



Universidade do Minho
Licenciatura em Engenharia Informática

Unidade Curricular de Bases de Dados

Ano Letivo de 2022/2023

BIBLIOTECA MARIA JUDITE

Ana Lopes(a10834)

Ema Martins(a97678)

Gonçalo Costa(a100824)

Henrique Malheiro(a97455)

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

Biblioteca Maria Judite

Ana Lopes a10834

Ema Martins a97678

Gonçalo Costa a100824

Henrique Malheiro a97455

Resumo

Este trabalho tem como objetivo interligar todos os pontos abordados na unidade curricular de Base de Dados, desde o planeamento à implementação de um sistema de gestão de base de dados.

O tema do nosso trabalho foca-se numa biblioteca com dificuldades na organização dos seus livros e salas de estudo, pois os seus sistemas de gestão eram muito arcaicos. Assim, desenvolveu-se um sistema de gestão de base de dados que pretende registar os funcionários da biblioteca, leitores e livros. Tem ainda como intuito fornecer informações sobre as salas de estudo disponíveis, livros reservados, livros em que a entrega esteja atrasada, entre outras funcionalidades.

Neste relatório são exibidas as várias fases do trabalho. Assim, primeiramente fez-se a contextualização e fundamentação, seguindo-se do levantamento de requisitos e organização dos mesmos. Com estas informações, foi possível criar o modelo conceptual. A partir deste, foi feito o modelo logico seguindo as regras de conversão abordadas na UC. Implementado em mySQL, criou-se uma base de dados. Fez-se ainda um script com possíveis manipulações da mesma.

Área de Aplicação: Base de dados, Sistemas de Bases de Dados, Informática, Desenvolvimento de software

Palavras-Chave: Modelo relacional, SQL, Entidades, Relacionamentos, Atributos, Base de Dados, Sistemas de Gestão de Base de Dados, Modelo Logico, Modelo Conceptual, Requisitos

Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	v
1. Definição do Sistema	1
1.1. Contexto de Aplicação e Fundamentação do Sistema	1
1.2. Motivação e Objetivos do Trabalho	1
1.3. Análise de Viabilidade do Processo	2
1.4. Recursos e Equipa de Trabalho	2
1.5. Plano de Execução do Projeto	3
1.6. Próximos Capítulos	3
2. Levantamento e Análise de Requisitos	4
2.1. Método de Levantamento e de Análise de Requisitos Adotados	4
2.2. Organização dos Requisitos Levantados	5
2.2.1. Requisitos de Descrição	6
2.2.2. Requisitos de Exploração	7
2.2.3. Requisitos de Controlo	8
2.1. Análise e Validação Geral dos Requisitos	8
3. Modelo Conceptual	9
3.1. Apresentação da Abordagem de Modelação Realizada	9
3.2. Identificação e Caracterização das Entidades	9
3.3. Identificação e Caracterização dos Relacionamentos	12
3.4. Identificação e Caracterização da Associação dos Atributos com as Entidades e Relacioname	13
3.5. Apresentação e Explicação do diagrama ER produzido	15
4. Modelação Lógica	16
4.1. Construção e Validação do Modelo de Dados Lógico	16
4.2. Normalização de Dados	17
4.3. Apresentação e explicação do modelo lógico produzido	18
4.4. A Validação do modelo com interrogações do utilizado	19
5. Implementação Física	20
5.1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	21

5.2	Tradução das interrogações do utilizador para SQL	22
5.3	Definição e caracterização das vistas de utilização em SQL	25
5.4	Indexação do sistema de dados	27
5.5	Procedimentos Implementados	28
5.6	Plano de segurança e recuperação de dados	29
6.	Implementação de um Sistema de Recolha de Dados	30
6.1.	Apresentação e modelo do sistema	31
7.	Implementação do sistema de painéis de análise	33
7.1	Definição e caracterização da vista de dados para análise	34
7.2	Povoamento das estruturas de dados para análise	35
7.3	Apresentação e caracterização dos dashboards implementados	36
	Conclusões e Trabalho Futuro	38
	Referências	39
	Anexos	40
I.	Anexo 1 – Script da Implementação Física	41
II.	Anexo 2 – Script do Povoamento da Base de Dados	47
III.	Anexo 3 – Script das Queries	50
IV.	Anexo 4 – Script das Restrições	61

Índice de Figuras

Figura 1 – Diagrama de Gent	3
Figura 2 – Inquérito realizado aos leitores da Biblioteca Maria Judite	5
Figura 3 – Modelo Conceptual	15
Figura 4 – Modelo Lógico	18
Figura 5 – Criação da Tabela Funcionário	22
Figura 6 – Função para saber o local de um certo Livro	23
Figura 7 – Query dos generos de livros mais existente	23
Figura 8 – Procedimento para alugar livros	24
Figura 9 – Query para verificar o género mais verificado	24
Figura 10 – Restrições Funcionário	25
Figura 11 – Restrições Leitor	25
Figura 12 – Vistas	26
Figura 13 – Procedimento para alugar uma sala de estudo	28
Figura 14 – Procedimento para adicionar um Leitor	29
Figura 15 – Implementação do Povoamento da Base de Dados	32
Figura 16 – Exemplo de conexão entre a Base de Dados e o Power BI	35
Figura 17 – Gráfico das somas de livros disponíveis e de livros Empréstados	36
Figura 18 – Gráfico das somas de salas de estudo livres e ocupadas	37

Índice de Tabelas

Tabela 1 - Requisitos de Descrição	6
Tabela 2 - Requisitos de Manipulação	7
Tabela 3 - Requisitos de Controlo	8
Tabela 4 - Entidades	11
Tabela 5 - Relacionamentos	13
Tabela 6 - Atributos	14

1. Definição do Sistema

Neste capítulo começa-se por falar do percurso da D. Francisca até esta ficar encarregue da biblioteca de Lousada. Fala-se ainda de alguns problemas que a biblioteca tem e de uma possível abordagem de solução dos mesmos, utilizando um sistema de gestão de base de dados. São ainda salientados os objetivos que se pretende alcançar com a sua implementação.

1.1. Contexto de Aplicação e Fundamentação do Sistema

Desde pequena que para Francisca de Andrade a biblioteca era o seu santuário. Adorava tanto ler como criar histórias. Viveu sempre rodeada de livros, muito por influência do seu avô que trabalhava numa biblioteca.

Contudo, mais velha, apercebeu-se que os livros não são acessíveis a todos. Seguindo o exemplo do seu avô, decidiu que queria trabalhar numa biblioteca e a ajudar a desenvolvê-la, de forma que a literatura chegasse cada vez a mais pessoas. Resolveu estudar Biblioteconomia e Administração Pública, duas áreas que a ajudariam a entender como gerir uma biblioteca e torná-la aconchegante. Com tal conhecimento, poderia transformar uma biblioteca num espaço moderno, inclusivo e cheio de oportunidades de aprendizagem para todas as pessoas da comunidade.

Foi com estes conhecimentos que Francisca ganhou um concurso da câmara municipal ficando assim encarregue da biblioteca municipal de Lousada, sua terra natal. As ideias visionárias da Francisca destacam-se e, desde a tomada de posse, o movimento da biblioteca tem aumentado de forma exponencial.

A fachada da biblioteca assemelha-se à dos edifícios do século XVI. Localiza-se na praça D. Rebelo Gomes, local muito conhecido por ter a histórica estátua de Camões. O edifício é constituído por três pisos. O primeiro piso é dedicado à leitura, onde se encontram os livros e alguns bancos e puffs para os leitores. O segundo piso é dedicado a salas de estudo individuais e coletivas, assim como, uma área reservada para computadores. O terceiro é uma arrecadação.

Apesar das ideias inovadoras da D. Francisca, a biblioteca é alvo de algumas queixas, nomeadamente por parte dos seus leitores, uma vez que acontecem frequentemente erros ligados às informações da biblioteca.

A gestão da biblioteca é feita através de um antigo sistema informático. Esse sistema apenas se encontra apto para registar os empréstimos dos livros e, mesmo assim, já não segue as novas regras implementadas na biblioteca.

A gestão das salas, individuais e coletivas, e da área de computadores é realizada em formato físico, tornando a reserva e controle um processo demorado e ineficaz.

1.2. Motivação e Objetivos do Trabalho

Com o número de empréstimos de livros a aumentar exponencialmente e o prazo dos empréstimos aumentados, o sistema de gestão de base de dados acabou por ficar desatualizado. Apesar

de não haver registo de perda de livros nem de documentos, é uma situação que a D. Francisca teme que aconteça.

Nas salas de estudo, além da reserva e o check in serem demorados, já houve conflitos por uma sala ter sido reservada por pessoas diferentes ao mesmo tempo.

Para e prevenir situações constrangedoras, consequentes da falta do registo adequado de dados das atividades realizadas na biblioteca, a D. Francisca, com base na sua formação académica, decidiu implementar um pequeno sistema que lhe permite guardar atempadamente toda a informação que necessita para fazer a melhor gestão dos recursos e controlo no quotidiano da biblioteca. Definiu um conjunto de objetivos que pretende alcançar com a implementação do futuro sistema da sua biblioteca, nomeadamente:

- Melhorar a sua capacidade de gestão de recursos e registo das atividades da biblioteca;
- Saber os documentos que foram emprestados e os que estão disponíveis;
- Melhorar o nível de controle sobre os empréstimos, evitando perdas de documentos;
- Saber, a cada momento, que salas estão a ser utilizadas/reservadas/disponíveis.

