



**Universidade do Minho**  
Mestrado Integrado em Engenharia Informática  
Licenciatura em Ciências da Computação

## **Unidade Curricular de Bases de Dados**

Ano Lectivo de 2020/2021

### **Hotel D'UMinho**

**Daniel Filipe Santos Sousa, 89562**

**Inês Sofia Paiva Bastos, 89522**

**José Pedro Fernandes Veloso, 89573**

**Lucie Fernandes Dantas, 89576**

dezembro, 2020

**BD**

Data de Recepção	
Responsável	
Avaliação	
Observações	

## **Hotel D'UMinho**

**Daniel Filipe Santos Sousa, 89562**

**Inês Sofia Paiva Bastos, 89522**

**José Pedro Fernandes Veloso, 89573**

**Lucie Fernandes Dantas, 89576**

Dezembro, 2020

## Resumo

Este relatório possui como finalidade a apresentação do nosso trabalho de grupo na elaboração de uma base de dados para o Hotel D'UMinho implementando o conhecimento teórico e prático abordado no decorrer da unidade curricular de Bases de Dados.

De forma a obter-se uma melhor gestão deste hotel desenvolvemos esta Base de Dados que incorpora todo um registo de clientes, reservas e respetivos quartos associados a estas.

Através do presente relatório são retratadas e registadas todas as etapas e pontos percorridos desde a criação do Modelo Conceptual à implementação do Modelo Físico.

Concluímos, através de vários testes com dados reais, que a Base de Dados desenvolvida cumpre os objetivos e necessidades pretendidas, tornando-se assim consistente.

**Área de Aplicação:** Desenho e arquitectura de Sistemas de Bases de Dados do Hotel D'UMinho

**Palavras-Chave:** Bases de Dados relacionais, Modelo conceptual, Entidades, Atributos, Relacionamentos, Modelo lógico, Modelo físico, Queries, MySQL, Entidades, Atributos, Relacionamentos, Queries

# Índice

Resumo	i
Índice de Figuras	iii
Índice de Tabelas	v
1.2. Fundamentação da implementação da base de dados	3
1.3. Análise da viabilidade do processo	3
1.4. Estrutura do Relatório	4
2. Levantamento e análise de requisitos	6
2.1. Método de levantamento e de análise de requisitos adotado	6
2.2. Requisitos levantados	6
3. Modelação Conceptual	9
3.1. Apresentação da abordagem de modelação realizada	9
3.2. Identificação e caracterização das entidades	9
3.3. Identificação e caracterização dos relacionamentos	10
3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	13
3.5. Detalhe ou generalização de entidades	16
3.6. Apresentação e explicação do diagrama ER	17
3.7. Validação do modelo de dados produzido	18
4. Modelação Lógica	20
4.1. Construção e validação do modelo de dados lógico	20
4.2. Desenho do modelo lógico	24
4.3. Validação do modelo através da normalização	25
4.4. Validação do modelo com interrogações do utilizador	26
4.5. Revisão do modelo lógico produzido	27
5. Implementação física	28
5.1. Seleção do sistema de gestão de bases de dados	28
5.2. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	28
5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos)	29
5.5. Estimativa do espaço em disco de base de dados e taxa de crescimento anual	35
5.6. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	37
5.8. Siglas e Acrónimos	39
5.9. Referências Bibliográficas	39
6. Conclusões e Trabalho Futuro	41
I. Anexo 1 – Script completo da criação do esquema físico	44
II. Anexo 2 – Script completo da criação do esquema físico	49
III. Anexo 3 – Exemplos de queries em SQL	58

# Índice de Figuras

Figura 1 – Relacionamento entre a entidade Tipo de Cliente e a entidade Cliente.	11
Figura 2 – Relacionamento entre a entidade Cliente e a entidade Reserva.	12
Figura 3 – Relacionamento entre a entidade Reserva e a entidade Serviço.	12
Figura 4 – Relacionamento entre a entidade Reserva e a entidade Quarto.	12
Figura 5 – Relacionamento entre a entidade Quarto e a entidade Tipo de Quarto.	13
Figura 6 – Diagrama ER da Base de Dados.	17
Figura 7 – Relacionamento entre as tabelas das entidades TipoDeCliente e Cliente.	21
Figura 8 – Relacionamento entre as tabelas das entidades Cliente e Reserva.	22
Figura 9 – Relacionamento entre as tabelas das entidades TipoDeQuarto e Quarto.	22
Figura 10 – Relacionamento entre as tabelas das entidades Reserva e Servico.	23
Figura 11 – Relacionamento entre as tabelas das entidades Reserva e Quarto.	24
Figura 12 – Esquema do modelo lógico da Base de Dados.	25
Figura 13 – Resolução da Query para obter o nome dos clientes e as suas cidades a partir do país.	29
Figura 14 – Resultado da Query da Figura 13, dado o país “Espanha”.	30
Figura 15 – Resolução da Query para obter os tipos de clientes.	30
Figura 16 – Resultado da Query da figura 15	31
Figura 17 - Resolução da Query sobre a listagem dos Quartos ocupados numa determinada data.	31
Figura 18 – Resultado da Query da figura 17 utilizando o mês de maio e ano de 2020	32
Figura 19 – Realização do procedure spTiposDeServicos que dado um serviço retorna os clientes que o usaram.	32
Figura 20 – Resultado do procedure spTiposDeServicos com o input campo de futebol.	32
Figura 21 – Resolução da function fuPagoCliente que dado um cliente retorna o valor total pago.	33
Figura 22 – Resultado da function fuPagoCliente com o argumento ‘Diogo Pereira Catalao Da Costa’.	33
Figura 23 – Realização do procedure spReservasPorAnoMes que dado um mês e um ano retorna os clientes que fizeram uma determinada reserva nesse tempo.	34
Figura 24 – Resultado do procedure spReservasPorAnoMes para o input do ano igual a 2020 e do mês de Maio.	34

Figura 25 - Código em SQL que permite verificar o tamanho em MB de cada tabela da Base de dados.	35
Figura 26 – Tamanho (em MB) de cada tabela.	36
Figura 27 – Tamanho total em disco da base de dados.	36
Figura 28 – Tamanho total em disco.	37
Figura 29 – Vista que permite o acesso aos cinco melhores clientes do Hotel.	37
Figura 30 – Vista que permite o acesso às reservas canceladas do Hotel.	38
Figura 31 – Vista que permite o acesso à listagem de quartos de valor menor ou igual a 50€.	38

## **Índice de Tabelas**

Tabela 1 – Preço respetivo de cada tipo de quarto por noite.	1
Tabela 2 – Preço respetivo de cada serviço extra adicional por noite.	2
Tabela 3 – Identificação e caracterização das entidades.	9
Tabela 4 – Identificação e caracterização dos relacionamentos.	11
Tabela 5 – Tabela de atribuição e caracterização do Tipo de Cliente.	13
Tabela 6 – Tabela de atribuição e caracterização do Cliente.	14
Tabela 7 – Tabela de atribuição e caracterização da Reserva.	14
Tabela 8 – Tabela de atribuição e caracterização do Serviço.	15
Tabela 9 – Tabela de atribuição e caracterização do Tipo de Quarto.	15
Tabela 10 – Tabela de atribuição e caracterização do Quarto.	16
Tabela 11 – Tabela de atribuição e caracterização do Preço (atributo de relacionamento entre as entidades Reserva e Quarto).	16

## 1. Definição do Sistema

Neste capítulo são evidenciados o enquadramento e a motivação deste trabalho prático de bases de dados, elucidando a importância das bases de dados como base de suporte à modelação de dados e gestão de informação, admitindo o contexto e a conveniência deste trabalho prático, e enquadrando a modelação levada a cabo num problema de análise concreto.

Tipo de Quarto	Preço do Quarto
Quarto Simples Standard	25€
Quarto Duplo Standard	45€
Quarto Triplo Standard	65€
Quarto Quádruplo Standard	95€
Quarto Suíte Standard	120€
Quarto Simples Deluxe	50€
Quarto Duplo Deluxe	65€
Quarto Triplo Deluxe	80€
Quarto Quádruplo Deluxe	100€
Quarto Suíte Deluxe	132€

Tabela 1 – Preço respetivo de cada tipo de quarto por noite.

<b>Serviço extra</b>	<b>Preço do serviço</b>
Mini-bar	10,54€
Acesso ao campo de futebol	35€
Pequeno almoço no quarto	27,45€
Acesso ao jacuzzi	40€
Acesso à piscina	20,57€
Acesso ao spa	56,38€
Limpeza diária do quarto	35€
Acesso ao casino	23€
Acesso à sauna	46€
Acesso ao ginásio	10€
Acesso à biblioteca	5,99€

Tabela 2– Preço respetivo de cada serviço extra adicional por noite.

## 1.1. Contexto de Aplicação do Sistema

O Hotel D'Uminho foi investido e idealizado com o objetivo de abranger todos os alunos da Universidade do Minho tal como todos os visitantes e turistas que pretendam frequentar alguma atividade na universidade, conhecer o estabelecimento de ensino ou ainda, simplesmente, familiares de alunos que tencionem acompanhar de mais perto o início desta jornada académica que, a cada ano letivo se revela uma nova etapa e, por vezes, de difícil adaptação para os alunos recentemente colocados.

Situado na rua da Sabedoria, paralelamente ao Campus de Gualtar (um dos Campus que constitui a universidade), foi construído em 2018 como exaltação ao seu reconhecimento pelo ranking internacional como melhor universidade portuguesa na área da engenharia.

Este Hotel faz-se representar especialmente pela imagem de “Prometeu Agrilhoado” (herói da mitologia grega que lutara pelo “fogo” do conhecimento), com o intuito de, juntamente com o conforto proporcionado, impulsionar os alunos e todos os que frequentam o espaço, a lutar pelos seus ideais e pelo conhecimento que vangloria a Universidade do Minho.

Com isto, um dos propósitos da criação e gestão deste hotel baseia-se na junção do útil ao agradável. Isto é, a tentativa de abranger e equilibrar o luxo e conforto da estadia dos hóspedes com o preço e localização privilegiada próxima da Universidade do Minho.

Especialmente desde 2018 a Universidade do Minho tem vindo a crescer e a conquistar cada vez mais reconhecimento. Consequentemente, a curiosidade e a vontade da população em visitar e frequentar a universidade advêm desse fator.

## **1.2. Fundamentação da implementação da base de dados**

Uma vez que é primordial o Hotel conseguir inovar e responder aos clientes de forma rápida e eficaz consoante os seus pedidos diversos, e encontrando nesta necessidade a fundamentação para o nosso trabalho prático, é essencial a criação de um novo sistema com o intuito de aprimorar a organização das reservas de quartos, bem como alguns dos seus serviços (nomeadamente limpeza diária do quarto, piscina, ginásio, massagem, etc).

Para que seja possível o alojamento no hotel, o cliente (aluno da universidade, familiar, turista ou mero visitante) terá de realizar uma reserva na qual terá de fornecer alguns dos seus dados pessoais como: o seu nome completo, o número de identificação fiscal, profissão (opcional), o seu endereço e o seu contacto. No momento da reserva o cliente também terá de mencionar os serviços associados ao hotel que pretende para que fique registado.

Este Hotel possui todas as possibilidades e condições para oferecer aos seus clientes uma estadia memorável, no entanto é de ter em atenção que, dado o estado pandémico que, tanto Portugal como os restantes países atravessam atualmente em consequência do vírus COVID-19, toda a base de reservas efetuadas por parte de clientes que não habitam em Portugal podem sofrer alterações.

Desta forma, foi sugerida a criação de uma base de dados que suporte um sistema do Hotel D'UMinho, exibindo com esse intuito uma análise criteriosa de requisitos, seguida da sua modelação e terminando com a implementação física da base de dados.

## **1.3. Análise da viabilidade do processo**

Tendo em conta a qualidade de oferta educativa, o resultante aumento da procura de ensino na Universidade do Minho e, em consequência, o crescente número de colocações por ano letivo (tal como referido pelo reitor da universidade “Preenchemos 98% das 2869 vagas a concurso (...), o que representa uma progressão importante face aos resultados do último ano”), foi motivado o desenvolvimento de um novo sistema que permitisse dar resposta rápida

e eficaz à reserva de quartos e serviços associados ao Hotel, acompanhando assim o crescimento do número de futuros hóspedes.

