoria dos livros. O objetivo é expressar quais os Funcionário que processa a entrada dos hóspedes



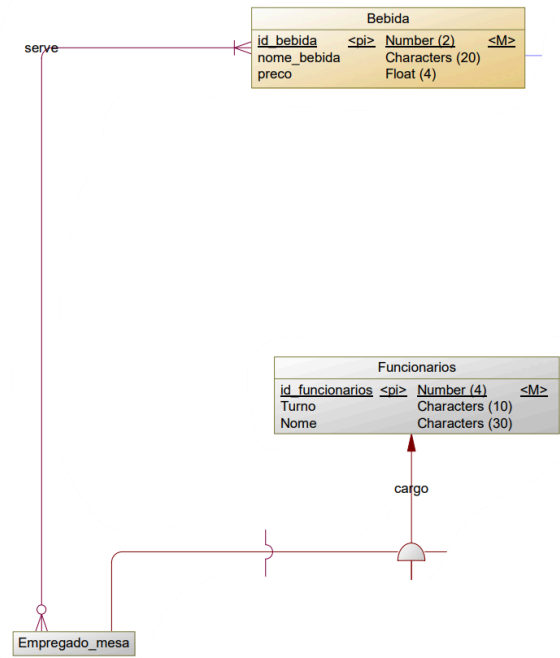
Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
BEBIDA	NÃO	M : N	NÃO	HOSPEDE	NÃO	N : P	NÃO	BUFFET
Observações								
<ul style="list-style-type: none"> <li>• Um hospede pode pedir várias bebidas.</li> <li>• Um hospede pode pedir vários buffets.</li> <li>• Uma bebida pode ser pedida por vários hospedes.</li> <li>• Um buffet pode ser pedido por vários hospedes.</li> </ul>								

Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_BEBIDA	N	S	PRIMARY KEY, identifica a bebida a ser pedida
ID_BUFFET	N	S	PRIMARY KEY, identifica o buffet a ser pedido
ID_HOSPEDE	N	S	PRIMARY KEY, identifica o hospede que pede

### Sec 3.2.12 - Relacionamento: Serve

Este relacionamento pretende expressar o relacionamento existente entre as Entidades EMPREGADO\_MESA e BEBIDA, no que concerne ao empregado de mesa que serve. O objectivo é saber qual o EMPREGADO\_MESA que servirá a BEBIDA.



Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
EMPREGADO_MESA	NÃO	N : M	SIM	BEBIDA
Observações				
<ul style="list-style-type: none"><li>Vários empregados de mesa servem várias bebidas.</li><li>Várias bebidas são obrigatoriamente servidas por vários empregados de mesa.</li></ul>				