E ainda:

- Conhecer melhor os seus leitores, através da implementação de mecanismos de análise da participação na biblioteca e de profiling de leitor;
- Conhecer as preferências da comunidade, como gênero, autor;
- Melhorar a qualidade do serviço da biblioteca.

1.3. Análise de Viabilidade do Processo

A D. Francisca acredita que com um meio mais eficaz consiga fazer com que não falem registos de empréstimos, salas bem como de leitores e funcionários. Conseguirá ainda:

- Saber, a cada momento, o que cada leitor requisitou;
- Saber quais os leitores que tem livros atrasados;
- Aumentar, logo no primeiro mês, cerca de 10% no movimento da biblioteca – uma vez que à uma maior facilidade de reversa das salas de estudo e na requisição dos documentos.

E ainda:

- Conhecer os horários de maior movimento;
- Conhecer os hábitos dos seus leitores.

1.4. Recursos e Equipa de Trabalho

Para tal implementação serão necessários recursos humanos como o pessoal da biblioteca, leitores e funcionários da empresa de desenvolvimento do sistema de gestão de base de dados. Serão ainda necessários recursos materiais de hardware e software.

A equipa de trabalho é composta pelo pessoal interno como a D. Francisca, a S. Maria Salgado e o Sr. José Amaro. Estes estarão encarregues do funcionamento da biblioteca, validação de serviços e atendimento dos leitores.

Juntamente, por uma equipa externa, composta pelo arquiteto e pelo engenheiro de base de dados da empresa a contratar, que farão o levantamento de requisitos, modelação e implementação do sistema de base de dados.

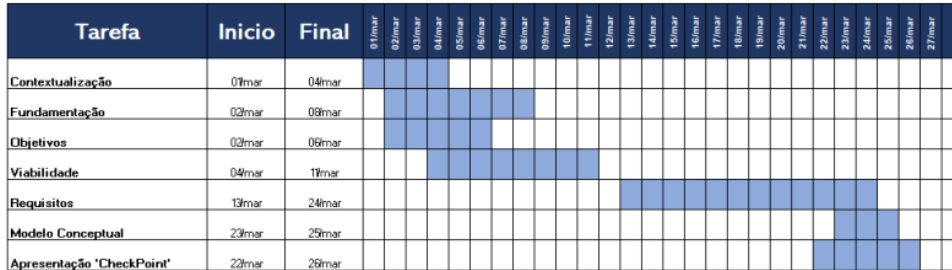
Foram ainda selecionados alguns leitores para responder inquéritos de opinião relacionados com a validação dos serviços da biblioteca.

1.5. Plano de Execução do Projeto

De forma a facilitar o planeamento da execução do projeto, foi realizado um diagrama de Gant.

DIAGRAMA DE GANTT

Primeira Parte



Segunda Parte

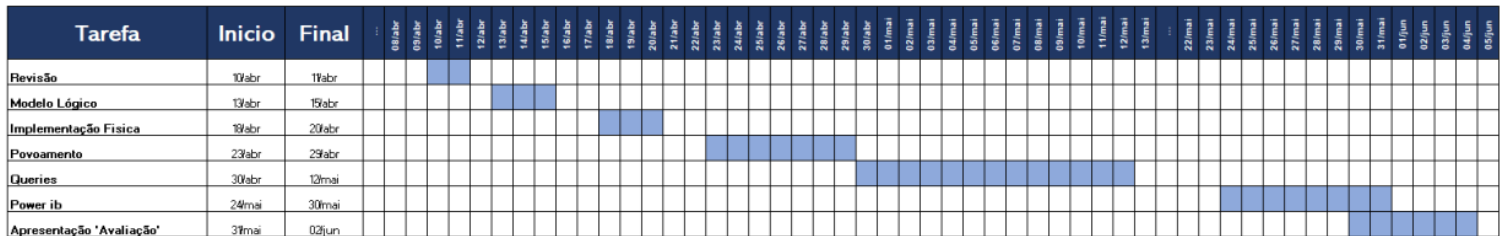


Figura 1 - Diagrama de Gant

1.6. Próximos Capítulos

Nos próximos capítulos será mostrada o processo de implementação do sistema de base de dados, começando pela modelação conceptual, passando para a modelação logica e terminando com a implementação física. Juntamente será mostrada como foi implementada as vistas de utilização e mecanismos de backup.

2. Levantamento e Análise de Requisitos

Nesta parte do relatório serão abordadas as diferentes formas de obtenção de requisitos. Após a sua recolha e análise, foram agrupados de acordo com o seu tipo, de forma a facilitar a implementação do modelo conceptual.

2.1. Método de Levantamento e de Análise de Requisitos Adotados

Utilizaram-se diversas estratégias para o levantamento de requisitos. Um deles foi a realização de inquéritos de opinião e de dados estatísticos. Com este recolheu-se informação sobre os horários de maior frequência como as áreas mais frequentadas, com o intuito de aprimorar as nossas estratégias nesses períodos. Recolheu-se ainda informação de possíveis constrangimentos que os leitores passaram de forma a procurar causas e posteriormente soluções para os mesmos.

Inquérito de Opinião
Este inquérito tem como objetivo recolher as opiniões dos leitores da Biblioteca Maria Judite

Quais as áreas que costuma mais utilizar: *

☐ Área de Estudo
☐ Área de leitura

A frequência que costuma utilizar essas áreas *

1 2 3 4 5 6
Apenas 1 dia ☐ ☐ ☐ ☐ ☐ ☐ Todos os dias

Já passou por algum constrangimento? *

☐ Sim
☐ Não

Se sim, qual? *

A sua resposta _____

Enviar Limpar formulário

Figura 2 - Inquérito realizado aos leitores da Biblioteca Maria Judite

Frequentemente um analista fez idas à biblioteca, nomeadamente nos horários mais frequentes recolhidos nos inquéritos, para observar o funcionamento e assim identificar possíveis novas formas de organização da biblioteca. Ainda foi analisado o antigo sistema de base de dados, utilizado pela biblioteca de forma a perceber as melhorias que poderiam ser implementadas nele e também tentar ajustar as estratégias de modo que o funcionamento da biblioteca não sege mudado na sua totalidade. Foram feitas entrevistas tanto a leitores, tentando dar maior foco nos mais frequentes, como a funcionários da biblioteca para perceber as suas opiniões acerca da organização e recolher sugestões de possíveis melhorias. Como planeado, foram realizadas reuniões e estas por sua vez foram complementadas pela análise da documentação da biblioteca.

2.2. Organização dos Requisitos Levantados

À medida que a informação foi recolhida e registada num documento. Esse documento foi revisto para evitar erros e inconsistências e mais tarde os requisitos foram agrupados em 3 categorias.

2.2.1.Requisitos de Descrição

Descrição				
Nr	Data e Hora	Descrição	Área	Analista
1	28/02/2023 12:05H	A cada leitor é lhe atribuído um id.	Biblioteca	D. Francisca
2	28/02/2023 12:05H	Para cada leitor é necessário guardar o nome, email, nº telemóvel, nº de cartão de cidadão e endereço.	Biblioteca	D. Francisca
3	28/02/2023 12:05H	Livro é caracterizado por um id único, autor, género e seção.	Biblioteca	D. Francisca
4	28/02/2023 12:05H	As salas de estudo são caracterizadas pelo id, estado, tipo e capacidade.	Biblioteca	D. Francisca
5	28/02/2023 12:05H	Os funcionários são caracterizados por um id único, nome, endereço, email, nº telemóvel.	Biblioteca	D. Francisca
6	05/03/2023 16:30H	Uma seção é composta por sala, corredor, estante e prateleira.	Biblioteca	D. Francisca
7	08/03/2023 12:32H	O tipo da sala de estudo pode ser individual ou grupo.	Biblioteca	D. Francisca
8	08/03/2023 12:32H	O estado das salas de estudo pode ser ocupado ou livre.	Biblioteca	D. Francisca
9	08/03/2023 12:32H	Guardar data no aluguer de um livro	Biblioteca	D. Francisca
10	08/03/2023 12:32H	Guardar o horário (dia e hora) da reserva de uma sala de estudo	Biblioteca	D. Francisca
11	08/03/2023 12:32H	O horário onde a biblioteca é mais frequentada é dado pelas horas de mais reservas de salas.	Biblioteca	D. Francisca
12	23/03/2023 18:40H	Na reserva de uma sala de estudo, tem de ser especificado o tipo (individual/grupo).	Biblioteca	D. Francisca

