



**Universidade do Minho**  
Licenciatura em Ciências da Computação

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

Ano Letivo de 2021/2022

### **Alojamento Local**

**David Machado, Inês Presa, Ivo Lima, Tiago Carriço**

Novembro, 2021

# BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Alojamento Local

**David Machado, Inês Presa, Ivo Lima, Tiago Carriço**  
**A91665, A90355, A90214, A91695**

Novembro, 2021

# 1 Resumo

Este relatório visa descrever o processo de implementação de uma base de dados para um estabelecimento de Alojamento Local, que permita facilitar a gestão do aluguer de alojamento temporário. O estabelecimento em questão insere-se na categoria de turismo rural, pelo que, apesar de ser uma instalação pequena, tem vindo a ter cada vez mais afluência de pessoas, pelo que precisa que os seus dados sejam mais bem organizados a fim de ter uma gestão mais eficiente.

Ao longo deste relatório pretende-se descrever todas as fases da implementação da base de dados, desde a sua idealização até à sua implantação física.

Primeiramente, é exposto o contexto do caso em questão, acompanhado da sua descrição e definição dos objetivos que se pretendem cumprir. S

Seguidamente, procede-se à exposição do método e respetivo levantamento e documentação dos requisitos determinados em conjunto com o senhor Luís, o dono do estabelecimento.

Numa seguinte fase, elabora-se o modelo concetual com recurso à ferramenta *brModelo*, que após estar definido é transposto para o respetivo modelo lógico, implementado utilizando o *MySQL Workbench*, que devido à sua versatilidade serviu também para a geração e implementação física da base de dados em questão.

Finalmente, é calculado o espaço ocupado pela base de dados e estimada a forma como esse valor vai evoluir.

Findado o processo a que nos propusemos neste trabalho, dá-se por terminado o presente relatório.

**Área de Aplicação:** Criação e Implementação de uma Base de Dados para Gestão de um Alojamento Local.

**Palavras-Chave:** Base de Dados, Entidade, Atributo, Relacionamento, Modelo Conceptual, Modelo Lógico, Modelo Físico, *MySQL*, *SQL*

## 2 Índice

1	Resumo	i
2	Índice	ii
3	Índice de Figuras	iv
4	Índice de Tabelas	v
1...	Introdução	1
1.1.	Contextualização	1
1.2.	Apresentação do Caso de Estudo	1
1.3.	Motivação e Objetivos	2
1.4.	Análise da Viabilidade do Processo	2
1.5.	Recursos e Equipa de Trabalho	2
1.6.	Estrutura do Relatório	3
2...	Levantamento e Análise de Requisitos	4
2.1.	Método de levantamento e de análise de requisitos adotado	4
2.2.	Organização dos requisitos levantados	4
2.2.1.	Requisitos de descrição	4
2.2.2.	Requisitos de exploração	4
2.2.3.	Requisitos de Controlo	5
2.2.4.	Análise e validação geral dos requisitos	5
3...	Modelo Conceptual	6
3.1.	Apresentação da abordagem de modelação realizada	6
3.2.	Identificação e caracterização das entidades	6
3.3.	Identificação e caracterização dos relacionamentos	6
3.4.	Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	8
3.5.	Apresentação e explicação do diagrama ER	9
3.6.	Validação do Modelo Conceptual	10
4...	Modelação Lógica	11
4.1.	Construção e validação do modelo de dados lógico	11
4.2.	Desenho do modelo lógico	12
4.3.	Validação do modelo com interrogações do utilizador	12
5...	Implementação Física	15
5.1.	Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	15
5.2.	Tradução das interrogações do utilizador para SQL	17
5.3.	Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	21
5.4.	Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)	21

5.5. Plano de segurança e recuperação de dados	24
6... Conclusões e Trabalho Futuro	25
5 Referências	26
6 Lista de Siglas e Acrónimos	27
7 Anexos	28
I. Anexo 1 - <i>Script</i> de Criação da Base de Dados	29
II. Anexo 2 - <i>Script</i> de Povoamento	32
III. Anexo 3 - <i>Script</i> de Implementação das Interrogações	46
IV. Anexo 4 - <i>Script</i> da Criação das Views	48

### 3 Índice de Figuras

Figura 1 - Relação entre Alojamento e Edifício	6
Figura 2 - Relação entre Cliente e Reserva	7
Figura 3 - Relação Funcionário Gerência	7
Figura 4 - Relação entre Reserva e Funcionário	7
Figure 5 - Relação entre Reserva e Alojamento	7
Figura 6 - Modelo Conceptual	9
Figura 7 - Modelo Lógico	12
Figura 8 - Árvore representativa da RE02	13
Figura 9 - Árvore representativa da RE03	13
Figura 10 - Árvore representativa do RE04	14
Figura 11 - Código <i>SQL</i> para criar tabela Edificio	15
Figura 12 - Código <i>SQL</i> para criar tabela Alojamento	15
Figura 13 - Código <i>SQL</i> para criar a tabela Funcionario	16
Figura 14 - Código <i>SQL</i> para criar tabela Cliente	16
Figura 15 - Código <i>SQL</i> para criar a tabela Reserva	16
Figura 16 - Código <i>SQL</i> para criar a tabela Reserva_Alojamento	17
Figura 17 - Código <i>SQL</i> para RE01	17
Figura 18 – Resultado da <i>Query</i> relativa a RE01	17
Figura 19 - Código <i>SQL</i> para RE02	18
Figura 20 - Resultado da <i>Query</i> relativa a RE02	18
Figura 21 - Código <i>SQL</i> para RE03	18
Figura 22 - Resultado da <i>Query</i> relativa a RE03	18
Figura 23 - Código <i>SQL</i> para RE04	19
Figura 24 - Resultado da <i>Query</i> relativa a RE04	19
Figura 25 - Código <i>SQL</i> para RE05	19
Figura 26 - Resultado da <i>Query</i> relativa a RE05	19
Figura 27 - Código <i>SQL</i> para RE06	20
Figura 28 - Resultado da <i>Query</i> relativa a RE06	20
Figura 29 - Código <i>SQL</i> para RE07	20
Figura 30 - Resultado da <i>Query</i> relativa a RE07	20
Figura 31 - Código <i>SQL</i> para RE08	20
Figura 32 - Resultado da <i>Query</i> relativa a RE08	20
Figura 33 - Código <i>SQL</i> para vista	21

## 4 Índice de Tabelas

Tabela 1 - Caracterização das Entidades	6
Tabela 2 - Atributos da entidade Cliente	8
Tabela 3 - Atributos da entidade Funcionário	8
Tabela 4 - Atributos da entidade Edifício	8
Tabela 5 - Atributos da entidade Alojamento	9
Tabela 6 - Atributos da entidade Reserva	9
Tabela 7- Referente à entidade Cliente	11
Tabela 8 - Referente à entidade Funcionario	11
Tabela 9 - Referente à entidade Edifício	11
Tabela 10 - Referente à entidade Alojamento	11
Tabela 11 - Referente à entidade Reserva	11
Tablela 12 - Referente à relação Reserva_Alojamento	11
Tabela 13 - Descrição dos tamanhos consoante o tipo	21
Tabela 14 - Tamanho (em bytes) dos atributos de Cliente	22
Tabela 15 - Tamanho (em <i>bytes</i> ) dos atributos de Funcionario	22
Tabela 16 - Tamanho (em <i>bytes</i> ) dos atributos de Edificio	22
Tabela 17 - Tamanho (em <i>bytes</i> ) dos atributos de Alojamento	22
Tabela 18 - Tamanho (em <i>bytes</i> ) dos atributos de Reserva	23
Tabela 19 - Tamanho (em <i>bytes</i> ) dos atributos de Reserva_Alojamento	23

# **1...Introdução**

## **1.1.Contextualização**

A interrupção da rotina de trabalho é imprescindível para o bem-estar físico e mental de qualquer pessoa, pelo que se torna crucial reservar alguns dias para descanso, já que é provado por vários estudos que as férias são importantes para a saúde cardiovascular, para melhorar os níveis de cortisol e a pressão arterial, entre outras coisas.

Assim, estabelecimentos de turismo são fortemente procurados e, em particular, hoje em dia as pessoas cada vez mais procuram passar férias fora da cidade em busca de tranquilidade, natureza, tradição e autenticidade, tornando assim o turismo rural uma prática cada vez mais frequente, uma vez que o campo é considerado por muitos como o “paraíso na terra”.

Portanto, ao longo dos últimos anos tem-se verificado um aumento progressivo da procura e frequência dos espaços rurais para o consumo e prática de atividades de turismo e lazer, especialmente por populações urbanas.

Sendo assim, a gestão de um estabelecimento de alojamento local representa um certo desafio, já que para além de ser importante manter um negócio viável, também é relevante respeitar as necessidades dos clientes e garantir uma estadia agradável.

## **1.2.Apresentação do Caso de Estudo**

O senhor Luís, que vive na aldeia das Eiras, a qual, devido à sua proximidade ao marco do quilómetro zero da Estrada Nacional 2, apresenta uma grande afluência de turistas, decidiu investir no seu terreno e aí construir dois edifícios para alojar viajantes. Nestes edifícios ele conseguiu implementar vários alojamentos, cada um com a sua própria cozinha, zona de entretenimento e conseguindo albergar grupos de até 6 pessoas.

Até ao momento, o senhor Luís regista os dados relativos ao seu negócio manualmente (com papel e caneta), mas com o aumento da afluência de pessoas ele decidiu que precisa de uma base de dados para registar e gerir melhor os alugueres dos seus estabelecimentos.



### **1.3.Motivação e Objetivos**

Uma grande motivação para o senhor Luís querer implementar uma base de dados é que desta forma todas as informações relativas à evolução do seu negócio ficarão devidamente registadas, uma vez que a base de dados garante a persistência dos dados, tornando assim mais fácil a sua passagem, um dia, para os seus descendentes.

Para além disso, este método permitirá uma melhor gestão das reservas dos diferentes espaços que o senhor Luís tem disponíveis.

Finalmente, já que o senhor Luís pensa um dia aumentar o seu negócio, os registos da base de dados permitir-lhe-ão avaliar qual será a forma mais rentável de o fazer.

Na sua base de dados ele pretende registar todos os clientes que passam pelo seu estabelecimento, informação quanto aos seus funcionários, registos de todos os quartos existentes, suas características e de todas as reservas.

### **1.4.Análise da Viabilidade do Processo**

Este projeto consiste na implementação de uma Base de Dados Relacional que permita analisar e relacionar os dados do Alojamento Local do senhor Luís com vista a melhor averiguar as necessidades de melhoramento dos serviços, uma vez que esta deverá possibilitar a manipulação e consulta dos dados de uma forma ágil e segura.

Este será um processo trabalhoso, mas que trará grandes frutos no futuro, uma vez que será substancialmente mais eficiente que o método atual, especialmente na atualização, consulta e tratamento dos dados, o que permitirá que o senhor Luís consiga alcançar os seus objetivos. Para além disso, será também um sistema mais fiável, visto que garantirá uma uniformização dos dados, sendo que qualquer interveniente que pretenda consultar ou alterar os dados o fará de uma forma mais segura.

### **1.5.Recursos e Equipa de Trabalho**

A equipa de trabalho reunida para a realização desta base de dados é constituída por 4 especialistas em Bases de Dados e o senhor Luís (especialista em hotelaria).

Quanto aos recursos usados, foram consultados sites de *booking* e informativos relativamente ao funcionamento de estabelecimentos de alojamento local.

## **1.6.Estrutura do Relatório**

Na Introdução dá-se a contextualização do projeto, seguido pela apresentação do caso de estudo bem como as motivações e objetivos.

Na segunda parte intitulada Levantamento e Análise de Requisitos, menciona-se a metodologia de levantamento e análise dos requisitos, seguida da apresentação dos requisitos de descrição, exploração e controlo, que serão analisados e validados.

