



Universidade do Minho
Licenciatura em Engenharia Informática

Unidade Curricular de Bases de Dados

Ano Lectivo de 2022/2023

Relatório Trabalho Prático

Hugo Dias a100758

Hugo Ramos a100644

Ricardo Jesus a100066

Rui Pinto a100659

Junho, 2023

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Hotel do Sr. Libério

Hugo Dias a100758

Hugo Ramos a100644

Ricardo Jesus a100066

Rui Pinto a100659

Resumo

Este relatório é feito no âmbito da Unidade Curricular de Bases de Dados, e é referente a um projeto requerido pela equipa docente.

O projeto visa a criação e implementação de um Sistema de Base de Dados, aplicado a um contexto fictício criado e idealizado pelo grupo que realizou o mesmo. Este contexto fictício envolve um negócio de pequena escala, ao qual pretendemos adicionar valor e facilidade com a implementação do Sistema sugerido.

Com esta implementação são tidos em conta todos os aspetos necessários. Inicialmente a idealização do caso de estudo, seguidamente o plano de execução, o levantamento de todos os requisitos necessários para que a sua criação seja facilitada, a criação dos modelos necessários, a implementação de um sistema de recolha de dados, e ainda a implementação de uma dashboard de ajuda à análise da base de dados por parte de um utilizador que não tem conhecimento acerca da parte de desenvolvimento relativo à mesma.

Área de Aplicação: Hotelaria.

Palavras-Chave: Turismo, Base de Dados

Índice

Resumo	3
Índice	4
Índice de figuras	6
Índice de tabelas	8
1. Definição do sistema	9
1.1. Contexto de aplicação e fundamentação do sistema	9
1.2. Motivação e Objetivos do Trabalho	10
1.3. Análise da viabilidade do processo	10
1.4. Recursos e Equipa de Trabalho	11
1.5. Plano de Execução do Projeto	12
2. Levantamento e análise de requisitos	12
2.1. Método de levantamento e de análise de requisitos adotado	12
2.2. Organização dos requisitos levantados	13
2.2.1. Requisitos de descrição	13
2.2.2. Requisitos de exploração	14
2.2.3 Requisitos de controlo	15
2.3. Análise e validação geral dos requisitos	15
3. Modelação conceptual	15
3.1. Abordagem da modelação realizada	15
3.2. Identificação e caracterização das entidades e respetivos atributos	16
3.3. Identificação e caracterização dos relacionamentos	20
3.4. Apresentação e explicação do diagrama ER produzido	20
3.5. Análise e Validação do Sistema	21
4. Modelação lógica	21
4.1. Construção e validação do modelo de dados lógico	21
4.2. Normalização de dados	25
4.3. Apresentação do modelo lógico produzido	26
4.4. Validação do modelo com interrogações do utilizador	27
4.5. Análise do sistema	27
5. Implementação Física	27
5.1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	28
5.2. Tradução das interrogações do utilizador para SQL (alguns exemplos)	34
5.3. Definição e caracterização das vistas de utilização em SQL	36
5.4. Cálculo do espaço da bases de dados (inicial e taxa de crescimento anual)	38

5.5. Plano de segurança e recuperação de dados	39
6. Implementação do Sistema de Recolha de Dados	39
6.1. Apresentação e modelo do sistema	39
6.2. Implementação do sistema de recolha	39
6.3. Funcionamento do sistema	40
7. Implementação do Sistema de Painéis de Análise	40
7.1. Definição e caracterização da vista de dados para análise	40
7.2. Povoamento das estruturas de dados para análise	40
7.3. Apresentação e caracterização dos dashboards implementados	41
8. Conclusões	41

Índice de Figuras

Figura 1 - Diagrama de Gantt (1ªparte)	12
Figura 2 - Diagrama de Gantt (2ªparte)	12
Figura 3 - Modelo Conceptual	21
Figura 4 – Modelo Lógico	26
Figura 5 - Criação tabela morada	28
Figura 6 - Criação Tabela cliente	29
Figura 7 - Criação tabela hotel	30
Figura 8 - Criação tabela regime	30
Figura 9 - Criação tabela reserva (1ª parte)	31
Figura 10 - Criação tabela reserva (2ªparte)	31
Figura 11 - Criação tabela funcionário	32
Figura 12 - Criação tabela quarto	33
Figura 13 - Criação tabela relação cliente-hotel	33
Figura 14 - Criação tabela relação reserva-quarto	34
Figura 15 - Agências Utilizadas	34
Figura 16 - Faturação da Cadeia Hoteleira	35
Figura 17 - Informações através do ID	35
Figura 18 - Média de Pessoas por Reserva	35
Figura 19 - Média de salários dos Funcionários	36
Figura 20 - Criação vista reserva	36
Figura 21 – Criação vista funcionário	36
Figura 22 - Criação utilizador para Recepcionistas	37

Figura 23 - Criação utilizador para Chefes de Departamento	37
Figura 24 - Criação utilizador para Administradores	37
Figura 25 - Criação permissões para Recepcionistas	38
Figura 26 - Criação permissões para Chefes de Departamento	38
Figura 27 - Criação permissões para Administradores	38
Figura 28 - Tamanho da base de dados	39
Figura 29 - Comando de recuperação de dados	39
Figura 30 - Script de recolha	40
Figura 31 - Dashboard Administrador	41

Índice de Tabelas

Tabela 1 - Caracterização de Entidades	16
Tabela 2 - Atributos da Entidade Cliente	17
Tabela 3 - Atributos da Entidade Reserva	17
Tabela 4 - Atributos da Entidade Quarto	18
Tabela 5 - Atributos da Entidade Hotel	19
Tabela 6 - Atributos da Entidade Funcionário	19
Tabela 7 - Tabela Cliente	22
Tabela 8 - Tabela Cliente_has_Hotel	22
Tabela 9 - Tabela Hotel	22
Tabela 10 - Tabela Funcionário	23
Tabela 11 - Tabela Reserva	23
Tabela 12 - Tabela Morada	24
Tabela 13 - Tabela Reserva_has_Quarto	24
Tabela 14 - Tabela Morada	24
Tabela 15 - Tabela Regime	24
Tabela 16 - Tabela Quarto	25

1. Definição do sistema

1.1 Contexto de aplicação e fundamentação do sistema

Contextualização

O Sr. Libério Palhinha nasceu e foi criado em Galafura, uma aldeia na região do Douro. Desde novo, ele trabalhou em diversas posições na indústria hoteleira, adquirindo assim um vasto conhecimento e habilidades práticas ao longo dos vastos anos na área.

Em 2007, o Sr. Libério decidiu pedir um empréstimo ao banco e, finalmente, concretizar o seu sonho de abrir o seu próprio hotel. Abriu o seu primeiro hotel em Vila Real, a sua ideia era suprir a crescente demanda por turismo na região, principalmente na altura das corridas que trazem turistas de todos os cantos do mundo. O Sr. Libério investiu muito tempo e dinheiro para que o seu hotel oferecesse um serviço de alta qualidade e acomodações de luxo, e com o tempo, tornou-se um ponto de referência na área do turismo na cidade.

Com o grande sucesso do seu primeiro hotel, o Sr. Libério resolveu expandir o seu negócio para além de Vila Real. Em 2018, inaugurou o segundo hotel da cadeia hoteleira à beira rio, nas margens do Rio Douro, na cidade de Peso da Régua. Este hotel com uma arquitetura elegante e espaços públicos sofisticados, oferece aos seus visitantes uma experiência inesquecível com as suas diversas atividades, como degustação de vinho e caminhadas pela região. Isto tudo acompanhado por uma vista deslumbrante do rio e das vinhas que rodeiam o hotel.

Fundamentação

Apesar do Sr. Libério já possuir uma cadeia hoteleira com 2 hotéis que até agora tinha corrido bem, o sistema utilizado pela empresa demonstrava algumas fraquezas, principalmente pelo facto de tudo funcionar através de um simples e único livro de registos para cada hotel, onde eram colocadas todas as informações de cada reserva. Ora isto era o primeiro sério problema de Libério, que impedia o crescimento exponencial dos hotéis. Outro dos problemas era o facto da publicidade do hotel ser praticamente inexistente, e a existência do hotel chegava ao cliente através do puro boca a boca.

Com a vinda da primeira vaga do COVID-19, Libério viu-se forçado a fechar portas por não haver praticamente clientes nenhuns, uma vez que o Turismo em Portugal praticamente cessou, e pior ainda nas áreas rurais como a que se encontrava o negócio do senhor Libério. Esta vaga passou e Libério voltou a abrir portas, mas ao contrário do que acontecia antigamente, o método de publicidade boca a boca acabou, já que as pessoas tinham ficado em casa nos últimos dois anos, e agora o mundo tinha-se tornado muito mais tecnológico, coisa que o senhor Libério sempre criou muita resistência contra.

Passados os primeiros meses Libério viu-se obrigado a ceder à pressão da família e vendeu 51% da empresa a um investidor japonês Hiroko Sushima, um homem conceituado no que toca ao turismo e indústria hoteleira. A primeira medida de Hiroko foi ordenar a criação de uma base de dados, para melhor organização, facilitação da gestão e acima de tudo um facilitamento extremo no que toca ao registo de cada reserva, o que permitia que fossem implementados sistemas de análise das informações dos clientes, para que fossem oferecidos os serviços que mais se adequam a cada tipo de cliente.