Tabela 1 - Requisitos de Descrição

2.2.2.Requisitos de Exploração

Manipulação				
Nr	Data e Hora	Descrição	Área	Analista
1	02/03/2023 19:00H	A qualquer momento dá para ver quantos livros estão emprestados/disponíveis/entrega atrasada.	Biblioteca	D. Francisca
2	02/03/2023 19:00H	A qualquer momento dá para ver quantos leitores estão a usar as salas de estudo e o tipo da sala (individual/de grupo).	Biblioteca	D. Francisca
3	05/03/2023 16:30H	Ver em que seção se encontra um determinado livro.	Biblioteca	D. Francisca
4	05/03/2023 16:30H	Ver os gêneros mais requisitados pelos leitores.	Biblioteca	D. Francisca
5	05/03/2023 16:30H	Ver o horário onde a biblioteca é mais frequentada pelos leitores.	Biblioteca	D. Francisca
6	05/03/2023 16:30H	Listagem dos livros de um determinado género.	Biblioteca	D. Francisca
7	05/03/2023 16:30H	Listagem dos livros de um autor à escolha.	Biblioteca	D. Francisca
8	05/03/2023 16:30H	Listagem das salas de estudo disponíveis.	Biblioteca	D. Francisca
9	11/03/2023 17:22H	Um livro pode ser disponibilizado por um funcionário.	Biblioteca	D. Francisca
10	11/03/2023 17:22H	A sala de estudo pode ser disponibilizada por um funcionário.	Biblioteca	D. Francisca
11	11/03/2023 17:22H	O leitor pode alugar livros.	Biblioteca	D. Francisca
12	11/03/2023 17:22H	O leitor pode requisitar salas de estudo.	Biblioteca	D. Francisca
13	11/03/2023 17:22H	Um funcionário pode registar um leitor na base de dados.	Biblioteca	D. Francisca

Tabela 2 - Requisitos de Manipulação

2.2.3.Requisitos de Controlo

Controlo				
Nr	Data e Hora	Descrição	Área	Analista
1	28/02/2023 12:05H	Cada livro está requisitado durante exatamente 25 dias.	Biblioteca	D. Francisca
2	28/02/2023 12:05H	Cada leitor só pode requisitar 5 livros em simultâneo.	Biblioteca	D. Francisca
3	28/02/2023 12:05H	Cada leitor só pode requisitar uma sala individual/de grupo por dia.	Biblioteca	D. Francisca
4	28/02/2023 12:05H	As capacidades das diversas salas de grupo variam entre 3 e 8 elementos.	Biblioteca	D. Francisca
5	28/02/2023 12:05H	Se um leitor não devolver o livro dentro do prazo, fica sujeito ao pagamento de uma coima.	Biblioteca	D. Francisca
6	28/02/2023 12:05H	Para a utilização das salas individuais/grupos é necessário a reserva.	Biblioteca	D. Francisca
7	02/03/2023 19:00H	As salas individual/grupos têm a duração de 2 horas.	Biblioteca	D. Francisca

Tabela 3 - Requisitos de Controlo

2.1. Análise e Validação Geral dos Requisitos

Esta fase do projeto exigiu bastante discussão para que desta forma fossem resolvidos os conflitos entre os diversos requisitos. Houve requisitos que foram retirados pois eram extremamente complicadas a sua implementação e pouca vantagem teria para o objetivo final, como, por exemplo, ver se num determinado momento estão a ser mais utilizadas salas de grupo ou individuais.

Apos várias conversações chegou-se a um acordo de requisitos entre as diversas partes assim a evitar conflitos. Pôde-se então prosseguir para a implementação do modelo conceptual.

3. Modelo Conceptual

3.1. Apresentação da Abordagem de Modelação Realizada

Posteriormente à recolha e análise aos requisitos, é a fase do planeamento do design da Base de Dados.

Para facilitar a comunicação entre ambas as partes, viu-se a necessidade de usar um modelo conceptual. Apesar de não técnico, é de fácil compreensão e permite uma comunicação clara sem ambiguidades. O modelo conceptual utilizado é **Diagrama ER**.

O **Diagrama ER** começou-se pela identificação das entidades e os seus relacionamentos. De seguida, adicionou-se os atributos, quer nas entidades como nos relacionamentos. E, por fim, inclui-se a cardinalidade dos relacionamentos.

Ao longo desta fase, a revisão e a testagem com a D. Francisca e com os requisitos é fundamental, para assegurar a coerência da futura base de dados.

3.2. Identificação e Caracterização das Entidades

Com base nos requisitos levantados, identificasse as seguintes entidades.

- **Leitor**

O leitor é um utilizador registado que tem permissão para aceder a serviços e recursos da biblioteca. Este é caracterizado por **id, nome, nº CC, e-mail, endereço (rua, localidade e código-postal) e números de telemóvel**.

Chaves Candidatas: id, nº CC

Chave Primária: id

Chave Alternativas: nº CC

- **Funcionário**

O funcionário é um indivíduo que trabalha na biblioteca e desempenha uma variedade de funções e responsabilidades para garantir o bom funcionamento e atendimento aos

leitores. Esta entidade é caracterizada **id, nome, e-mail, endereço (rua, localidade e código-postal) e número de telemóvel.**

Chaves Candidatas: id

Chave Primária: id

Chave Alternativas: nenhuma

- **Livro**

A entidade Livro representa o livro que integra a coleção de livros da biblioteca. É caracterizada pelo **id, título, autor, género e local (sala, corredor, secção).**

Chaves Candidatas: id

Chave Primária: id

Chave Alternativas: nenhuma

- **Sala de Estudo**

A entidade Sala de Estudo representa a sala que biblioteca disponibiliza aos leitores. É caracterizada pelo **número de sala, tipo (individual, grupo), capacidade e estado.**

Chaves Candidatas: número da sala

Chave Primária: número da sala

Chave Alternativas: nenhuma

- **ReservaLivro**

A entidade Reserva Livro representa o empréstimo do livro a um leitor, feito por um funcionário. É caracterizada por **id** e a **data do empréstimo.**

Chaves Candidatas: id

Chave Primária: id

Chave Alternativas: nenhuma

- **ReservaSala**

A entidade Reserva Sala representa a reserva de uma sala para utilização de um ou mais leitores, feita por um funcionário. É caracterizada por **id, data do empréstimo e horário (inicial e final).**

Chaves Candidatas: id

Chave Primária: id

Chave Alternativas: nenhuma

Entidades			
Designação	Descrição	Sinónimos	Ocorrência
Livro	Esta entidade acolhe informação sobre todos os livros existentes na biblioteca.		Aparecem de forma única da base de dados
Funcionário	Esta entidade gere as salas de estudos e os livros. Além disso, cria os leitores na BD.		Aparecem de forma única da base de dados
Salas de Estudo	Acolhe a informação sobre todas as salas de estudo existente na biblioteca.		Aparecem de forma única da base de dados
Leitor	Acolhe a informação sobre todos os leitores existentes na biblioteca.		Aparecem de forma única da base de dados
Reserva Sala	Acolhe o tipo da sala reservada(individual/grupo) e o horário (dia e hora).		Aparecem de forma única da base de dados
Reserva Livro	Acolhe o dia da reserva do livro.	Aluguer	Aparecem de forma única da base de dados

Tabela 4 - Entidades

3.3. Identificação e Caracterização dos Relacionamentos

Relacionamentos					
Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade	Descrição
Leitor	1	Faz	n	Reserva Livro	Um leitor pode fazer reservas de livros.
Leitor	1	Faz	n	Reserva Sala	Um leitor pode reservar uma sala de estudo, individual ou de grupo.
Funcionário	1	registra	n	Reserva Sala	O funcionário faz o empréstimo de um livro solicitado pelo leitor.
Funcionário	1	registra	n	Reserva Livro	O funcionário faz a reserva de uma sala de estudo solicitada pelo leitor.
Funcionário	1	registra	n	Leitor	O funcionário pode registrar vários leitores na base de dados.
Reserva Livro	n	contem	m	Livro	A Reserva Livro contem os livros emprestados aos leitores.
Reserva Sala	n	contem	1	Salas de Estudo	A Reserva Sala contem as salas reservadas por leitores.