Como referido, a implementação deste Sistema de Base de Dados (SBD), mais especificamente um Sistema de Base de Dados Relacional (SBDR), é motivado pelo aumento da procura da oferta de ensino superior facultada, em particular, pela Universidade do Minho.

Um dos principais pontos a questionar é, como mencionado no próprio capítulo, a viabilidade do projeto e os proveitos que esta base de dados pode oferecer assim que concluída.

Primeiramente é importante entender o conceito de base de dados e o seu propósito associado. Teoricamente pode afirmar-se que uma base de dados é um conjunto estruturado de informação, partilhável e sujeita a um controlo central.

Uma vez que se processarmos os dados de forma manual ou até mesmo num sistema que não numa base de dados, pode ocorrer a repetição e aglomeração desnecessária de informação, constatamos que, ao implementar um sistema de gestão e desenvolvendo uma base de dados são várias as vantagens que esta pode proporcionar: desde o poder de controlo da redundância ao controlo de acesso à informação. Tendo sempre em mente as regras de normalização, é conseguida uma redução da acumulação de informação evitando assim a redundância.

Assim sendo, é obtida uma maior rapidez no acesso a toda a informação ainda como uma maior facilidade na alteração dos dados ou mesmo na recuperação destes, caso necessário.

## **1.4. Estrutura do Relatório**

Este subcapítulo tem como propósito a apresentação e breve descrição sobre o conteúdo que cada um dos capítulos abrange visando a melhor compreensão da estrutura do presente relatório.

Neste primeiro capítulo é feita toda uma introdução e enquadramento do trabalho onde é evidenciado o caso de estudo e a contextualização do problema assim como uma exposição das motivações e objetivos a alcançar com o mesmo.

O segundo capítulo trata os requisitos e todo o trabalho envolvido, desde o seu levantamento, identificação do respetivo método utilizado e do tipo de requisitos em questão, finalizando-se com uma análise destes.

O terceiro capítulo é referente ao processo de desenvolvimento do modelo de dados conceptual onde são, portanto, identificadas as entidades (principais elementos de informação do problema), os atributos, os variados relacionamentos e onde é também apresentado o diagrama ER.

No quarto capítulo, é dada a tradução do modelo conceptual da base de dados para o modelo lógico, modelo este onde são também tratados os relacionamentos e as entidades.

Finalizando, no quinto capítulo é utilizado o sistema de bases de dados MySQL com o fim de se dar também uma tradução porém, agora, do modelo conceptual para o modelo físico.

Demonstra-se que há várias maneiras de responder precisamente aos requisitos recorrendo à linguagem SQL. E, quanto à implementação física de uma base de dados, são também analisados aspectos particulares como é o caso do espaço em disco necessário para o suporte da base de dados.

## **2. Levantamento e análise de requisitos**

### **2.1. Método de levantamento e de análise de requisitos adotado**

Com o intuito de proceder com o levantamento de dados para a realização desta base de dados foi efetuada uma entrevista ao proprietário do Hotel D'UMinho. A fim de conseguir uma boa margem de sustento a nível de dados credíveis para esta base de dados foi ainda recolhida, com o reitor da Universidade do Minho Rui Vieira de Castro, informação sobre o aumento do número de colocações na universidade.

O principal objetivo a alcançar consistia em melhorar a qualidade de resposta/do serviço uma vez que eram as principais preocupações transmitidas na entrevista. Então, um dos principais propósitos do requerimento da base de dados foi a facilidade com que é feita a gestão das reservas de maneira a que fosse melhorado o controle sobre os quartos livres e ocupados num determinado momento/data. Isto é, para que não houvesse falhas e não ocorresse, por exemplo, mais do que uma reserva para o mesmo quarto na mesma data.

Desta forma, reuniram-se algumas condições de modo a satisfazer por parte da base de dados a criação de alguns requisitos.

### **2.2. Requisitos levantados**

Esta implementação de base de dados pretende armazenar as reservas de quartos e serviços associados ao hotel, para melhorar e facilitar a gestão e o processo de reservas.

#### **2.2.1. Requisitos de descrição**

- Cliente

Cada cliente pode efetuar várias reservas, mas para isso precisa primeiro de fornecer alguns dos seus dados pessoais como o nome, idade, o seu número de identificação fiscal (NIF), o seu endereço e profissão. Estes dados serão utilizados para poder concluir uma

reserva e os clientes conseguirem, então, ter acesso a todos os serviços que o hotel tem ao seu dispor e também de forma a poder entrar em contacto com estes, em caso de necessidade. A cada cliente está associado um número identificador único.

- Quarto

Cada quarto é caracterizado por um número que o identifica de forma única podendo ou não estar ocupado.

- Reserva

A uma reserva está associado um identificador, o dia em que esta foi efetuada, e as respetivas datas de início e fim da estadia do cliente.

- Tipo de quarto

Cada quarto terá uma descrição (por exemplo sobre a vista do mesmo), um número distinto que o identifica e distingue dos restantes e um preço a ele associado.

- Valor dos quartos

Cada tipo de quarto tem associado a si um preço correspondente

### **2.2.2. Requisitos de exploração**

- Obter a listagem dos clientes e das respetivas cidades dado um determinado país
- Obter os diferentes tipos de clientes
- Obter o número dos quartos reservados numa determinada data.
- Obter a listagem de clientes que reservaram um determinado serviço
- Obter o que foi pago por um determinado cliente
- Obter os clientes que fizeram uma reserva numa determinada data

### **2.2.3. Requisitos de controlo**

- O preço associado a cada reserva depende dos serviços pretendidos e da duração da estadia
- Um quarto não pode ser reservado mais do que uma vez para a mesma data, isto é: estando ocupado não pode ser reservado.

### **2.3. Análise e validação geral dos requisitos**

Após uma análise cuidada dos dados recolhidos na entrevista com o proprietário do Hotel, tornou-se possível a elaboração de uma lista de requisitos essenciais para a criação de uma base de dados, que permitisse uma mais fácil resolução dos problemas do Hotel D'UMinho. Como foi dito anteriormente, o objetivo principal era a criação um sistema de modo a gerir de um modo mais eficiente as reservas do Hotel, simplificando o sistema anterior tornando a reserva um processo otimizado. Através do apontamento destes requisitos consegue-se obter uma maior fluidez, simplificação e eficácia da base de dados. Posto isto, partimos para a próxima etapa, o desenvolvimento do modelo conceptual.

## **3. Modelação Conceptual**

### **3.1. Apresentação da abordagem de modelação realizada**

Neste capítulo são apresentadas todas as etapas realizadas para desenvolver e concluir o modelo conceptual da base de dados com base na lista de requisitos apresentada anteriormente no presente relatório. Estas etapas serão exibidas em notação Chen nos subcapítulos posteriores, nomeadamente a identificação de entidades, dos respetivos relacionamentos entre estas e ainda dos atributos.

A fim de concluir este capítulo de apresentação do modelo conceptual da base de dados dá-se a validação do modelo de dados produzido.

### **3.2. Identificação e caracterização das entidades**

Neste subcapítulo, como o título do mesmo sugere, é tratada a identificação e a caracterização das várias entidades da base de dados desenvolvida, em tabelando-as, de modo a explanar o modelo apresentado.

Entidade	Descrição	Alias	Ocorrência
Tipo de Cliente	Termo geral que contém a informação associada a tipo de Cliente	-----	A cada tipo de cliente associa-se a sua descrição e o seu código único identificador

Cliente	Termo geral que contém a informação associada a cada cliente do Hotel	-----	Cada cliente é associado aos seus dados: nome, idade, endereço (Cidade, País e Rua), o contacto (Email e Telefone), o número de identificação fiscal, profissão e um código único identificador para conseguir proceder com a sua reserva e ter acesso aos serviços do hotel
Reserva	Termo geral que contém a informação associada a cada reserva de um quarto/serviço, por parte de um cliente	-----	Cada reserva é associada à sua data de início e final correspondente, um código único identificador e a data em que foi efetuada.
Quarto	Termo geral que contém a informação associada ao quarto reservado	-----	A cada quarto associa-se um andar, a ocupação máxima, a descrição
Tipo de Quarto	Termo geral que contém a informação associada a cada tipo de quarto reservado	-----	A cada tipo de quarto associa-se o seu código único identificador, a sua descrição e o seu valor de reserva.

Tabela 3 – Identificação e caracterização das entidades.

### 3.3. Identificação e caracterização dos relacionamentos

Neste subcapítulo, são apresentados e explanados os relacionamentos da base de dados desenvolvida. De forma a obter uma melhor compreensão

decompomos a sua caracterização em dois momentos: primeiro apresentando uma tabela e posteriormente clarificando os relacionamentos de forma breve.

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade (Relacionada)
Tipo de Cliente	1	Contém	N	Cliente
Cliente	1	Efetua	N	Reserva
Reserva	N	Contém	N	Serviço
Reserva	N	Possui	N	Quarto
Tipo de Quarto	1	Contém	N	Quarto

Tabela 4 -Identificação e caracterização dos relacionamentos.

Para uma melhor compreensão de cada relacionamento entre as respetivas entidades, apresentamos uma breve explicação:

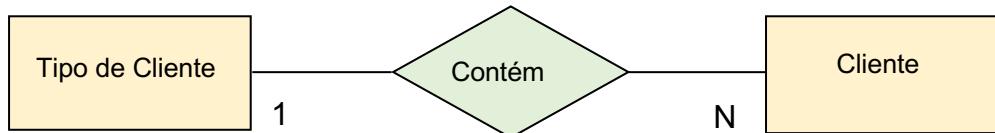


Figura 1 - Relacionamento entre a entidade Tipo de Cliente e a entidade Cliente.

Vários clientes podem estar contidos num só tipo de cliente, assim como um tipo de cliente contém vários clientes. Assim, existe um relacionamento de cardinalidade 1: N entre a entidade Tipo de Cliente e a entidade (relacionada) Cliente, como pode ser observado na Figura 1.

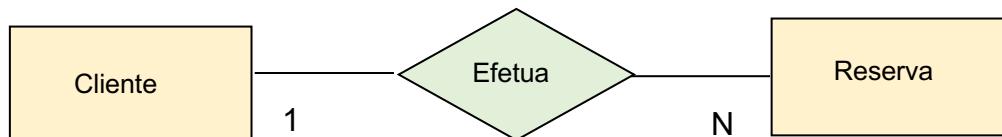


Figura 2 - Relacionamento entre a entidade Cliente e a entidade Reserva.

Várias reservas podem ser efetuadas apenas por um cliente, assim como um só cliente pode efetuar várias reservas. Assim, existe um relacionamento de cardinalidade 1: N entre a entidade Cliente e a entidade (relacionada) Reserva, como pode ser observado na Figura 2.

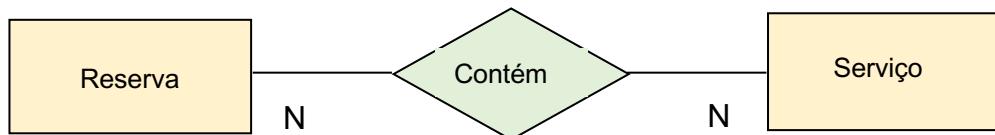


Figura 3 - Relacionamento entre a entidade Reserva e a entidade Serviço.

Vários serviços podem estar contidos em várias reservas, assim como várias reservas podem conter vários serviços. Assim, existe um relacionamento de cardinalidade N: N entre a entidade Reserva e a entidade (relacionada) Serviço, como pode ser observado na Figura 3.

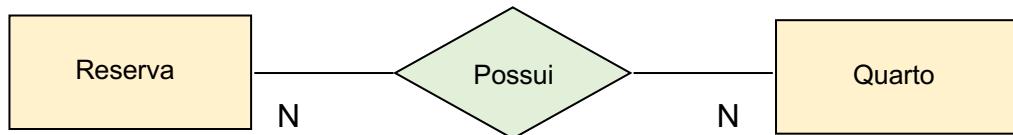


Figura 4 - Relacionamento entre a entidade Reserva e a entidade Quarto.