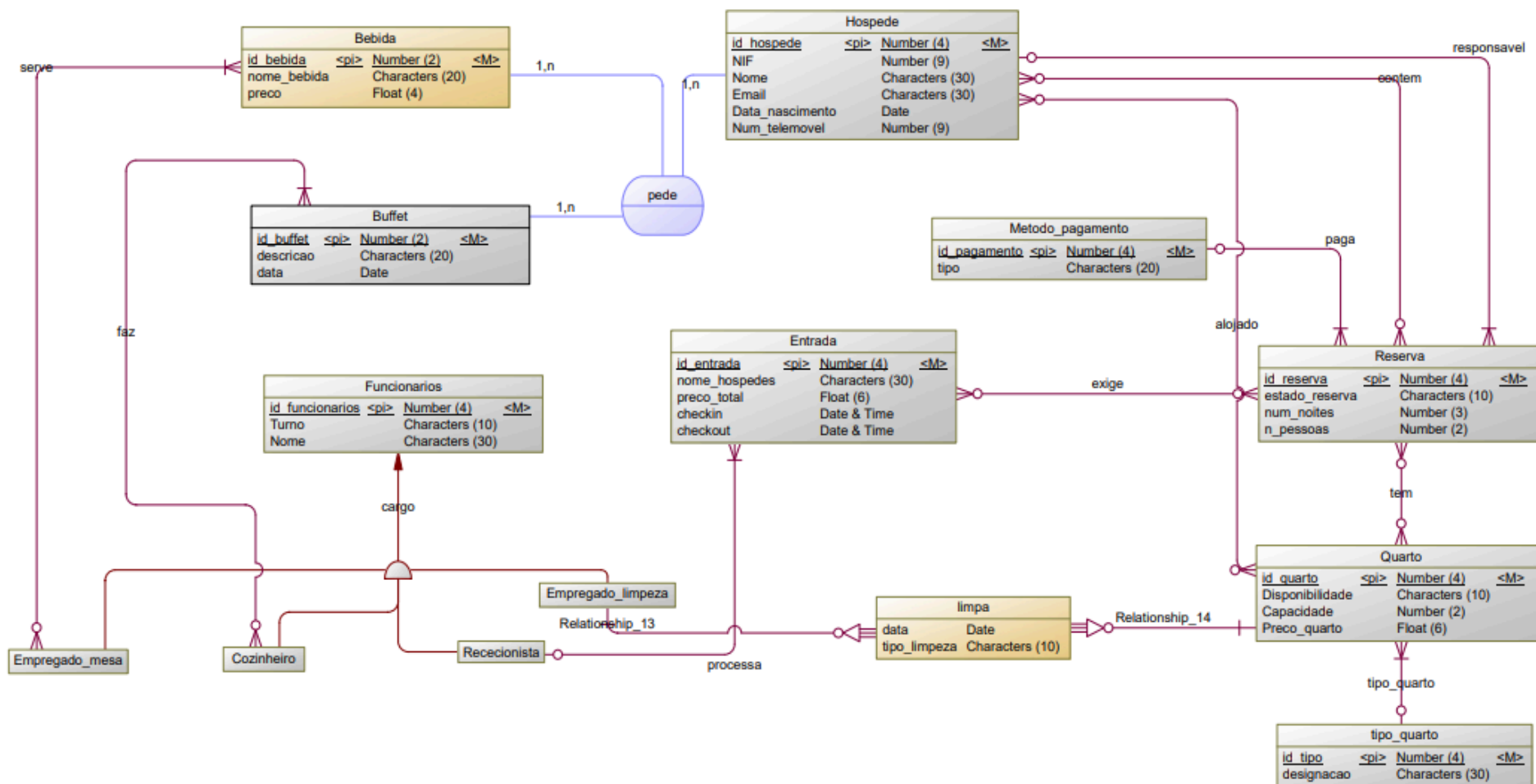
Atributos relevantes do Relacionamento:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
ID_BEBIDA	N	S	Indica a bebida a ser servida
NOME_BEBIDA	N	N	Indica o nome da bebida



## Sec 3.3 - Diagrama do Modelo Conceptual

O modelo conceptual de Entidade/Relacionamento completo é o seguinte:







Nesta secção apresentam-se as instruções SQL necessárias para criar as tabelas descritas anteriormente na secção 7.1 no SGBDR Oracle. O código SQL apresentado permite criar as tabelas, as restrições de integridade suportadas pelo SGBD, assim como as validações de dados definidas e valores por omissão.

```
/*=====*/  
/* DBMS name:   ORACLE Version 11g           */  
/* Created on:  13/12/2024 18:56:07          */  
/*=====*/
```

```
alter table ALOJADO  
  drop constraint FK_ALOJADO_ALOJADO_HOSPEDE;  
  
alter table ALOJADO  
  drop constraint FK_ALOJADO_ALOJADO2_QUARTO;  
  
alter table CONTEM  
  drop constraint FK_CONTEM_CONTEM_RESERVA;  
  
alter table CONTEM  
  drop constraint FK_CONTEM_CONTEM2_HOSPEDE;  
  
alter table COZINHEIRO  
  drop constraint FK_COZINHEI_CARGO3_FUNCIONA;  
  
alter table EMPREGADO_LIMPEZA  
  drop constraint FK_EMPREGAD_CARGO2_FUNCIONA;  
  
alter table EMPREGADO_MESA  
  drop constraint FK_EMPREGAD_CARGO4_FUNCIONA;  
  
alter table ENTRADA  
  drop constraint FK_ENTRADA_PROCESSA_RECECION;  
  
alter table EXIGE  
  drop constraint FK_EXIGE_EXIGE_ENTRADA;  
  
alter table EXIGE  
  drop constraint FK_EXIGE_EXIGE2_RESERVA;  
  
alter table FAZ  
  drop constraint FK_FAZ_FAZ_COZINHEI;
```

```
alter table FAZ
  drop constraint FK_FAZ_FAZ2_BUFFET;

alter table LIMPA
  drop constraint FK_LIMPA_RELATIONS_EMPREGAD;

alter table LIMPA
  drop constraint FK_LIMPA_RELATIONS_QUARTO;

alter table PEDE
  drop constraint FK_PEDE_PEDE_HOSPEDE;

alter table PEDE
  drop constraint FK_PEDE_PEDE2_BEBIDA;

alter table PEDE
  drop constraint FK_PEDE_PEDE3_BUFFET;

alter table QUARTO
  drop constraint FK_QUARTO_RELATIONS_TIPO_QUA;

alter table RECECIONISTA
  drop constraint FK_RECECION_CARGO_FUNCIONA;

alter table RESERVA
  drop constraint FK_RESERVA_PAGA_METODO_P;

alter table RESERVA
  drop constraint FK_RESERVA_RESPONSAV_HOSPEDE;

alter table SERVE
  drop constraint FK_SERVE_SERVE_EMPREGAD;

alter table SERVE
  drop constraint FK_SERVE_SERVE2_BEBIDA;

alter table TEM
  drop constraint FK_TEM_TEM_RESERVA;

alter table TEM
  drop constraint FK_TEM_TEM2_QUARTO;

drop index ALOJADO2_FK;

drop index ALOJADO_FK;
```

```
drop table ALOJADO cascade constraints;  
drop table BEBIDA cascade constraints;  
drop table BUFFET cascade constraints;  
drop index CONTEM2_FK;  
drop index CONTEM_FK;  
drop table CONTEM cascade constraints;  
drop table COZINHEIRO cascade constraints;  
drop table EMPREGADO_LIMPEZA cascade constraints;  
drop table EMPREGADO_MESA cascade constraints;  
drop index PROCESSA_FK;  
drop table ENTRADA cascade constraints;  
drop index EXIGE2_FK;  
drop index EXIGE_FK;  
drop table EXIGE cascade constraints;  
drop index FAZ2_FK;  
drop index FAZ_FK;  
drop table FAZ cascade constraints;  
drop table FUNCIONARIOS cascade constraints;  
drop table HOSPEDE cascade constraints;  
drop index RELATIONSHIP_14_FK;  
drop index RELATIONSHIP_13_FK;  
drop table LIMPA cascade constraints;  
drop table METODO_PAGAMENTO cascade constraints;  
drop index ASSOCIATION_3_FK;
```

```
drop index ASSOCIATION_2_FK;

drop index ASSOCIATION_1_FK;

drop table PEDE cascade constraints;