1.2 Motivação e Objetivos do Trabalho

O Sr. Hiroko pediu ajuda ao primo de Libério Palinha, Sérgio Mallandro, dono de uma empresa de desenvolvimento e implementação de Software, que já tinha uma vasta experiência na área de sistemas de base de dados, tendo já implementado com bastante sucesso este tipo de sistemas em variadíssimos negócios no campo da restauração. Já que anteriormente tudo era feito manualmente através de papel e lápis, a criação da base de dados ajudará a automatizar tudo isto. De modo a ajudar o seu primo, Sérgio listou uma série de objetivos que considerou importantes para a base de dados da cadeia hoteleira de Libério. Estes são:

- Conseguir armazenar todas as informações acerca de todas as características de cada reserva
- Ter acesso às informações de cada reserva quando for necessário
- Organizar e melhorar as vendas da cadeia hoteleira como um todo
- Através de sistemas de análises das informações registadas em cada reserva, ver que tipo de reserva é mais comum ser feita por um tipo específico de cliente.
- Aumentar a rapidez da reserva
- Saber com mais facilidade, quantos quartos estão disponíveis no momento que for necessário.
- Melhorar a gestão de toda a instituição controlando melhor os gastos e aprovisionando melhor todos os quartos necessários para cada dia
- Registo de reservas.
- Armazenamento de todas as informações relativas à reserva.
- Facilitamento do acesso às informações relativas às reservas.
- Organização e melhoramento das vendas da cadeia hoteleira.
- Melhor gestão dos contratos dos funcionários.
- Aumento da rapidez do processamento de uma reserva.
- Melhoria da qualidade do serviço prestado ao cliente.
- Melhor aprovisionamento do serviço de quartos.
- Armazenamento de informações relativas ao cliente.
- Capacidade de entendimento das tendências do cliente.
- Simplificação do tratamento de burocracias

1.3 Análise da viabilidade do processo

- Através de sistemas de análise do cliente sugerir o melhor serviço para o tipo específico de cliente.
- Facilitar o acesso às informações relativas ao número de quartos disponíveis.
- Aumentar em 80% a rapidez do processamento de uma reserva.

- Melhorar a gestão dos funcionários diminuindo em 60% a carga horária gasta neste parâmetro.
- Permitir uma exponenciação do negócio sem diminuição da qualidade do serviço prestado.
- Facilitar o aprovisionamento de todas as atividades
- Aumentar a produtividade em 440%.

1.4 Recursos e equipa de trabalho

Recursos:

Humanos

- Funcionários do Hotel, Funcionários da Empresa do Sr. Sérgio Mallandro, Clientes.

Materiais

- Hardware (1 servidor para cada hotel, 3 computadores (2 para o resort e 1 para o hotel da cidade)).
- Software (Sistema de Base de Dados, aplicação de interação com a Base de Dados, Aplicação de registos de reservas).

Equipa de Trabalho:

Pessoal Interno

- Empregados, Sr. Libério, Sr. Hiroko Sushima,
- Registo de clientes, atendimento ao cliente, serviço de quartos, tratamento da piscina, serviço de bar, (...)

Pessoal Externo

- Engenheiro de Base de Dados, Programadores, Sr. Sérgio Mallandro
- Levantamento de requisitos, modelação conceptual do sistema, implementação do sistema, (...)

1.5 Plano de execução do projeto

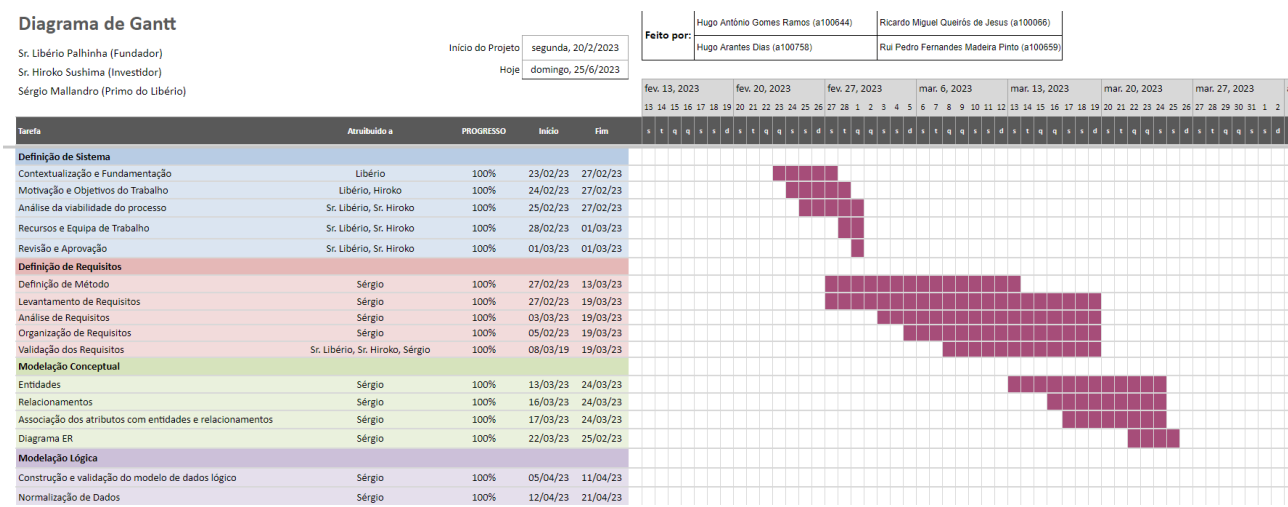


Figura 1 - Diagrama de Gantt (Parte 1)

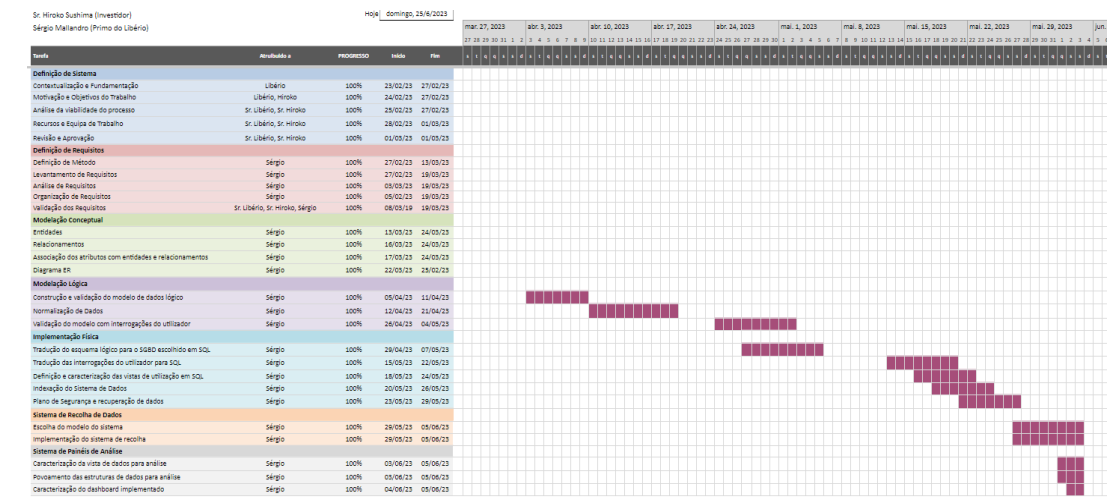


Figura 2 - Diagrama de Gantt (Parte 2)

2. Levantamento e análise de requisitos

2.1. Método de levantamento e análise de requisitos adotados

De forma a proceder ao levantamento de requisitos necessários para a criação da base de dados foram delineadas as seguintes tarefas pelo senhor Sérgio:

- Reunião com Libério e Hiroko para identificação dos processos que ocorrem no hotel.
- Identificação das entidades do hotel.
- Observação da informação necessária para a realização destes processos.

- Análise das informações dos Livros de Registos de modo a perceber o essencial em relação ao registo de cada reserva.
- Telefonema aos principais hóspedes dos últimos meses para perceber a sua opinião.

2.2. Organização dos requisitos levantados

2.2.1. Requisitos de descrição

- O sistema deve manter um registo de todos os quartos que cada hotel possui
- Cada quarto deve ter associado a si uma série de atributos: o seu id, e informações relativas a se tem ou não, televisão, varanda, cofre, se está ou não disponível e o nome do hotel a que pertence
- O sistema deve ter um registo de todos os empregados que trabalham no hotel.
- Cada funcionário tem uma série de atributos: o seu id, o seu nome, o seu cargo, a sua data de nascimento, o salário, o nome do hotel onde trabalha, e ainda a sua morada.
- Cada um dos hotéis têm associados o seu nome, telefone, número de lugares de estacionamento, número de quartos, número de piscinas, uma informação relativa a se tem ou não spa e ginásio, a hora de abertura e fecho, e ainda a sua morada.
- O sistema deve manter o registo de todas as reservas já feitas, e todas as informações relativas às mesmas.
- Cada reserva tem como atributos, o seu id, uma informação de se foi cancelada ou não, a agência pela qual foi feita a reserva, o número de lugares de estacionamento a que tem direito, data do Check-In e Check-Out, data em que foi feita, método de pagamento usado, preço, número de crianças, número de bebés, número de adultos, o nome do Hotel a que se refere, o id do cliente que efetivou a reserva e regime que esta comporta, podendo este ser apenas pequeno almoço, meia pensão ou pensão completa.
- O sistema deve manter o registo de todos os clientes que já frequentaram a cadeia de hotéis e todas as informações relativas aos mesmos.
- Cada cliente tem uma série de atributos, como o seu id, nome, email, número de cartão de crédito, idade, nif e a sua morada. Sendo que o email, número de cartão de crédito e NIF são opcionais.