No terceiro capítulo, são tratados os aspetos relativos à modelação das entidades que irão integrar o modelo conceptual. Estas serão identificadas e caracterizadas em termos de atributos, o mesmo acontecerá com os relacionamentos. Na próxima fase, é apresentado o diagrama ER que caracterizará o sistema que passará por um processo de validação, tendo em conta os requisitos apresentados no capítulo acima.

Ao longo do quarto capítulo, é feita a conversão do modelo conceptual para a sua versão lógica que será validado através de interrogações do utilizador.

Para concluir o desenvolvimento da base de dados é feita uma conversão do modelo lógico para uma versão física em *SQL*.

Para finalizar, na Conclusão e Trabalho Futuro, apresentam-se ilações resultantes da elaboração do trabalho e possíveis formas de aprimoramento do projeto realizado.

## **2...Levantamento e Análise de Requisitos**

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

O método adotado para o levantamento de requisitos foi a realização de várias entrevistas ao senhor Luís. Durante estas reuniões compilamos um conjunto de exigências que serão enunciadas nesta secção do relatório.

Para além disso, foi feita também alguma pesquisa no campo do turismo rural e do alojamento local, para uma melhor compreensão desta área e das necessidades relacionadas com a gestão e funcionamento de um estabelecimento deste cariz.

### **2.2.Organização dos requisitos levantados**

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

RD01 – Um edifício é identificado por um código, devendo ter também uma referência à sua morada, que deve conter informação quanto à rua, código-postal e localidade;

RD02 – Um alojamento é identificado por um código, devendo ter também uma referência ao seu número, edifício onde se encontra (identificado pelo seu código único), preço base, lotação máxima e número de quartos;

RD03 – Um funcionário é identificado por um código, devendo ter também uma referência ao seu nome, número de telemóvel, email, salário mensal e supervisor (identificado pelo seu código único);

RD04 – Um cliente é identificado por um código, devendo ter também uma referência ao seu nome, data de nascimento, email e número de telemóvel;

RD05 – Uma reserva é identificada por um código, devendo ter também uma referência ao funcionário responsável pela mesma, cliente que a efetuou (ambos identificados pelo respetivo código único) data de entrada e saída do alojamento, preço total da estadia, número de adultos e número de crianças.

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

RE01 – Listar os clientes que frequentaram o estabelecimento e respetivos contactos;

RE02 – Listar os edifícios e respetivos alojamentos, fornecendo informação quanto ao número de quartos e lotação máxima;

RE03 – Listar todas as reservas realizadas num certo quarto fornecendo informação quanto às datas de entrada e saída;

RE04 – Listar todos os funcionários e respetivo responsável;

- RE05 – Calcular o número de vezes que cada alojamento foi reservado;
- RE06 – Calcular o número de pedidos que cada funcionário processou;
- RE07 – Listar quantos quartos estão disponíveis até um certo preço;
- RE08 – Listar os clientes por ordem decrescente de dinheiro gasto;

### **2.2.3.Requisitos de Controlo**

- RC01 – Uma reserva só pode ser processada por um funcionário;
- RC02 – Uma reserva só pode ser feita por um cliente;
- RC03 – Uma reserva pode ser referente a vários alojamentos;
- RC04 – O funcionário pode criar novas fichas de cliente;
- RC05 – Na reserva são indicados tanto os dados do cliente como os do funcionário;
- RC06 – Alojamentos no mesmo edifício têm números diferentes;
- RC07 – Apenas existe uma única reserva para cada quarto num determinado intervalo de tempo.

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

O processo de análise e validação de requisitos ocorre no ato de desenho e posterior implementação de um sistema de gestão de bases de dados, tendo essas modelações uma relação direta com os critérios definidos nesta fase.

O processo de análise provém do processo de levantamento que possui três fases, os requisitos de descrição, os requisitos de exploração e os requisitos de controlo.

Com os requisitos de descrição, focamo-nos em analisar quais os elementos indispensáveis para que o alojamento do Senhor Luís funcione e na maneira como estes se deviam relacionar entre si, refletindo-se nos requisitos apresentados acima.

De seguida, deliberou-se quais informações seriam mais relevantes extrair, tendo em conta os elementos e relacionamentos apresentados na fase anterior.

Passada a fase dos requisitos de exploração entramos na etapa final da análise, os requisitos de controlo, onde estabelecemos as restrições de administração da base de dados, de forma a criar coerência entre todo o sistema.

Já no processo de validação verificou-se a consistência, precisão e contextualização dos requisitos expostos no processo de análise por forma a evitar falhas na modelação dos requisitos, o pouco realismo que tem como fator a não concordância do requisito com o problema e assim por diante. As técnicas utilizadas foram as usuais a reconsideração, testagem e análise dos modelos relativos às condições levantadas, qualificando-os quanto à redundância, consistência, compreensibilidade, rastreabilidade e conformidade com as normas impostas pelo cliente.

## 3...Modelo Conceptual

### 3.1.Apresentação da abordagem de modelação realizada

Na criação do modelo conceptual perdemos grande parte do tempo a estruturar e corrigir os pedidos do Sr. Luís, uma vez que a sua boa implementação evitou erros na parte lógica e física. Para a realização do mesmo utilizamos o software *brModelo* que permitiu uma modelação *Top Down* onde começamos por identificar as entidades e depois as relações com as devidas restrições e caracterizações dos atributos, tendo em especial atenção as relações N para N.

### 3.2.Identificação e caracterização das entidades

Entidade	Caracterização
Cliente	Termo que contém a informação sobre as pessoas que tenham feito alugueres no alojamento
Funcionario	Termo que contém a informação sobre os funcionários que trabalham no alojamento
Edifício	Termo que contém a informação de cada edifício do alojamento
Alojamento	Termo que contém a informação de cada acomodação do alojamento
Reserva	Termo que contém a informação de cada reserva efetuada no alojamento

Tabela 1 - Caracterização das Entidades

### 3.3.Identificação e caracterização dos relacionamentos

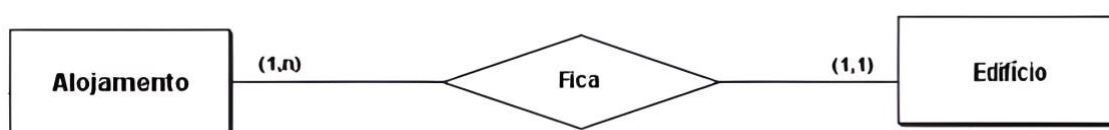


Figura 1 - Relação entre Alojamento e Edifício

Um Edifício pode ter 1 ou N Alojamentos e um Alojamento só pode estar num único Edifício. Logo temos uma cardinalidade de 1 para N.



Figura 2 - Relação entre Cliente e Reserva

Uma Reserva pode ser feita por 1 único Cliente, mas um Cliente efetuar 1 ou N Reservas. Logo temos uma cardinalidade de 1 para N.

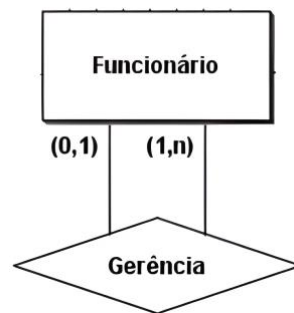


Figura 3 - Relação Funcionário Gerência

Uma Gerência pode ter 1 ou N Funcionário e um Funcionário pode ou não ter uma única Gerência. Logo temos uma cardinalidade de 0 para N.

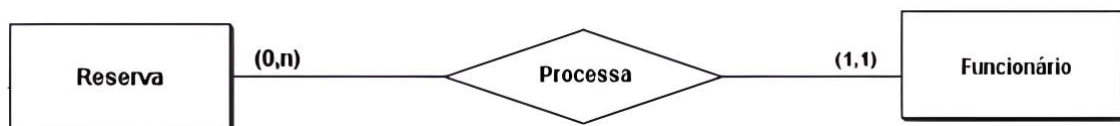


Figura 4 - Relação entre Reserva e Funcionário

Um Funcionário pode fazer 0 ou N Reservas, mas as Reservas só podem ser feitas por um único Funcionário. Logo temos uma cardinalidade de 0 para N.



Figure 5 - Relação entre Reserva e Alojamento

Um Alojamento pode fazer 0 ou N Reservas e as Reservas podem ser de 1 ou N Alojamentos. Logo temos uma cardinalidade de N para N.

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

Atributo	Tipo	Descrição
Id_Cliente	Chave	Número único que identifica o cliente
Nome	Simple	Nome do cliente
Data_Nascimento	Simple	Data de nascimento do cliente
E-mail	Simple	E-mail do cliente
Telemovel	Simple	Número de telemóvel do cliente

Tabela 2 - Atributos da entidade Cliente

Atributo	Tipo	Descrição
Id_Funcionario	Chave	Número único que identifica o funcionário
Nome	Simple	Nome do funcionário
Telemovel	Simple	Número de telemóvel do funcionário
E-mail	Simple	E-mail do funcionário
Salario	Simple	Salário do funcionário

Tabela 3 - Atributos da entidade Funcionário

Atributo	Tipo	Descrição
Id_Edificio	Chave	Número único que identifica o edifício
Rua\Morada	Composta	Rua referente à morada do edifício
Localidade\Morada	Composta	Localidade onde se encontra o edifício
Codigo_Postal\Morada	Composta	Código Postal do edifício

Tabela 4 - Atributos da entidade Edifício

Atributo	Tipo	Descrição
Id_Alojamento	Chave	Número único que identifica o alojamento
Numero	Simple	Número da porta do quarto
Numero_Quartos	Simple	Número de quartos que existem no alojamento
Preco_Base	Simple	Preço base do quarto

Lotacao	Simple	Lotação máxima do quarto
---------	--------	--------------------------

Tabela 5 - Atributos da entidade Alojamento

Atributo	Tipo	Descrição
Id_Reserva	Chave	Número único que identifica a reserva
Adultos	Simple	Número de adultos que vão frequentar o quarto
Crianças	Simple	Número de crianças que vão frequentar o quarto
DataInicio	Simple	Data e hora em que a reserva foi feita.
DataFim	Simple	Data e hora de quando acaba a reserva do quarto.
Preco	Simple	Número decimal referente ao valor pago pelo cliente no momento da reserva.

Tabela 6 - Atributos da entidade Reserva

### 3.5. Apresentação e explicação do diagrama ER

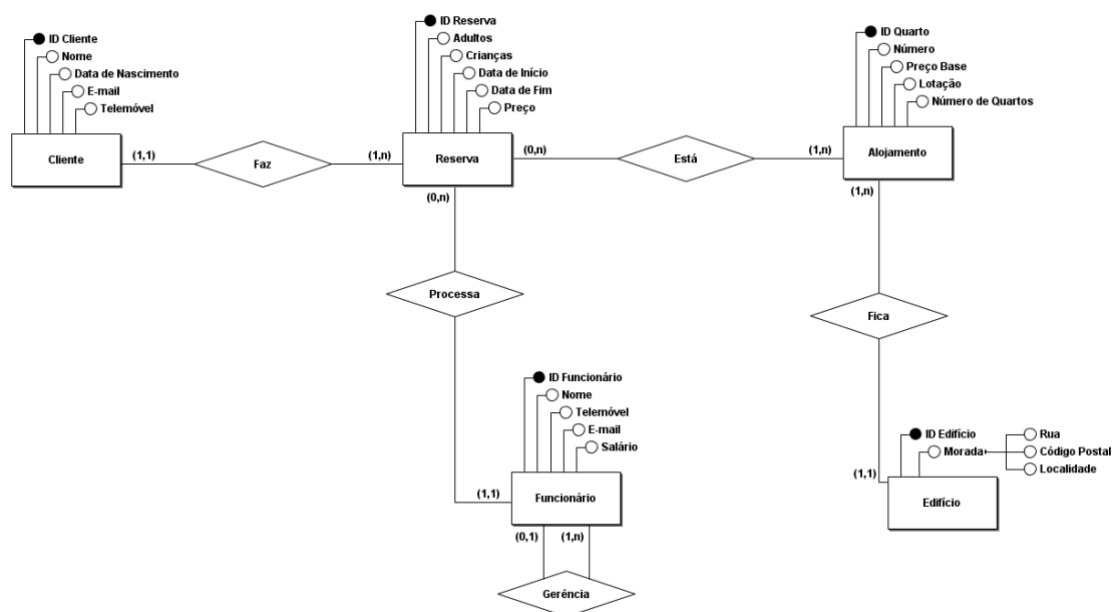


Figura 6 - Modelo Conceptual



### **3.6. Validação do Modelo Conceptual**

Após um longo período de apreciação por parte do Sr. Luís acabamos por chegar a um esquema em que todos os requisitos de descrição estavam a ser satisfeitos. Com a sua aprovação pudemos finalmente passar para a criação do modelo lógico.