drop index RELATIONSHIP_3_FK;

drop table QUARTO cascade constraints;

drop table RECECIONISTA cascade constraints;

drop index RESPONSAVEL_FK;

drop index PAGA_FK;

drop table RESERVA cascade constraints;

drop index SERVE_FK;

drop table SERVE cascade constraints;

drop index TEM2_FK;

drop index TEM_FK;

drop table TEM cascade constraints;

drop table TIPO_QUARTO cascade constraints;

/*=====*/
/* Table: ALOJADO */
/*=====*/
create table ALOJADO
(
  ID_HOSPEDE      NUMBER(4)      not null,
  ID_QUARTO       NUMBER(4)      not null,
  constraint PK_ALOJADO primary key (ID_HOSPEDE, ID_QUARTO)
);

/*=====*/
/* Index: ALOJADO_FK */
/*=====*/
create index ALOJADO_FK on ALOJADO (
  ID_HOSPEDE ASC
);
```

```
/*=====*/
/* Index: ALOJADO2_FK                               */
/*=====*/
create index ALOJADO2_FK on ALOJADO (
  ID_QUARTO ASC
);

/*=====*/
/* Table: BEBIDA                                     */
/*=====*/
create table BEBIDA
(
  ID_BEBIDA      NUMBER(2)      not null,
  NOME_BEBIDA    CHAR(20),
  PRECO          FLOAT(4),
  constraint PK_BEBIDA primary key (ID_BEBIDA)
);

/*=====*/
/* Table: BUFFET                                     */
/*=====*/
create table BUFFET
(
  ID_BUFFET      NUMBER(2)      not null,
  DESCRICAO      CHAR(20),
  DATA          DATE,
  constraint PK_BUFFET primary key (ID_BUFFET)
);

/*=====*/
/* Table: CONTEM                                     */
/*=====*/
create table CONTEM
(
  ID_RESERVA     NUMBER(4)      not null,
  ID_HOSPEDE     NUMBER(4)      not null,
  constraint PK_CONTEM primary key (ID_RESERVA, ID_HOSPEDE)
);

/*=====*/
/* Index: CONTEM_FK                               */
/*=====*/
```

```
create index CONTEM_FK on CONTEM (
  ID_RESERVA ASC
);

/*=====*/
/* Index: CONTEM2_FK                      */
/*=====*/
create index CONTEM2_FK on CONTEM (
  ID_HOSPEDE ASC
);

/*=====*/
/* Table: COZINHEIRO                      */
/*=====*/
create table COZINHEIRO
(
  ID_FUNCIONARIOS  NUMBER(4)      not null,
  constraint PK_COZINHEIRO primary key (ID_FUNCIONARIOS)
);

/*=====*/
/* Table: EMPREGADO_LIMPEZA                */
/*=====*/
create table EMPREGADO_LIMPEZA
(
  ID_FUNCIONARIOS  NUMBER(4)      not null,
  constraint PK_EMPREGADO_LIMPEZA primary key (ID_FUNCIONARIOS)
);

/*=====*/
/* Table: EMPREGADO_MESA                   */
/*=====*/
create table EMPREGADO_MESA
(
  ID_FUNCIONARIOS  NUMBER(4)      not null,
  constraint PK_EMPREGADO_MESA primary key (ID_FUNCIONARIOS)
);

/*=====*/
/* Table: ENTRADA                          */
/*=====*/
create table ENTRADA
(
  ID_ENTRADA       NUMBER(4)      not null,
```

```
ID_FUNCIONARIOS    NUMBER(4),
NOME_HOSPEDES      CHAR(30),
PRECO_TOTAL        FLOAT(6),
CHECKIN            DATE,
CHECKOUT           DATE,
constraint PK_ENTRADA primary key (ID_ENTRADA)
);

/*=====*/
/* Index: PROCESSA_FK                               */
/*=====*/
create index PROCESSA_FK on ENTRADA (
  ID_FUNCIONARIOS ASC
);

/*=====*/
/* Table: EXIGE                                     */
/*=====*/
create table EXIGE
(
  ID_ENTRADA        NUMBER(4)        not null,
  ID_RESERVA        NUMBER(4)        not null,
  constraint PK_EXIGE primary key (ID_ENTRADA, ID_RESERVA)