Tabela 5 - Relacionamentos

3.4. Identificação e Caracterização da Associação dos Atributos com as Entidades e Relacionamentos

Entidade	Atributo	Tipo de Dados	Descrição	Exemplo
Leitor	Nome	VARCHAR (30)	Nome completo do/a leitor/a	António Pereira Costa
	ID	INT	Identificador do cliente	1
	Email	VARCHAR (30)	Endereço de email do/a leitor/a	antonpc@gmail.com
	Nº do CC	INT	Nº ID Civil	31063815
	Nº Telemóvel	INT	Nº de contacto do cliente	967123123
	Endereço		Endereço do cliente	
	Rua	VARCHAR (45)	Nome da rua, nº da porta e/ou andar	Rua do Caire, porta 7, 2ºesq
	Localidade	VARCHAR (15)	Nome da localidade	Braga
	Código Postal	VARCHAR (15)	Código Postal	4715-123
Livro	Autor	VARCHAR (25)	Autor do livro	Fernando Pessoa
	Género	VARCHAR (20)	Género literário do livro	Romance
	Título	VARCHAR (25)	Título do livro	Mensagem
	ID	INT	Identificador do livro	1307
	Local		Local da biblioteca onde o livro se encontra	
	Sala	INT	Nº da sala	7
	Corredor	INT	Nº/Nome do corredor	2
	Secção	VARCHAR (15)	Secção	3
Funcionário	Nome	VARCHAR (30)	Nome completo do/a funcionário/a	Joana Pimenta Magalhães
	ID	INT	Identificador do/a funcionário/a	5
	Email	VARCHAR (35)	Endereço de email do/a funcionário/a	joanihapimmg@gmail.com
	Nº Telemóvel	INT	Nº de contacto do/a funcionário/a	932789987
	Endereço		Endereço do/a funcionário/a	
	Rua	VARCHAR (45)	Nome da rua, nº da porta e/ou andar	Rua Dionisio III
	Localidade	VARCHAR (15)	Nome da localidade	Braga

Entidade	Atributo	Tipo de Dados	Descrição	Exemplo
	Código Postal	VARCHAR (15)	Código Postal	4715-321
Sala de Estudo	Nº da sala	INT	Nº da sala (identificador)	2
	Estado	ENUM ('Ocupado', 'Livre')	Ocupação atual da sala (Ocupada/Livre)	Ocupada
	Tipo	ENUM ('Individual', 'Grupo')	Tipo de serviço da sala(individual/grupo)	Individual
	Capacidade	INT	Capacidade máxima da sala	12
Reserva Livro	Dia	DATE	Dia que o leitor requisitou o livro	2023-03-23
	ID	INT	Identificador da reserva	12
Reserva Sala	Tipo	ENUM ('Individual', 'Grupo')	Tipo de serviço da sala	Individual
	Dia	DATE	Dia para qual o cliente reservou a sala	2023-03-23
	Horário		Horário em que o leitor requisitou a sala	
	Hora de Entrada	TIME	Horas para quais o leitor entrou na sala	14:00
	Horas de Saída	TIME	Horas para quais o leitor saiu da sala	16:00

Tabela 6 - Atributos

3.5. Apresentação e Explicação do diagrama ER produzido

Um as informações recolhidas anteriormente, foi possível construir, com o auxílio da ferramenta BR modelo, o seguinte modelo conceptual. Este foi âmbito de várias discussões até se chegar a acordo final.

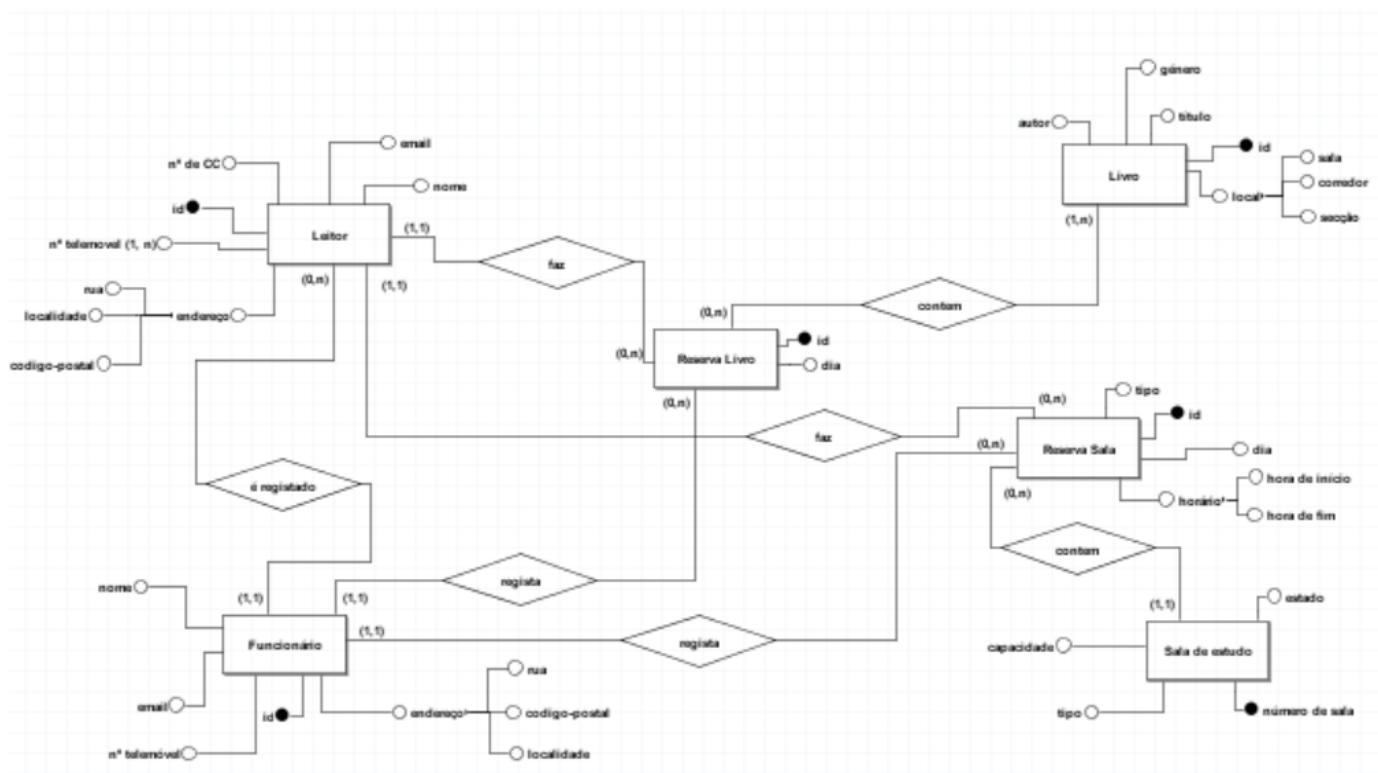


Figura 3 - Modelo Conceptual

Podemos então perceber que leitor somente pode ser registado na base de dados por um funcionário. O leitor pode requisitar tanto livros como salas de estudo.

A organização da biblioteca centra-se nos funcionários, uma vez que estes, além de registarem os leitores, são os responsáveis pela reserva de salas de estudo e empréstimos de livros.

Para se reservar um livro, é necessário termos informações sobre o leitor que pretende requisitá-lo, o funcionário responsável pela verificação e registo da reserva, e dados do livro que vai ser emprestado.

A reserva de sala de estudo é em todo semelhante à reserva de livros, apenas difere na parte que em vez de termos informação sobre o livro a ser pedido, temos informações sobre a sala de estudo requerida.

Para que as reservas sejam efetuadas tem de existir livro e salas de estudo na base de dados, pelo que se cria duas entidades para tal.

4. Modelação Lógica

Neste capítulo abordaremos o processo da construção do modelo lógico, tendo em conta a normalização de dados. Posteriormente será explicado o modelo lógico obtido. Serão ainda apresentadas interrogações feitas por parte do utilizador.

4.1. Construção e Validação do Modelo de Dados Lógico

Para construir o modelo logico foi preciso seguir as regras de derivação do modelo relacional, abordadas na UC. Iniciamos esse processo identificando as entidades e os atributos que lhes correspondem. Em seguida, aplicamos as seguintes regras:

- **Relacionamento binário de grau 1:N:** Neste caso é necessário duas entidades logicas, correspondendo cada uma a uma das entidades envolvidas no relacionamento. A entidade cujo cardinalidade é N, recebe a chave primária da outra entidade como atributo.

Ocorrências:

“é registado” entre Funcionário (1) e Leitor(N);

“registra” entre Funcionário (1) e Reserva Livro(N);

“registra” entre Funcionário (1) e Reserva Sala(N);

“contem” entre Sala de Estudo (1) e Reserva Sala (N);

“faz”, entre Leitor (1) e Reserva Livro(N);

“faz” entre Leitor (1) e Reserva Sala(N).

- **Relacionamento binário de grau N:M:** Nesta situação, são necessárias três entidades logicas. Duas deles correspondem às entidades iniciais e a terceira ao relacionamento. Esta última recebe como atributos as chaves primárias das outras duas entidades.

Ocorrências:

“contem” entre Livro (N) e Reserva Livro (M);

- **Atributos compostos** resultam numa entidade, e a entidade que lhe deu origem fica com a chave desta nova tabela.

Este processo resultou nas seguintes entidades logicas:

- Leitor = {Id, nome, email, nº do CC ,nº telemóvel, rua, localidade, código_postal, funcionário_id}
- Nº telemóvel = {nºtelemovel, leitor_id}
- Reserva Livro = {id, dia, funcionário_id, leitor_id}
- Reserva Sala_Estudo = {id, tipo, dia, hora de estrada, hora de saída, leitor_id, funcionário_id, sala de estudo numero da sala}
- Sala de Estudo = {numero da sala, tipo, capacidade, estado}
- ReservaLivro_Livro = {reserva livro_id, livro_id}
- Livro = {id, género, autor, titulo, sala, corredor, secção}
- Funcionário = {Id, nome, email, nº telemóvel, rua, localidade, código_postal}

4.2. Normalização de Dados

A normalização tem como objetivo minimizar a duplicação de dados nas tabelas. A nossa base de dados encontra-se normalizada uma vez o valor de cada atributo é constituído apenas por uns valores simples do seu domínio e o domínio de cada atributo de uma relação apenas contem valores atómicos. Além disso, numa tabela nenhum atributo não primo é funcionalmente dependente num subconjunto próprio de qualquer chave candidata e não existem atributos que não contribuam para a descrição da chave primária.