2.2.2. Requisitos de exploração

Consulta:

- Deve ser possível recolher uma lista acerca das reservas que foram feitas, apresentando detalhes acerca de quantos quartos foram vendidos, no espaço de tempo que for pretendido.
- O sistema deve conseguir distinguir a quantidade de reservas que foi feita em cada agência de viagens, e as que foram feitas na casa, no espaço de tempo que for pretendido.
- O sistema tem de ser capaz de verificar quanto tempo um certo hóspede esteve no hotel.
- O sistema deve ser capaz de verificar médias de adultos/crianças/bebés há por reserva de quarto.
- O sistema deve ser capaz de verificar quais são os meses mais concorridos e que são menos concorridos.
- O sistema deve ser capaz de comparar o número de reservas que houve entre os dois hotéis entre duas datas.
- O sistema deve ser capaz de obter uma média do preço de todas as reservas.
- O sistema deve ser capaz de obter uma lista de nomes dos clientes que já frequentaram os hotéis.
- O sistema deve ser capaz de obter qual é a média de tempo de estadia do hotel.
- Através do número da reserva deve ser possível obter todas as informações da mesma.
- O sistema deve ser capaz de fazer uma média de quantos estacionamento são pedidos por reserva.
- O sistema deve ser capaz de através do id de um funcionário consultar as suas informações.
- O sistema deve ser capaz de saber quanto a cadeira hoteleira faturou num intervalo de tempo.
- O sistema deve ser capaz de obter uma média do salário gasto nos funcionários.
- O sistema deve ser capaz de fazer uma média de idades dos funcionários.

Atualização:

- O sistema deve ter a possibilidade de atualização, inserção ou remoção de um funcionário.
- O sistema deve ter a possibilidade de atualização, inserção ou remoção de uma reserva.
- O sistema deve ter a possibilidade de atualização, inserção ou remoção de um cliente.
- O sistema deve ter a possibilidade de atualização, inserção ou remoção de um quarto.
- O sistema deve ter a possibilidade de atualização de um hotel.

2.2.3. Requisitos de Controlo

- Os rececionistas do hotel apenas podem consultar e adicionar registos de reservas ao sistema, não podendo alterar nada.
- Apenas os administradores podem atualizar ou remover um novo funcionário.
- Apenas os administradores podem consultar informações em relação a um funcionário.
- Apenas os administradores podem inserir um novo funcionário.
- Apenas os administradores podem consultar estatísticas dos funcionários.
- Apenas administradores e chefes do departamento podem alterar informações das reservas.
- Os rececionistas do hotel conseguem consultar informações acerca dos clientes.
- Os rececionistas do hotel conseguem registar um novo cliente.
- Apenas os administradores podem consultar estatísticas dos clientes.
- Apenas os administradores podem consultar estatísticas das reservas.

2.3. Análise e validação geral dos requisitos

No seguimento do levantamento e organização de todos os requisitos, os mesmos passaram por uma rigorosa análise por parte de Libério, Hiroko, Sérgio e a sua equipa de programadores, de modo a serem validados para que o processo pudesse ser continuado. Foram corrigidos erros e adicionados parâmetros que o senhor Libério achava essenciais para o funcionamento do Hotel.

Finalmente após todo um processo de análise e validação, os requisitos foram aprovados e o projeto foi prosseguido.

3. Modelação Conceptual

3.1. Abordagem da modelação realizada

Para o modelo conceptual a equipa que anteriormente fez o levantamento dos requisitos definiu os seguintes passos a seguir pela ordem definida.

- **Definição das Entidades** - Identificação das entidades essenciais e claramente definidas no problema em questão.
- **Definição dos Atributos das Entidades** - Identificação dos atributos de que cada entidade precisa para que haja uma maior efetividade na base de dados.
- **Definição dos Relacionamentos entre Entidades** - Identificação das relações existentes entre as várias entidades para funcionamento da base de dados.

- **Construção do modelo conceptual** - construção do diagrama com a ferramenta escolhida sendo o BrModelo.

O modelo conceptual consiste numa representação gráfica de todas as entidades com os seus respetivos atributos, onde estão também apresentadas as relações que estas entidades têm entre si.

3.2. Identificação e caracterização das entidades

Com a análise dos requisitos de descrição podemos concluir que seriam necessárias as seguintes entidades, com os respetivos atributos referentes a cada uma das entidades.

Tabela de Caracterização de Entidades

Entidade	Descrição	Sinónimos	Ocorrência
Cliente	Informações sobre os clientes que frequentam ou que já frequentaram os hotéis.	-----	Os clientes têm um registo único, que é obrigatório, para poder ser registada uma reserva.
Reserva	Representa as reservas de quartos dos hotéis, contendo informações sobre o regime, nome do hotel do quarto reservado, etc...	-----	Cada reserva tem um número único, que é atribuído no momento do seu registo.
Quarto	Representa a informação sobre os quartos disponíveis para reserva nos hotéis.	-----	Cada quarto é representado pelo seu número de quarto, que é único.
Hotel	Contém informações sobre os dois hotéis.	-----	Cada um dos dois hotéis tem um nome que identifica cada um.
Funcionário	Contém informações sobre os funcionários dos hotéis.	-----	Cada um dos funcionários é identificado por um "id" único.

Tabela 1 - Caracterização de Entidades

Tabelas de Associação dos Atributos às Entidades

Entidade	Atributos	Descrição	Domínio e Tamanho	Nul o	Exemplo
Cliente	idCliente	Identificador único do cliente	INT	Não	21
	nome	Nome completo	VARCHAR (45)	Não	Pedro Simão
	nrCartaoCredito	Número do cartão de crédito	INT	Sim	4325 65433 33333
	NIF	Número de contribuinte	INT	Sim	266554222
	Email	Endereço de email	VARCHAR (45)	Sim	pedro@gmail.com
	Morada <i>Rua</i> <i>Código-Postal</i> <i>Localidade</i> <i>Porta</i>	Morada do cliente	VARCHAR(45) INT VARCHAR(45) INT	Não	Avenida da Liberdade 2410-105 Leiria 69

Tabela 2 - Atributos da Entidade Cliente

Entidade	Atributos	Descrição	Domínio e Tamanho	Nulo	Exemplo
Reserva	idReserva	Identificador da reserva (Chave primária)	INT	Não	2313
	cancelamento	Se a reserva já foi cancelada ou não	TINYINT	Não	0
	regime <i>peqAlm</i> <i>meiaPensao</i> <i>pensaoCompleta</i>	Tipo de regime	TINYINT TINYINT TINYINT	Não	Meia Pensão
	preco	Preço total	DOUBLE	Não	850,50
	nrLugaresEstacionamento	Número de lugares de estacionamento atribuídos	INT	Não	2
	nrAdultos	Número de adultos	INT	Não	1

	nrCrianças	Número de crianças	INT	Não	2
	nrBebes	Número de bebês	INT	Não	1
	agencia	Nome da agência que tratou da reserva	VARCHAR (45)	Sim	Agência Abreu
	metodoPagamento	Nome do método de pagamento utilizado	VARCHAR (45)	Não	Multibanco
	dataPagamento	Data em que foi efetuado o pagamento	DATE	Não	20-02-2021
	checkIn	Data de check-in	DATE	Não	20-07-2021
	checkOut	Data de check-out	DATE	Não	27-07-2021

Tabela 3 - Atributos da Entidade Reserva

Entidade	Atributos	Descrição	Domínio e Tamanho	Nulo	Exemplo
Quarto	idQuarto	Número único identificador do quarto	INT	Não	123
	televisao	Se o quarto tem televisão	TINYINT	Não	1
	varanda	Se o quarto tem acesso a varanda	TINYINT	Não	1
	cofre	Se o quarto tem cofre	TINYINT	Não	0
	disponibilidade	Data em que o quarto tem disponibilidade para ser reservado	DATE	Não	20-06-2022

Tabela 4 - Atributos da Entidade Quarto

Entidade	Atributos	Descrição	Domínio e Tamanho	Nulo	Exemplo
Hotel	nome	Nome do hotel	VARCHAR (45)	Não	City Hotel
	telefone	Número de telefone	INT	Não	259923222
	nrQuartos	Número de quartos	INT	Não	215
	nrLugaresEstacionamento	Número de estacionamentos	INT	Não	300
	nrPiscinas	Número de piscinas	INT	Não	25
	spa	Se o hotel tem spa	TINYINT	Não	1
	horaAbertura	Horário de serviço do hotel	VARCHAR(8)	Não	09:00 23:00
	horaFecho	Se tem ginásio	VARCHAR(8)	Não	1
	morada rua código-postal localidade porta	Morada do Hotel	VARCHAR(45) INT VARCHAR(45) INT	Não	Avenida 1º de Maio 5000-651 Vila-Real 53
	ginasio	Se o hotel tem ginásio	TINYINT	Não	1

Tabela 5 - Atributos da Entidade Hotel

Entidade	Atributos	Descrição	Domínio e Tamanho	Nulo	Exemplo
Funcionário	idFuncionário	Identificador único do funcionário	INT	Não	231
	nome	Nome do Hotel onde o funcionário trabalha	VARCHAR (45)	Não	City Hotel
	salário	Salário do funcionário	DOUBLE	Não	Ricardo Cabeças
	função	Função exercida pelo funcionário	VARCHAR (45)	Não	Rececionista
	idade	Idade do funcionário	INT	Não	15-01-1995

	morada <i>Rua</i> <i>Código-Postal</i> <i>Localidade</i> <i>Porta</i>	Morada do funcionário	VARCHAR(45) INT VARCHAR(45) INT	Não	Rua de Barros 4710-058 Gualtar 55
--	--	-----------------------	--	-----	--

Tabela 6 - Atributos da Entidade Funcionário

3.3. Identificação e caracterização dos relacionamentos

Após estabelecer as entidades do sistema é possível identificar os relacionamentos entre as mesmas.