Vários quartos podem ser associados a várias reservas, assim como várias reservas podem possuir vários quartos. Assim, existe um relacionamento de cardinalidade N: N entre a entidade Reserva e a entidade (relacionada) Quarto, como pode ser observado na Figura 4.

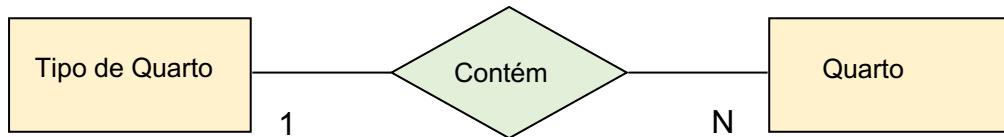


Figura 5 - Relacionamento entre a entidade Quarto e a entidade Tipo de Quarto.

Vários quartos podem estar contidos num tipo de quarto, assim como um tipo de quarto contém vários quartos. Assim, existe um relacionamento de cardinalidade 1: N entre a entidade Tipo de Quarto e a entidade (relacionada) Quarto, como pode ser observado na Figura 5.

### **3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos**

#### **Tipo de Cliente**

Atributo	Descrição	NULL	Multi – Valor	Composto
Código	Código identificador do tipo de Cliente	Não	Não	Não
Descrição	Descrição do tipo de cliente	Não	Não	Não

Tabela 5 - Tabela de atribuição e caracterização do Tipo de Cliente.

### Cliente

Atributo	Descrição	NULL	Multi – Valor	Composto
Id	Identificador do Cliente	Não	Não	Não
Nome	Nome do Cliente	Não	Não	Não
Contacto	Contacto do Cliente	Não	Não	Sim
NIF	Número de Identificação Fiscal do Cliente	Não	Não	Não
Endereço	Morada do Cliente	Não	Não	Sim
Idade	Idade do Cliente	Sim	Não	Não
Profissão	Profissão do Cliente	Sim	Não	Não

Tabela 6 – Tabela de atribuição e caracterização do Cliente

### Reserva

Atributo	Descrição	NULL	Multi – Valor	Composto
Código	Identificador do Cliente	Não	Não	Não
Data de Reserva	Data em que o cliente realiza a reserva	Não	Não	Não
Data Check-In	Data em que o cliente inicia a hospedagem	Não	Não	Não

Data Check-Out	Data em que o cliente termina a sua hospedagem	Não	Não	Não
----------------	--	-----	-----	-----

Tabela 7 – Tabela de atribuição e caracterização da Reserva.

### Serviço

Atributo	Descrição	NULL	Multi – Valor	Composto
Código	Identificador do serviço	Não	Não	Não
Tipo de Serviço	Serviço a ser efetuado	Não	Não	Não
Descrição	Descrição do serviço	Não	Não	Não
Valor	Preço do serviço	Não	Não	Não

Tabela 8 – Tabela de atribuição e caracterização do Serviço.

### Tipo de Quarto

Atributo	Descrição	NULL	Multi – Valor	Composto
Código	Código que identifica o tipo de quarto	Não	Não	Não
Descrição	Descrição do tipo de quarto	Não	Não	Não
Valor	Preço do tipo de quarto	Não	Não	Não

Tabela 9 – Tabela de atribuição e caracterização do Tipo de Quarto.

## Quarto

Atributo	Descrição	NULL	Multi – Valor	Composto
Número	Número do quarto	Não	Não	Não
Andar	Andar em que o quarto se encontra	Não	Não	Não
Ocupação máxima	Ocupação máxima do quarto	Não	Não	Não
Descrição	Descrição do quarto	Não	Não	Não

Tabela 10 – Tabela de atribuição e caracterização do Quarto.

Há ainda um atributo “Preço” que é o atributo de relacionamento entre as entidades Reserva e Quarto.

Atributo	Descrição	NULL	Multi - Valor	Composto
Preço	Preço da reserva	Não	Não	Não

Tabela 11 – Tabela de atribuição e caracterização do Preço (atributo de relacionamento entre as entidades Reserva e Quarto)

## 3.5. Detalhe ou generalização de entidades

No nosso projeto não foi utilizada generalização nem especialização de entidades no Modelo Conceptual.

### 3.6. Apresentação e explicação do diagrama ER

Apresenta-se neste subcapítulo o diagrama ER após a validação do modelo conceptual e a análise de todos os seus constituintes.

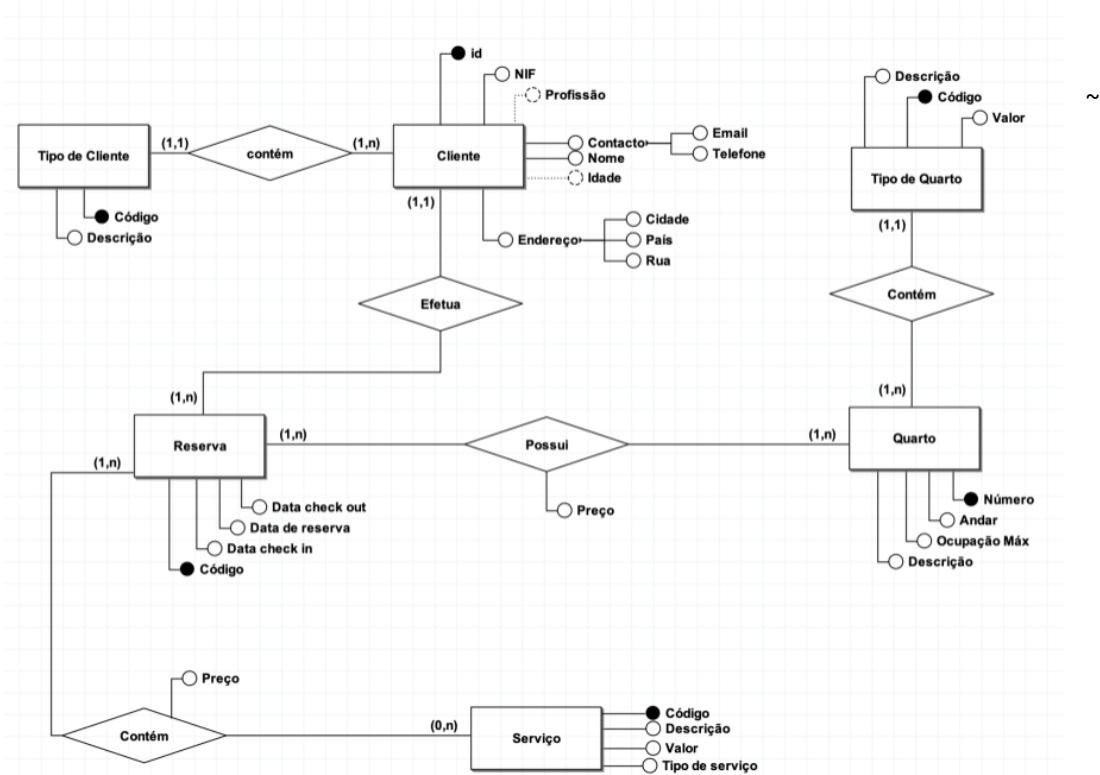


Figura 6 – Diagrama ER da Base de Dados.

Nesta figura são apresentadas todas as entidades, atributos e relacionamentos, base para a criação de uma base de dados, explicados num capítulo anterior.

### **3.7. Validação do modelo de dados produzido**

É fundamental confirmar se as transações descritas nos requisitos são plausíveis e possíveis a fim de validar o modelo conceptual. Com esse intuito, são apresentados os dados e a sua validade correspondente justificada.

#### **a) Consulta dos dados de um cliente**

Na entidade “Cliente” são guardados todos os dados associados a este, desde o seu nome e contacto, e percorrendo outros dados pessoais necessários ao registo da reserva como por exemplo a sua morada.

#### **b) Associação de um cliente ao tipo de cliente que lhe corresponde**

Uma vez que temos um relacionamento “Contém” entre a entidade “Cliente” e a entidade “Tipo de Cliente”, é-nos fornecida a relação entre o cliente e o tipo de cliente de acordo com os dados associados ao cliente em questão.

#### **c) Associação de uma reserva a um cliente**

Uma vez que temos um relacionamento “Efetua” entre as entidades “Cliente” e “Reserva”, é clara a associação entre o cliente e a reserva correspondente.

#### **d) Consulta dos dados de uma reserva**

Na entidade “Reserva” são guardados todos os dados associados a esta, desde a data em que foi efetuada e a data de início da hospedagem do cliente à data da sua saída.

#### **e) Consulta do preço associado a um quarto**

Uma vez que temos um relacionamento “Possui” entre a entidade “Reserva” e a entidade “Serviço” e que essa relação tem um preço para cada quarto, é possível saber o preço do quarto no momento em que a reserva foi realizada.

#### **f) Consulta dos dados de um quarto**

Na entidade “Quarto” é guardada a informação correspondente a cada quarto, desde o seu número e andar em que se encontra à sua descrição e ocupação máxima.

#### **g) Consulta dos dados de cada tipo de quarto**

Na entidade “Tipo de Quarto” é guardada toda a informação sobre os diferentes tipos de quarto que existem.

**h) Associação de um quarto ao tipo de quarto que lhe corresponde**

Uma vez que temos um relacionamento “Contém” entre a entidade “Quarto” e a entidade “Tipo de Quarto”, é-nos fornecida a relação entre o quarto e o tipo de quarto de acordo com os dados associados ao quarto em questão.

## **4. Modelação Lógica**

### **4.1. Construção e validação do modelo de dados lógico**

Após a conclusão da modelação conceptual desenvolvida para a nossa base de dados passamos à tradução do modelo conceptual para o modelo lógico. Este foi construído através do modelo conceptual sendo cada entidade representada por uma tabela e ainda todos os relacionamentos de muitos para muitos, ou seja de n para n. Neste modelo lógico iremos proceder com a sua validação e assim verificar se, com os requisitos inicialmente estabelecidos, a estrutura está correta e é capaz de suportar as transações necessárias.

#### **4.1.1. Entidades**

Analizando esta base de dados conclui-se que há seis entidades e que cada uma originará uma tabela no modelo lógico.

- TipoDeCliente (Codigo, Descricao)
  - Chave primária: Código
- Cliente (Id, Nome, Idade, Endereco, NIF, Profissao, TipoDeCliente\_Codigo)
  - Chave primária: Id
  - Chave estrangeira: TipoDeCliente\_Codigo
- Reserva (Codigo, DataCheckOut, Data CheckIn, DataReserva, Cliente\_Id)
  - Chave primária: Código
  - Chave estrangeira: Cliente\_Id
- Servico (Codigo, Descricao, Valor, Tipo\_De\_Servico)
  - Chave primária: Código

- TipoDeQuarto (Codigo, Descricao, Valor)
  - Chave primária: Codigo
- Quarto (Numero, Andar, Ocupacao\_Maxima, Descricao, TipoDeQuarto\_Codigo)
  - Chave primária: Numero
  - Chave estrangeira: TipoDeQuarto\_Codigo

#### 4.1.2. Relacionamentos 1:N

Na base de dados em apresentação existem três relacionamentos 1:N.

Temos como primeira ocorrência de relacionamento 1:N, o relacionamento entre a entidade “TipoDeCliente” e a entidade “Cliente”. Na representação deste relacionamento é feita uma cópia da chave primária da entidade “TipoDeCliente” para a entidade (relacionada) “Cliente”, que irá atuar como uma chave estrangeira.

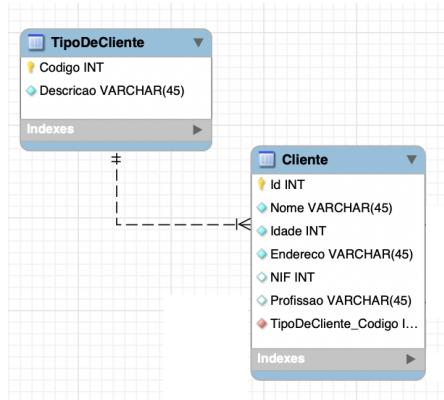


Figura 7 - Relacionamento entre as tabelas das entidades TipoDeCliente e Cliente.