## 4...Modelação Lógica

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

Para se dar conversão do modelo de dados conceptual para o modelo de dados lógico fizemos o agrupamento de cada entidade com os seus atributos numa tabela. Após este procedimento, prestou-se grande atenção às regras de derivação dos relacionamentos do modelo conceptual, concluindo-se assim a construção do modelo lógico.

A entidade Cliente dá origem à tabela:

Cliente (Id_Cliente, Nome, Data_Nascimento, E-mail, Telemovel)
Chave Primária: Id_Cliente

Tabela 7- Referente à entidade Cliente

A entidade Funcionario dá origem à tabela:

Funcionario (Id_Funcionario, Nome, Telemovel, E-mail, Salario, Gerente)	
Chave Primária: Id_Funcionario	Chave Estrangeira: Gerente

Tabela 8 - Referente à entidade Funcionario

A entidade Edificio dá origem à tabela:

Edificio (Id_Edificio, Rua/Morada, Localidade /Morada, Codigo_Postal /Morada)
Chave Primária: Id_Edificio

Tabela 9 - Referente à entidade Edifício

A entidade Alojamento dá origem à tabela:

Alojamento (Id_Alojamento, Numero, Edificio, Preco_Base, Lotacao, Numero_Quartos)	
Chave Primária: Id_Alojamento	Chave Estrangeira: Edificio

Tabela 10 - Referente à entidade Alojamento

A entidade Reserva dá origem à tabela: ola

Reserva(Id_Reserva, Funcionario, Cliente, DataInicio, DataFim, Adultos, Crianças)	
Chave Primária: Id_Reserva	Chaves Estrangeiras: Funcionario, Cliente

Tabela 11 - Referente à entidade Reserva

A relação de N para N entre Reserva e Alojamento origina a tabela:

Reserva_Alojamento(Reserva, Alojamento)
Chaves Estrangeiras: Reserva, Alojamento

Tablela 12 - Referente à relação Reserva\_Alojamento

## 4.2.Desenho do modelo lógico

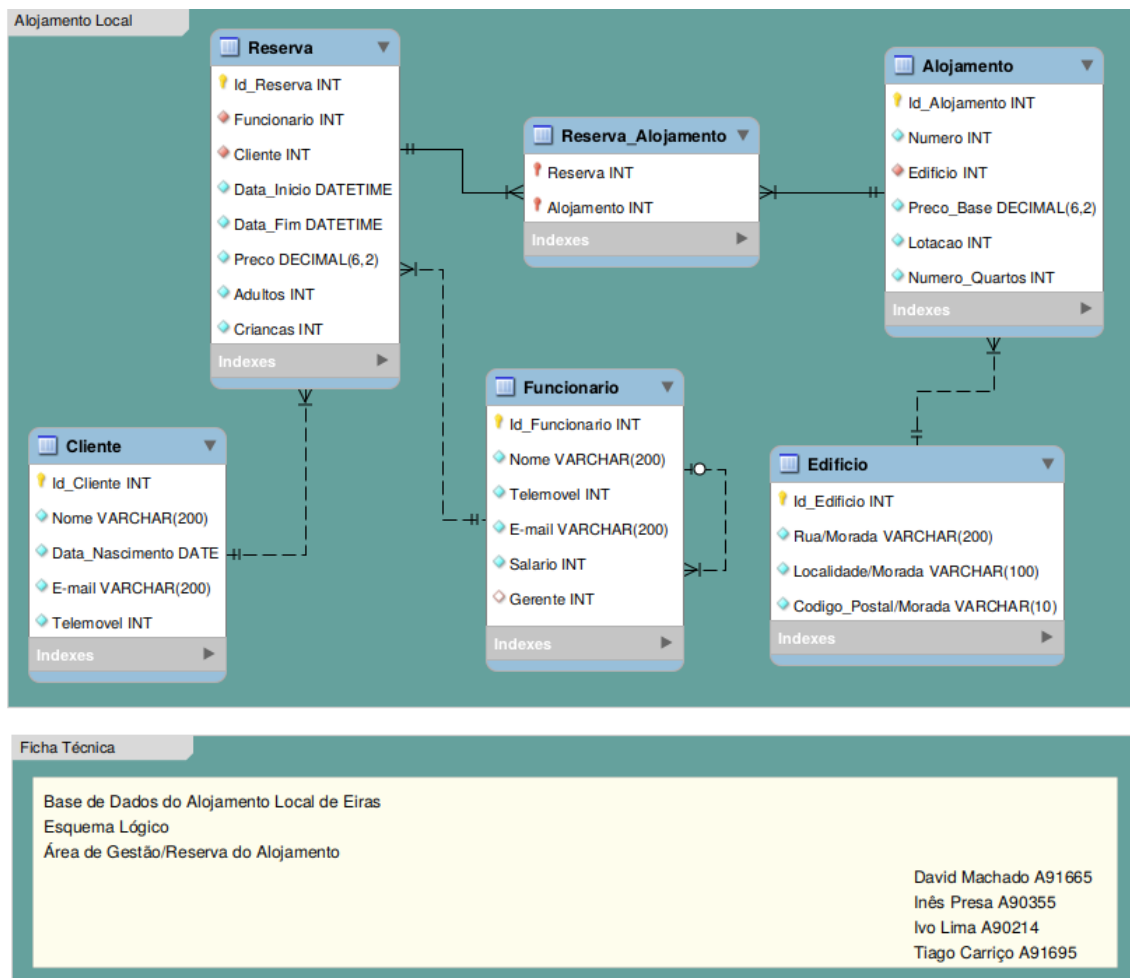
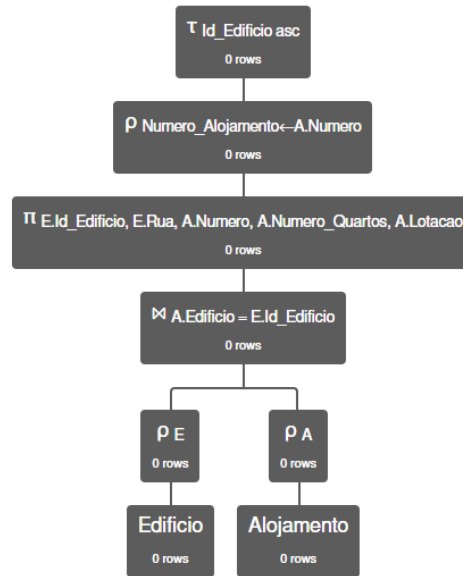


Figura 7 - Modelo Lógico

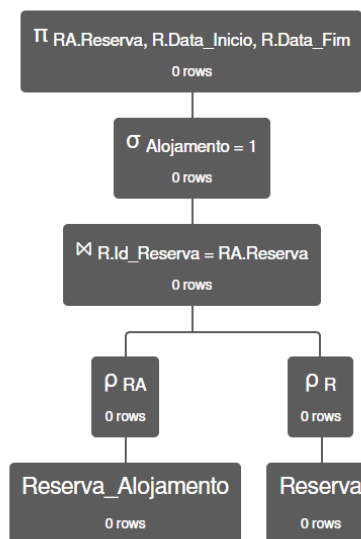
## 4.3.Validação do modelo com interrogações do utilizador

Como forma de validação do modelo lógico, testamos se seria possível satisfazer as interrogações definidas sob forma de requisitos de exploração. Para tal, procedeu-se à transformação de algumas interrogações para álgebra relacional de modo que a ferramenta *RelaX – Relational Algebra Calculator* as conseguisse interpretar. Mas este programa possui algumas limitações no que toca à disponibilização de operadores de lógica extendida, o que acabou por limitar a quantidade de interrogações que conseguimos modelar, visto que muitos operadores disponibilizados pelo *MySQL* não tinham equivalente lógico no *RelaX*.



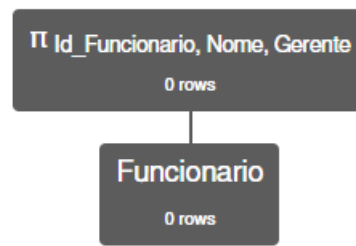
$\tau_{Id\_Edificio\ asc} \rho_{Numero\_Alojamento \leq A.Numero} \pi_{E.Id\_Edificio, E.Rua, A.Numero, A.Numero\_Quartos, A.Lotacao} ( \rho_E Edificio \bowtie_{A.Edificio = E.Id\_Edificio} \rho_A Alojamento )$

Figura 8 - Árvore representativa da RE02



$\pi_{RA.Reserva, R.Data\_Inicio, R.Data\_Fim} \sigma_{Alojamento = 1} ( \rho_{RA} Reserva\_Alojamento \bowtie_{R.Id\_Reserva = RA.Reserva} \rho_R Reserva )$

Figura 9 - Árvore representativa da RE03



$\pi$  Id\_Funcionario, Nome, Gerente Funcionario

Figura 10 - Árvore representativa do RE04

## 5...Implementação Física

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

Uma vez que o sistema de gestão de bases de dados utilizado durante as aulas foi o MySQL, decidimos que também deveríamos adotá-lo para a implementação deste projeto, já que estávamos mais familiarizados com o seu funcionamento e consideramos que é uma ferramenta bastante intuitiva.

O *MySQL* é um Sistema de Gestão de Bases de Dados *open-source* que assenta sobre um paradigma relacional. Sendo um dos sistemas mais utilizados em todo o mundo, devido à sua facilidade de utilização e elevado desempenho.

O *engine* utilizado será o *InnoDB*.