Cliente-Reserva (faz) - 1: N; P:T

- Uma reserva está sempre associada a um cliente, mas um Cliente não tem necessariamente de estar relacionado a uma reserva.

Reserva-Quarto (tem) - N:M; T:P

- As reservas têm de estar obrigatoriamente associadas a um ou mais quartos, mas um ou mais quartos não têm necessariamente de estar relacionados com uma reserva, pois um quarto pode não ter nenhuma reserva associada ao mesmo, mas o quarto existe na mesma.

Hotel-Quarto (é constituído) - 1: N; T:T

- Um hotel tem necessariamente que estar associado a um quarto, e um quarto tem de estar obrigatoriamente associado a um hotel.

Funcionário-Hotel (trabalha) - N:1; T:T

- Um funcionário tem obrigatoriamente de estar associado a um hotel que, no nosso caso, é o hotel no qual trabalho e, um hotel, deve ter pelo menos, um funcionário associado.

Cliente-Hotel (associado a) - N:M; T:P

- Um cliente tem de estar associado pelo menos a um hotel, mas um hotel não tem obrigatoriamente de ter clientes associados a si.

3.4. Apresentação e explicação do diagrama ER produzido

O diagrama produzido nas seguintes figuras foi elaborado na plataforma BrModelo, e foi tido em conta a identificação de todas as entidades, atributos e relacionamentos descritos anteriormente.

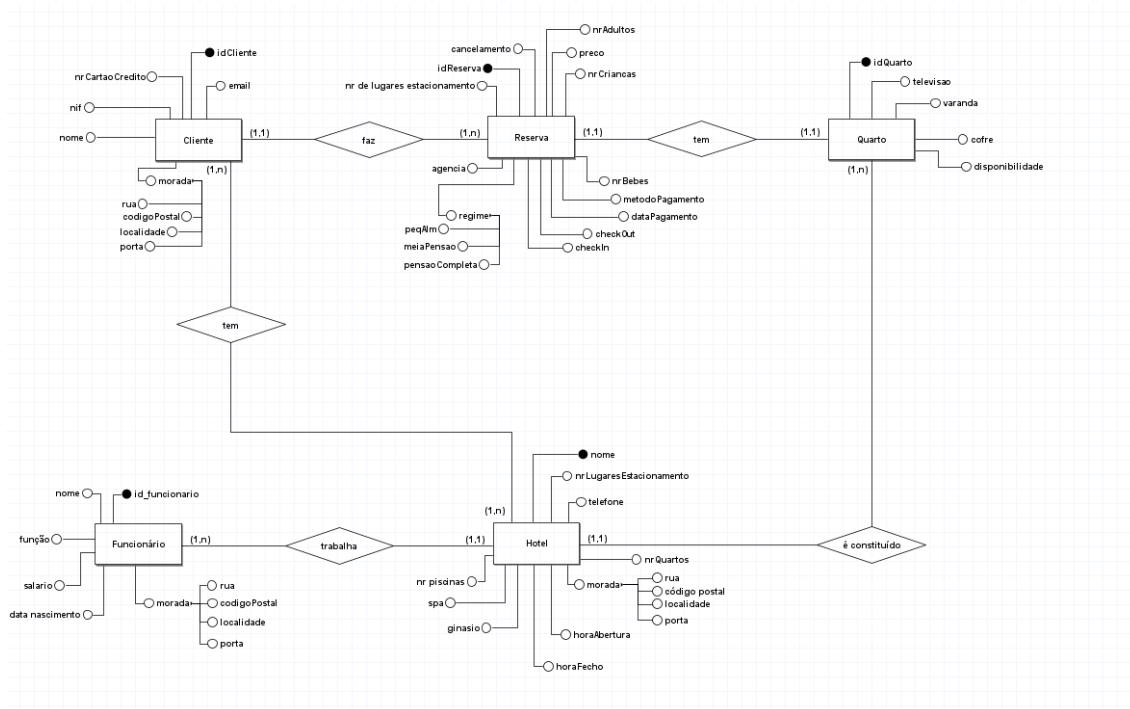


Figura 3 - Modelo Conceptual

3.5. Análise e Validação do Sistema

Como forma de validação do modelo conceptual construído pela equipa de Sérgio, foi marcada uma reunião com o mesmo para que se verificasse se todos os relacionamentos e entidades se encontravam bem representadas de acordo com o que foi discutido nas reuniões anteriores. Nesta reunião final detetou-se que seria necessário tratar a Entidade Cliente como um cliente do hotel e não apenas como o cliente que fazia a reserva no hotel. Ora isto fez com que fosse necessário adicionar o relacionamento entre as entidades Hotel e Cliente.

4. Modelação lógica

4.1. Construção e validação do modelo de dados lógico

Foi elaborado um modelo lógico com base no Modelo Conceptual criado anteriormente. Durante esta fase, as entidades, relacionamentos e entidades-relacionamento são convertidos em tabelas, seguindo a abordagem do modelo relacional. Em cada uma destas tabelas é necessário proceder à identificação da chave primária, dos seus atributos e, em alguns casos, às suas chaves estrangeiras.

O nosso modelo lógico é constituído então, pelas seguintes tabelas:

Cliente	
idCliente INT NOT NULL	Chave Primária
FK_idMorada INT NOT NULL	Chave Estrangeira (Hotel)
email VARCHAR (45)	Atributo
nome VARCHAR (45) NOT NULL	Atributo
nrCartaoCredito INT	Atributo
idade INT NOT NULL	Atributo

Tabela 7 - Tabela Cliente

Cliente_has_Hotel	
FK_idCliente INT NOT NULL	Chave Estrangeira (Cliente)
FK_nomeHotel VARCHAR (45) NOT NULL	Chave Estrangeira (Hotel)

Tabela 8 - Tabela Cliente_has_Hotel

Hotel	
nome VARCHAR (45) NOT NULL	Chave Primária
FK_idMorada INT NOT NULL	Chave Estrangeira (Morada)
telefone INT NOT NULL	Atributo
nrLugaresEstacionamento INT NOT NULL	Atributo
nrQuartos INT NOT NULL	Atributo
nrPiscinas INT NOT NULL	Atributo
spa TINYINT NOT NULL	Atributo
ginasio TINYINT NOT NULL	Atributo
horaAbertura VARCHAR (8) NOT NULL	Atributo
horaFecho VARCHAR (8) NOT NULL	Atributo

Tabela 9 - Tabela Hotel

Funcionário	
idFuncionário INT NOT NULL	Chave Primária
FK_nomeHotel VARCHAR (45) NOT NULL	Chave Estrangeira (Hotel)
FK_idMorada INT NOT NULL	Chave Estrangeira (Morada)
nome VARCHAR (45) NOT NULL	Atributo
funcao VARCHAR (45) NOT NULL	Atributo
dataNascimento DATE NOT NULL	Atributo
salario DOUBLE NOT NULL	Atributo

Tabela 10 - Tabela Funcionário

Reserva	
idReserva INT NOT NULL	Chave Primária
FK_nomeHotel VARCHAR (45) NOT NULL	Chave Estrangeira (Hotel)
FK_idCliente INT NOT NULL	Chave Estrangeira (Cliente)
FK_idRegime INT NOT NULL	Chave Estrangeira (Regime)
cancelamento TINYINT NOT NULL	Atributo
agencia VARCHAR (45) NOT NULL	Atributo
nrLugaresEstacionamento INT NOT NULL	Atributo
checkIn DATE NOT NULL	Atributo
checkOut DATE NOT NULL	Atributo
dataPagamento DATE NOT NULL	Atributo
metodoPagamento VARCHAR (45) NOT NULL	Atributo
preco DOUBLE NOT NULL	Atributo
nrCrianças INT NOT NULL	Atributo
nrBebes INT NOT NULL	Atributo
nrAdultos INT NOT NULL	Atributo