O segundo relacionamento 1:N que ocorre é entre a entidade “Cliente” e a entidade “Reserva”. Na representação deste relacionamento é feita uma cópia da chave primária da entidade “Cliente” para a entidade (relacionada) “Reserva”, que irá atuar como uma chave estrangeira.

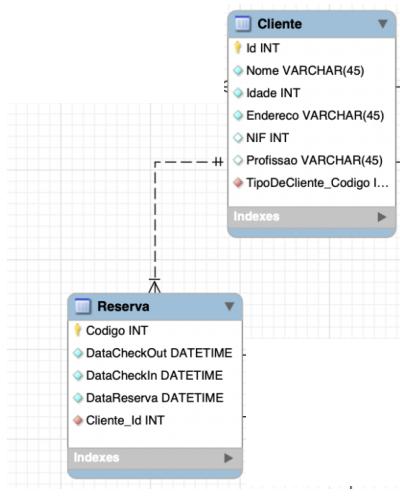


Figura 8 - Relacionamento entre as tabelas das entidades Cliente e Reserva.

O terceiro relacionamento 1:N que ocorre é entre a entidade “TipoDeQuarto” e a entidade “Quarto”. Na representação deste relacionamento é feita uma cópia da chave primária da entidade “TipoDeQuarto” para a entidade (relacionada) “Quarto”, que irá atuar como uma chave estrangeira.

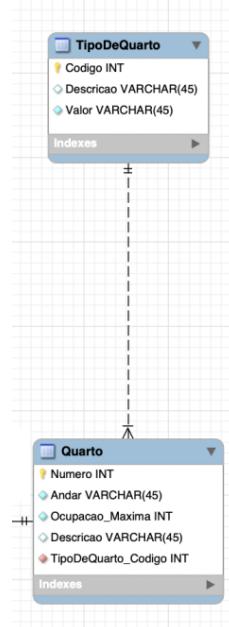


Figura 9 - Relacionamento entre as tabelas das entidades TipoDeQuarto e Quarto.

#### 4.1.3. Relacionamentos N:M

Na base de dados em apresentação existem dois relacionamentos N:M.

Quando temos relacionamentos N:M, não se consegue definir imediatamente onde colocar a chave estrangeira. A fim de resolver esta dificuldade cria-se uma nova tabela que represente o relacionamento, incluindo esta atributos que possam fazer parte dele e fazê-lo então uma cópia da chave primária das entidades que fazem parte do relacionamento para a tabela, atuando como chaves estrangeiras.

Temos como primeiro relacionamento N:M o relacionamento entre a entidade “Reserva” e a entidade “Serviço”, para o qual foi criada uma tabela “ReservaServiço”. Esta tabela funciona como um desenrolar do relacionamento N:M que antigamente tínhamos, convertendo-o num relacionamento 1:N e M:1. Assim, a chave primária da nova tabela criada passa a ser uma chave constituída pelas chaves primárias pertencentes às duas tabelas que lhe dão origem.

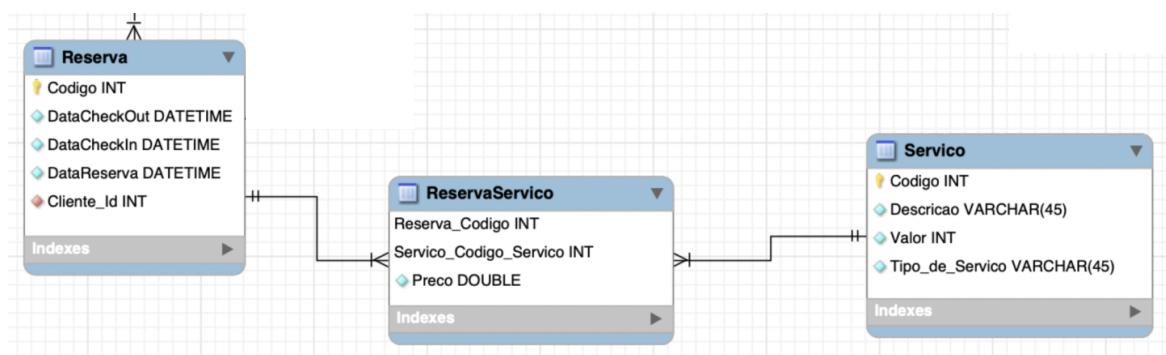


Figura 10 - Relacionamento entre as tabelas das entidades Reserva e Servico.

Como segundo relacionamento N:N temos o relacionamento entre a entidade “Reserva” e a entidade “Quarto”, para o qual foi criada uma tabela “ReservaQuarto”. Assim como no primeiro relacionamento N:N explanado, esta tabela funciona como um desenrolar do relacionamento N:N que antigamente tínhamos, convertendo-o num relacionamento 1:N e N:1. Assim, a chave primária da nova tabela criada passa a ser uma chave constituída pelas chaves primárias pertencentes às duas tabelas que lhe dão origem.

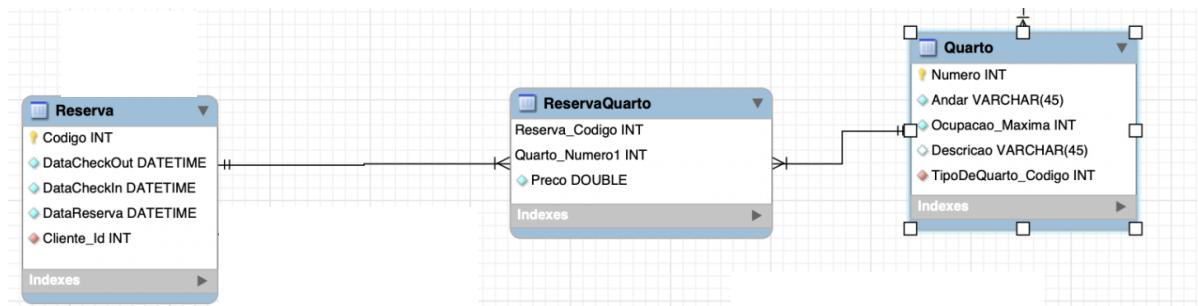


Figura 11 - Relacionamento entre as tabelas das entidades Reserva e Quarto.

#### 4.1.4. Relacionamentos 1:1

Na base de dados em apresentação não existem relacionamentos 1:1.

Um relacionamento 1:1 consiste num relacionamento entre duas entidades em que ambas têm uma entidade única.

#### 4.2. Desenho do modelo lógico

Após a apresentação da construção e validação do modelo lógico assim como a análise e interpretação de todos os componentes lógicos, expomos o respetivo esquema neste subcapítulo.

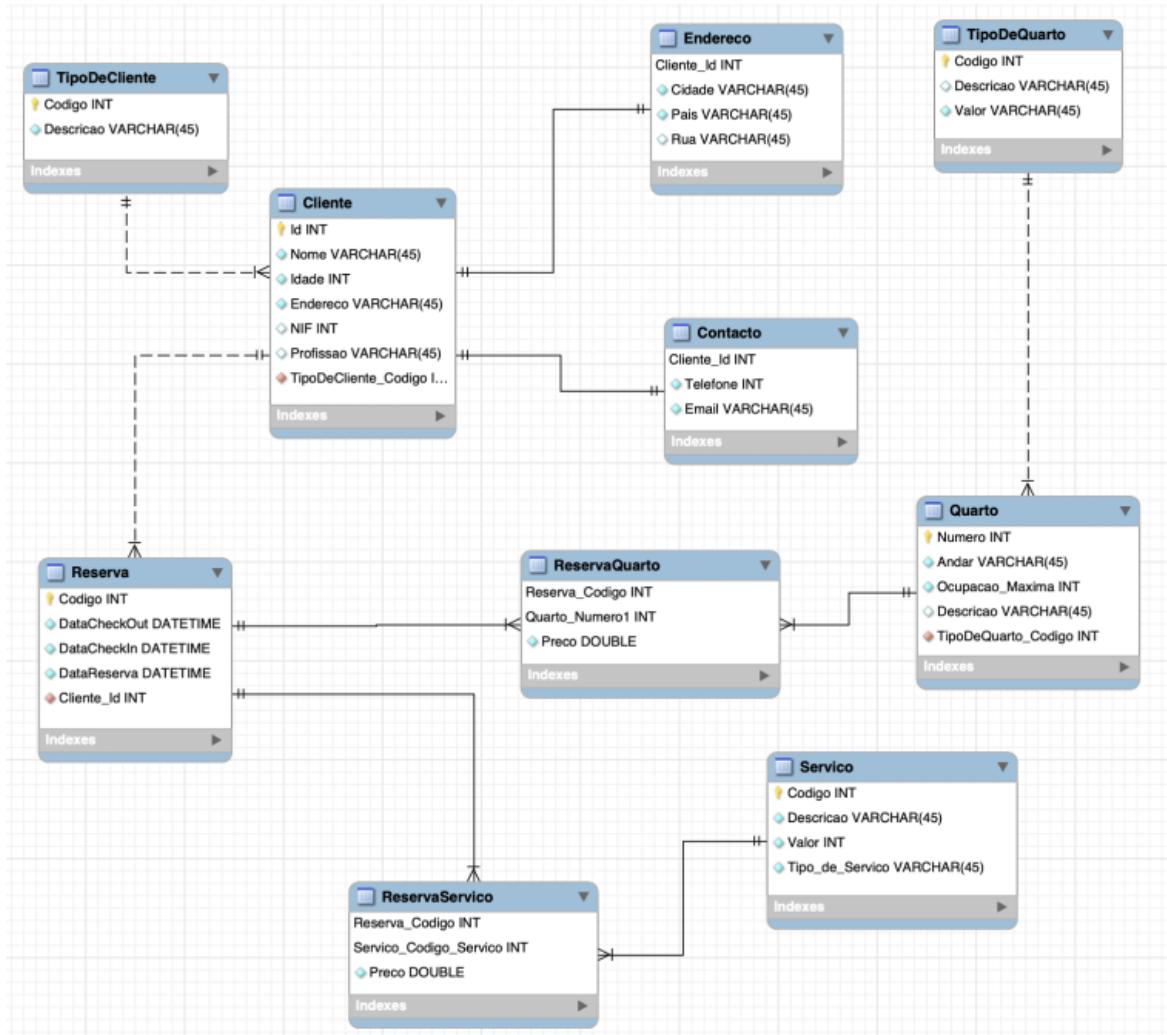


Figura 12 – Esquema do modelo lógico da Base de Dados.

#### 4.3. Validação do modelo através da normalização

Perante a decisão de desenvolvimento de uma base de dados há alguns aspetos que temos em mente e que destacamos pelo seu nível de importância ao implementar um sistema de gestão e ao desenvolver uma base de dados. É o caso das vantagens que esta nos pode oferecer. No entanto, e indo de encontro à menção das conveniências de uma base de dados (pois estas podem variar entre o poder de controlo da redundância e o controlo de acesso à informação), há imensos aspetos, principalmente os que relacionam a redundância, a integridade e a complexidade dos

dados, que podem surgir e levantar problemas caso não sejam tomados os cuidados necessários.

Abordando o conceito de “normalização” através de uma perspetiva mais teórica podemos concentrar-nos na ideia de que a normalização é, como referido nas aulas teóricas, um processo de remoção de dados redundantes das tabelas que a constituem, com o intuito de aumentar a eficiência de armazenamento e a integridade dos dados. De forma a conseguir certos objetivos de competência, desde o controlo efetivo da redundância dos dados, garantia de integridade dos dados e eficiência devem ser seguidas as regras de normalização – formas normais – que consistem em regras para a verificação da eficiência da base de dados.

Assim, consegue obter-se uma maior rapidez no acesso a toda a informação e maior facilidade na alteração de dados ou recuperação destes em caso de falha de sistema.