- Edifício

```
-- -----  
-- Table AlojamentoLocal.Edificio  
-- -----  
CREATE TABLE IF NOT EXISTS AlojamentoLocal.Edificio (  
  Id_Edificio INT NOT NULL AUTO_INCREMENT,  
  `Rua/Morada` VARCHAR(200) NOT NULL,  
  `Localidade/Morada` VARCHAR(100) NOT NULL,  
  `Codigo_Postal/Morada` VARCHAR(10) NOT NULL,  
  PRIMARY KEY (Id_Edificio))  
ENGINE = InnoDB;
```

Figura 11 - Código SQL para criar tabela Edificio

- Alojamento

```
-- -----  
-- Table AlojamentoLocal.Alojamento  
-- -----  
CREATE TABLE IF NOT EXISTS AlojamentoLocal.Alojamento (  
  Id_Alojamento INT NOT NULL AUTO_INCREMENT,  
  Numero INT NOT NULL,  
  Edificio INT NOT NULL,  
  Preco_Base DECIMAL(6,2) NOT NULL,  
  Lotacao INT NOT NULL,  
  Numero_Quartos INT NOT NULL,  
  INDEX fk_Quartos_Edificio1_idx (Edificio ASC),  
  PRIMARY KEY (Id_Alojamento),  
  CONSTRAINT fk_Quartos_Edificio1  
    FOREIGN KEY (Edificio)  
    REFERENCES AlojamentoLocal.Edificio (Id_Edificio)  
  )  
ENGINE = InnoDB;
```

Figura 12 - Código SQL para criar tabela Alojamento

- **Funcionário**

```

-----
-- Table AlojamentoLocal.Funcionario
-----
CREATE TABLE IF NOT EXISTS AlojamentoLocal.Funcionario (
  Id_Funcionario INT NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(200) NOT NULL,
  Telemovel INT NOT NULL,
  `E-mail` VARCHAR(200) NOT NULL,
  Salario INT NOT NULL,
  Gerente INT,
  PRIMARY KEY (Id_Funcionario),
  INDEX funcionario_gerente_idx (Gerente ASC),
  CONSTRAINT funcionario_gerente
    FOREIGN KEY (Gerente)
    REFERENCES AlojamentoLocal.Funcionario (Id_Funcionario)
)
ENGINE = InnoDB;

```

Figura 13 - Código SQL para criar a tabela Funcionario

- **Cliente**

```

-----
-- Table AlojamentoLocal.Cliente
-----
CREATE TABLE IF NOT EXISTS AlojamentoLocal.Cliente (
  Id_Cliente INT NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(200) NOT NULL,
  Data_Nascimento DATE NOT NULL,
  `E-mail` VARCHAR(200) NOT NULL,
  Telemovel INT NOT NULL,
  PRIMARY KEY (Id_Cliente))
ENGINE = InnoDB;

```

Figura 14 - Código SQL para criar tabela Cliente

- **Reserva**

```

-----
-- Table AlojamentoLocal.Reserva
-----
CREATE TABLE IF NOT EXISTS AlojamentoLocal.Reserva (
  Id_Reserva INT NOT NULL AUTO_INCREMENT,
  Funcionario INT NOT NULL,
  Cliente INT NOT NULL,
  Data_Inicio DATETIME NOT NULL,
  Data_Fim DATETIME NOT NULL,
  Preco DECIMAL(6,2) NOT NULL,
  Adultos INT NOT NULL,
  Crianças INT NOT NULL,
  PRIMARY KEY (Id_Reserva),
  INDEX fk_Reservas_Funcionarios1_idx (Funcionario ASC),
  INDEX fk_Reserva_Cliente1_idx (Cliente ASC),
  CONSTRAINT fk_Reservas_Funcionarios1
    FOREIGN KEY (Funcionario)
    REFERENCES AlojamentoLocal.Funcionario (Id_Funcionario),
  CONSTRAINT fk_Reserva_Cliente1
    FOREIGN KEY (Cliente)
    REFERENCES AlojamentoLocal.Cliente (Id_Cliente)
)
ENGINE = InnoDB;

```

Figura 15 - Código SQL para criar a tabela Reserva

- **Reserva\_Alojamento**

```

-----
-- Table AlojamentoLocal.Reserva_Alojamento
-----
CREATE TABLE IF NOT EXISTS AlojamentoLocal.Reserva_Alojamento (
  Reserva INT NOT NULL,
  Alojamento INT NOT NULL,
  PRIMARY KEY (Reserva, Alojamento),
  INDEX fk_Reserva_Quarto_Quarto1_idx (Alojamento ASC),
  CONSTRAINT fk_Reserva_Quarto_Reserva1
    FOREIGN KEY (Reserva)
      REFERENCES AlojamentoLocal.Reserva (Id_Reserva),
  CONSTRAINT fk_Reserva_Quarto_Quarto1
    FOREIGN KEY (Alojamento)
      REFERENCES AlojamentoLocal.Alojamento (Id_Alojamento)
)
ENGINE = InnoDB;

```

Figura 16 - Código SQL para criar a tabela Reserva\_Alojamento

## 5.2.Tradução das interrogações do utilizador para SQL

Nesta secção será apresentado o código SQL que permite obter resposta a cada requisito de exploração e exemplos da resposta com o povoamento atual.

- **RE01 - Listar os clientes que frequentaram o estabelecimento e respetivos contactos**

```

SELECT Id_Cliente, Nome, `E-mail`, Telemovel FROM Cliente
WHERE Id_Cliente IN
(SELECT Cliente FROM Reserva);

```

Figura 17 - Código SQL para RE01

Id_Cliente	Nome	E-mail	Telemovel	
1	Luísa Matos	pvicente@hotmail.com	927594361	
2	Camila-Carlota Pacheco	yarasoares@clix.pt	206650137	
3	Francisco da Oliveira	omelo@sapo.pt	919864375	
4	William Lourenço	salomecastro@sapo.pt	921059978	
5	António Nunes	luciacorreia@gmail.com	932470127	
6	Joaquim Lourenço	wbarbosa@gmail.com	913505744	
7	Henrique Lima	enzo76@hotmail.com	969660401	
8	Irina Valente	araujopedro@sapo.pt	923680584	
9	William Ramos	freitasirina@sapo.pt	964586291	
10	Amélia Torres-Branco	fgaspar@clix.pt	930950625	

Figura 18 – Resultado da Query relativa a RE01



- **RE02 – Listar os edifícios e respetivos alojamentos, fornecendo informação quanto ao número de quartos e lotação máxima**

```
SELECT E.Id_Edificio, E.`Rua/Morada`, A.Numero AS Numero_Alojamento, A.Numero Quartos, A.Lotacao FROM Edificio AS E
INNER JOIN Alojamento AS A
ON A.Edificio = E.Id_Edificio
ORDER BY Id_Edificio;
```

Figura 19 - Código SQL para RE02

Id_Edificio	Rua/Morada	Numero_Alojamento	Numero Quartos	Lotacao	
1	Largo de Soares, 488	1	3	6	
1	Largo de Soares, 488	2	2	4	
1	Largo de Soares, 488	3	2	4	
2	Largo de Loureiro, 6	1	1	2	
2	Largo de Loureiro, 6	2	1	2	
2	Largo de Loureiro, 6	3	2	4	
2	Largo de Loureiro, 6	4	2	4	

Figura 20 - Resultado da Query relativa a RE02

- **RE03 – Listar todas as reservas realizadas num certo quarto fornecendo informação quanto às datas de entrada e saída**

```
SELECT RA.Reserva, R.Data_Inicio, R.Data_Fim FROM Reserva_Alojamento AS RA
INNER JOIN Reserva AS R
ON R.Id_Reserva = RA.Reserva
WHERE Alojamento = 1;
```

Figura 21 - Código SQL para RE03

Reserva	Data_Inicio	Data_Fim	
2	2020-01-09 00:00:00	2020-01-12 00:00:00	
4	2020-01-15 00:00:00	2020-01-18 00:00:00	
6	2020-01-21 00:00:00	2020-01-24 00:00:00	
7	2020-01-30 00:00:00	2020-02-02 00:00:00	
9	2020-02-06 00:00:00	2020-02-09 00:00:00	
10	2020-02-07 00:00:00	2020-02-10 00:00:00	
11	2020-02-12 00:00:00	2020-02-15 00:00:00	
23	2020-03-30 00:00:00	2020-04-02 00:00:00	
25	2020-04-12 00:00:00	2020-04-15 00:00:00	
26	2020-04-11 00:00:00	2020-04-14 00:00:00	

Figura 22 - Resultado da Query relativa a RE03

- **RE04 – Listar todos os funcionários e respetivo responsável**

```
SELECT Id_Funcionario, Nome, Gerente FROM Funcionario;
```

Figura 23 - Código SQL para RE04

Reserva	Data_Inicio	Data_Fim	
2	2020-01-09 00:00:00	2020-01-12 00:00:00	
4	2020-01-15 00:00:00	2020-01-18 00:00:00	
6	2020-01-21 00:00:00	2020-01-24 00:00:00	
7	2020-01-30 00:00:00	2020-02-02 00:00:00	
9	2020-02-06 00:00:00	2020-02-09 00:00:00	
10	2020-02-07 00:00:00	2020-02-10 00:00:00	
11	2020-02-12 00:00:00	2020-02-15 00:00:00	
23	2020-03-30 00:00:00	2020-04-02 00:00:00	
25	2020-04-12 00:00:00	2020-04-15 00:00:00	
26	2020-04-11 00:00:00	2020-04-14 00:00:00	

Figura 24 - Resultado da Query relativa a RE04

- **RE05 – Calcular o número de vezes que cada alojamento foi reservado**

```
SELECT P.Id_Alojamento, P.Numero, P.Edificio, COUNT(P.Reserva) AS Total_Reservas FROM
(SELECT A.Id_Alojamento, A.Numero, A.Edificio, RA.Reserva FROM Reserva_Alojamento AS RA
INNER JOIN Alojamento AS A
ON A.Id_Alojamento = RA.Alojamento) AS P
GROUP BY Id_Alojamento;
```

Figura 25 - Código SQL para RE05

Reserva	Data_Inicio	Data_Fim	
2	2020-01-09 00:00:00	2020-01-12 00:00:00	
4	2020-01-15 00:00:00	2020-01-18 00:00:00	
6	2020-01-21 00:00:00	2020-01-24 00:00:00	
7	2020-01-30 00:00:00	2020-02-02 00:00:00	
9	2020-02-06 00:00:00	2020-02-09 00:00:00	
10	2020-02-07 00:00:00	2020-02-10 00:00:00	
11	2020-02-12 00:00:00	2020-02-15 00:00:00	
23	2020-03-30 00:00:00	2020-04-02 00:00:00	
25	2020-04-12 00:00:00	2020-04-15 00:00:00	
26	2020-04-11 00:00:00	2020-04-14 00:00:00	
29	2020-04-24 00:00:00	2020-04-27 00:00:00	

Figura 26 - Resultado da Query relativa a RE05

- **RE06 – Calcular o número de pedidos que cada funcionário processou**

```
SELECT R.Funcionario, F.Nome, COUNT(R.Id_Reserva) AS Total_Reservas FROM Reserva AS R
INNER JOIN Funcionario AS F
ON F.Id_Funcionario = R.Funcionario
GROUP BY Funcionario;
```

Figura 27 - Código SQL para RE06

Funcionario	Nome	Total_Reservas	
1	Luís Presa	53	
2	Samuel Gaspar	37	
3	Renata Alves	44	
4	Patrícia Guerreiro	56	

Figura 28 - Resultado da Query relativa a RE06

- **RE07 – Listar quantos quartos estão disponíveis até um certo preço**

```
SELECT COUNT(Id_Alojamento) FROM Alojamento WHERE Preco_Base <= 100;
```

Figura 29 - Código SQL para RE07

COUNT(Id_Alojamento)	
2	

Figura 30 - Resultado da Query relativa a RE07

- **RE08 – Listar os clientes por ordem decrescente de dinheiro gasto**

```
SELECT R.Cliente, C.Nome, SUM(R.Preco) AS Total_Gasto FROM Reserva AS R
INNER JOIN Cliente AS C
ON C.Id_Cliente = R.Cliente
GROUP BY R.Cliente
ORDER BY Total_Gasto DESC;
```

Figura 31 - Código SQL para RE08

Cliente	Nome	Total_Gasto	
95	Brian Cardoso-Macedo	619.00	
99	Catarina Neto	611.00	
5	António Nunes	610.00	
66	Helena Amorim	604.00	
15	Andreia Ribeiro-Valente	591.00	
58	Kyara Esteves	588.00	
54	Patrícia Carvalho	588.00	
3	Francisco da Oliveira	583.00	
28	Helena Martins	576.00	
32	Rúben Cardoso	574.00	

Figura 32 - Resultado da Query relativa a RE08

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

- Vista sobre o número de vezes que cada alojamento foi reservado

```
-- V01 - Calcular o número de vezes que cada alojamento foi reservado
CREATE VIEW V01 AS
SELECT P.Id_Alojamento, P.Numero, P.Edificio, COUNT(P.Reserva) AS Total_Reservas FROM
(SELECT A.Id_Alojamento, A.Numero, A.Edificio, RA.Reserva FROM Reserva_Alojamento AS RA
INNER JOIN Alojamento AS A
ON A.Id_Alojamento = RA.Alojamento) AS P
GROUP BY Id_Alojamento;
```

Figura 33 - Código SQL para vista

## 5.4. Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)

De forma a estimar o espaço gasto por uma entrada de cada entidade na nossa Base de Dados foi necessário calcular quantos *bytes* são gastos por cada tipo de dados. Para tal, foram consultadas as secções 11.1, 11.2, 11.3 e 11.7 do *MySQL 8.0 Reference Manual*.