Existem exemplos de normalização 1FN, como a criação da tabela telemóveis que permitiu que não existissem copias desnecessárias na tabela Leitor. Podemos verificar que a 2FN foi aplicada na ReservaLivro_Livro pois na entidade não existem atributos específicos do livro, o que levaria a uma dependência parcial, uma vez que os precisava de um atributo que faz parte da chave primaria. Um exemplo da utilização da 3FN é o facto de a Reserva Sala não ter informação da sala além do id da mesma, evitando dependências transitivas.

4.3. Apresentação e explicação do modelo lógico produzido

O seguinte modelo logico foi obtido, em mysql workbench, através da conversão pelos mecanismos mencionados.

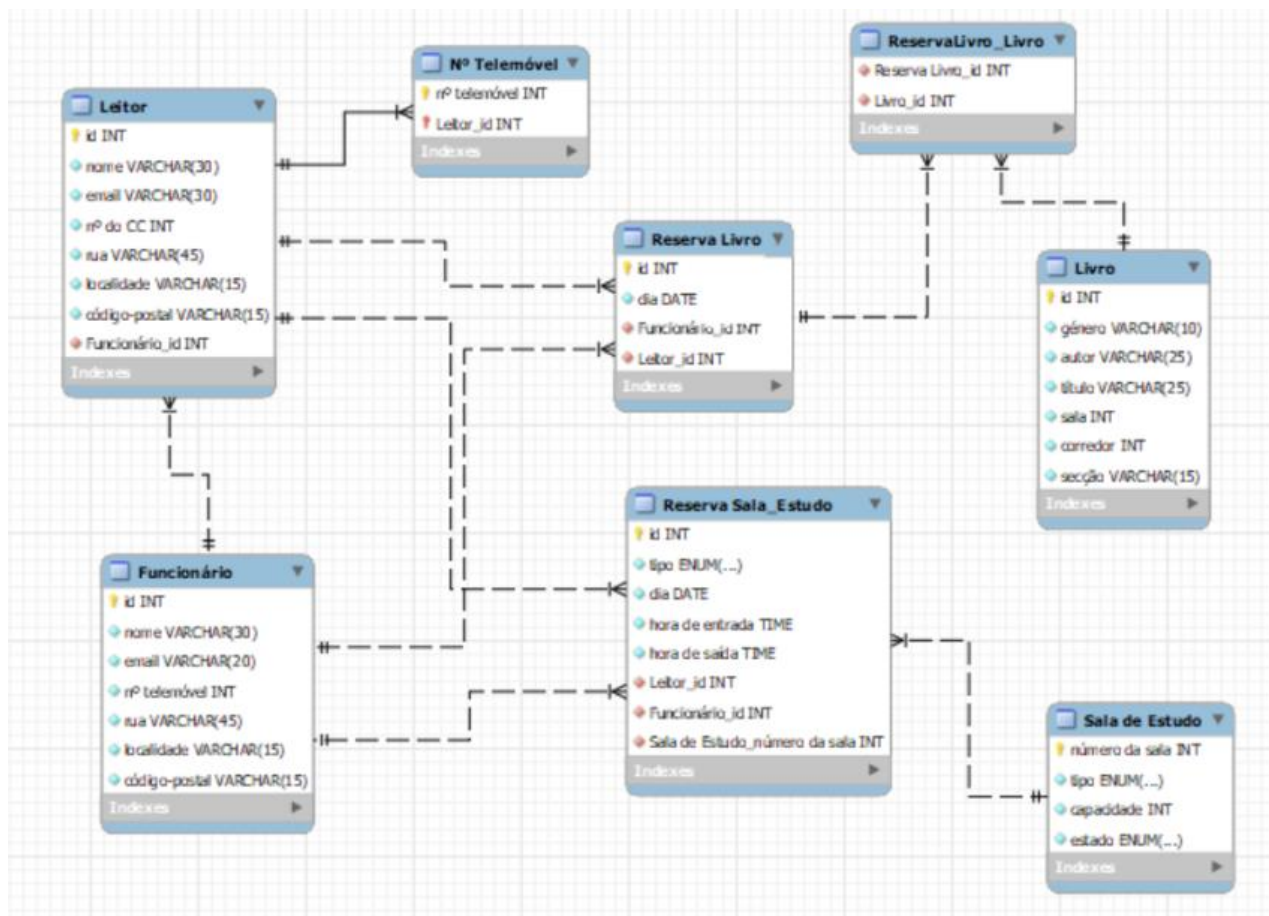


Figura 4 - Modelo Lógico

Neste formato, uma linha corresponde a um registo na base de dados e as colunas são relativas a um atributo. As pesquisas nas tabelas são essencialmente feitas através das chaves estrangeiras que surgiram das derivações aplicadas.

4.4. A Validação do modelo com interrogações do utilizado

Para validar o nosso modelo, fizemos interrogações com o objetivo de perceber que obtínhamos os resultados pretendidos. Segue-se um exemplo:

- **Consultar o género mais requisitado**

Começou-se por seleccionar todos os livros que estavam presentes tanto na tabela Livro e ReservaLivro_Livro. Seguidamente agruparam-se as linhas por género e ordenou-se os grupos por ordem decrescente de quantidade de linhas. O género mais requisitado era apresentado na primeira linha, após a ordenação.

Podemos concluir com estes testes a integridade do nosso sistema de base de dados, pelo que prosseguimos para a próxima fase.

5. Implementação Física

Nesta secção serão abordados todos os tópicos referentes à implementação física, desde a sua criação da base de dados, ao seu povoamento e manipulação. Ainda serão abordados tópicos acerca de backup bem como gerenciamento de permissões.

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

De forma a facilitar a tradução dos dados do modelo logico para a implementação física, decidimos fazer o modelo diretamente no mysql Workbench, uma vez que este possui a ferramenta Forward Engineering que transforma automaticamente o modelo num script. Para um mais fácil manuseamento das diversas tabelas e respetivos atributos, decidiu-se apenas retocar o script. Bastou executar o código para termos a nossa base de dados criada.

5.2 Tradução das interrogações do utilizador para SQL

- 6D - Os funcionários são caracterizados por um id único, nome, endereço, email, nºtelemovel.

```
CREATE TABLE IF NOT EXISTS Funcionário (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(30) NOT NULL,  
    email VARCHAR(35) NOT NULL,  
    nº_telemóvel INT UNSIGNED NOT NULL,  
    rua VARCHAR(45) NOT NULL,  
    localidade VARCHAR(15) NOT NULL,  
    codigo_postal VARCHAR(15) NOT NULL,  
    PRIMARY KEY (id),  
    UNIQUE INDEX id_UNIQUE (id ASC) VISIBLE)  
ENGINE = InnoDB;
```

Figura 5 – Criação da Tabela Funcionário

- 3M - Ver em que local se encontra um determinado livro.

```

DELIMITER $$
CREATE FUNCTION getlivrolocal (input VARCHAR(25))
  RETURNS VARCHAR(100)
  DETERMINISTIC
  BEGIN
    DECLARE output VARCHAR(100);

    SELECT CONCAT('Sala: ', sala, ' Corredor: ', corredor, ' Secção: ', secção)
    INTO output
    FROM Livro
    WHERE título = input;

    RETURN output;
  END $$
DELIMITER ;

```

Figura 6 - Função para saber o local de um certo Livro

- 10M – Qual o género de livros mais presente na biblioteca

```

SELECT género
  FROM Livro
 GROUP BY género
 ORDER BY count(*) DESC
 LIMIT 1;

```

Figura 7 - Query dos generos de livros mais existente

- 8C – O leitor pode alugar livros

```

9 ----- CONTROLO -----
10
11 DELIMITER $$
12 CREATE PROCEDURE addSalaEstudoGrupo(IN numero INT, IN hora_inicio TIME, IN hora_final TIME, IN id_funcionario INT, IN leitor INT)
13 BEGIN
14     DECLARE sala INT UNSIGNED;
15     DECLARE conta_funcionarios INT UNSIGNED;
16     DECLARE conta_leitor INT UNSIGNED;
17
18     -- Verificar se o leitor existe
19     SELECT COUNT(*) INTO conta_leitor FROM Reserva_Sala_Estudo WHERE Leitor_id = leitor and dia=curdate();
20
21     IF conta_leitor > 0 THEN
22         SELECT 'O leitor ja reserrvou hoje' AS mensagem;
23     ELSE
24         IF numero <3 or numero >8 THEN
25             SELECT 'Número de pessoas errado' AS mensagem;
26         ELSE
27             IF TIMESTAMPDIFF(HOUR, hora_inicio, hora_final) > 2 THEN
28                 SELECT 'Deve-se alugar menos tempo para a sala' AS mensagem;
29             ELSE
30                 -- Obtém o número da sala livre
31                 SELECT número_da_sala INTO sala FROM Sala_de_Estudo WHERE estado = 'Livre' AND tipo = 'Grupo' LIMIT 1;
32
33                 IF sala IS NOT NULL THEN
34                     -- Sala está livre, atualiza o estado para ocupado
35                     UPDATE Sala_de_Estudo SET estado = 'Ocupada' WHERE número_da_sala = sala;
36                     INSERT INTO Reserva_Sala_Estudo (tipo, data_reserva, hora_inicio, hora_final, id_leitor, sala) VALUES ('Grupo', CURRENT_DATE(), hora_inicio, hora_final, leitor, sala);
37                     SELECT CONCAT('A sala ', sala, ' está agora ocupada.') AS mensagem;
38                 ELSE
39                     -- Todas as salas estão ocupadas
40                     SELECT 'Todas as salas de estudo em grupo estão ocupadas.' AS mensagem;
41                 END IF;
42             END IF;
43         END IF;
44     END IF;
45 END $$
46 DELIMITER ;
47