);

/*=====*/
/* Index: EXIGE_FK                               */
/*=====*/
create index EXIGE_FK on EXIGE (
  ID_ENTRADA ASC
);

/*=====*/
/* Index: EXIGE2_FK                               */
/*=====*/
create index EXIGE2_FK on EXIGE (
  ID_RESERVA ASC
);

/*=====*/
/* Table: FAZ                                     */
/*=====*/
create table FAZ
(
```

```
ID_FUNCIONARIOS  NUMBER(4)      not null,
ID_BUFFET        NUMBER(2)      not null,
constraint PK_FAZ primary key (ID_FUNCIONARIOS, ID_BUFFET)
);

/*=====*/
/* Index: FAZ_FK                                */
/*=====*/
create index FAZ_FK on FAZ (
  ID_FUNCIONARIOS ASC
);

/*=====*/
/* Index: FAZ2_FK                                */
/*=====*/
create index FAZ2_FK on FAZ (
  ID_BUFFET ASC
);

/*=====*/
/* Table: FUNCIONARIOS                          */
/*=====*/
create table FUNCIONARIOS
(
  ID_FUNCIONARIOS  NUMBER(4)      not null,
  TURNO            CHAR(10),
  NOME             CHAR(30),
  constraint PK_FUNCIONARIOS primary key (ID_FUNCIONARIOS)
);

/*=====*/
/* Table: HOSPEDE                                */
/*=====*/
create table HOSPEDE
(
  ID_HOSPEDE       NUMBER(4)      not null,
  NIF              NUMBER(9),
  NOME             CHAR(30),
  EMAIL           CHAR(30),
  DATA_NASCIMENTO DATE,
  NUM_TELEMOVEL    NUMBER(9),
  constraint PK_HOSPEDE primary key (ID_HOSPEDE)
);
```



```
/*=====*/
/* Table: LIMPA */
/*=====*/
create table LIMPA
(
  ID_FUNCIONARIOS  NUMBER(4)      not null,
  ID_QUARTO        NUMBER(4)      not null,
  DATA            DATE,
  TIPO_LIMPEZA     CHAR(10)
);

/*=====*/
/* Index: RELATIONSHIP_13_FK */
/*=====*/
create index RELATIONSHIP_13_FK on LIMPA (
  ID_FUNCIONARIOS ASC
);

/*=====*/
/* Index: RELATIONSHIP_14_FK */
/*=====*/
create index RELATIONSHIP_14_FK on LIMPA (
  ID_QUARTO ASC
);

/*=====*/
/* Table: METODO_PAGAMENTO */
/*=====*/
create table METODO_PAGAMENTO
(
  ID_PAGAMENTO     NUMBER(4)      not null,
  TIPO             CHAR(20),
  constraint PK_METODO_PAGAMENTO primary key (ID_PAGAMENTO)
);

/*=====*/
/* Table: PEDE */
/*=====*/
create table PEDE
(
  ID_HOSPEDE       NUMBER(4)      not null,
  ID_BEBIDA        NUMBER(2)      not null,
```

```

    ID_BUFFET      NUMBER(2)      not null,
    constraint PK_PEDE primary key (ID_HOSPEDE, ID_BEBIDA, ID_BUFFET)
);

/*=====*/
/* Index: ASSOCIATION_1_FK                      */
/*=====*/
create index ASSOCIATION_1_FK on PEDE (
    ID_HOSPEDE ASC
);

/*=====*/
/* Index: ASSOCIATION_2_FK                      */
/*=====*/
create index ASSOCIATION_2_FK on PEDE (
    ID_BEBIDA ASC
);

/*=====*/
/* Index: ASSOCIATION_3_FK                      */
/*=====*/
create index ASSOCIATION_3_FK on PEDE (
    ID_BUFFET ASC
);

/*=====*/
/* Table: QUARTO                                */
/*=====*/
create table QUARTO
(
    ID_QUARTO      NUMBER(4)      not null,
    ID_TIPO        NUMBER(4),
    DISPONIBILIDADE CHAR(10),
    CAPACIDADE     NUMBER(2),
    PRECO_QUARTO   FLOAT(6),
    constraint PK_QUARTO primary key (ID_QUARTO)
);

/*=====*/
/* Index: RELATIONSHIP_3_FK                    */
/*=====*/
create index RELATIONSHIP_3_FK on QUARTO (
    ID_TIPO ASC
);