Dessa consulta, estabeleceram-se as seguintes métricas de *bytes* por tipo de dado:

Tipo	Tamanho
Int	4 bytes
Float	4 bytes
Datetime	5 bytes
Date	3 bytes
Decimal (X, Y)	8 bytes*
Varchar (N)	N+1 ou 2xN+1 ou 2xN+2**

Tabela 13 - Descrição dos tamanhos consoante o tipo

\*Quando  $X - Y = 9$  e  $Y < 9$ , para valores superiores devemos consultar a tabela disponível na secção 11.7 do manual.

\*\*N+1 bytes caso a *string* esteja codificada em *ASCII*,  $2*N + 1$  caso a codificação seja mais complexa, mas o valor de  $2*N$  seja menor que 255 e  $2*N + 2$  caso  $2*N$  maior que 255;

Tendo em conta os valores definidos, explicitam-se os valores máximos que uma entrada de cada entidade e relação com atributos ocupará.

Atributo	Tipo	Tamanho
Id_Cliente	INT	4 bytes

Nome	VARCHAR (200)	402 bytes
Data_Nascimento	DATE	3 bytes
E-mail	VARCHAR (200)	402 bytes
Telemovel	INT	4 bytes

Tabela 14 - Tamanho (em bytes) dos atributos de Cliente

Uma entrada na tabela Cliente ocupa um total de 815 *bytes*.

Atributo	Tipo	Tamanho
Id_Funcionario	INT	4 bytes
Nome	VARCHAR (200)	402 bytes
Telemovel	INT	4 bytes
E-mail	VARCHAR (200)	402 bytes
Salario	INT	4 bytes
Gerente	INT	4 bytes

Tabela 15 - Tamanho (em *bytes*) dos atributos de Funcionario

Uma entrada na tabela Funcionario ocupa um total de 820 *bytes*.

Atributo	Tipo	Tamanho
Id_Edificio	INT	4 bytes
Rua/Morada	VARCHAR (200)	402 bytes
Localidade/Morada	VARCHAR (100)	201 bytes
Codigo_Postal/Morada	VARCHAR (10)	21 bytes

Tabela 16 - Tamanho (em *bytes*) dos atributos de Edificio

Uma entrada na tabela Edificio ocupa um total de 628 *bytes*.

Atributo	Tipo	Tamanho
Id_Alojamento	INT	4 bytes
Numero	INT	4 bytes
Edificio	INT	4 bytes
Preco_Base	DECIMAL (6,2)	8 bytes
Lotacao	INT	4 bytes
Numero_Quartos	INT	4 bytes

Tabela 17 - Tamanho (em *bytes*) dos atributos de Alojamento

Uma entrada na tabela Alojamento ocupa um total de 28 *bytes*.

Atributo	Tipo	Tamanho
Id_Reserva	INT	4 bytes
Funcionario	INT	4 bytes
Cliente	INT	4 bytes
DataInicio	DATETIME	5 bytes
DataFim	DATETIME	5 bytes
Preco	DECIMAL (6,2)	8 bytes
Adultos	INT	4 bytes
Crianças	INT	4 bytes

Tabela 18 - Tamanho (em *bytes*) dos atributos de Reserva

Uma entrada na tabela Reserva ocupa um total de 38 *bytes*.

Atributo	Tipo	Tamanho
Reserva	INT	4 bytes
Alojamento	INT	4 bytes

Tabela 19 - Tamanho (em *bytes*) dos atributos de Reserva\_Alojamento

Uma entrada na tabela Reserva\_Alojamento ocupa um total de 8 *bytes*.

Usando estes valores calculados, pode-se obter uma estimativa de espaço ocupado pela base de dados sem povoamento de 2337 *bytes*.

Tendo em conta o contexto do nosso projeto que é um alojamento local, é de esperar que as entidades que tenham uma quantidade considerável de inserções na base de dados anualmente sejam o Cliente e a Reserva.

Assumindo um crescimento anual de 5% no número de clientes e tendo-se atualmente 100 clientes no sistema, este configura-se como um crescimento de  $(100 * 0.05) * 815 \approx 4\,075$  *bytes*, ou seja, 4.1 KB.

Tendo em conta os novos clientes e os eventuais clientes que possam voltar, pode-se assumir uma taxa de crescimento anual de 8% no número de reservas e tendo-se atualmente 190 reservas no sistema, este configura-se como um crescimento de  $(190 * 0.08) * 38 \approx 578$  *bytes*.

O crescimento relativo às restantes entidades revela-se bastante imprevisível, no entanto é possível assumir que a sua variação será reduzida o suficiente de modo que não terá um impacto significativo no crescimento da base de dados.

De acordo com a informação acima mencionada, o crescimento total anual traduz-se no valor de  $4\,075 + 578 = 4\,653$  *bytes*, ou seja, 4.7 KB por ano.

## 5.5.Plano de segurança e recuperação de dados

Quando o assunto é a gestão de dados a precaução é obrigatória pois a perda de qualquer tipo de dados pode colocar em risco todo o negócio. Como tal, foram sugeridas uma série de ações e diretrizes que vão assegurar o funcionamento da Base de Dados por forma a minimizar os danos causados por ataques cibernéticos, erro humano, entre outros.

Mas aquelas que mais se destacam são:

- Um documento onde é indicada a tolerância máxima relativamente à destruição de dados e quais os meios mais afetados por esse incidente.
- Um documento que contenha uma lista de prioridades relativamente à informação que estava na Base de Dados.
- Uma medida de *Backups* Periódicos para uma *Cloud* fornecida por terceiros ou um Sistema Físico com um grande aglomerado de Discos Rígidos.

## 6...Conclusões e Trabalho Futuro

Durante a realização deste trabalho constatamos que a parte fundamental para que uma base de dados seja implementada com sucesso é a de recolha e organização dos requisitos, pois sem isso é difícil construir uma base de dados coerente, robusta e segura. Para além de que, se esta fase não for realizada meteticulosamente, posteriormente surgirão desafios e questões que poderiam ter sido evitadas.

Dito isto, consideramos que, apesar de termos dado a devida atenção a estes factos, existem ainda alguns melhoramentos que podem ser aplicados ao projeto final, nomeadamente uma melhor caracterização das várias entidades, que teria como consequência a criação de mais atributos. Ao rever a estrutura da base de dados chegamos também à conclusão que seria mais correto ter um atributo “preço” na relação Reserva\_Alojamento que permitisse o cálculo do Preço da reserva através da soma desses campos. Para além disso, seria também importante garantir uma maior uniformidade do registo de certos dados, como é o exemplo do atributo localidade que, para evitar confusões devido a abreviamentos de nomes deveria ter a sua própria tabela onde se encontravam listadas todas as localidades com um identificador para que seja usado na tabela Edificio como chave estrangeira.

Não obstante, concluímos que os objetivos deste trabalho foram cumpridos, sendo que desenvolvemos uma base de dados apta a sustentar o estabelecimento de Alojamento Local do senhor Luís e que corresponde às suas necessidades.



## 5 Referências

Turismo de Portugal -

[http://business.turismodeportugal.pt/pt/Planear\\_Iniciar/Como\\_comecar/Alojamento\\_Local/Paginas/default.aspx](http://business.turismodeportugal.pt/pt/Planear_Iniciar/Como_comecar/Alojamento_Local/Paginas/default.aspx)

MySQL 8.0 Reference Manual - <https://dev.mysql.com/doc/refman/8.0/en/>

RelaX - <https://dbis-uibk.github.io/relax/landing>

## 6 Lista de Siglas e Acrónimos

<b>BD</b>	Base de Dados
<b>SQL</b>	<i>Structured Query Language</i>
<b>ER</b>	Entidade Relação
<b>Sr</b>	Senhor

## **7 Anexos**

## I. Anexo 1 - *Script* de Criação da Base de Dados

```
-- -----,
-- Schema AlojamentoLocal
-- -----

CREATE SCHEMA IF NOT EXISTS AlojamentoLocal DEFAULT CHARACTER SET utf8 ;
USE AlojamentoLocal ;

-- -----
-- Table AlojamentoLocal.Edificio
-- -----

CREATE TABLE IF NOT EXISTS AlojamentoLocal.Edificio (
  Id_Edificio INT NOT NULL AUTO_INCREMENT,
  `Rua/Morada` VARCHAR(200) NOT NULL,
  `Localidade/Morada` VARCHAR(100) NOT NULL,
  `Codigo_Postal/Morada` VARCHAR(10) NOT NULL,
  PRIMARY KEY (Id_Edificio))
ENGINE = InnoDB;

-- -----
-- Table AlojamentoLocal.Alojamento
-- -----

CREATE TABLE IF NOT EXISTS AlojamentoLocal.Alojamento (
  Id_Alojamento INT NOT NULL AUTO_INCREMENT,
  Numero INT NOT NULL,
  Edificio INT NOT NULL,
  Preco_Base DECIMAL(6,2) NOT NULL,
  Lotacao INT NOT NULL,
  Numero_Quartos INT NOT NULL,
  INDEX fk_Quartos_Edificio1_idx (Edificio ASC),
  PRIMARY KEY (Id_Alojamento),
  CONSTRAINT fk_Quartos_Edificio1
  FOREIGN KEY (Edificio)
  REFERENCES AlojamentoLocal.Edificio (Id_Edificio)
)
ENGINE = InnoDB;
```

```
-- Table AlojamentoLocal.Funcionario
-----

CREATE TABLE IF NOT EXISTS AlojamentoLocal.Funcionario (
  Id_Funcionario INT NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(200) NOT NULL,
  Telemovel INT NOT NULL,
  `E-mail` VARCHAR(200) NOT NULL,
  Salario INT NOT NULL,
  Gerente INT,
  PRIMARY KEY (Id_Funcionario),
  INDEX funcionario_gerente_idx (Gerente ASC),
  CONSTRAINT funcionario_gerente
  FOREIGN KEY (Gerente)
  REFERENCES AlojamentoLocal.Funcionario (Id_Funcionario)
)
ENGINE = InnoDB;
```

```
-- Table AlojamentoLocal.Cliente
-----

CREATE TABLE IF NOT EXISTS AlojamentoLocal.Cliente (
  Id_Cliente INT NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(200) NOT NULL,
  Data_Nascimento DATE NOT NULL,
  `E-mail` VARCHAR(200) NOT NULL,
  Telemovel INT NOT NULL,
  PRIMARY KEY (Id_Cliente))
ENGINE = InnoDB;
```

```
-- Table AlojamentoLocal.Reserva
-----

CREATE TABLE IF NOT EXISTS AlojamentoLocal.Reserva (
  Id_Reserva INT NOT NULL AUTO_INCREMENT,
  Funcionario INT NOT NULL,
  Cliente INT NOT NULL,
```

```

Data_Inicio DATETIME NOT NULL,
Data_Fim DATETIME NOT NULL,
Preco DECIMAL(6,2) NOT NULL,
Adultos INT NOT NULL,
Crianças INT NOT NULL,
PRIMARY KEY (Id_Reserva),
INDEX fk_Reservas_Funcionarios1_idx (Funcionario ASC),
INDEX fk_Reserva_Cliente1_idx (Cliente ASC),
CONSTRAINT fk_Reservas_Funcionarios1
FOREIGN KEY (Funcionario)
REFERENCES AlojamentoLocal.Funcionario (Id_Funcionario),
CONSTRAINT fk_Reserva_Cliente1
FOREIGN KEY (Cliente)
REFERENCES AlojamentoLocal.Cliente (Id_Cliente)
)
ENGINE = InnoDB;

-----
-- Table AlojamentoLocal.Reserva_Alojamento
-----

CREATE TABLE IF NOT EXISTS AlojamentoLocal.Reserva_Alojamento (
Reserva INT NOT NULL,
Alojamento INT NOT NULL,
PRIMARY KEY (Reserva, Alojamento),
INDEX fk_Reserva_Quarto_Quarto1_idx (Alojamento ASC),
CONSTRAINT fk_Reserva_Quarto_Reserva1
FOREIGN KEY (Reserva)
REFERENCES AlojamentoLocal.Reserva (Id_Reserva),
CONSTRAINT fk_Reserva_Quarto_Quarto1
FOREIGN KEY (Alojamento)
REFERENCES AlojamentoLocal.Alojamento (Id_Alojamento)
)
ENGINE = InnoDB;