#### **4.4. Validação do modelo com interrogações do utilizador**

De modo a verificar a validade do modelo lógico desenvolvido tendo como base algumas interrogações do utilizador optamos por seletar as interrogações de maior importância referidas nos requisitos de exploração da base de dados e passar à explanação da satisfação destas por parte do modelo lógico desenvolvido.

##### **a) Obter a listagem dos clientes (incluindo os seus nomes e cidades) relacionados a um determinado país**

Nesta interrogação começa-se por fazer a junção das tabelas Endereço e Cliente utilizando a chave estrangeira TipoDeCliente\_Código do Cliente e a chave primária Cliente\_Id do Endereço. Após a junção destas tabelas obtém-se a tabela resultante que irá fazer a seleção dos clientes cujo país é o desejado, ou seja é o país que vamos querer procurar.

Para finalizar, faz-se uma projeção dos atributos nome do Cliente e da sua cidade.

**b) Obter o tipo de clientes**

Nesta interrogação usa-se a união das tabelas Cliente e tipo de cliente usando a chave primária Código do tipo de cliente e a chave estrangeira TipoDeCliente\_codigo na tabela cliente. Após o ‘inner join’ das tabelas obtem-se uma outra tabela que contém todos os clientes com o mesmo código de tipo de cliente, ou seja, se o argumento for estudante a tabela resultante será todos os clientes estudantes do hotel. Sobre este resultado temos o nome, id, idade, profissao de todos os clientes de cada tipo em específico.

## **4.5. Revisão do modelo lógico produzido**

Sabe-se que o modelo lógico é consideravelmente mais complexo do que o modelo conceptual desenvolvido anteriormente. Por isto mesmo, introduzimo-lo com uma breve apresentação geral, análise e interpretação para que se possa confirmar que o modelo cumpre os objetivos e necessidades requeridas.

Após a apresentação e verificação de todas as entidades do modelo presentes no modelo lógico, de todos os atributos associados necessários de modo a armazenar toda a informação vital e de obter resposta a todos os objetivos, e da utilização da normalização onde foi possível verificar que as relações são capazes de suportar os requisitos, verifica-se a validade deste modelo, capaz de cumprir as exigências requeridas.

## **5. Implementação física**

### **5.1. Seleção do sistema de gestão de bases de dados**

Foi necessário decidir qual o Sistema de gestão de base de dados a utilizar após desenhamos os modelos conceptual e lógico da base de dados. O mais adequado seria um Sistema de gestão que fosse gratuito, de fácil utilização e principalmente que tivesse um bom desempenho. De acordo com as necessidades, um Sistema de bases de dados relacional seria o mais apropriado.

O facto desta base de dados ser de tamanho reduzido facilitou também a escolha do Sistema de gestão de base de dados a recorrer uma vez que não é necessária uma ferramenta tão complexa.

Considerando todas as necessidades referidas decidimos então optar pelo MySQL, por se revelar um excelente suporte com funcionalidades testadas em situações de utilização extremas e uma multitude de ferramentas compatíveis com ele, tais como o MySQL Workbench.

### **5.2. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL**

Nesta etapa, como o título do capítulo refere, temos de tratar a tradução do modelo lógico para MySQL. Tendo em conta o sistema de gestão de bases de dados pela qual optamos para a implementação da base de dados, esta tradução é um processo direto uma vez que o MySQL é um DBMS ( DataBase Management System) relacional.

Para uma aproveitação máxima das funcionalidades do MySQL decidimos também, durante todo o processo de desenvolvimento, utilizar a ferramenta de desenho, desenvolvimento e administração de base de dados, MySQL Workbench.

Uma vez que criamos o nosso modelo lógico nesta ferramenta, utilizamos o Forward Engineering que esta nos disponibiliza. Tal como o seu nome indica, permite-nos automatizar o processo top-down de construção de componentes de baixo nível, neste caso a implementação no sistema de gestão de base de dados, através de uma abstração de alto nível, o modelo lógico.

Uma vez que conseguimos o código obtido, este foi revisto e executado, criando assim a base de dados.

### 5.3. Tradução das interrogações do utilizador para SQL (alguns exemplos)

O primeiro requisito de exploração consiste na obtenção da listagem de clientes (obtendo os seus nomes e as suas cidades) a partir do país a que estão associados. Para isso, foi criada uma procedure que retorna o nome do cliente e a cidade do seu endereço.

```
--  
-- Criacao de um procedimento para nos indicar a cidade e o nome do cliente  
-- dado um pais.  
DELIMITER $$  
CREATE PROCEDURE spCidadeNome  
    (IN pais VARCHAR(100))  
BEGIN  
    SELECT E.Cidade AS Cidade,  
        C.nome AS NomeCliente  
    FROM Cliente AS C INNER JOIN Endereco AS E  
        ON C.id=E.Cliente_Id  
        WHERE E.Pais = pais;  
END $$
```

Figura 13 - Resolução da Query para obter o nome dos Clientes e as suas cidades a partir do país.

	Cidade	NomeCliente
▶	Madrid	Rocío Garcia Camacha
	Barcelona	Matteo Pietro Dybala
	Sevilha	Alessio Cunha Batina
	Granada	Teresa Seco
	Madrid	Javi Puertes
	Barcelona	Alvarez de Lucas

Figura 14 - Resultado da Query da Figura 13, dado o país “Espanha”.

O segundo requisito de exploração indica-nos os tipos de clientes existentes na nossa base de dados. Para isso, foi criada uma procedure que retorna uma tabela com todos os atributos do cliente.

```
--  
-- Criacao de um procedimento para nos indicar os tipos dos Clientes.  
DELIMITER $$  
CREATE PROCEDURE spTiposDeClientes  
(IN cod INT)  
BEGIN  
    SELECT C.Id AS Id, C.nome AS nome, C.Idade AS Idade,  
    C.Profissao AS Profissao, TC.Descricao AS TipoDeCliente  
    FROM cliente AS C INNER JOIN tipodecliente AS TC  
    ON C.TipoDeCliente_Codigo=TC.Codigo  
    WHERE C.TipoDeCliente_Codigo = cod;  
  
END $$
```

Figura 15 – Resolução da Query para obter os tipo de clientes.

	Id	nome	Idade	Profissao	TipoDeCliente
▶	1	António Oliveira e Castro	53	Empresário	Pais de alunos
	12	Antonio Jose da Costa Afonseca	62	Militar	Pais de alunos
	13	Anabela da Silva Sousa	58	Empresaria	Pais de alunos
	17	Alessio Cunha Batina	55	Eletricista	Pais de alunos
	19	Luis Faria Macedo	65	Dentista	Pais de alunos
	26	Jorge Vieira	54	Tecnico	Pais de alunos
	27	Elvira Loureiro	57	Gestora	Pais de alunos

Figura 16 – Resultado da Query da figura 15.

O terceiro requisito de exploração consiste na obtenção do número dos quartos reservados numa determinada data. Para isso, foi criada uma procedure que retorna o número do quarto.

```
--  
-- Criacao de um procedimento para nos indicar o numero dos quartos reservados  
-- numa Data.  
DELIMITER $$  
CREATE PROCEDURE spNumeroDeQuartosData  
    (IN Mes INT, IN Ano INT)  
BEGIN  
    SELECT C.Id, C.Nome, RE.DataCheckIn, rq.Quarto_Numer01 AS NumeroQuarto  
    FROM reserva AS RE INNER JOIN cliente AS C  
        ON RE.Cliente_Id = C.Id  
        INNER JOIN reservaquearto AS rq  
            ON re.Codigo = rq.Reserva_Codigo  
    WHERE  
        MONTH(RE.DataCheckIn) = Mes AND  
        YEAR(RE.DataCheckIn) = Ano  
        GROUP BY NumeroQuarto;  
END $$
```

Figura 17 – Resolução da Query sobre a Listagem dos Quartos ocupados numa determinada data.

	Id	Nome	DataCheckIn	NumeroQuarto
▶	4	Rocío Garcia Camacha	2020-05-05 00:00:00	31
	23	Constanca Cunha	2020-05-11 00:00:00	32
	18	Teresa Seco	2020-05-14 00:00:00	33

Figura 18 – Resultado da Query da figura 17 utilizando o mês de maio e ano de 2020.

O quarto requisito de exploração consiste na obtenção dos clientes que usaram um determinado serviço. Para isso, foi criada uma procedure que recebe um serviço e retorna o código Id, nome, profissão e idade.

```
-- 
-- Criacao de um procedimento para nos indicar os Clientes que reservaram
-- um determinado serviço
DELIMITER $$

CREATE PROCEDURE spTiposDeServicos
    (IN codServiço INT)
BEGIN
    SELECT C.Id AS Id, C.nome AS nome, C.Idade AS Idade,
    C.Profissao AS Profissao
    FROM cliente AS C INNER JOIN reserva AS R
    ON C.Id=R.Cliente_Id
    INNER JOIN reservaservico AS RS
    ON R.Codigo=RS.Reserva_Codigo
    INNER JOIN servico AS S
    ON RS.Servico_Codigo_Servico=S.Codigo
    WHERE S.Codigo = codServiço;
END $$
```

Figura 19 – Realização do procedure spTiposDeServicos que dado um serviço retorna os clientes que o usaram.

	Id	nome	Idade	Profissao
▶	4	Rocío Garcia Camacha	21	Estudante
	13	Anabela da Silva Sousa	58	Empresaria
	6	Matteo Pietro Dybala	24	Estudante

Figura 20 - Resultado do procedure spTiposDeServicos com o input campo de futebol.

O quinto requisito de exploração consiste na obtenção do valor total que um cliente gastou no hotel. Para isso, foi criada uma function que recebe um cliente e retorna o valor total pago por este.

```
-- F U N C O E S
-- Criacao de uma função para nos indicar o que foi pago para um
-- dado cliente.
DELIMITER $$

CREATE FUNCTION fuPagoCliente
(cliente1 VARCHAR(100))
RETURNS VARCHAR(100)
DETERMINISTIC

BEGIN
DECLARE pagar VARCHAR(100);
SELECT SUM(re.preco) INTO pagar
FROM cliente AS C
INNER JOIN reserva AS R
ON C.id = R.Cliente_Id
INNER JOIN reservaquarto AS re
ON R.Codigo = re.Reserva_Codigo
WHERE C.nome =cliente1;
RETURN pagar;
END $$
```

Figura 21 – Realização da function fuPagoCliente que dado um cliente retorna o valor total pago.

	fuPagoCliente('Diogo Pereira Catalao Da Costa')
▶	50

Figura 22 – Resultado da function fuPagoCliente com o argumento ('Diogo Pereira Catalao Da Costa)

O sexto requisito de exploração consiste na obtenção dos clientes que fizeram uma reserve num determinado ano e mês. Para isso, foi criada uma procedure que recebe um ano e um mes e retorna os clientes nesse ano e nesse mes.

```

-- P R O C E D I M E N T O S
-- Criacao de um procedimento para nos indicar os Clientes que reservaram
-- numa Data.

DELIMITER $$

CREATE PROCEDURE spReservasPorAnoMes
    (IN Mes INT, IN Ano INT)
BEGIN
    SELECT C.Id, C.nome, RE.DataCheckIn
    FROM reserva AS RE INNER JOIN cliente AS C
        ON RE.Cliente_Id = C.Id
    WHERE
        MONTH(RE.DataCheckIn) = Mes AND
        YEAR(RE.DataCheckIn) = Ano;

END $$
```

Figura 23 - Realização do procedure spReservasPorAnoMes que dado um mês e um ano retorna os clientes que fizeram uma determinada reserve nesse tempo.

	Id	nome	DataCheckIn
▶	4	Rocío Garcia Camacha	2020-05-05 00:00:00
	23	Constanca Cunha	2020-05-11 00:00:00
	18	Teresa Seco	2020-05-14 00:00:00

Figura 24 – Resultado do procedure spReservasPorAnoMes para o input do ano igual a 2020 e do mes de Maio.