Tabela 11 - Tabela Reserva

Morada	
idMorada INT NOT NULL	Chave Primária
rua VARCHAR (45) NOT NULL	Atributo
codigoPostal INT NOT NULL	Atributo
localidade VARCHAR (45) NOT NULL	Atributo
porta INT	Atributo

Tabela 12 - Tabela Morada

Reserva_has_Quarto	
FK_idReserva INT NOT NULL	Chave Estrangeira (Reserva)
FK_idQuarto INT NOT NULL	Chave Estrangeira (Quarto)

Tabela 13 - Tabela Reserva_has_Quarto

Morada	
idMorada INT NOT NULL	Chave Primária
rua VARCHAR (45) NOT NULL	Atributo
codigoPostal INT NOT NULL	Atributo
localidade VARCHAR (45) NOT NULL	Atributo
porta INT	Atributo

Tabela 14 - Tabela Morada

Regime	
id INT NOT NULL	Chave Primária
peqAlm TINYINT NOT NULL	Atributo
meiaPensao TINYINT NOTNULL	Atributo
pensaoCompleta TINYINT	Atributo

Tabela 15 - Tabela Regime

Quarto	
idQuarto INT NOT NULL	Chave Primária
FK_nomeHotel VARCHAR (45) NOT NULL	Chave Estrangeira (Hotel)
televisao TINYINT (1) NOT NULL	Atributo
varanda TINYINT (1) NOT NULL	Atributo
cofre TINYINT (1) NOT NULL	Atributo
FK_nomeHotel VARCHAR (45) NOT NULL	Atributo

Tabela 16 - Tabela Quarto

Há bastantes diferenças quanto ao modelo conceptual, dado que a lógica por detrás do modelo lógico é diferente da lógica do modelo conceptual. Essas diferenças deram-se por exemplo na criação de tabelas intermediárias no modelo lógico que correspondem a atributos multivalorados no modelo conceptual e também devido à necessidade de criação de uma tabela extra para cada relação N: N no modelo conceptual.

Para cada atributo composto no modelo conceptual, como no caso do atributo '*morada*', presente nas entidades 'Cliente', 'Hotel' e 'Funcionário' há a necessidade de criar uma tabela, neste caso, a tabela 'Morada', logo uma ligação N:1 entre as tabelas 'Cliente', 'Hotel' e 'Funcionário' e a tabela 'Morada', o mesmo se aplica de forma análoga à tabela 'Regime' que foi criada também por 'Regime' ser um atributo composto no modelo conceptual.

No caso de cada relação N: N no modelo conceptual, como temos por exemplo entre as entidades 'Cliente' e 'Hotel', é necessário criar uma tabela intermediária, no nosso caso a tabela 'Cliente_has_Hotel', criando duas ligações de 1: N entre cada tabela e a tabela intermediária. Esta tabela intermediária contém duas chaves estrangeiras que referenciam cada uma, a uma das tabelas 'Cliente' e 'Hotel'.

Como tabelas criadas devido a atributos compostos temos:

- Morada;
- Regime;

Como tabelas criadas devido a relações de N: N entre entidades:

- Cliente_has_Hotel;
- Reserva_has_Quarto

4.2. Normalização de dados

A normalização tem como objetivo principal fazer o isolamento dos dados de modo que os processos de inserção, remoção sejam realizados sem qualquer tipo de anomalia de forma a garantir a integridade dos dados.

Para efetuar a avaliação do modelo, verificamos se o mesmo atende a três regras conhecidas como formas normais.

A **Primeira Forma Normal (1FN)** estabelece a necessidade de que os atributos sejam atômicos, ou seja, as tabelas devem evitar a repetição de valores e não devem conter atributos multivalorados. Ao realizar uma análise das tabelas do nosso modelo, verificamos que essa condição é satisfeita, pois os atributos multivalorados são separados numa outra tabela e são também integrados através de uma chave estrangeira, como por exemplo no caso das tabelas 'Morada' e 'Regime'. É também de salientar que todas as tabelas presentes no nosso modelo contêm chaves primárias.

Para atingir a **Segunda Forma Normal (2FN)**, é necessário garantir que a tabela já esteja na Primeira Forma Normal. A Segunda Forma Normal estabelece que os atributos não-chave devem depender unicamente da chave primária da tabela de modo a evitar redundâncias e inconsistências. No nosso modelo, consideremos por exemplo, o atributo 'telefone' da tabela 'Hotel', neste caso, o atributo depende única e exclusivamente da chave primária 'nome' atribuído ao hotel. O mesmo acontece para os restantes atributos presentes no nosso modelo.

Por último, a **Terceira Forma Normal (3FN)** estipula que, uma vez cumpridas as duas formas normais anteriores, é necessário garantir que não haja dependências entre atributos, ou seja, nenhum atributo deve ser derivado de outro. No nosso modelo, após ser realizada uma análise do mesmo, verificamos que não há dependências entre atributos, logo, conseguimos com que modelo satisfaça a Terceira Forma Normal.

Com tudo isto, podemos afirmar que o modelo se encontra de acordo com as regras de normalização.

4.3. Apresentação do modelo lógico produzido

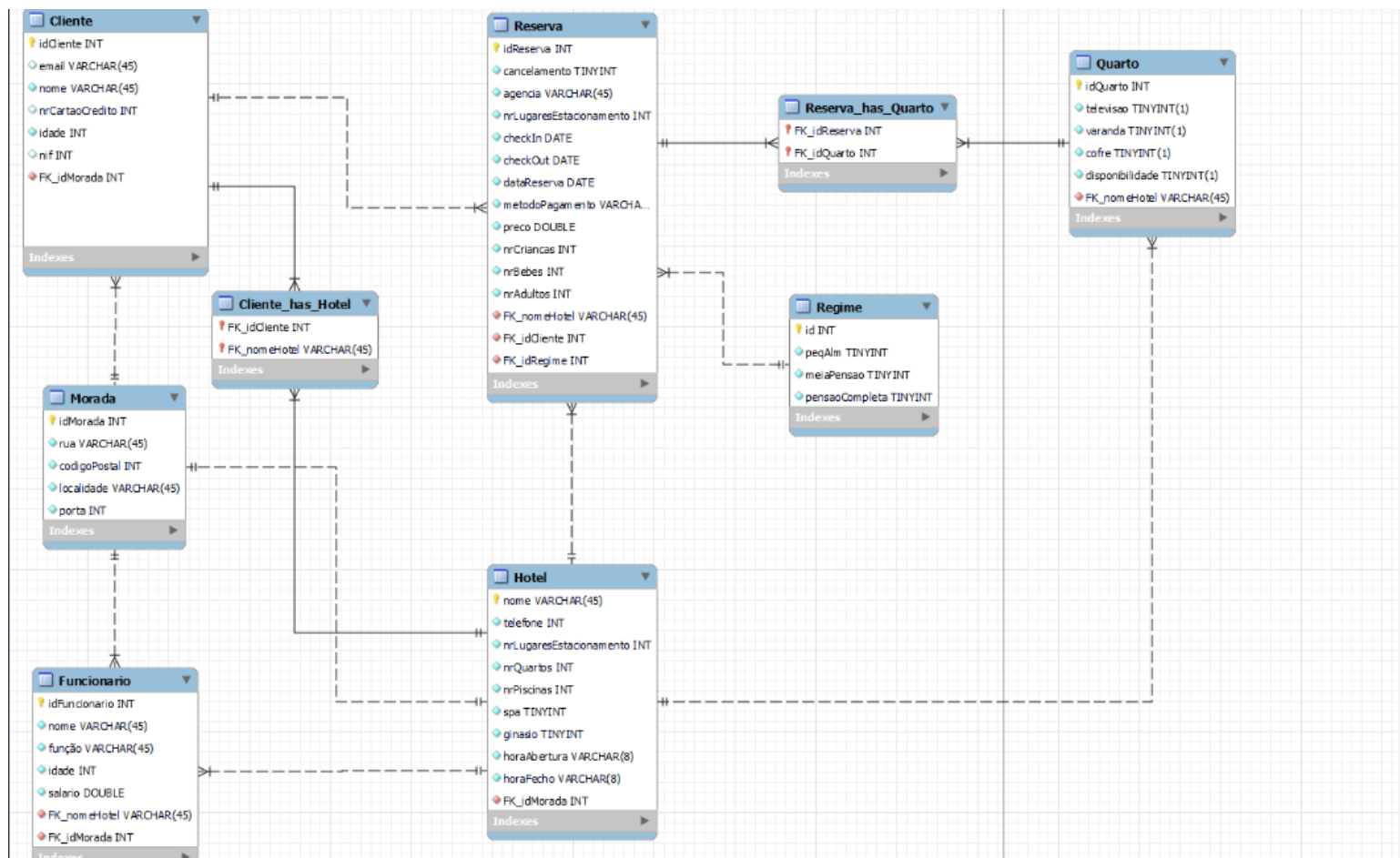


Figura 4 - Modelo Lógico

4.4. Validação do modelo com interrogações do utilizador

- **Quartos Reservados** - Apresenta todos os quartos que foram reservados dadas duas datas.
- **Agências Utilizadas** - Apresenta quantas reservas foram efetuadas em cada agência existente no sistema.
- **Informações Reserva** - Dado o id de uma reserva, apresenta as suas informações.
- **Informações Funcionário** - Dado o id de um funcionário, apresenta as suas informações.
- **Faturamento Total** - Dadas duas datas calcula o faturamento total do hotel nesse intervalo.
- **Média Estacionamentos por Reserva** - Calcula a média de estacionamentos requisitados por reserva.
- **Meses mais concorridos** - Apresenta o mês mais concorrido e o menos concorrido.
- **Duração Estadia** - Dado o id de um Cliente, apresenta todas as suas estadias com a respetiva duração.
- **Média Pessoas por Reserva** - Calcula a média de pessoas por reserva.
- **Média Salários de Funcionários** - Calcula a média dos salários dos funcionários.
- **Média Preço por Reserva** - Calcula a média dos preços de todas as reservas.
- **Média Idades de Funcionários** - Calcula a média de idade dos funcionários.
- **Média Estadia Hotel** - Dado o nome do hotel, retorna a média de dias que os hóspedes costumam ficar no mesmo.