```

## II. Anexo 2 - Script de Povoamento

```
INSERT INTO Cliente (`Nome`, `Data_Nascimento`, `E-mail`, `Telemovel`) VALUES
('Luísa Matos', '1986-02-18', 'pvicente@hotmail.com', '927594361'),
('Camila-Carlota Pacheco', '1961-07-13', 'yarasoaes@clix.pt', '206650137'),
('Francisco da Oliveira', '1971-10-04', 'omelo@sapo.pt', '919864375'),
('William Lourenço', '1971-06-29', 'salomecastro@sapo.pt', '921059978'),
('António Nunes', '1972-12-31', 'luciacorreia@gmail.com', '932470127'),
('Joaquim Lourenço', '1995-11-27', 'wbarbosa@gmail.com', '913505744'),
('Henrique Lima', '1970-08-12', 'enzo76@hotmail.com', '969660401'),
('Irina Valente', '1997-12-29', 'araujopedro@sapo.pt', '923680584'),
('William Ramos', '1963-07-21', 'freitasirina@sapo.pt', '964586291'),
('Amélia Torres-Branco', '1989-04-10', 'fgaspar@clix.pt', '930950625'),
('Marco Gaspar', '1967-10-26', 'paulobatista@hotmail.com', '963309395'),
('António Moreira', '1972-02-10', 'nlourenco@gmail.com', '202244711'),
('Valentim Torres', '1991-02-07', 'valenteleonardo@gmail.com', '925799606'),
('Eduarda Almeida', '1998-08-19', 'mhenriques@gmail.com', '921981027'),
('Andreia Ribeiro-Valente', '1996-12-09', 'andreiaribeiro@sapo.pt', '929033797'),
('Guilherme Soares-Barros', '1990-05-26', 'alvaro46@gmail.com', '925918535'),
('Alice Sousa', '1988-11-08', 'brian19@clix.pt', '915589070'),
('Salvador Barros', '1960-12-02', 'yassuncao@hotmail.com', '228747271'),
('Ângelo Cunha', '1961-09-26', 'henriquesyara@gmail.com', '916981337'),
('Tiago Lima', '1996-02-01', 'jbarros@sapo.pt', '967470428'),
('Manuel Abreu', '1973-09-02', 'rribeiro@clix.pt', '912297254'),
('Xavier Figueiredo', '1996-02-29', 'lmaia@clix.pt', '910451671'),
('Manuel Alves', '1992-04-09', 'ubaptista@gmail.com', '934232925'),
('Erika Anjos', '1983-07-03', 'melissa61@sapo.pt', '935538573'),
('Júlia Figueiredo', '1973-07-27', 'larissa85@clix.pt', '910031563'),
('Lourenço Cruz', '1986-06-24', 'camposismael@gmail.com', '930404583'),
('Erica Leal', '1992-01-20', 'madalenaleal@clix.pt', '923840187'),
('Helena Martins', '1970-05-27', 'ecoelho@hotmail.com', '916922318'),
('Lia Costa', '1990-05-28', 'alvaromoura@hotmail.com', '968588376'),
('Núria Almeida', '1979-05-03', 'franciscopacheco@clix.pt', '913848406'),
('Laura Lopes', '1995-05-07', 'alexandrarodrigues@clix.pt', '933801398'),
('Rúben Cardoso', '1984-05-12', 'nmachado@gmail.com', '233777589'),
('Alice Macedo-Pinho', '1975-10-14', 'bcruz@sapo.pt', '918484723'),
('Miguel Costa', '1962-03-13', 'ritavicente@hotmail.com', '204465859'),
('Soraia Campos', '1979-06-15', 'iribeiro@hotmail.com', '914316145'),
```

('Daniela do Garcia', '1997-12-04', 'pnogueira@gmail.com', '929343907'),  
('Carlota de Santos', '1994-04-24', 'kloureiro@gmail.com', '963077833'),  
('Gil Fernandes', '1991-02-25', 'marco00@gmail.com', '913659217'),  
('Clara Oliveira', '1972-02-02', 'patricia76@sapo.pt', '960938558'),  
('Tiago Cunha-Leite', '1985-08-06', 'alexandre04@clix.pt', '964039712'),  
('Jaime Nunes', '1993-04-18', 'csousa@clix.pt', '933115240'),  
('Wilson Tavares', '1961-08-01', 'anadomingues@sapo.pt', '211861840'),  
('Flor-Marta Amaral', '1969-07-29', 'castrowilliam@hotmail.com', '910188440'),  
('Pilar Monteiro', '1984-08-12', 'martinsruben@clix.pt', '931861456'),  
('Ana Fonseca', '1966-11-24', 'samuelalves@gmail.com', '270125735'),  
('Sérgio Andrade', '1987-01-02', 'carvalhonicole@gmail.com', '918412698'),  
('Clara Matos', '1994-02-18', 'qfonseca@sapo.pt', '910194454'),  
('Isabel do Monteiro', '1968-08-01', 'hbarbosa@gmail.com', '913274920'),  
('Vítor Simões', '2000-07-06', 'eva64@sapo.pt', '922853622'),  
('Renato Rocha', '1983-11-14', 'carolinaalves@clix.pt', '928504487'),  
('Sebastião Rodrigues', '1969-08-04', 'danielacosta@clix.pt', '914907383'),  
('Pilar Vicente', '1978-06-25', 'floureiro@gmail.com', '967518290'),  
('Filipe Moura', '1996-02-27', 'tmarques@hotmail.com', '965952381'),  
('Patrícia Carvalho', '1985-11-27', 'marcio56@sapo.pt', '290456657'),  
('Érica Macedo-Esteves', '1974-05-22', 'azevedocesar@hotmail.com', '916917990'),  
('Noah Araújo', '1997-06-28', 'evacunha@clix.pt', '967496147'),  
('Kevin Macedo', '1980-09-27', 'helena12@gmail.com', '275657910'),  
('Kyara Esteves', '1986-09-19', 'antunessoraia@hotmail.com', '915648529'),  
('Mauro Alves', '1982-09-10', 'bferreira@clix.pt', '918308511'),  
('Carlota Anjos-Henriques', '1990-10-26', 'sergio42@sapo.pt', '914904775'),  
('Caetana Torres', '1991-03-15', 'fariarafeela@clix.pt', '939715716'),  
('Pedro Martins', '1998-03-12', 'britomia@gmail.com', '967199347'),  
('Mia Nascimento', '1966-06-03', 'nevesleonardo@hotmail.com', '920057099'),  
('Joaquim de Oliveira', '1973-07-03', 'paulonascimento@clix.pt', '911758406'),  
('Cristiano Rodrigues', '1960-11-11', 'maurofonseca@sapo.pt', '923547159'),  
('Helena Amorim', '1966-07-12', 'utorres@clix.pt', '930241029'),  
('Gil Silva', '1961-06-26', 'cgomes@clix.pt', '275008492'),  
('Irina Nogueira', '1999-02-01', 'jpacheco@sapo.pt', '915248152'),  
('Emanuel Rodrigues', '1982-06-29', 'nadia21@clix.pt', '937734083'),  
('Emma Pires', '1981-10-18', 'fonsecajuliana@clix.pt', '926653113'),  
('Iara Matos', '1984-10-12', 'cmagalhaes@clix.pt', '960834491'),  
('Kévim Pires', '1960-05-20', 'igor18@gmail.com', '919922428'),  
('Ariana Martins', '1963-05-20', 'araujomicael@clix.pt', '965386662'),



('Ângelo Vicente', '1974-09-07', 'danjos@gmail.com', '924768548'),  
 ('Alícia Amorim', '1973-12-20', 'azevedodenis@gmail.com', '917282705'),  
 ('Frederico do Moreira', '1990-11-22', 'kpereira@hotmail.com', '913907205'),  
 ('Leonardo Mota-Barbosa', '1963-01-08', 'csantos@gmail.com', '910412735'),  
 ('Helena do Matias', '1978-01-11', 'abreuines@hotmail.com', '969131533'),  
 ('Isaac Loureiro', '1971-02-25', 'maiadaniela@gmail.com', '965686276'),  
 ('Mateus Gomes', '1971-04-13', 'xaviercastro@sapo.pt', '916283263'),  
 ('Ariana Coelho', '1989-06-04', 'acunha@clix.pt', '933495270'),  
 ('Inês Lourenço', '1977-02-04', 'goncalo41@sapo.pt', '967137409'),  
 ('Lorena da Pinheiro', '1996-07-21', 'rodrigueseduarda@hotmail.com', '205377744'),  
 ('Juliana-Kyara Matos', '1962-12-18', 'areis@gmail.com', '967939728'),  
 ('Vasco do Moraes', '1983-07-03', 'matiasleite@clix.pt', '912270877'),  
 ('Gustavo Monteiro', '1963-04-20', 'sousavitor@gmail.com', '257084838'),  
 ('Renata Antunes', '1976-09-22', 'batistajorge@gmail.com', '932988072'),  
 ('João Pacheco', '1990-12-18', 'eborges@hotmail.com', '961382860'),  
 ('Ismael Lima', '1982-12-12', 'cbranco@hotmail.com', '915169068'),  
 ('Tiago Guerreiro', '1994-07-07', 'juliana55@hotmail.com', '913158169'),  
 ('Patrícia do Santos', '1966-12-05', 'emilianeto@hotmail.com', '935466156'),  
 ('Ângela Figueiredo-Henriques', '1973-03-17', 'vnunes@hotmail.com', '968224481'),  
 ('Paulo Guerreiro', '1993-11-12', 'guerreirodiana@hotmail.com', '913698384'),  
 ('Tiago Pereira', '1971-10-31', 'kelly65@sapo.pt', '215870619'),  
 ('Brian Cardoso-Macedo', '1973-07-27', 'vbranco@gmail.com', '299758158'),  
 ('Mélanie-Irina Pacheco', '1960-05-30', 'maurotorres@hotmail.com', '929249492'),  
 ('Diogo do Pinho', '1997-10-14', 'machadorui@hotmail.com', '234254577'),  
 ('Jaime Ribeiro', '1995-03-25', 'amaralluna@hotmail.com', '922114335'),  
 ('Catarina Neto', '1961-05-07', 'ismael50@clix.pt', '917484449'),  
 ('Duarte Simões', '1999-04-11', 'zpinto@hotmail.com', '925859308');