#### 5.4. Escolha, definição e caracterização de índices em SQL (alguns exemplos)

Devido à nossa base de dados ser reduzida, o índice não é relevante para o desenvolvimento do nosso trabalho. Se a nossa Base de Dados fosse povoada com centenas ou milhares de dados, aí seria importante implementar os índices de modo a uma ter uma maior acessibilidade para os mesmos.

## 5.5. Estimativa do espaço em disco de base de dados e taxa de crescimento anual

Com o desenvolvimento de uma base de dados é necessário ter em conta o espaço em disco que está ou será utilizado para armazenar todos os dados que pretendemos, de forma a garantir um funcionamento adequado da base de dados (uma performance desejada). Cada registo de informação de uma determinada tabela ocupa espaço físico na memória, dependendo do tipo de dados a que este está associado.

Assim, de forma a calcular o tamanho da base de dados, estima-se o tamanho de cada tabela e, após esta estimativa, somam-se os valores obtidos de forma a conseguirmos obter um valor total.

Para tal, utilizou-se o seguinte código SQL:

```
SELECT TABLE_NAME 'Nome Tabela',table_rows 'Entradas',
       data_length /1024/1024 'Dados(MB)', INDEX_LENGTH/1024/1024 'Indices(MB)',
       ROUND((DATA_LENGTH + INDEX_LENGTH) / 1024 / 1024,5) 'Size(MB)'
  FROM information_schema.tables
 WHERE table_schema = "hotel";
```

Figura 25 - Código em SQL que permite verificar o tamanho em MB de cada tabela da base de dados.

Em que se obteve a seguinte tabela:

	Nome Tabela	Entradas	Dados(MB)	Indices(MB)	Size(MB)
▶	diente	30	0.01562500	0.01562500	0.03125
	contacto	30	0.01562500	0.00000000	0.01563
	endereco	30	0.01562500	0.00000000	0.01563
	quarto	28	0.01562500	0.01562500	0.03125
	reserva	20	0.01562500	0.01562500	0.03125
	reservaquarto	20	0.01562500	0.01562500	0.03125
	reservaservico	19	0.01562500	0.03125000	0.04688
	servico	11	0.01562500	0.00000000	0.01563
	tipodecliente	5	0.01562500	0.00000000	0.01563
	tipodequarto	10	0.01562500	0.00000000	0.01563
	vwmelhoresdi...	NULL	NULL	NULL	NULL

Figura 26 - Tamanho (em MB) de cada tabela.

Depois foi executada a consulta do espaço total da base de dados:

```
SELECT table_schema "hotel",
       sum( data_length + index_length ) / 1024 / 1024 "Data Base Size in MB"
  FROM information_schema.TABLES GROUP BY table_schema like 'hotel';
```

Figura 27 – Tamanho total em disco da base de dados.

Onde obtivemos o seguinte resultado:

	hotel	Data Base Size in MB
▶	mysql	10.15625000
	hotel	0.25000000

Figura 28 – Tamanho total em disco.

Com isto concluímos que a base de dados inicial irá ocupar um total de 0,250 MB.

Após recolhidas as informações necessárias juntamente com o proprietário do Hotel soube-se que a taxa de crescimento anual do Hotel D'UMinho é aproximadamente 40%.

Tendo em mente este valor e fazendo os cálculos estima-se que o tamanho inicial da base de dados irá aumentar cerca de 0,0812 MB por ano.

## 5.6. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)

De modo a integrar nas informações do Hotel o top 5 de melhores clientes, isto é, os 5 clientes do Hotel que usufruem de mais serviços no Hotel investindo uma maior quantia monetária, decidiu-se criar uma vista. Neste caso em específico, a vista permite o acesso aos cinco melhores clientes do hotel.

```
-- Criacao da vista "vwMelhorClientes"
CREATE VIEW vwMelhoresClientes AS
    SELECT
        C.Id AS id,
        C.nome AS Nome,
        COUNT(R.Codigo) AS Reserva,
        SUM(re.preco) AS TotalPago
    FROM cliente AS C
        INNER JOIN reserva AS R
            ON C.id = R.Cliente_Id
                INNER JOIN reservaquarto AS re
                    ON R.Codigo = re.Reserva_Codigo
    GROUP BY C.id
    ORDER BY TotalPago DESC
    LIMIT 5;
```

Figura 29 - Vista que permite o acesso aos cinco melhores clientes do Hotel.

Criamos ainda uma segunda vista, que nos permite o acesso às reservas canceladas uma vez que, como mencionado na introdução do presente relatório, por motivos pandémicos a probabilidade de cancelamento de reservas pode aumentar. Uma vez que não se prevêm, atualmente, uma melhoria do estado a nível mundial, não é possível conseguir datas para adiar as reservas e, por isso, a única solução consiste no cancelamento destas.

```
-- Criacao da vista "vwReservasCanceladas" onde o resultado é uma lista de reservas canceladas
CREATE VIEW vwReservasCanceladas AS
    SELECT R.Codigo 'Codigo Reserva', C.Id, C.nome 'Nome', R.DataCheckIn, R.DataCheckOut
    FROM reserva AS R INNER JOIN cliente AS C
    ON R.Cliente_Id = C.Id
    WHERE R.Observacoes LIKE 'Cancelada';
```

Figura 30 - Vista que permite o acesso às reservas canceladas do Hotel.

Criamos uma terceira e última vista, que nos permite o acesso à listagem de quarto de valor menor ou igual a 50 euros.

```
-- 
-- Quartos de valor igual ou abaixo de 50 euros.
CREATE VIEW vwQuartosMaisBaratos AS
    SELECT Q.Numero, Q.Andar, Q.Ocupacao_Maxima AS 'Ocupacao Maxima', TQ.Valor
    FROM quarto AS Q INNER JOIN tipodequarto AS TQ
    ON Q.TipoDeQuarto_Codigo=TQ.Codigo
    WHERE TQ.Valor <= 50
    ORDER BY TQ.Valor ASC;
```

Figura 31 – Vista que permite o acesso à listagem de quartos de valor menor ou igual a 50€.

## **5.7. Revisão do Sistema implementado**

Revistos os requisitos apresentados no subcapítulo 2.2., verificou-se com o utilizador que todos estão corretamente implementados no modelo físico. Em particular, os requisitos de descrição listados em 2.2.1. foram corretamente traduzidos, pelo modelo Lógico, que foi validado em 4.4., para o modelo físico. Os seis requisitos de exploração listados em 2.2.2. encontram-se implementados e validados, pela respetiva ordem, no subcapítulo 5.3. Finalmente os requisitos de controlo apresentados em 2.2.3.

Para além disso, várias questões pertinentes apenas à implementação física da base de dados puderam agora também ser validadas com sucesso pelo utilizador, tais como as estimativas de espaço em disco, que tomando como valores de entrada as projeções de novas reservas, permitiu calcular o tamanho necessário para armazenar em disco uma base de dados que cumpra todas as exigências do utilizado.

## **5.8. Siglas e Acrónimos**

<b>ER</b>	Entidade-Relacionamento
<b>DBMS</b>	DataBase management system
<b>SQL</b>	Structured Query Language

## **5.9. Referências Bibliográficas**

<https://www.techopedia.com/definition/19445/forward-engineering> acedido em 28/11/2020

C.J. Date, 2003, *An Introduction to Database Systems - Eighth Edition*



## 6. Conclusões e Trabalho Futuro

Para concluir este projeto apresentamos uma apreciação crítica do trabalho realizado.

Começamos por fazer uma observação breve e por considerar todo o nosso modelo conceptual. A nível conceptual, poderiam ter sido exploradas novas funcionalidades que o hotel tem a capacidade de oferecer, e que, no entanto, não foram expostas nos requisitos. Serve como exemplo as refeições que o hotel tem para oferecer e ainda a possibilidade de prescrição a um cartão gold para todos os estudantes e clientes habituais do hotel de forma a obterem benefícios monetários.

A implementação da base de dados proposta neste relatório resulta de uma abordagem que nos pareceu adequada à resolução do problema, onde, na nossa opinião, os principais objetivos foram alcançados.

No entanto, analisando criteriosamente a nossa base de dados, assim como a existência de pontos bons, é também clara a percepção da existência de pontos mais fracos.

Com a realização deste trabalho conseguimos acompanhar todas as etapas necessárias ao desenvolvimento de uma base de dados, desde a mínima análise e levantamento de requisitos até à implementação de um modelo físico com a capacidade de suportar as transações de informação pretendidas.

Se olharmos para a análise e levantamento de requisitos, este permitiu-nos o desenvolvimento da capacidade de solucionar questões relacionadas com o problema.

Através do modelo conceptual conseguimos alcançar uma solução viável para o problema, com a capacidade de responder a todos os requisitos apresentados no capítulo 2.

Olhando de seguida para o desenvolvimento do modelo lógico, neste aprofundamos todos os conceitos do modelo conceptual e aproximando-o de um modelo de programação orientada a SGBD.

Quanto à tradução do modelo lógico para físico, permitiu-nos o aprofundamento de conhecimentos de SQL através da criação do esquema físico da

base de dados, povoamento de tabelas necessário e consulta de informação pertinente através de queries.

Finalizando, e agora com uma perspetiva mais abrangente, conseguimos concluir que as bases de dados são imprescindíveis para o tratamento da informação e gestão da mesma, constituindo uma tecnologia bastante proveitosa principalmente na atualidade que nos oferece cada vez mais quantidade de informação de recorrente necessidade de armazenamento e gestão.



## I. Anexo 1 – Script completo da criação do esquema físico

```
-- MySQL Workbench Forward Engineering
```

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO
_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- -----  
-- Schema hotel  
-----
```

```
-- -----  
-- Schema hotel  
-----
```

```
CREATE SCHEMA IF NOT EXISTS `hotel` DEFAULT CHARACTER SET utf8 ;
USE `hotel` ;
```

```
-- -----  
-- Table `hotel`.`TipoDeCliente`  
-----
```

```
CREATE TABLE IF NOT EXISTS `hotel`.`TipoDeCliente` (
`Codigo` INT NOT NULL COMMENT 'Código do tipo do cliente',
`Descricao` VARCHAR(45) NOT NULL,
PRIMARY KEY (`Codigo`))
ENGINE = InnoDB;
```

```
-- -----  
-- Table `hotel`.`Cliente`
```

```

-----  

CREATE TABLE IF NOT EXISTS `hotel`.`Cliente` (  

  `Id` INT NOT NULL COMMENT 'Identificador do Cliente\n',  

  `Nome` VARCHAR(45) NOT NULL COMMENT 'Nome do Cliente\n',  

  `Idade` INT NOT NULL COMMENT 'Idade do Cliente\n',  

  `Endereco` VARCHAR(45) NOT NULL COMMENT 'Endereço do Cliente\n',  

  `NIF` INT NULL COMMENT 'NIF do Cliente\n',  

  `Profissao` VARCHAR(45) NULL COMMENT 'Profissão do Cliente\n',  

  `TipoDeCliente_Codigo` INT NOT NULL,  

  PRIMARY KEY (`Id`),  

  INDEX `fk_Cliente_TipoDeCliente1_idx` (`TipoDeCliente_Codigo` ASC) VISIBLE,  

  CONSTRAINT `fk_Cliente_TipoDeCliente1`  

    FOREIGN KEY (`TipoDeCliente_Codigo`)  

    REFERENCES `hotel`.`TipoDeCliente` (`Codigo`)  

    ON DELETE NO ACTION  

    ON UPDATE NO ACTION)  

ENGINE = InnoDB;

```

```

-----  

-- Table `hotel`.`Reserva`  

-----  

CREATE TABLE IF NOT EXISTS `hotel`.`Reserva` (  

  `Codigo` INT NOT NULL,  

  `DataCheckOut` DATETIME NOT NULL,  

  `DataCheckIn` DATETIME NOT NULL,  

  `DataReserva` DATETIME NOT NULL,  

  `Cliente_Id` INT NOT NULL,  

  PRIMARY KEY (`Codigo`),  

  INDEX `fk_Reserva_Cliente_idx` (`Cliente_Id` ASC) VISIBLE,  

  CONSTRAINT `fk_Reserva_Cliente`  

    FOREIGN KEY (`Cliente_Id`)  