```

```
/*=====*/
/* Table: RECECIONISTA */
/*=====*/
create table RECECIONISTA
(
  ID_FUNCIONARIOS  NUMBER(4)      not null,
  constraint PK_RECECIONISTA primary key (ID_FUNCIONARIOS)
);

/*=====*/
/* Table: RESERVA */
/*=====*/
create table RESERVA
(
  ID_RESERVA      NUMBER(4)      not null,
  ID_PAGAMENTO    NUMBER(4),
  ID_HOSPEDE     NUMBER(4),
  ESTADO_RESERVA CHAR(10),
  NUM_NOITES     NUMBER(3),
  N_PESSOAS      NUMBER(2),
  constraint PK_RESERVA primary key (ID_RESERVA)
);

/*=====*/
/* Index: PAGA_FK */
/*=====*/
create index PAGA_FK on RESERVA (
  ID_PAGAMENTO ASC
);

/*=====*/
/* Index: RESPOSAVEL_FK */
/*=====*/
create index RESPOSAVEL_FK on RESERVA (
  ID_HOSPEDE ASC
);

/*=====*/
/* Table: SERVE */
/*=====*/
create table SERVE
(
  ID_FUNCIONARIOS  NUMBER(4)      not null,
```

```
        ID_BEBIDA      NUMBER(2)      not null,
        constraint PK_SERVE primary key (ID_FUNCIONARIOS, ID_BEBIDA)
    );

/*=====*/
/* Index: SERVE_FK                      */
/*=====*/
create index SERVE_FK on SERVE (
    ID_FUNCIONARIOS ASC
);

/*=====*/
/* Table: TEM                          */
/*=====*/
create table TEM
(
    ID_RESERVA      NUMBER(4)      not null,
    ID_QUARTO       NUMBER(4)      not null,
    constraint PK_TEM primary key (ID_RESERVA, ID_QUARTO)
);

/*=====*/
/* Index: TEM_FK                      */
/*=====*/
create index TEM_FK on TEM (
    ID_RESERVA ASC
);

/*=====*/
/* Index: TEM2_FK                     */
/*=====*/
create index TEM2_FK on TEM (
    ID_QUARTO ASC
);

/*=====*/
/* Table: TIPO_QUARTO                  */
/*=====*/
create table TIPO_QUARTO
(
    ID_TIPO         NUMBER(4)      not null,
    DESIGNACAO      CHAR(30),
    constraint PK_TIPO_QUARTO primary key (ID_TIPO)
);
```

```
alter table ALOJADO
add constraint FK_ALOJADO_ALOJADO_HOSPEDE foreign key (ID_HOSPEDE)
references HOSPEDE (ID_HOSPEDE);

alter table ALOJADO
add constraint FK_ALOJADO_ALOJADO2_QUARTO foreign key (ID_QUARTO)
references QUARTO (ID_QUARTO);

alter table CONTEM
add constraint FK_CONTEM_CONTEM_RESERVA foreign key (ID_RESERVA)
references RESERVA (ID_RESERVA);

alter table CONTEM
add constraint FK_CONTEM_CONTEM2_HOSPEDE foreign key (ID_HOSPEDE)
references HOSPEDE (ID_HOSPEDE);

alter table COZINHEIRO
add constraint FK_COZINHEI_CARGO3_FUNCIONA foreign key (ID_FUNCIONARIOS)
references FUNCIONARIOS (ID_FUNCIONARIOS);

alter table EMPREGADO_LIMPEZA
add constraint FK_EMPREGAD_CARGO2_FUNCIONA foreign key (ID_FUNCIONARIOS)
references FUNCIONARIOS (ID_FUNCIONARIOS);

alter table EMPREGADO_MESA
add constraint FK_EMPREGAD_CARGO4_FUNCIONA foreign key (ID_FUNCIONARIOS)
references FUNCIONARIOS (ID_FUNCIONARIOS);

alter table ENTRADA
add constraint FK_ENTRADA_PROCESSA_RECECION foreign key (ID_FUNCIONARIOS)
references RECECIONISTA (ID_FUNCIONARIOS);

alter table EXIGE