```

Figura 8 - Procedimento para alugar livros

- 4M – Ver o género mais requisitado pelos leitores

```

select Livro.género from livro
    join ReservaLivro_Livro where (ReservaLivro_Livro.Livro_id=Livro.id)
GROUP BY género
ORDER BY count(*) DESC
LIMIT 1;

```

Figura 9 – Query para verificar o género mais verificado

É importante realçar que isto é apenas um pequeno exemplo de algumas queries, funções e procedures que nós implementamos, temos **bastantes** mais que pretendemos mostrar na apresentação do projeto, não colocamos todas pois assim o relatório ficaria demasiado extenso e cansativo. Assim como em seguida iremos apenas mostrar alguns exemplos de views criadas com o intuito de restringir o acesso a certos métodos.

5.3 Definição e caracterização das vistas de utilização em SQL

Criação da vista do funcionário e atribuição das suas permissões.

```
create user 'funcionario'@'localhost' identified by 'password';
grant select,insert,update,delete on EventsWorkbench.* to 'funcionario'@'localhost';
grant execute on procedure getlivrosgenero to 'funcionario'@'localhost';
grant execute on procedure getlivrosautor to 'funcionario'@'localhost';
grant execute on procedure getLivrosDaSeccao to 'funcionario'@'localhost';
grant execute on procedure addSalaEstudoGrupo to 'funcionario'@'localhost';
grant execute on procedure addSalaEstudoIndividual to 'funcionario'@'localhost';
grant execute on procedure addLivro to 'funcionario'@'localhost';
grant execute on procedure addLeitor to 'funcionario'@'localhost';
grant execute on procedure addFuncionario to 'funcionario'@'localhost';
grant select on getLivrosEmprestados to 'funcionario'@'localhost';
grant select on getLivrosDisponiveis to 'funcionario'@'localhost';
grant select on getLivrosAtrasados to 'funcionario'@'localhost';
grant select on getSalasEstudoUsadasDesdeSempre to 'funcionario'@'localhost';
grant select on getGeneroMaisPresenteNaBiblio to 'funcionario'@'localhost';
grant select on getHorarioMaisFrequente to 'funcionario'@'localhost';
grant select on getQuantidadeLivros to 'funcionario'@'localhost';
grant select on getSalasEstudoLivres to 'funcionario'@'localhost';
```

Figura 10 – Restrições Funcionário

Criação da vista do funcionário e atribuição das suas permissões.

```
create user 'leitor'@'localhost' identified by 'password';
grant select,insert,update,delete on EventsWorkbench.* to 'leitor'@'localhost';
grant select on getGeneroMaisPresenteNaBiblio to 'funcionario'@'localhost';
grant select on getSalasEstudoLivres to 'funcionario'@'localhost';
grant select on getMaisRequisitado to 'funcionario'@'localhost';
grant select on getLivrosDiferentes to 'funcionario'@'localhost';
grant select on getAutoresDiferentes to 'funcionario'@'localhost';
```

Figura 11 - Restrições Leitor

Alguns exemplos de criação de views criadas.

```
5 -- gêneros mais presente na biblioteca
6 CREATE view getGeneroMaisPresenteNaBiblio as
7     SELECT gênero
8     FROM Livro
9     GROUP BY gênero
0     ORDER BY count(*) DESC
1     LIMIT 1;
2
3 -- horário onde a biblioteca é mais frequentada pelos leitores.
4 CREATE view getHorarioMaisFrequente as
5     SELECT HOUR(hora_de_entrada)
6     FROM Reserva_Sala_Estudo
7     GROUP BY HOUR(hora_de_entrada)
8     ORDER BY count(*) DESC
9     LIMIT 1;
```

Figura 12 - Vistas

5.4 Indexação do sistema de dados

Uma vez que as nossas queries são simples e os seus tempos de execução são reduzidos, não sentimos a necessidade de utilizar indexação para além da que o Foward Engeneering criou, nomeadamente, para as chaves primárias.

5.5 Procedimentos Implementados

Alguns exemplos de procedimentos

```
9  ----- CONTROLO -----
10
11 DELIMITER $$
12 CREATE PROCEDURE addSalaEstudoGrupo(IN numero INT, IN hora_inicio TIME, IN hora_final TIME, IN id_funcionario INT, IN leitor INT)
13 BEGIN
14     DECLARE sala INT UNSIGNED;
15     DECLARE conta_funcionarios INT UNSIGNED;
16     DECLARE conta_leitor INT UNSIGNED;
17
18     -- Verificar se o leitor existe
19     SELECT COUNT(*) INTO conta_leitor FROM Reserva_Sala_Estudo WHERE Leitor_id = leitor and dia=curdate();
20
21     IF conta_leitor > 0 THEN
22         SELECT 'O leitor ja reserrvou hoje' AS mensagem;
23     ELSE
24         IF numero <3 or numero >8 THEN
25             SELECT 'Número de pessoas errado' AS mensagem;
26         ELSE
27             IF TIMESTAMPDIFF(HOUR, hora_inicio, hora_final) > 2 THEN
28                 SELECT 'Deve-se alugar menos tempo para a sala' AS mensagem;
29             ELSE
30                 -- Obtém o número da sala livre
31                 SELECT número_da_sala INTO sala FROM Sala_de_Estudo WHERE estado = 'Livre' AND tipo = 'Grupo' LIMIT 1;
32
33                 IF sala IS NOT NULL THEN
34                     -- Sala está livre, atualiza o estado para ocupado
35                     UPDATE Sala_de_Estudo SET estado = 'Ocupada' WHERE número_da_sala = sala;
36                     INSERT INTO Reserva_Sala_Estudo (tipo, data_reserva, hora_inicio, hora_final, id_leitor, sala) VALUES ('Grupo',
37                     SELECT CONCAT('A sala ', sala, ' está agora ocupada.') AS mensagem;
38                 ELSE
39                     -- Todas as salas estão ocupadas
40                     SELECT 'Todas as salas de estudo em grupo estão ocupadas.' AS mensagem;
41                 END IF;
42             END IF;
43         END IF;
44     END IF;
45 END $$
46 DELIMITER ;
47
48
```

Figura 13 – Procedimento para alugar uma sala de estudo

```

314
315 DELIMITER $$
316 CREATE PROCEDURE addLeitor(
317     IN p_nome VARCHAR(30),
318     IN p_email VARCHAR(30),
319     IN p_n_do_CC INT,
320     IN p_rua VARCHAR(45),
321     IN p_localidade VARCHAR(15),
322     IN p_codigo_postal VARCHAR(15),
323     IN p_funcionário_id INT
324 )
325 BEGIN
326     INSERT INTO Leitor (nome, email, n°_do_CC, rua, localidade, codigo_postal, funcionário_id)
327     VALUES (p_nome, p_email, p_n_do_CC, p_rua, p_localidade, p_codigo_postal, p_funcionário_id);
328 END $$
329 DELIMITER ;
---
```

Figura 14 – Procedimento para adicionar um Leitor

Mais uma vez vale relembrar que isto são apenas exemplos de alguns procedures que fizemos, pois temos diversos mais!

5.6 Plano de segurança e recuperação de dados

Começou-se por criar o ficheiro backup.sql através do **comando mysqldump -u root -p Biblioteca_MariaJudite > ./backup.sql**, onde ficará guardado o backup da base de dados. Seguidamente criou-se o ficheiro backup_script.sh com unicamente com o comando acima mencionado. Para esse ficheiro poder ser executado, utilizou-se o comando **chmod +x backup_script.sh**. Para que este seja executado todos os dias às duas da manhã, colocou-se o comando **0 2 * * * /home/<nome> /Trabalho_BD/backup_script.sh** no crontab (ficheiro com as tarefas a ser executadas).

Este método foi nos inspirado pois é o mais utilizado por diferentes aplicação (por exemplo Whatsapp) e foi nos recomendado também o mesmo\.

6. Implementação de um Sistema de Recolha de Dados.

Neste capítulo fala-se sobre como fizemos um ficheiro python para fazer o povoamento da base de dados. A seguir segue a explicação de como de como foi implementado.

Utilizamos python pelo simples motivo da sua fácil implementação, tanto em termos de escrever código, como conexão com o Mysql server.

6.1. Apresentação e modelo do sistema

De maneira a que o povoamento de uma base de dados seja possível através de dados em ficheiros exteriores ao script da base de dados, decidiu-se implementar uma conexão entre essa base de dados e os ficheiros, num programa python.