INSERT INTO Funcionario (`Nome`, `Telemovel`, `E-mail`, `Salario`, `Gerente`) VALUES  
 ('Luís Presa', '964800687', 'lpresa@sapo.com', 1515, NULL),  
 ('Samuel Gaspar', '967522310', 'batistamauro@gmail.com', 665, 1),  
 ('Renata Alves', '290778354', 'vcarneiro@hotmail.com', 665, 1),  
 ('Patrícia Guerreiro', '240716218', 'mateuspinheiro@clix.pt', 665, 1);

INSERT INTO Edificio (`Rua/Morada`, `Localidade/Morada`, `Codigo\_Postal/Morada`) VALUES  
 ('Largo de Soares, 488', 'Fafe', '9845-670'),  
 ('Largo de Loureiro, 6', 'Arganil', '4911-782');

```

INSERT INTO Alojamento (`Edificio`, `Numero`, `Preco_Base`, `Lotacao`, `Numero_Quartos`) VALUES
(1, 1, 190, 6, 3),
(1, 2, 130, 4, 2),
(1, 3, 130, 4, 2),
(2, 1, 70, 2, 1),
(2, 2, 70, 2, 1),
(2, 3, 130, 4, 2),
(2, 4, 130, 4, 2);

```

```

INSERT INTO Reserva (`Funcionario`, `Cliente`, `Data_Inicio`, `Data_Fim`, `Preco`, `Adultos`, `Crianças`)
VALUES
(2, 1, '2020-01-05', '2020-01-08', 166, 1, 3),
(1, 2, '2020-01-09', '2020-01-12', 205, 3, 2),
(2, 3, '2020-01-12', '2020-01-15', 170, 1, 0),
(3, 3, '2020-01-15', '2020-01-18', 238, 3, 3),
(4, 3, '2020-01-17', '2020-01-20', 175, 2, 0),
(2, 4, '2020-01-21', '2020-01-24', 208, 3, 3),
(1, 5, '2020-01-30', '2020-02-02', 211, 3, 2),
(3, 5, '2020-01-30', '2020-02-02', 174, 1, 2),
(3, 5, '2020-02-06', '2020-02-09', 225, 2, 3),
(1, 6, '2020-02-07', '2020-02-10', 226, 3, 2),
(2, 6, '2020-02-12', '2020-02-15', 240, 3, 2),
(1, 7, '2020-02-16', '2020-02-19', 172, 2, 0),
(4, 8, '2020-02-22', '2020-02-25', 147, 1, 2),
(4, 8, '2020-02-25', '2020-02-28', 151, 2, 1),
(1, 8, '2020-03-01', '2020-03-04', 161, 1, 0),
(3, 9, '2020-03-01', '2020-03-04', 171, 2, 0),
(4, 10, '2020-03-09', '2020-03-12', 146, 3, 0),
(1, 10, '2020-03-15', '2020-03-18', 176, 1, 1),
(2, 11, '2020-03-14', '2020-03-17', 104, 1, 0),
(4, 12, '2020-03-23', '2020-03-26', 164, 3, 1),
(4, 13, '2020-03-23', '2020-03-26', 160, 1, 2),
(1, 13, '2020-03-26', '2020-03-29', 154, 1, 2),
(4, 13, '2020-03-30', '2020-04-02', 207, 3, 0),
(3, 14, '2020-04-03', '2020-04-06', 180, 1, 1),
(4, 15, '2020-04-12', '2020-04-15', 206, 2, 3),
(1, 15, '2020-04-11', '2020-04-14', 235, 2, 2),
(2, 15, '2020-04-15', '2020-04-18', 150, 2, 2),

```

(1, 16, '2020-04-22', '2020-04-25', 149, 1, 2),  
(3, 16, '2020-04-24', '2020-04-27', 219, 3, 0),  
(3, 17, '2020-04-28', '2020-05-01', 179, 1, 2),  
(1, 17, '2020-04-30', '2020-05-03', 147, 1, 1),  
(4, 18, '2020-05-07', '2020-05-10', 151, 1, 3),  
(1, 18, '2020-05-08', '2020-05-11', 227, 3, 3),  
(3, 18, '2020-05-17', '2020-05-20', 149, 1, 2),  
(4, 19, '2020-05-20', '2020-05-23', 110, 1, 0),  
(4, 19, '2020-05-20', '2020-05-23', 234, 3, 3),  
(3, 20, '2020-05-26', '2020-05-29', 218, 2, 3),  
(2, 21, '2020-05-28', '2020-05-31', 237, 2, 0),  
(3, 22, '2020-06-05', '2020-06-08', 148, 2, 1),  
(1, 23, '2020-06-10', '2020-06-13', 236, 2, 2),  
(4, 23, '2020-06-15', '2020-06-18', 169, 1, 3),  
(2, 24, '2020-06-16', '2020-06-19', 220, 1, 3),  
(1, 25, '2020-06-22', '2020-06-25', 218, 3, 3),  
(3, 25, '2020-06-21', '2020-06-24', 210, 3, 2),  
(4, 26, '2020-06-26', '2020-06-29', 176, 2, 0),  
(1, 26, '2020-06-29', '2020-07-02', 237, 3, 3),  
(3, 27, '2020-07-05', '2020-07-08', 155, 3, 1),  
(2, 27, '2020-07-12', '2020-07-15', 158, 1, 1),  
(3, 27, '2020-07-15', '2020-07-18', 174, 1, 1),  
(4, 28, '2020-07-19', '2020-07-22', 220, 3, 2),  
(1, 28, '2020-07-25', '2020-07-28', 207, 3, 3),  
(3, 28, '2020-07-25', '2020-07-28', 149, 2, 0),  
(2, 29, '2020-07-29', '2020-08-01', 151, 1, 0),  
(2, 29, '2020-08-06', '2020-08-09', 155, 3, 0),  
(4, 29, '2020-08-10', '2020-08-13', 234, 2, 3),  
(3, 30, '2020-08-12', '2020-08-15', 159, 2, 1),  
(4, 30, '2020-08-15', '2020-08-18', 102, 1, 1),  
(4, 31, '2020-08-21', '2020-08-24', 156, 1, 2),  
(4, 31, '2020-08-25', '2020-08-28', 218, 3, 1),  
(4, 31, '2020-08-26', '2020-08-29', 147, 1, 3),  
(1, 32, '2020-08-29', '2020-09-01', 233, 3, 0),  
(1, 32, '2020-09-07', '2020-09-10', 175, 2, 1),  
(4, 32, '2020-09-07', '2020-09-10', 166, 3, 1),  
(2, 33, '2020-09-12', '2020-09-15', 165, 2, 2),  
(2, 33, '2020-09-16', '2020-09-19', 236, 2, 1),

(1, 33, '2020-09-23', '2020-09-26', 158, 2, 0),  
(1, 34, '2020-09-27', '2020-09-30', 160, 2, 0),  
(4, 35, '2020-09-26', '2020-09-29', 211, 3, 3),  
(4, 35, '2020-10-01', '2020-10-04', 145, 1, 2),  
(4, 36, '2020-10-09', '2020-10-12', 237, 2, 3),  
(2, 37, '2020-10-10', '2020-10-13', 157, 3, 1),  
(2, 37, '2020-10-16', '2020-10-19', 227, 3, 3),  
(4, 38, '2020-10-17', '2020-10-20', 221, 2, 3),  
(3, 38, '2020-10-25', '2020-10-28', 209, 3, 3),  
(3, 38, '2020-10-24', '2020-10-27', 89, 2, 0),  
(2, 39, '2020-11-01', '2020-11-04', 146, 3, 1),  
(3, 39, '2020-11-06', '2020-11-09', 220, 2, 3),  
(4, 39, '2020-11-05', '2020-11-08', 93, 1, 0),  
(3, 40, '2020-11-13', '2020-11-16', 180, 2, 1),  
(3, 41, '2020-11-17', '2020-11-20', 160, 1, 3),  
(2, 42, '2020-11-16', '2020-11-19', 169, 2, 1),  
(2, 43, '2020-11-20', '2020-11-23', 165, 1, 3),  
(1, 43, '2020-11-25', '2020-11-28', 230, 3, 0),  
(4, 44, '2020-11-29', '2020-12-02', 147, 2, 1),  
(1, 44, '2020-12-05', '2020-12-08', 221, 3, 3),  
(1, 44, '2020-12-07', '2020-12-10', 172, 1, 0),  
(3, 45, '2020-12-16', '2020-12-19', 155, 1, 2),  
(4, 45, '2020-12-19', '2020-12-22', 157, 3, 1),  
(4, 46, '2020-12-22', '2020-12-25', 212, 2, 3),  
(4, 47, '2020-12-22', '2020-12-25', 93, 1, 1),  
(4, 48, '2020-12-30', '2021-01-02', 147, 1, 2),  
(1, 49, '2020-12-30', '2021-01-02', 117, 1, 1),  
(2, 50, '2021-01-05', '2021-01-08', 175, 1, 2),  
(1, 50, '2021-01-12', '2021-01-15', 153, 2, 2),  
(1, 50, '2021-01-11', '2021-01-14', 148, 3, 0),  
(4, 51, '2021-01-18', '2021-01-21', 220, 3, 1),  
(4, 52, '2021-01-25', '2021-01-28', 180, 2, 1),  
(4, 52, '2021-01-23', '2021-01-26', 165, 1, 3),  
(4, 53, '2021-01-29', '2021-02-01', 176, 2, 1),  
(2, 53, '2021-02-01', '2021-02-04', 173, 2, 0),  
(1, 54, '2021-02-08', '2021-02-11', 207, 2, 3),  
(1, 54, '2021-02-10', '2021-02-13', 228, 1, 2),  
(3, 54, '2021-02-14', '2021-02-17', 153, 2, 2),

(1, 55, '2021-02-22', '2021-02-25', 207, 3, 3),  
(2, 56, '2021-02-26', '2021-03-01', 169, 2, 2),  
(3, 56, '2021-03-01', '2021-03-04', 153, 3, 1),  
(3, 57, '2021-02-28', '2021-03-03', 210, 2, 1),  
(2, 57, '2021-03-10', '2021-03-13', 229, 2, 1),  
(2, 58, '2021-03-09', '2021-03-12', 148, 1, 2),  
(4, 58, '2021-03-17', '2021-03-20', 216, 3, 2),  
(1, 58, '2021-03-16', '2021-03-19', 224, 2, 3),  
(2, 59, '2021-03-25', '2021-03-28', 165, 1, 1),  
(2, 60, '2021-03-24', '2021-03-27', 227, 1, 1),  
(4, 60, '2021-03-29', '2021-04-01', 160, 2, 1),  
(1, 61, '2021-04-07', '2021-04-10', 171, 1, 3),  
(1, 61, '2021-04-11', '2021-04-14', 175, 1, 2),  
(3, 62, '2021-04-14', '2021-04-17', 108, 1, 1),  
(1, 63, '2021-04-15', '2021-04-18', 179, 1, 3),  
(3, 64, '2021-04-22', '2021-04-25', 233, 1, 3),  
(4, 64, '2021-04-21', '2021-04-24', 157, 1, 3),  
(3, 64, '2021-04-30', '2021-05-03', 153, 3, 1),  
(2, 65, '2021-05-04', '2021-05-07', 118, 2, 0),  
(3, 65, '2021-05-08', '2021-05-11', 157, 2, 0),  
(4, 65, '2021-05-09', '2021-05-12', 148, 2, 0),  
(2, 66, '2021-05-14', '2021-05-17', 235, 3, 2),  
(2, 66, '2021-05-20', '2021-05-23', 213, 3, 0),  
(3, 66, '2021-05-25', '2021-05-28', 156, 2, 1),  
(3, 67, '2021-05-29', '2021-06-01', 220, 3, 1),  
(3, 67, '2021-06-02', '2021-06-05', 224, 3, 3),  
(3, 68, '2021-06-05', '2021-06-08', 146, 3, 0),  
(2, 69, '2021-06-07', '2021-06-10', 150, 2, 1),  
(4, 70, '2021-06-08', '2021-06-11', 171, 2, 1),  
(4, 71, '2021-06-14', '2021-06-17', 228, 3, 3),  
(2, 72, '2021-06-19', '2021-06-22', 156, 3, 0),  
(2, 72, '2021-06-23', '2021-06-26', 209, 3, 3),  
(3, 73, '2021-06-28', '2021-07-01', 155, 1, 3),  
(1, 73, '2021-06-29', '2021-07-02', 214, 3, 2),  
(4, 73, '2021-07-04', '2021-07-07', 155, 2, 2),  
(4, 74, '2021-07-08', '2021-07-11', 154, 2, 1),  
(1, 74, '2021-07-12', '2021-07-15', 100, 2, 0),  
(3, 75, '2021-07-15', '2021-07-18', 210, 1, 3),