    REFERENCES `hotel`.`Cliente` (`Id`)  

    ON DELETE NO ACTION  

    ON UPDATE NO ACTION)  

ENGINE = InnoDB;

```

```

-----  

-- Table `hotel`.`Endereco`  

-----
```

```
--  
CREATE TABLE IF NOT EXISTS `hotel`.`Endereco` (  
    `Cliente_Id` INT NOT NULL,  
    `Cidade` VARCHAR(45) NOT NULL,  
    `Pais` VARCHAR(45) NOT NULL,  
    `Rua` VARCHAR(45) NULL,  
    PRIMARY KEY (`Cliente_Id`),  
    CONSTRAINT `fk_Endereco_Cliente1`  
        FOREIGN KEY (`Cliente_Id`)  
        REFERENCES `hotel`.`Cliente` (`Id`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
--  
-- Table `hotel`.`Contacto`  
--  
CREATE TABLE IF NOT EXISTS `hotel`.`Contacto` (  
    `Cliente_Id` INT NOT NULL,  
    `Telefone` INT NOT NULL,  
    `Email` VARCHAR(45) NOT NULL,  
    PRIMARY KEY (`Cliente_Id`),  
    CONSTRAINT `fk_Contacto_Cliente1`  
        FOREIGN KEY (`Cliente_Id`)  
        REFERENCES `hotel`.`Cliente` (`Id`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
--  
-- Table `hotel`.`TipoDeQuarto`  
--  
CREATE TABLE IF NOT EXISTS `hotel`.`TipoDeQuarto` (  
    `Codigo` INT NOT NULL,  
    `Descricao` VARCHAR(45) NULL,  
    `Valor` VARCHAR(45) NOT NULL,  
    PRIMARY KEY (`Codigo`))  
ENGINE = InnoDB;
```

```

-----  

-- Table `hotel`.`Quarto`  

-----  

CREATE TABLE IF NOT EXISTS `hotel`.`Quarto` (  

    `Numero` INT NOT NULL,  

    `Andar` VARCHAR(45) NOT NULL,  

    `Ocupacao_Maxima` INT NOT NULL,  

    `Descricao` VARCHAR(45) NULL,  

    `TipoDeQuarto_Codigo` INT NOT NULL,  

    PRIMARY KEY (`Numero`),  

    INDEX `fk_Quarto_TipoDeQuarto1_idx` (`TipoDeQuarto_Codigo` ASC) VISIBLE,  

    CONSTRAINT `fk_Quarto_TipoDeQuarto1`  

        FOREIGN KEY (`TipoDeQuarto_Codigo`)  

        REFERENCES `hotel`.`TipoDeQuarto` (`Codigo`)  

        ON DELETE NO ACTION  

        ON UPDATE NO ACTION)  

ENGINE = InnoDB;

```

```

-----  

-- Table `hotel`.`ReservaQuarto`  

-----  

CREATE TABLE IF NOT EXISTS `hotel`.`ReservaQuarto` (  

    `Reserva_Codigo` INT NOT NULL,  

    `Quarto_Numer01` INT NOT NULL,  

    `Preco` DOUBLE NOT NULL,  

    PRIMARY KEY (`Reserva_Codigo`, `Quarto_Numer01`),  

    INDEX `fk_Reserva Quarto_Quarto1_idx` (`Quarto_Numer01` ASC) VISIBLE,  

    CONSTRAINT `fk_Reserva Quarto_Reserva1`  

        FOREIGN KEY (`Reserva_Codigo`)  

        REFERENCES `hotel`.`Reserva` (`Codigo`)  

        ON DELETE NO ACTION  

        ON UPDATE NO ACTION,  

    CONSTRAINT `fk_Reserva Quarto_Quarto1`  

        FOREIGN KEY (`Quarto_Numer01`)  

        REFERENCES `hotel`.`Quarto` (`Numero`)  

        ON DELETE NO ACTION  

        ON UPDATE NO ACTION)

```

```

ENGINE = InnoDB;

-----
-- Table `hotel`.`Servico`

CREATE TABLE IF NOT EXISTS `hotel`.`Servico` (
  `Codigo` INT NOT NULL,
  `Descricao` VARCHAR(45) NOT NULL,
  `Valor` INT NOT NULL,
  `Tipo_de_Servico` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Codigo`))
ENGINE = InnoDB;

-----
-- Table `hotel`.`ReservaServico`

CREATE TABLE IF NOT EXISTS `hotel`.`ReservaServico` (
  `Reserva_Codigo` INT NOT NULL,
  `Servico_Codigo_Servico` INT NOT NULL,
  `Preco` DOUBLE NOT NULL,
  PRIMARY KEY (`Reserva_Codigo`, `Servico_Codigo_Servico`),
  INDEX `fk_ReservaServiço_Reserva1_idx` (`Reserva_Codigo` ASC) VISIBLE,
  INDEX `fk_ReservaServiço_Serviço1_idx` (`Servico_Codigo_Servico` ASC) VISIBLE,
  CONSTRAINT `fk_ReservaServiço_Reserva1`
    FOREIGN KEY (`Reserva_Codigo`)
    REFERENCES `hotel`.`Reserva` (`Codigo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_ReservaServiço_Serviço1`
    FOREIGN KEY (`Servico_Codigo_Servico`)
    REFERENCES `hotel`.`Servico` (`Codigo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

## **II. Anexo 2 – Script completo da criação do esquema físico**

-- Implementação e Exploração de Sistemas de Bases de Dados Relacionais  
-- O caso de Estudo do "Hotel D'UMinho"  
-- Povoamento da Base de Dados Relacional  
--  
-- Obs.: Todos os objetos e elementos de dados manipulados pelos scripts SQL  
-- apresentados neste ficheiro são fictícios, tendo sido criados com o único  
-- objetivo de suportar a demonstração dos diversos tópicos e  
-- exercícios apresentadas e analisados nas aulas de bases de dados.

-- Indicação do esquema da base de dados de trabalho.

```
-----  
USE `hotel` ;
```

--  
-- Permissão para fazer operações de remoção de dados.  
SET SQL\_SAFE\_UPDATES = 0;

-- Povoamento da Tabela `hotel`.`TipoDeCliente`

```
-----  
-- DELETE FROM `hotel`.`TipoDeCliente`;
```

```
-- SELECT * FROM `hotel`.`TipoDeCliente`;
```

```
INSERT INTO `hotel`.`TipoDeCliente`
```

(Codigo,Descricao)

VALUES

('1','Estudante atual da Universidade'),

('2','Estudante de erasmus'),

('3','Futuro estudante na universidade'),

('4','Atividade lúdica na universidade'),

('5','Pais de alunos');

-- Povoamento da Tabela `hotel`.`Cliente`

-- DELETE FROM `hotel`.`Cliente`;

-- SELECT \* FROM `hotel`.`Cliente`;

INSERT INTO Cliente

(Id, Nome, Idade, Endereco, NIF, Profissao, TipoDeCliente\_Codigo)

VALUES

('1', 'António Oliveira e Castro', '53', '1', '999321678', 'Empresário', '5'),  
('2', 'Paula Maria da Silva Antunes', '38', '2', '893475978', 'Professora', '4'),  
('3', 'Grethel Pagny Tourret', '34', '3', '736497283', 'Engenheira de Polímeros', '4'),  
('4', 'Rocío Garcia Camacha', '21', '4', '937482903', 'Estudante', '1'),  
('5', 'Diogo Pereira Catalão da Costa', '28', '5', '837462836', 'Veterinário', '3'),  
('6', 'Matteo Pietro Dybala', '24', '6', '354723947', 'Estudante', '2'),  
('7', 'José João Marques Gouveia', '28', '7', '673549074', 'Médico', '4'),  
('8', 'Zacarias Alfredo Monteiro', '31', '8', '673298467', 'Professor', '4'),  
('9', 'Graziela Pietra Antonella', '24', '9', '746352839', 'Estudante', '2'),  
('10', 'Diego Murilo', '45', '10', '373649278', 'Professor', '4'),  
('11', 'Caroline Anita Fisz', '24', '11', '837263740', 'Estudante', '2'),  
('12', 'Antonio Jose da Costa Afonseca', '62', '12', '198747389', 'Militar', '5'),  
('13', 'Anabela da Silva Sousa', '58', '13', '284736256', 'Empresaria', '5'),  
('14', 'Ricardo Joao Monteiro Abelheira', '35', '14', '182736483', 'Engenheiro Informatico', '4'),  
('15', 'Maria Beatriz Almeida DEca', '19', '15', '183746287', 'Estudante', '1'),  
('16', 'Alessandro Enrico Dante', '23', '16', '873746183', 'Estudante', '2'),  
('17', 'Alessio Cunha Batina', '55', '17', '986734872', 'Eletricista', '5'),  
('18', 'Teresa Seco', '22', '18', '897423748', 'Estudante', '2'),  
('19', 'Luis Faria Macedo', '65', '19', '764326738', 'Dentista', '5'),  
('20', 'Marta Côncio', '22', '20', '378624763', 'Estudante', '3'),  
('21', 'Roberto Cavalli', '46', '21', '349283756', 'Contabilista', '4'),  
('22', 'Javi Puertes', '17', '22', '876473928', 'Estudante', '3'),  
('23', 'Constanca Cunha', '18', '23', '897764637', 'Estudante', '1'),  
('24', 'Goncalo Ribeiro', '35', '24', '786439287', 'Escritor', '4'),  
('25', 'Alvarez de Lucas', '18', '25', '875374892', 'Estudante', '3'),  
('26', 'Jorge Vieira', '54', '26', '876435748', 'Tecnico', '5'),  
('27', 'Elvira Loureiro', '57', '27', '874928378', 'Gestora', '5'),  
('28', 'Gervasio Gomes', '42', '28', '463748963', 'Professor', '4'),

('29', 'Roberto Carlos', '35', '29', '398756382', 'Engenheiro de Telecomunicoes', '4'),  
('30', 'Rodrigo Araujo Pereira', '18', '30', '783946102', 'Estudante', '1');

```
-- DELETE FROM `hotel`.`Endereco`;  
-- SELECT * FROM `hotel`.`Endereco`;
```

```
INSERT INTO Endereco  
    (Cliente_id,Cidade,Pais,Rua)  
VALUES  
    ('1', 'Porto', 'Portugal', 'Rua da Juventude'),  
    ('2', 'Lisboa', 'Portugal', 'Rua dos Pássaros'),  
    ('3', 'Paris', 'França', 'Rue de La Vie'),  
    ('4', 'Madrid', 'Espanha', 'Plaza del Real'),  
    ('5', 'Beja', 'Portugal', 'Avenida da Felicidade'),  
    ('6', 'Barcelona', 'Espanha', 'Plaza del Cariño'),  
    ('7', 'Viana do Castelo', 'Portugal', 'Rua do Zé da Esquina'),  
    ('8', 'Guarda', 'Portugal', 'Rua Do Monte de Baixo'),  
    ('9', 'Florencia', 'Italia', 'Strada di Mozzarela'),  
    ('10', 'Paris', 'França', 'Rue Croissant'),  
    ('11', 'Rio de Janeiro', 'Brasil', 'Rua de Dezembro'),  
    ('12', 'Coimbra', 'Portugal', 'Avenida da Boavista'),  
    ('13', 'Guimaraes', 'Portugal', 'Rua Juvenil'),  
    ('14', 'Aveiro', 'Portugal', 'Rua de Veneza'),  
    ('15', 'Braga', 'Portugal', 'Rua da Boanova'),  
    ('16', 'Milao', 'Italia', 'Strada di Buongiorno'),  
    ('17', 'Sevilha', 'Espanha', 'Calle del Bailado'),  
    ('18', 'Granada', 'Espanha', 'Calle del Hombre'),  
    ('19', 'Braga', 'Portugal', 'Rua Feliz'),  
    ('20', 'Funchal', 'Portugal', 'Rua das Almas'),  
    ('21', 'Veneza', 'Italia', 'Piazza di Spaghetti'),  
    ('22', 'Madrid', 'Espanha', 'Plaza del Puto Amo'),  
    ('23', 'Braga', 'Portugal', 'Avenida da Foz'),  
    ('24', 'Braga', 'Portugal', 'Rua das Cores'),  
    ('25', 'Barcelona', 'Espanha', 'Plaza del Tigre'),  
    ('26', 'Porto', 'Portugal', 'Rua do Compromisso'),  
    ('27', 'Sintra', 'Portugal', 'Rua Daqui'),  
    ('28', 'Braga', 'Portugal', 'Rua Feliz'),
```