6.2. Implementação do sistema de recolha

```
oamento_csv_to_sql.py
import os
import csv
import mysql.connector

# Configurações do banco de dados
db_config = {
    'host': 'localhost',
    'user': 'root',
    'password': 'Henriquenuno16',
    'database': 'Biblioteca_MariaJudite'
}

# Diretório dos arquivos CSV
csv_directory = os.getcwd()

# Conectar ao banco de dados
conn = mysql.connector.connect(**db_config)
cursor = conn.cursor()

# Função para inserir dados na tabela
def insert_data(table_name, data):
    placeholders = ', '.join(['%s'] * len(data[0]))
    columns = ', '.join(data[0].keys())
    query = f"INSERT INTO {table_name} ({columns}) VALUES ({placeholders})"
    values = [tuple(row.values()) for row in data]
    cursor.executemany(query, values)
    conn.commit()

# Definir a ordem das tabelas (de pais para filhos)
table_order = ['Funcionário', 'Leitor', 'Nº_Telemóvel', 'Sala_de_Estudo', 'Livro', 'Reserva_Livro', 'ReservaLivro_Livro']
```

```

# Povoamento_csv_to_sql.py
# Percorrer as tabelas na ordem especificada
for table_name in table_order:
    filename = table_name + '.csv'
    csv_path = os.path.join(csv_directory, filename)

    # Verificar se o arquivo CSV existe
    if not os.path.exists(csv_path):
        print(f"Arquivo CSV {filename} não encontrado.")
        continue

    # Ler dados do arquivo CSV
    with open(csv_path, 'r') as csv_file:
        csv_reader = csv.DictReader(csv_file)
        data = [row for row in csv_reader]

    # Inserir dados na tabela correspondente
    try:
        insert_data(table_name, data)
        print(f"Dados inseridos na tabela {table_name}.")
    except mysql.connector.Error as err:
        print(f"Erro ao inserir dados na tabela {table_name}: {err}")
    except Exception as e:
        print(f"Erro: {e}")

# Fechar a conexão com o banco de dados
cursor.close()
conn.close()

```

Figura 15 - Implementação do Povoamento da Base de Dados

Começamos por fazer um script em python que transformasse todos os nossos dados recolhidos em ficheiros csv.

Para a Implementação do Sistema de Recolha de Dados, decidiu-se criar um script em Python que faz a conexão entre diversos ficheiros “.csv” e a base de dados.

Os diversos “.csv” contêm os valores que são desejados inserir nas diversas tabelas da base de dados.

Desta maneira, podemos então fazer o povoamento da base de dados a partir de ficheiros “.csv”.

6.3. Funcionamento do Sistema

De forma objetiva a implementação do sistema funciona da seguinte forma: começamos por guardar em “db_config” as configurações do mysql server e da base de dados a serem utilizados. Depois, em “csv_directory” guardamos a diretoria para a pasta atual, pois os “.csv” e o “.py” estão na mesma diretoria, fazendo então a conexão com a base de dados.

Os “.csv” contêm os valores das diversas tabelas da base de dados, sendo os seus nomes o respetivo nome da tabela na base de dados.

A função “insert_data” vai para todos os “.csv” e respetivas tabelas realizar o commando mysql “INSERT INTO”, de maneira a carregar as diversas entradas para a sua respetiva tabela na base de dados.

Por fim, fechamos a conexão com a base de dados.

7. Implementação do sistema de painéis de análise

Nesta secção será abordado como se utilizou a ferramenta Power BI para criar painéis de análise de dado.

Decidimos usar o Power BI devido à sua recomendação pelo Professor nas aulas Teóricas.

7.1 Definição e caracterização da vista de dados para análise

O Power BI é uma plataforma de análise e visualização de dados. Esta permite a criação rápida de gráficos e tabelas utilizando, por exemplo, uma base de dados como fonte de dados para manipulação.

Com a base de dados da biblioteca Maria Judite implementada, foi possível criar gráficos estatísticos como os seguintes.

Neste caso, apenas foi realizada uma vista uma vez que foi a do funcionário, uma vez que o que contém mais informações a serem analisadas.

7.2 Povoamento das estruturas de dados para análise

As estruturas de dados para análise foram as tabelas da nossa base de dados e os seus procedimentos. Desta forma, foi necessário estabelecer uma ligação entre o server do Mysql server e o Power BI, a partir de um connector/NET, que nos possibilitou a transição de dados do server para o Power BI.

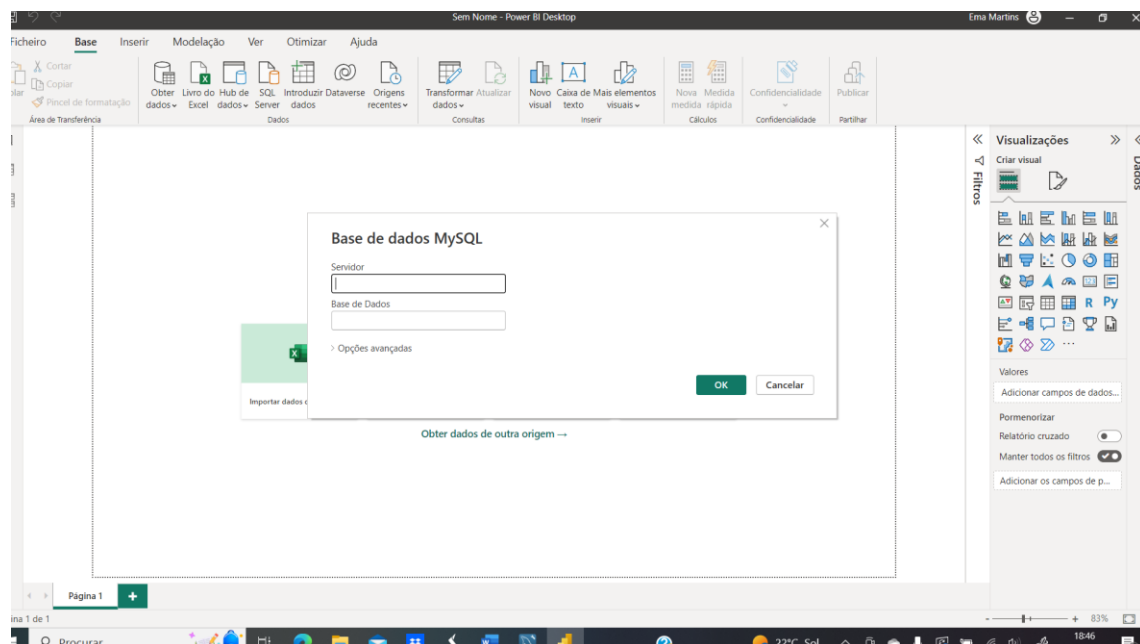


Figura 16 - Exemplo de conexão entre a Base de Dados e o Power BI

7.3 Apresentação e caracterização dos dashboards implementados

Uma vez que os nossos dados para análise se centravam em percentagens, decidimos criar alguns gráficos circulares que representassem estados da biblioteca. Seguem alguns exemplos.