4.5. Análise do Sistema

Com o objetivo de validar o sistema, Sérgio reuniu com a sua equipa de modo a garantir que todas as formas normais eram cumpridas, para que o modelo não prejudicasse a futura expansão do negócio, e permitisse que todos os requisitos fossem igualmente cumpridos.

5. Implementação Física

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

Como forma de traduzir o esquema lógico, foi escolhida a ferramenta MySQL através do MySQL Workbench. A criação das tabelas foi feita com ajuda que a ferramenta disponibiliza de tradução do modelo lógico para código SQL.

```
-- -----  
-- Schema Hotel  
-- -----  
  
DROP SCHEMA IF EXISTS `Hotel` ;  
  
-- -----  
-- Schema Hotel  
-- -----  
  
CREATE SCHEMA IF NOT EXISTS `Hotel` DEFAULT CHARACTER SET utf8 ;  
USE `Hotel` ;  
  
CREATE TABLE IF NOT EXISTS `Hotel`.`Morada` (  
    `idMorada` INT NOT NULL,  
    `rua` VARCHAR(45) NOT NULL,  
    `codigoPostal` INT NOT NULL,  
    `localidade` VARCHAR(45) NOT NULL,  
    `porta` INT NOT NULL,  
    PRIMARY KEY (`idMorada`));
```

Figura 5 - Criação tabela morada

```
CREATE TABLE IF NOT EXISTS `Hotel`.`Cliente` (  
  `idCliente` INT NOT NULL,  
  `email` VARCHAR(45) NULL,  
  `nome` VARCHAR(45) NOT NULL,  
  `nrCartaoCredito` INT NULL,  
  `idade` INT NOT NULL,  
  `nif` INT NULL,  
  `FK_idMorada` INT NOT NULL,  
  PRIMARY KEY (`idCliente`),  
  INDEX `fk_Cliente_Morada1_idx` (`FK_idMorada` ASC) VISIBLE,  
  CONSTRAINT `fk_Cliente_Morada1`  
    FOREIGN KEY (`FK_idMorada`)  
    REFERENCES `Hotel`.`Morada` (`idMorada`));
```

Figura 6 - Criação tabela cliente

```

CREATE TABLE IF NOT EXISTS `Hotel`.`Hotel` (
  `nome` VARCHAR(45) NOT NULL,
  `telefone` INT NOT NULL,
  `nrLugaresEstacionamento` INT NOT NULL,
  `nrQuartos` INT NOT NULL,
  `nrPiscinas` INT NOT NULL,
  `spa` TINYINT NOT NULL,
  `ginasio` TINYINT NOT NULL,
  `horaAbertura` VARCHAR(8) NOT NULL,
  `horaFecho` VARCHAR(8) NOT NULL,
  `FK_idMorada` INT NOT NULL,
  PRIMARY KEY (`nome`),
  INDEX `fk_Hotel_Morada1_idx` (`FK_idMorada` ASC) VISIBLE,
  CONSTRAINT `fk_Hotel_Morada1`
    FOREIGN KEY (`FK_idMorada`)
    REFERENCES `Hotel`.`Morada` (`idMorada`));

```

Figura 7 - Criação tabela hotel

```

CREATE TABLE IF NOT EXISTS `Hotel`.`Regime` (
  `id` INT NOT NULL,
  `peqAlm` TINYINT NOT NULL DEFAULT 0,
  `meiaPensao` TINYINT NOT NULL,
  `pensaoCompleta` TINYINT NOT NULL,
  PRIMARY KEY (`id`));

```

Figura 8 - Criação tabela regime

```

CREATE TABLE IF NOT EXISTS `Hotel`.`Reserva` (
  `idReserva` INT NOT NULL,
  `cancelamento` TINYINT NOT NULL,
  `agencia` VARCHAR(45) NOT NULL,
  `nrLugaresEstacionamento` INT NOT NULL,
  `checkIn` DATE NOT NULL,
  `checkOut` DATE NOT NULL,
  `dataReserva` DATE NOT NULL,
  `metodoPagamento` VARCHAR(45) NOT NULL,
  `preco` DOUBLE NOT NULL,
  `nrAdultos` INT NOT NULL,
  `nrCrianças` INT NOT NULL,
  `nrBebes` INT NOT NULL,
  `FK_nomeHotel` VARCHAR(45) NOT NULL,
  `FK_idCliente` INT NOT NULL,
  `FK_idRegime` INT NOT NULL,

```

Figura 9 - Criação tabela reserva (1ªparte)

```

PRIMARY KEY (`idReserva`),
INDEX `fk_Reserva_Cliente1_idx` (`FK_idCliente` ASC) VISIBLE,
INDEX `fk_Reserva_Hotel1_idx` (`FK_nomeHotel` ASC) VISIBLE,
INDEX `fk_Reserva_Regime1_idx` (`FK_idRegime` ASC) VISIBLE,
CONSTRAINT `fk_Reserva_Cliente1`
  FOREIGN KEY (`FK_idCliente`)
  REFERENCES `Hotel`.`Cliente` (`idCliente`),
CONSTRAINT `fk_Reserva_Hotel1`
  FOREIGN KEY (`FK_nomeHotel`)
  REFERENCES `Hotel`.`Hotel` (`nome`),
CONSTRAINT `fk_Reserva_Regime1`
  FOREIGN KEY (`FK_idRegime`)
  REFERENCES `Hotel`.`Regime` (`id`));

```

Figura 10 - Criação tabela reserva (2ªparte)

```

CREATE TABLE IF NOT EXISTS `Hotel`.`Funcionario` (
  `idFuncionario` INT NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `função` VARCHAR(45) NOT NULL,
  `idade` INT NOT NULL,
  `salario` DOUBLE NOT NULL,
  `FK_nomeHotel` VARCHAR(45) NOT NULL,
  `FK_idMorada` INT NOT NULL,
  PRIMARY KEY (`idFuncionario`),
  INDEX `fk_Funcionario_Hotel1_idx` (`FK_nomeHotel` ASC) VISIBLE,
  INDEX `fk_Funcionario_Morada1_idx` (`FK_idMorada` ASC) VISIBLE,
  CONSTRAINT `fk_Funcionario_Hotel1`
    FOREIGN KEY (`FK_nomeHotel`)
      REFERENCES `Hotel`.`Hotel` (`nome`),
  CONSTRAINT `fk_Funcionario_Morada1`
    FOREIGN KEY (`FK_idMorada`)
      REFERENCES `Hotel`.`Morada` (`idMorada`));

```

Figura 11 - Criação tabela funcionário


```

CREATE TABLE IF NOT EXISTS `Hotel`.`Quarto` (
  `idQuarto` INT NOT NULL,
  `televisao` TINYINT(1) NOT NULL,
  `varanda` TINYINT(1) NOT NULL,
  `cofre` TINYINT(1) NOT NULL,
  `disponibilidade` TINYINT(1) NOT NULL,
  `FK_nomeHotel` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idQuarto`),
  INDEX `fk_Quarto_Hotel1_idx` (`FK_nomeHotel` ASC) VISIBLE,
  CONSTRAINT `fk_Quarto_Hotel1`
    FOREIGN KEY (`FK_nomeHotel`)
    REFERENCES `Hotel`.`Hotel` (`nome`));

```

Figura 12 - Criação tabela quarto

```

CREATE TABLE IF NOT EXISTS `Hotel`.`Cliente_has_Hotel` (
  `FK_idCliente` INT NOT NULL,
  `FK_nomeHotel` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`FK_idCliente`, `FK_nomeHotel`),
  INDEX `fk_Cliente_has_Hotel_Hotel1_idx` (`FK_nomeHotel` ASC) VISIBLE,
  INDEX `fk_Cliente_has_Hotel_Cliente1_idx` (`FK_idCliente` ASC) VISIBLE,
  CONSTRAINT `fk_Cliente_has_Hotel_Cliente1`
    FOREIGN KEY (`FK_idCliente`)
    REFERENCES `Hotel`.`Cliente` (`idCliente`),
  CONSTRAINT `fk_Cliente_has_Hotel_Hotel1`
    FOREIGN KEY (`FK_nomeHotel`)
    REFERENCES `Hotel`.`Hotel` (`nome`));