(4, 76, '2021-07-18', '2021-07-21', 145, 2, 0),  
(2, 77, '2021-07-27', '2021-07-30', 177, 3, 0),  
(1, 77, '2021-07-29', '2021-08-01', 156, 3, 0),  
(3, 77, '2021-08-03', '2021-08-06', 238, 1, 3),  
(1, 78, '2021-08-04', '2021-08-07', 228, 1, 1),  
(3, 79, '2021-08-09', '2021-08-12', 159, 2, 1),  
(3, 79, '2021-08-15', '2021-08-18', 215, 2, 3),  
(3, 79, '2021-08-18', '2021-08-21', 161, 1, 3),  
(3, 80, '2021-08-22', '2021-08-25', 173, 2, 0),  
(2, 81, '2021-08-27', '2021-08-30', 208, 2, 0),  
(1, 81, '2021-08-29', '2021-09-01', 97, 1, 1),  
(1, 82, '2021-09-01', '2021-09-04', 217, 2, 3),  
(2, 83, '2021-09-04', '2021-09-07', 165, 1, 3),  
(1, 84, '2021-09-14', '2021-09-17', 156, 1, 2),  
(4, 84, '2021-09-12', '2021-09-15', 215, 3, 3),  
(4, 85, '2021-09-20', '2021-09-23', 240, 3, 2),  
(4, 85, '2021-09-26', '2021-09-29', 209, 3, 3),  
(1, 86, '2021-09-29', '2021-10-02', 180, 1, 2),  
(4, 86, '2021-09-29', '2021-10-02', 178, 2, 2),  
(4, 86, '2021-10-06', '2021-10-09', 165, 2, 2),  
(4, 87, '2021-10-11', '2021-10-14', 212, 3, 2),  
(4, 87, '2021-10-16', '2021-10-19', 89, 1, 1),  
(4, 88, '2021-10-17', '2021-10-20', 213, 1, 2),  
(1, 88, '2021-10-21', '2021-10-24', 149, 2, 2),  
(3, 89, '2021-10-26', '2021-10-29', 224, 1, 0),  
(4, 90, '2021-10-27', '2021-10-30', 156, 2, 2),  
(1, 90, '2021-10-30', '2021-11-02', 145, 2, 0),  
(2, 90, '2021-11-05', '2021-11-08', 236, 3, 3),  
(1, 91, '2021-11-09', '2021-11-12', 215, 3, 2),  
(3, 92, '2021-11-17', '2021-11-20', 173, 3, 1),  
(2, 93, '2021-11-16', '2021-11-19', 218, 2, 2),  
(1, 93, '2021-11-25', '2021-11-28', 168, 3, 0),  
(3, 94, '2021-11-28', '2021-12-01', 166, 2, 0),  
(1, 94, '2021-11-29', '2021-12-02', 167, 2, 0),  
(2, 94, '2021-12-07', '2021-12-10', 168, 3, 0),  
(1, 95, '2021-12-10', '2021-12-13', 238, 2, 3),  
(4, 95, '2021-12-13', '2021-12-16', 147, 2, 2),  
(4, 95, '2021-12-18', '2021-12-21', 234, 3, 3),

```
(4, 96, '2021-12-20', '2021-12-23', 237, 3, 1),
(3, 97, '2021-12-21', '2021-12-24', 164, 2, 2),
(1, 97, '2021-12-29', '2022-01-01', 208, 2, 3),
(1, 98, '2022-01-01', '2022-01-04', 166, 2, 1),
(1, 98, '2022-01-05', '2022-01-08', 148, 1, 0),
(1, 99, '2022-01-06', '2022-01-09', 227, 2, 0),
(1, 99, '2022-01-16', '2022-01-19', 158, 1, 2),
(4, 99, '2022-01-17', '2022-01-20', 226, 3, 2),
(3, 100, '2022-01-23', '2022-01-26', 171, 1, 3),
(1, 100, '2022-01-23', '2022-01-26', 210, 3, 0),
(1, 100, '2022-01-30', '2022-02-02', 160, 1, 1);
```

```
INSERT INTO Reserva_Alojamento (`Reserva`, `Alojamento`) VALUES
```

```
(1, 3),
(2, 1),
(3, 2),
(4, 1),
(5, 2),
(6, 1),
(7, 1),
(8, 7),
(9, 1),
(10, 1),
(11, 1),
(12, 6),
(13, 3),
(14, 6),
(15, 2),
(16, 3),
(17, 3),
(18, 2),
(19, 5),
(20, 7),
(21, 2),
(22, 2),
(23, 1),
(24, 3),
(25, 1),
```

(26, 1),  
(27, 7),  
(28, 3),  
(29, 1),  
(30, 3),  
(31, 3),  
(32, 2),  
(33, 1),  
(34, 3),  
(35, 4),  
(36, 1),  
(37, 1),  
(38, 1),  
(39, 3),  
(40, 1),  
(41, 6),  
(42, 1),  
(43, 1),  
(44, 1),  
(45, 2),  
(46, 1),  
(47, 7),  
(48, 6),  
(49, 2),  
(50, 1),  
(51, 1),  
(52, 3),  
(53, 3),  
(54, 7),  
(55, 1),  
(56, 2),  
(57, 4),  
(58, 3),  
(59, 1),  
(60, 2),  
(61, 1),  
(62, 6),  
(63, 3),



(64, 6),  
(65, 1),  
(66, 6),  
(67, 7),  
(68, 1),  
(69, 7),  
(70, 1),  
(71, 3),  
(72, 1),  
(73, 1),  
(74, 1),  
(75, 5),  
(76, 3),  
(77, 1),  
(78, 4),  
(79, 2),  
(80, 7),  
(81, 2),  
(82, 2),  
(83, 1),  
(84, 3),  
(85, 1),  
(86, 7),  
(87, 3),  
(88, 2),  
(89, 1),  
(90, 4),  
(91, 7),  
(92, 5),  
(93, 3),  
(94, 7),  
(95, 2),  
(96, 1),  
(97, 6),  
(98, 2),  
(99, 6),  
(100, 2),  
(101, 1),

(102, 1),  
(103, 2),  
(104, 1),  
(105, 2),  
(106, 3),  
(107, 1),  
(108, 1),  
(109, 6),  
(110, 1),  
(111, 1),  
(112, 2),  
(113, 1),  
(114, 3),  
(115, 2),  
(116, 6),  
(117, 5),  
(118, 3),  
(119, 1),  
(120, 7),  
(121, 3),  
(122, 4),  
(123, 7),  
(124, 7),  
(125, 1),  
(126, 1),  
(127, 7),  
(128, 1),  
(129, 1),  
(130, 2),  
(131, 3),  
(132, 7),  
(133, 1),  
(134, 2),  
(135, 1),  
(136, 6),  
(137, 1),  
(138, 7),  
(139, 3),

(140, 5),  
(141, 1),  
(142, 7),  
(143, 6),  
(144, 2),  
(145, 1),  
(146, 1),  
(147, 2),  
(148, 1),  
(149, 3),  
(150, 6),  
(151, 1),  
(152, 5),  
(153, 1),  
(154, 7),  
(155, 6),  
(156, 1),  
(157, 1),  
(158, 1),  
(159, 7),  
(160, 6),  
(161, 6),  
(162, 1),  
(163, 5),  
(164, 1),  
(165, 3),  
(166, 1),  
(167, 7),  
(168, 2),  
(169, 1),  
(170, 1),  
(171, 3),  
(172, 1),  
(173, 7),  
(174, 2),  
(175, 7),  
(176, 2),  
(177, 1),

(178, 3),  
(179, 1),  
(180, 1),  
(181, 3),  
(182, 1),  
(183, 2),  
(184, 2),  
(185, 1),  
(186, 7),  
(187, 1),  
(188, 7),  
(189, 1),  
(190, 2);

### III. Anexo 3 - *Script* de Implementação das Interrogações

USE AlojamentoLocal;

-- RE01 - Listar os clientes que frequentaram o estabelecimento;

SELECT Id\_Cliente, Nome, `E-mail`, Telemovel FROM Cliente

WHERE Id\_Cliente IN

(SELECT Cliente FROM Reserva);

-- RE02 – Listar os edifícios e respetivos alojamentos

SELECT E.Id\_Edificio, E.`Rua/Morada`, A.Numero AS Numero\_Alojamento, A.Numero\_Quartos, A.Lotacao

FROM Edificio AS E

INNER JOIN Alojamento AS A

ON A.Edificio = E.Id\_Edificio

ORDER BY Id\_Edificio;

-- RE03 - Listar todas as reservas realizadas num certo alojamento

SELECT RA.Reserva, R.Data\_Inicio, R.Data\_Fim FROM Reserva\_Alojamento AS RA

INNER JOIN Reserva AS R

ON R.Id\_Reserva = RA.Reserva

WHERE Alojamento = 1;

-- RE04 – Listar todos os funcionários e respetivo responsável

SELECT Id\_Funcionario, Nome, Gerente FROM Funcionario;

-- RE05 – Calcular o número de vezes que cada alojamento foi reservado

SELECT P.Id\_Alojamento, P.Numero, P.Edificio, COUNT(P.Reserva) AS Total\_Reservas FROM

(SELECT A.Id\_Alojamento, A.Numero, A.Edificio, RA.Reserva FROM Reserva\_Alojamento AS RA

INNER JOIN Alojamento AS A

ON A.Id\_Alojamento = RA.Alojamento) AS P

GROUP BY Id\_Alojamento;

-- RE06 – Calcular o número de pedidos que um funcionário processou

SELECT R.Funcionario, F.Nome, COUNT(R.Id\_Reserva) AS Total\_Reservas FROM Reserva AS R

INNER JOIN Funcionario AS F

ON F.Id\_Funcionario = R.Funcionario

GROUP BY Funcionario;

-- RE07 – Listar quantos quartos estão disponíveis até um certo preço  
SELECT COUNT(Id\_Alojamento) FROM Alojamento WHERE Preco\_Base <= 100;

-- RE08 - Listar os clientes por ordem decrescente de dinheiro gasto  
SELECT R.Cliente, C.Nome, SUM(R.Preco) AS Total\_Gasto FROM Reserva AS R  
INNER JOIN Cliente AS C  
ON C.Id\_Cliente = R.Cliente  
GROUP BY R.Cliente  
ORDER BY Total\_Gasto DESC;

## IV. Anexo 4 - *Script* da Criação das Views

```
USE AlojamentoLocal;
```

```
-- V01 - Calcular o número de vezes que cada alojamento foi reservado
```

```
CREATE VIEW V01 AS
```

```
SELECT P.Id_Alojamento, P.Numero, P.Edificio, COUNT(P.Reserva) AS Total_Reservas FROM
```

```
(SELECT A.Id_Alojamento, A.Numero, A.Edificio, RA.Reserva FROM Reserva_Alojamento AS RA
```

```
INNER JOIN Alojamento AS A
```

```
ON A.Id_Alojamento = RA.Alojamento) AS P
```

```
GROUP BY Id_Alojamento;
```