('29', 'Zaragoza', 'Espanha', 'Plaza del Pilar'),  
(‘30’, ‘Porto’, ‘Portugal’, ‘Praça da Misericordia’);

-- Povoamento da Tabela `hotel`.`Contacto`

-- DELETE FROM `hotel`.`Cliente`;

-- SELECT \* FROM `hotel`.`Contacto`;

INSERT INTO `hotel`.`Contacto`

(Cliente\_Id, Email, Telefone )

VALUES

(‘1’, ‘antonio@gmail.com’, ‘953647382’),  
(‘2’, ‘paula@gmail.com’, ‘915638462’),  
(‘3’, ‘tourretgretel@gmail.com’, ‘467283946’),  
(‘4’, ‘rocio@gmail.com’, ‘874635270’),  
(‘5’, ‘dioguinhoc37@gmail.com’, ‘946730274’),  
(‘6’, ‘matteo@hotmail.com’, ‘678354673’),  
(‘7’, ‘jjgouveia@gmail.com’, ‘926492856’),  
(‘8’, ‘zaca@gmail.com’, ‘983748392’),  
(‘9’, ‘antonella@hotmail.com’, ‘637482936’),  
(‘10’, ‘diego@gmail.com’, ‘876337483’),  
(‘11’, ‘anita253@hotmail.com’, ‘273526389’),  
(‘12’, ‘ajc@outlook.com’, ‘624637285’ ),  
(‘13’, ‘anabela.sousa@gmail.com’, ‘588782934’),  
(‘14’, ‘ricas@gmail.com’, ‘358758494’),  
(‘15’, ‘mariazinha@hotmail.com’, ‘198748491’),  
(‘16’, ‘alessandro@gmail.com’, ‘897348392’),  
(‘17’, ‘alessiobatina@hotmail.com’, ‘558973485’ ),  
(‘18’, ‘teresaseco@gmail.com’, ‘229872543’),  
(‘19’, ‘luis@gmail.com’, ‘659487554’),  
(‘20’, ‘martinha@outlook@gmail.com’, ‘227382943’),  
(‘21’, ‘robertinho@gmail.com’, ‘46738494’),  
(‘22’, ‘javi@gmail.com’, ‘1772839483’),  
(‘23’, ‘constanca@hotmail.com’, ‘188374931’),  
(‘24’, ‘goncaloribeiro@gmail.com’, ‘357392834’),  
(‘25’, ‘alvarez@hotmail.com’, ‘189883933’),  
(‘26’, ‘jorginho@outlook.com’, ‘548973543’),  
(‘27’, ‘elvira@gmail.com’, ‘578374835’),

```
('28', 'gervasio@gmail.com', '423847392'),  
('29', 'robertocarlos@outlook.com', '358493844'),  
('30', 'rodri@gmail.com', '188984914');
```

```
-- Povoamento da Tabela `hotel`.`TipoDeQuarto`
```

```
-- DELETE FROM `hotel`.`TipoDeQuarto`;  
-- SELECT * FROM `hotel`.`TipoDeQuarto`;  
INSERT INTO `hotel`.`TipoDeQuarto`  
    (Codigo,Valor,Descricao)  
VALUES  
    ('1', '25', 'Quarto Simples Standard'),  
    ('2', '45', 'Quarto Duplo Standard'),  
    ('3', '65', 'Quarto Triplo Standard'),  
    ('4', '95', 'Quarto Quadruplo Standard'),  
    ('5', '120', 'Quarto Suite Standard'),  
    ('6', '50', 'Quarto Simples Deluxe'),  
    ('7', '65', 'Quarto Duplo Deluxe'),  
    ('8', '80', 'Quarto Triplo Deluxe'),  
    ('9', '100', 'Quarto Quadruplo Deluxe'),  
    ('10', '132', 'Quarto Suite Deluxe');
```

```
-- Povoamento da Tabela `hotel`.`Quarto`
```

```
-- DELETE FROM `hotel`.`Quarto`;  
-- SELECT * FROM `hotel`.`Quarto`;  
INSERT INTO `hotel`.`Quarto`  
    (Numero,Andar,Ocupacao_Maxima,TipoDeQuarto_Codigo,Descricao)  
VALUES  
    ('11','1','2','1',NULL),  
    ('12','1','2','2',NULL),  
    ('13','1','1','2',NULL),  
    ('14','1','2','1',NULL),  
    ('15','1','2','1',NULL),
```

```

        ('21','2','3','1','Quarto com pouca luminosidade'),
        ('22','2','2','2',NULL),
        ('23','2','3','3',NULL),
        ('24','2','4','3',NULL),
        ('25','2','2','3',NULL),
        ('31','3','4','4',NULL),
        ('32','3','2','4',NULL),
        ('33','3','3','5',NULL),
        ('34','3','4','4',NULL),
        ('35','3','1','4',NULL),
        ('41','4','2','6',NULL),
        ('42','4','3','7',NULL),
        ('43','4','4','6','Otima vista para a Universidade do Minho'),
        ('44','4','5','6',NULL),
        ('45','4','3','7',NULL),
        ('51','5','3','8',NULL),
        ('52','5','2','9',NULL),
        ('53','5','2','9',NULL),
        ('54','5','2','8',NULL),
        ('55','5','1','8',NULL),
        ('61','6','4','9',NULL),
        ('62','6','2','9',NULL),
        ('63','6','2','10','Melhor quarto do hotel');

```

-- Povoamento da Tabela `hotel`.`Reserva`

```

-----  

-- DELETE FROM `hotel`.`Reserva`;  

-- SELECT * FROM `hotel`.`Reserva`;  

INSERT INTO `hotel`.`Reserva`  

(Codigo,DataCheckIn,DataCheckOut,DataReserva,Cliente_id)  

VALUES  

        ('111','2020-02-26', '2020-02.28', '2020-02-14', '4'),  

        ('112','2020-05-05', '2020-05-07', '2020-02-14', '4'),  

        ('113','2020-07-14', '2020-07-16', '2020-02-14', '4'),  

        ('114','2020-06-05', '2020-06-10', '2020-05-04', '2'),  

        ('115','2020-07-05', '2020-07-10', '2020-06-10', '27'),  

        ('116','2020-01-03', '2020-01-10', '2019-11-23', '3'),  

        ('117','2020-10-14', '2020-10-16', '2020-10-14', '7'),  

        ('118','2020-11-21', '2020-11-23', '2020-10-16', '3'),  

        ('119','2019-12-01', '2019-12-05', '2019-11-15', '24'),
```

```

('120','2020-04-11', '2020-04-17', '2020-02-18', '13'),
    ('121','2020-06-05', '2020-06-10', '2020-05-04', '20'),
    ('122','2020-01-05', '2020-01-10', '2019-12-10', '10'),
    ('123','2020-01-03', '2020-01-10', '2019-11-23', '11'),
    ('124','2020-06-14', '2020-06-16', '2020-04-14', '8'),
    ('125','2020-11-21', '2020-11-23', '2020-10-16', '18'),
    ('126','2019-12-01', '2019-12-05', '2019-11-15', '6'),
    ('127','2020-05-11', '2020-09-17', '2020-04-18', '23'),
        ('128','2020-05-14', '2020-05-16', '2020-04-14', '18'),
    ('129','2020-11-21', '2020-11-23', '2020-10-16', '1'),
    ('130','2019-10-01', '2019-10-05', '2019-10-01', '5');

```

-- Povoamento da Tabela `hotel`.`ReservaQuarto`

-- DELETE FROM `hotel`.`ReservaQuarto`;

-- SELECT \* FROM `hotel`.`ReservaQuarto`;

INSERT INTO `hotel`.`ReservaQuarto`

(Reserva\_Codigo,Quarto\_Numero1,Preco)

VALUES

```

('111','42','65'),
('112','31','95'),
    ('113','61','100'),
    ('114','45','65'),
    ('115','21','25'),
    ('116','11','25'),
    ('117','62','100'),
    ('118','51','80'),
    ('119','35','95'),
    ('120','54','80'),
    ('121','13','45'),
    ('122','34','95'),
    ('123','45','65'),
    ('124','53','100'),
    ('125','63','132'),
    ('126','14','25'),
    ('127','32','95'),
        ('128','33','120'),
    ('129','62','100'),

```

```
('130','41','50');
```

```
-- Povoamento da Tabela `hotel`.`Servico`
```

```
-- DELETE FROM `hotel`.`Servico`;
```

```
-- SELECT * FROM `hotel`.`Servico`;
```

```
INSERT INTO `hotel`.`Servico`
```

```
(Codigo,Descricao,Valor,Tipo_de_Servico)
```

```
VALUES
```

```
('1','Acesso ao mini bar no seu quarto','10.54','MiniBar'),
```

```
('2','Acesso ao campo de futebol','35.00','Campo de futebol'),
```

```
('3','Estadia com pequeno almoço servido no quarto','27.45','Pequeno Almoco na Cama'),
```

```
('4','Acesso ao jacuzzi do hotel','40.00','Jacuzzi'),
```

```
('5','Acesso a piscina do hotel','20.57','Piscina'),
```

```
('6','Acesso ao spa do hotel','56.38','Spa'),
```

```
('7','Limpeza de quarto diaria','35.00','Limpeza Diária'),
```

```
('8','Acesso ao casino do hotel','23.00','Casino'),
```

```
('9','Acesso a sauna do hotel','46.00','Sauna'),
```

```
('10','Acesso ao ginasio do hotel','10.00','Ginasio'),
```

```
('11','Acesso a biblioteca do hotel','05.99','Biblioteca');
```

```
-- Povoamento da Tabela `hotel`.`ReservaServico`
```

```
-- DELETE FROM `hotel`.`ReservaServico`;
```

```
-- SELECT * FROM `hotel`.`ReservaServico`;
```

```
INSERT INTO `hotel`.`ReservaServico`
```

```
(Servico_Codigo_Servico,Reserva_Codigo,Preco)
```

```
VALUES
```

```
('4','111','10.54'),
```

```
('2','112','35.00'),
```

```
('7','113','27.45'),
```

```
('1','114','40.00'),
```

```
('9','116','10.57'),
```

```
('5','117','56.38'),
```

```
('3','118','35.00'),
```

```
('8','119','15.99'),
```

('2','120','35.00'),  
('6','121','56.38'),  
('4','122','10.54'),  
('10','123','10.00'),  
('8','124','23.00'),  
('1','125','10.54'),  
('2','126','35.00'),  
('8','127','23.00'),  
    ('5','128','20.57'),  
('7','129','35.00'),  
('4','130','40.00');

---

### **III. Anexo 3 – Exemplos de queries em SQL**

Neste anexo são apresentados exemplos de resoluções de algumas queries em SQL, assim como os seus resultados.

```
-- Consulta de todo o conteúdo da tabela "cliente".
SELECT *
    FROM cliente;

-- Consulta de todo o conteúdo da tabela "contacto".
SELECT *
    FROM contacto;

-- Consulta de todo o conteúdo da tabela "endereco".
SELECT *
    FROM endereco;

-- Consulta de todo o conteúdo da tabela "quarto".
SELECT *
    FROM quarto;

-- Consulta de todo o conteúdo da tabela "reserva".
SELECT *
    FROM reserva;
```

```
-- Consulta de todo o conteúdo da tabela "reservaservico".
SELECT *
FROM reservaservico;

-- Consulta de todo o conteúdo da tabela "servico".
SELECT *
FROM servico;

-- Consulta de todo o conteúdo da tabela "Cliente".
SELECT *
FROM tipodequarto;
```