```

Figura 13 - Criação tabela relação cliente-hotel

```

CREATE TABLE IF NOT EXISTS `Hotel`.`Reserva_has_Quarto` (
  `FK_idReserva` INT NOT NULL,
  `FK_idQuarto` INT NOT NULL,
  PRIMARY KEY (`FK_idReserva`, `FK_idQuarto`),
  INDEX `fk_Reserva_has_Quarto_Quarto1_idx` (`FK_idQuarto` ASC) VISIBLE,
  INDEX `fk_Reserva_has_Quarto_Reserva1_idx` (`FK_idReserva` ASC) VISIBLE,
  CONSTRAINT `fk_Reserva_has_Quarto_Reserva1`
    FOREIGN KEY (`FK_idReserva`)
      REFERENCES `Hotel`.`Reserva` (`idReserva`),
  CONSTRAINT `fk_Reserva_has_Quarto_Quarto1`
    FOREIGN KEY (`FK_idQuarto`)
      REFERENCES `Hotel`.`Quarto` (`idQuarto`));

```

Figura 14 - Criação tabela relação reserva-quarto

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

```

Delimiter $$
Create procedure agencias_utilizadas()
BEGIN
  SELECT agencia, COUNT(*) AS total_utilizacoes
  FROM Reserva
  GROUP BY agencia;
End $$

Delimiter ;

```

Figura 15 - Agências Utilizadas

```

DELIMITER //
CREATE FUNCTION calcularFaturamentoCadeiaHoteleira(dataInicio DATE, dataFim DATE)
RETURNS DOUBLE
BEGIN
    DECLARE faturamento DOUBLE;

    SELECT SUM(preco) INTO faturamento
    FROM reserva
    WHERE dataReserva BETWEEN dataInicio AND dataFim;

    RETURN faturamento;
END //
DELIMITER ;

```

Figura 16 - Faturação da Cadeia Hoteleira

```

DELIMITER //
CREATE PROCEDURE obterInformacoesFuncionario(numeroFuncionario INT)
BEGIN
    SELECT *
    FROM Funcionario
    WHERE idFuncionario = numeroFuncionario;
END //
DELIMITER ;

```

Figura 17 - Informações através do ID

```

DELIMITER //
CREATE FUNCTION calcularMediaPessoasPorReserva()
RETURNS DECIMAL(10,2)
BEGIN
    DECLARE mediaPessoas DECIMAL(10,2);

    SELECT AVG(nrCrianças + nrBebes + nrAdultos) INTO mediaPessoas
    FROM Reserva;

    RETURN mediaPessoas;
END //
DELIMITER ;

```

Figura 18 - Média de Pessoas por Reserva

```

DELIMITER //
CREATE FUNCTION calcularMediaSalariosFuncionarios()
RETURNS DECIMAL(10,2)
BEGIN
    DECLARE mediaSalarios DECIMAL(10,2);

    SELECT AVG(salario) INTO mediaSalarios
    FROM Funcionario;

    RETURN mediaSalarios;
END //
DELIMITER ;

```

Figura 19 - Média de salários dos Funcionários

5.3. Definição e caracterização das vistas de utilização em SQL

Criação das vistas

```

CREATE VIEW reserva_details_view AS
SELECT Reserva.idReserva, cliente.nome AS cliente_nome, Cliente.email
AS cliente_email, Regime.peqAlm
AS Pequeno_Almoço, Regime.meiaPensao
AS Meia_pensao, Regime.pensaoCompleta
AS Pensao_Completa, Reserva_has_Quarto.FK_idQuarto
AS id_Quarto, Quarto.FK_nomeHotel
FROM Reserva
JOIN Cliente
ON Reserva.FK_idCliente = Cliente.idCliente
JOIN Regime
ON Reserva.FK_idRegime = Regime.id
JOIN Reserva_has_Quarto
ON Reserva.idReserva = Reserva_has_Quarto.FK_idReserva
JOIN Quarto
ON Reserva_has_Quarto.FK_idQuarto = Quarto.idQuarto;

```

Figura 20 - Criação vista reserva

```

CREATE VIEW funcionario_informacoes_view AS
SELECT funcionario.nome AS funcionario_nome, funcionario.função, funcionario.salario, funcionario.FK_nomeHotel
AS hotel_nome, Morada.rua, Morada.porta, Morada.codigoPostal, Morada.localidade
FROM Funcionario
JOIN Morada
ON Funcionario.FK_idMorada = Morada.idMorada;

```

Figura 21 – Criação vista funcionário

Criação

```
-- rececionistas
CREATE USER 'José Pinto'@'localhost'
  IDENTIFIED BY 'passJosePinto';

CREATE USER 'Filipa Marrão'@'localhost'
  IDENTIFIED BY 'passFilipaMarrao';
```

Figura 22 - Criação utilizador para Rececionistas

```
-- chefes departamento
CREATE USER 'Joaquim Paiva'@'localhost'
  IDENTIFIED BY 'passJoaquimPaiva';

CREATE USER 'Pedro Vaz'@'localhost'
  IDENTIFIED BY 'passPedroVaz';
```

Figura 23 - Criação utilizador para Chefes de Departamento

```
-- administradores
CREATE USER 'Rui Pinto'@'12.0.1.2'
  IDENTIFIED BY 'passRuiPinto';

CREATE USER 'António Silva'@'17.0.5.4'
  IDENTIFIED BY 'passAntonioSilva';
```

Figura 24 - Criação utilizador para Administradores

Permissões

```
-- rececionistas conseguem ter acesso às informações de um cliente
grant select (idCliente,email,nome,nrCartaoCredito,idade,nif,FK_idMorada)
  on hotel.cliente
  to 'José Pinto'@'localhost';

grant select (idCliente,email,nome,nrCartaoCredito,idade,nif,FK_idMorada)
  on hotel.cliente
  to 'Filipa Marrão'@'localhost';
```

```

-- rececionistas conseguem registar um novo cliente
grant insert (idCliente,email,nome,nrCartaoCredito,idade,nif,FK_idMorada)
on hotel.cliente
to 'José Pinto'@'localhost';

grant insert (idCliente,email,nome,nrCartaoCredito,idade,nif,FK_idMorada)
on hotel.cliente
to 'Filipa Marrão'@'localhost';

-- rececionistas podem consultar e adicionar registos de reservas ao sistema
grant insert (idReserva,cancelamento,agencia,nrLugaresEstacionamento,checkIn,checkOut,dataReserva,metodoPagamento,preco,nrAdultos,nrCrianças,nrBebes,FK_nomeHotel,FK_idCliente,FK_idRegime)
on hotel.reserva
to 'José Pinto'@'localhost';

grant insert (idReserva,cancelamento,agencia,nrLugaresEstacionamento,checkIn,checkOut,dataReserva,metodoPagamento,preco,nrAdultos,nrCrianças,nrBebes,FK_nomeHotel,FK_idCliente,FK_idRegime)
on hotel.reserva
to 'Filipa Marrão'@'localhost';

grant select (idReserva,cancelamento,agencia,nrLugaresEstacionamento,checkIn,checkOut,dataReserva,metodoPagamento,preco,nrAdultos,nrCrianças,nrBebes,FK_nomeHotel,FK_idCliente,FK_idRegime)
on hotel.reserva
to 'José Pinto'@'localhost';

grant select (idReserva,cancelamento,agencia,nrLugaresEstacionamento,checkIn,checkOut,dataReserva,metodoPagamento,preco,nrAdultos,nrCrianças,nrBebes,FK_nomeHotel,FK_idCliente,FK_idRegime)
on hotel.reserva
to 'Filipa Marrão'@'localhost';

```

Figura 25 - Criação permissões para Rececionistas

```

-- chefes de departamento podem alterar informações da reserva
grant update (idReserva,cancelamento,agencia,nrLugaresEstacionamento,checkIn,checkOut,dataReserva,metodoPagamento,preco,nrAdultos,nrCrianças,nrBebes,FK_nomeHotel,FK_idCliente,FK_idRegime)
on hotel.reserva
to 'Joaquim Paiva'@'localhost';

grant update (idReserva,cancelamento,agencia,nrLugaresEstacionamento,checkIn,checkOut,dataReserva,metodoPagamento,preco,nrAdultos,nrCrianças,nrBebes,FK_nomeHotel,FK_idCliente,FK_idRegime)
on hotel.reserva
to 'Pedro Vaz'@'localhost';

```

Figura 26 - Criação permissões para Chefes de Departamento

```

-- administradores têm todas as permissões
grant all privileges
on hotel.*
to 'Rui Pinto'@'12.0.1.2';

grant all privileges
on hotel.*
to 'António Silva'@'17.0.5.4';

```

Figura 27 - Criação permissões para Administradores

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

A base de dados tem um tamanho de 336.0 KiB e o crescimento anual da mesma vai variar conforme o número de reservas e o número de clientes registados na mesma.