add constraint FK_EXIGE_EXIGE_ENTRADA foreign key (ID_ENTRADA)
references ENTRADA (ID_ENTRADA);

alter table EXIGE
add constraint FK_EXIGE_EXIGE2_RESERVA foreign key (ID_RESERVA)
references RESERVA (ID_RESERVA);

alter table FAZ
add constraint FK_FAZ_FAZ_COZINHEI foreign key (ID_FUNCIONARIOS)
references COZINHEIRO (ID_FUNCIONARIOS);
```

```
alter table FAZ
add constraint FK_FAZ_FAZ2_BUFFET foreign key (ID_BUFFET)
references BUFFET (ID_BUFFET);

alter table LIMPA
add constraint FK_LIMPA_RELATIONS_EMPREGAD foreign key (ID_FUNCIONARIOS)
references EMPREGADO_LIMPEZA (ID_FUNCIONARIOS);

alter table LIMPA
add constraint FK_LIMPA_RELATIONS_QUARTO foreign key (ID_QUARTO)
references QUARTO (ID_QUARTO);

alter table PEDE
add constraint FK_PEDE_PEDE_HOSPEDE foreign key (ID_HOSPEDE)
references HOSPEDE (ID_HOSPEDE);

alter table PEDE
add constraint FK_PEDE_PEDE2_BEBIDA foreign key (ID_BEBIDA)
references BEBIDA (ID_BEBIDA);

alter table PEDE
add constraint FK_PEDE_PEDE3_BUFFET foreign key (ID_BUFFET)
references BUFFET (ID_BUFFET);

alter table QUARTO
add constraint FK_QUARTO_RELATIONS_TIPO_QUA foreign key (ID_TIPO)
references TIPO_QUARTO (ID_TIPO);

alter table RECECIONISTA
add constraint FK_RECECION_CARGO_FUNCIONA foreign key (ID_FUNCIONARIOS)
references FUNCIONARIOS (ID_FUNCIONARIOS);

alter table RESERVA
add constraint FK_RESERVA_PAGA_METODO_P foreign key (ID_PAGAMENTO)
references METODO_PAGAMENTO (ID_PAGAMENTO);

alter table RESERVA
add constraint FK_RESERVA_RESPONSAV_HOSPEDE foreign key (ID_HOSPEDE)
references HOSPEDE (ID_HOSPEDE);

alter table SERVE
add constraint FK_SERVE_SERVE_EMPREGAD foreign key (ID_FUNCIONARIOS)
references EMPREGADO_MESA (ID_FUNCIONARIOS);
```

```
alter table SERVE  
add constraint FK_SERVE_SERVE2_BEBIDA foreign key (ID_BEBIDA)  
references BEBIDA (ID_BEBIDA);
```

```
alter table TEM  
add constraint FK_TEM_TEM_RESERVA foreign key (ID_RESERVA)  
references RESERVA (ID_RESERVA);
```

```
alter table TEM  
add constraint FK_TEM_TEM2_QUARTO foreign key (ID_QUARTO)  
references QUARTO (ID_QUARTO);
```

## Cap 5 - Conclusões

Em resumo, este trabalho propõe criar uma base de dados e um sistema para ajudar na gestão do hotel. Analisamos a situação atual, onde o hotel pretende gerir melhor os seus recursos humanos e mantimentos, de forma a fornecer um melhor serviço para os clientes. A solução sugerida procura criar uma base de dados para melhorar a eficiência do hotel.

A base de dados inclui informações importantes sobre reservas, hóspedes, funcionários e outros aspectos hoteleiros. A proposta destaca a importância de uma abordagem integrada para resolver problemas atuais e futuros.

Concluindo, a ideia é melhorar a gestão do hotel, proporcionando uma ferramenta eficaz para a organização e tomada de decisões .

## Referências Bibliográficas

## Anexos