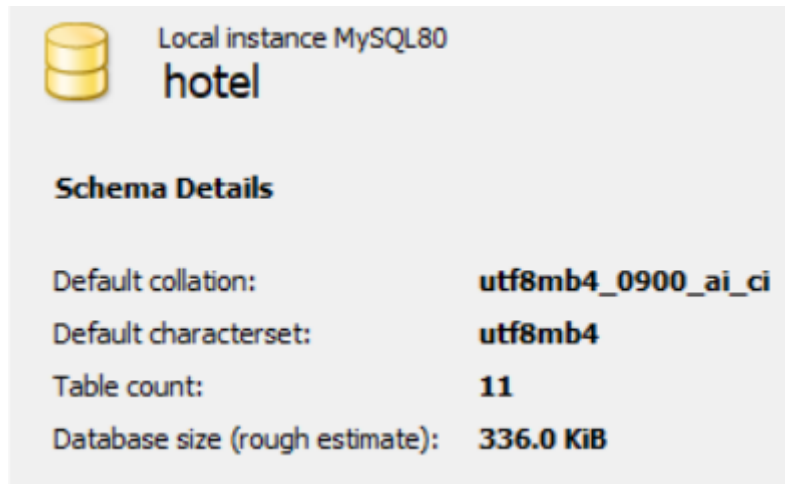


Figura 28 - Tamanho da base de dados

5.5. Plano de segurança e recuperação de dados

O plano de segurança em questão que foi implementado para prevenir qualquer tipo de desastre relacionado à perda de dados da base de dados. Desta forma podemos sempre ter um backUP que nos permite voltar à versão em que os dados não foram adulterados e invalidados. No fim de cada dia de trabalho, foi sugerido pela equipa de Sérgio à equipa do hotel que executasse sempre este comando, para que o backup fosse efetuado diariamente.

```
C:\Users\rfern\OneDrive\Ambiente de Trabalho\Uni projects\BD>"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqldump.exe" --user=root --password=ruipinto2023 --host=localhost --port=3306 --result-file="C:\Users\rfern\OneDrive\Ambiente de Trabalho\Uni projects\BD\backup\backup.sql" hotel
```

Figura 29 - Comando recuperação de dados

6. Implementação do Sistema de Recolha de Dados

6.1. Apresentação e modelo do sistema

Inicialmente a base de dados foi povoada com os dados que a cadeia hoteleira tinha armazenados em livros, acerca de cada entidade identificada nos parâmetros anteriores. Além disto foi criado utilizando a linguagem python, um script que permite que após isto a recolha de dados seja feita automaticamente, bastando apenas acrescentar a informação que se pretende adicionar à base de dados a um ficheiro JSON, e executar o script.

6.2. Implementação do sistema de recolha

```

scriptPovoamento.py > ...
1  import csv
2  import mysql.connector
3
4  cnx = mysql.connector.connect(host='127.0.0.1', user='root', password='ruipinto2023', database='hotel')
5  cursor = cnx.cursor()
6
7  def povoar_tabela(nome_tabela, nome_arquivo_csv):
8
9      with open(nome_arquivo_csv, 'r') as file:
10         csv_data = csv.reader(file)
11         next(csv_data)
12
13         for row in csv_data:
14             values = [f'"{value}"' if isinstance(value, str) else str(value) for value in row]
15
16             query = f"INSERT INTO {nome_tabela} VALUES ({', '.join(values)})"
17             cursor.execute(query)
18
19     print(f'Dados inseridos na tabela {nome_tabela} com sucesso!')
20
21     povoar_tabela('Cliente', 'C:\\Users\\rfern\\OneDrive\\Ambiente de Trabalho\\Uni projects\\BD\\clientes.csv')
22
23     cnx.commit()
24     cursor.close()
25     cnx.close()

```

Figura 30 - Script de recolha

6.3. Funcionamento do sistema

O sistema funcionaria então da seguinte forma: sempre que se deseje acrescentar uma informação relativa a uma tabela da base de dados, deve ser feita a escrita no ficheiro JSON correspondente à tabela em questão, por exemplo imaginando que se queira adicionar uma reserva, coloca-se todas as informações relativas à tabela das reservas, e depois é executado o script. Assim que o script é executado, a tabela é atualizada com a informação adicionada e assim a recolha dos novos dados é efetuada com sucesso.

7. Implementação do Sistema de Painéis de Análise

Como forma de tornar mais fácil a análise dos dados inseridos na base de dados, foi implementado um sistema de Painéis de Análise com a ajuda da ferramenta PowerBI. Com a qual a base de dados foi conectada para que houvesse esta recolha automática.

7.1. Definição e caracterização da vista de dados para análise

Foi apenas criada uma visão, a visão do administrador, que comporta todas as informações que a equipa de implementação, juntamente com o Sr. Libério, julgaram importantes ter sempre em destaque. Destas destacamos, o salário que está a ser gasto atualmente em funcionários, a faturação total que a cadeia hoteleira tem no momento e os meses em que o hotel é mais e menos requisitado.

7.2. Povoamento das estruturas de dados para análise

Primeiramente foi povoada a base de dados referida anteriormente, e de seguida efetuou-se a conexão entre o Server MySQL com o powerBI, para que este tivesse acesso a todos os dados introduzidos na

base de dados em questão e de forma a que sempre que a base de dados for atualizada, o Painel de Análise também recebe estas atualizações.

7.3. Apresentação e caracterização do dashboard implementado

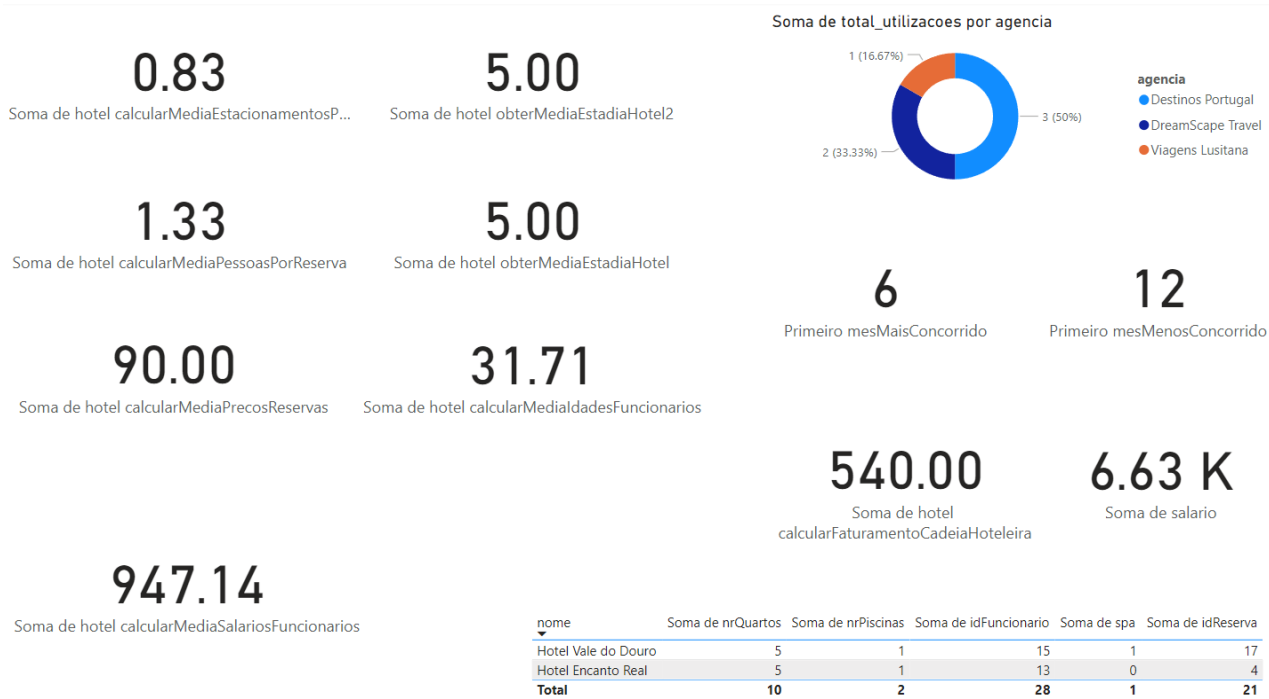


Figura 31 - Dashboard Administrador

8. Conclusões e Trabalho Futuro

Finalmente, após muito trabalho, o Sr. Libério foi obrigado a concordar, que realmente a introdução das tecnologias providenciadas pelo seu primo Sérgio ajudaram bastante a melhorar a gestão do seu negócio e automatizar todos os processos pelos quais a cadeia hoteleira tem de passar no seu quotidiano.

Como forma de melhoria do sistema implementado nesta cadeia hoteleira, Sérgio juntamente com a sua equipa, definiram que ainda havia muitas melhorias passíveis de serem feitas, das quais se destacam: a criação de uma aplicação que facilitasse a escrita no ficheiro JSON, para que os funcionários não se tivessem de preocupar com a sintaxe complexa que o ficheiro JSON possui. E além disso esta aplicação também seria responsável na automatização de toda a recolha de dados feita na base de dados, que atualmente não funciona na totalidade, já que cada vez que se quer inserir dados no sistema é necessário executar manualmente o script.

Ficou também por fazer a criação de mais vistas da dashboard, por exemplo uma vista para cada função dos funcionários, outra a gestão financeira e por exemplo outra para gestão dos quartos. Aliado a estas vistas, estaria a criação de várias funções que respondessem às questões que cada vista necessitaria para um funcionamento mais fluido de todas as operações.