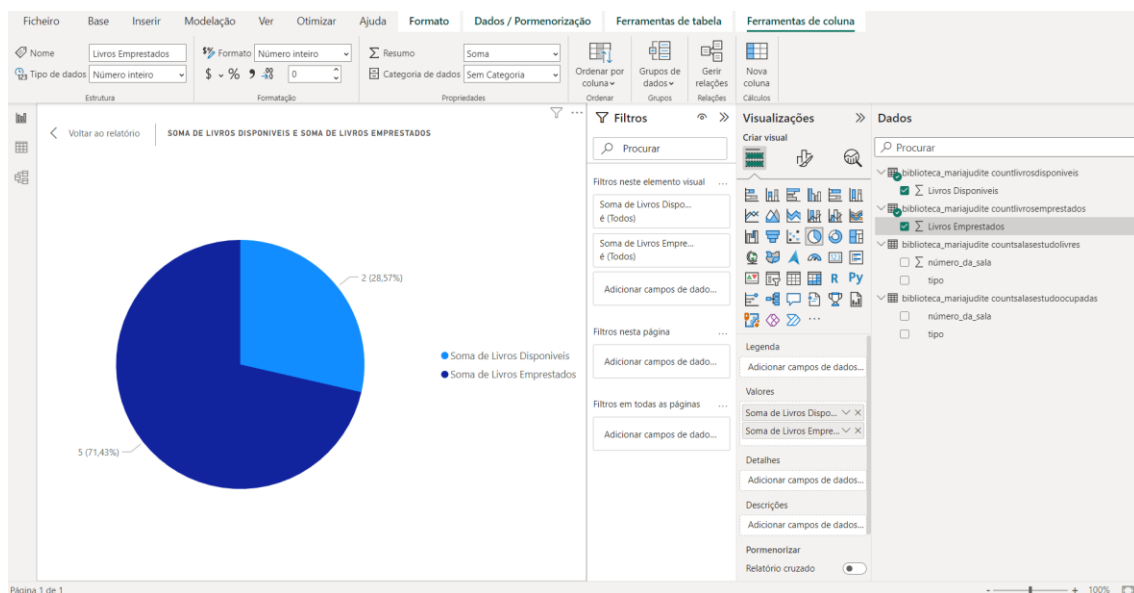


Figura 17 - Gráfico das somas de livros disponíveis e de livros Emprestados

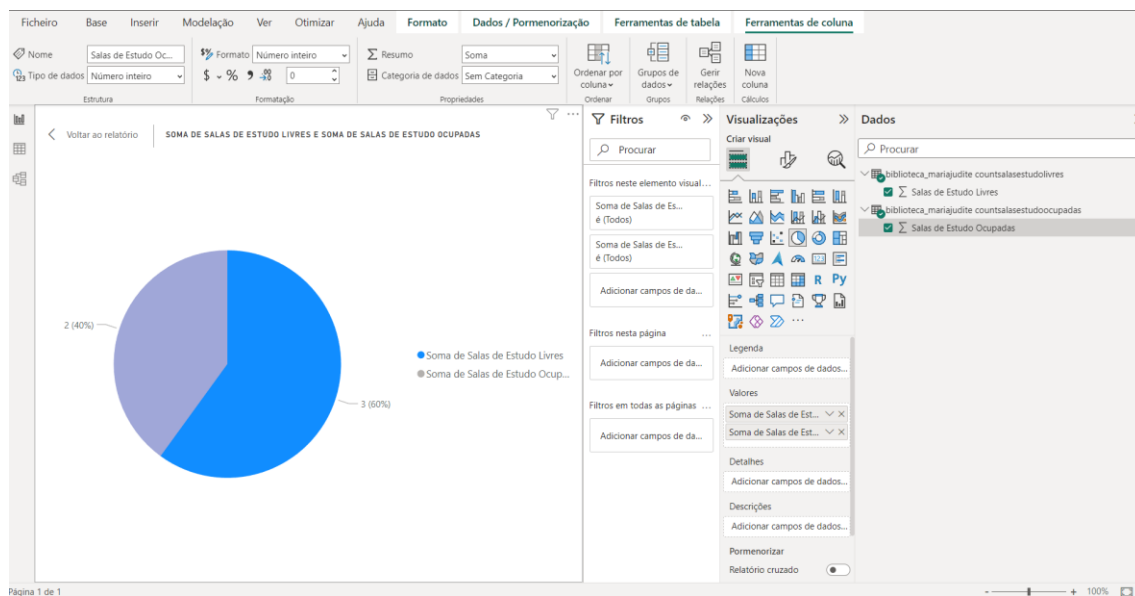


Figura 18 - Gráfico das somas de salas de estudo livres e ocupadas

Conclusões e Trabalho Futuro

Este trabalho foi importante para cimentar e aprofundar os conhecimentos obtidos na UC. Sentimos alguma dificuldade em certos tópicos do relatório, devido principalmente à nossa in experiência com o mesmo, como interrogações dos utilizadores e controlo de restrições, uma vez que achamos que não foram aprofundados o suficiente nas aulas para podermos responder corretamente a um trabalho com tamanha complexidade. Assim como no Power BI, porem reconhecemos após a sua aplicação que pode ser bastante útil para a análise de dados, bem como backups de bases de dados ou até povoamentos baseados em ficheiros externos. Pretende-se no futuro fazer uma interface gráfica para a biblioteca de forma a ser mais fácil de manipular os dados por parte do utilizador. Pretende-se ainda adicionar mais funcionalidades tanto para o leitor como para o funcionário assim como aprofundar as restrições de acessos a cada um e o encapsulamento da informação.

Referências

Mysql. (2018). MySql. [Online]. [26 November 2018]. Available from: <https://dev.mysql.com/doc/refman/8.0/en/>.

Connolly, Thomas, and Carolyn Begg. "MySQL and BR Model: A Practical Guide to Database Design and Implementation." Publisher Name, Year.

Ullman, Larry. "Building Web Applications with BR Model and MySQL." Publisher Name, Year.

https://sqlzoo.net/wiki/SQL_Tutorial

Anexos

<<Os anexos deverão ser utilizados para a inclusão de informação adicional necessária para uma melhor compreensão do relatório o para complementar tópicos, secções ou assuntos abordados. Os anexos criados deverão ser numerados e possuir uma designação. Estes dados permitirão complementar o Índice geral do relatório relativamente à enumeração e apresentação dos diversos anexos.>>

I. Anexo 1 – Script da Implementação Física

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----
-- Schema Biblioteca_MariaJudite
-- -----

-- -----
-- Schema Biblioteca_MariaJudite
-- -----

CREATE SCHEMA IF NOT EXISTS Biblioteca_MariaJudite;
USE Biblioteca_MariaJudite ;

-- -----
-- Table Funcionário
-- -----

CREATE TABLE IF NOT EXISTS Funcionário (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  nome VARCHAR(30) NOT NULL,
  email VARCHAR(35) NOT NULL,
  nº_telemóvel INT UNSIGNED NOT NULL,
  rua VARCHAR(45) NOT NULL,
  localidade VARCHAR(15) NOT NULL,
  codigo_postal VARCHAR(15) NOT NULL,
  PRIMARY KEY (id),
  UNIQUE INDEX id_UNIQUE (id ASC) VISIBLE)
```



```
ENGINE = InnoDB;
```

```
-- -----  
-- Table Leitor  
-- -----
```

```
CREATE TABLE IF NOT EXISTS Leitor (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(30) NOT NULL,  
    email VARCHAR(30) NOT NULL,  
    n°_do_CC INT UNSIGNED NOT NULL,  
    rua VARCHAR(45) NOT NULL,  
    localidade VARCHAR(15) NOT NULL,  
    codigo_postal VARCHAR(15) NOT NULL,  
    Funcionário_id INT UNSIGNED NOT NULL,  
    PRIMARY KEY (id),  
    UNIQUE INDEX id_UNIQUE (id ASC) VISIBLE,  
    UNIQUE INDEX n°_do_CC_UNIQUE (n°_do_CC ASC) VISIBLE,  
    INDEX fk_Leitor_Funcionário1_idx (Funcionário_id ASC) VISIBLE,  
    CONSTRAINT fk_Leitor_Funcionário1  
        FOREIGN KEY (Funcionário_id)  
        REFERENCES Funcionário (id)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table N° Telemóvel  
-- -----
```

```
CREATE TABLE IF NOT EXISTS N°_Telemóvel (  
    n°_telemóvel INT UNSIGNED NOT NULL,  
    Leitor_id INT UNSIGNED NOT NULL,  
    PRIMARY KEY (Leitor_id, n°_telemóvel),  
    INDEX fk_N°_Telemóvel_Leitor1_idx (Leitor_id ASC) VISIBLE,  
    CONSTRAINT fk_N°_Telemóvel_Leitor1  
        FOREIGN KEY (Leitor_id)  
        REFERENCES Leitor (id)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;
```

```
-- -----  
-- Table Sala de Estudo  
-- -----
```

```
CREATE TABLE IF NOT EXISTS Sala_de_Estudo (  
    número_da_sala INT UNSIGNED NOT NULL,  
    tipo ENUM('Individual', 'Grupo') NOT NULL,  
    capacidade INT UNSIGNED NOT NULL,  
    estado ENUM('Ocupada', 'Livre') NOT NULL,  
    PRIMARY KEY (número_da_sala),  
    UNIQUE INDEX número_da_sala_UNIQUE (número_da_sala ASC) VISIBLE)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table Reserva Sala_Estudo  
-- -----
```

```
CREATE TABLE IF NOT EXISTS Reserva_Sala_Estudo (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    tipo ENUM('Individual', 'Grupo') NOT NULL,  
    dia DATE NOT NULL,  
    hora_de_entrada TIME NOT NULL,  
    hora_de_saída TIME NOT NULL,  
    Leitor_id INT UNSIGNED NOT NULL,  
    Funcionário_id INT UNSIGNED NOT NULL,  
    Sala_de_Estudo_número_da_sala INT UNSIGNED NOT NULL,  
    PRIMARY KEY (id),  
    UNIQUE INDEX id_UNIQUE (id ASC) VISIBLE,  
    INDEX fk_Reserva_Sala_Estudo_Leitor1_idx (Leitor_id ASC) VISIBLE,  
    INDEX  fk_Reserva_Sala_Estudo_Funcionário1_idx (Funcionário_id ASC)  
VISIBLE,  
    INDEX                                     fk_Reserva_Sala_Estudo_Sala_de_Estudo1_idx  
(Sala_de_Estudo_número_da_sala ASC) VISIBLE,  
    CONSTRAINT fk_Reserva_Sala_Estudo_Leitor1  
        FOREIGN KEY (Leitor_id)  
        REFERENCES Leitor (id)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,
```

```

CONSTRAINT fk_Reserva_Sala_Estudo_Funcionário1
    FOREIGN KEY (Funcionário_id)
    REFERENCES Funcionário (id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT fk_Reserva_Sala_Estudo_Sala_de_Estudo1
    FOREIGN KEY (Sala_de_Estudo_número_da_sala)
    REFERENCES Sala_de_Estudo (número_da_sala)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table Reserva Livro
-- -----

CREATE TABLE IF NOT EXISTS Reserva_Livro (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    dia DATE NOT NULL,
    Funcionário_id INT UNSIGNED NOT NULL,
    Leitor_id INT UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    UNIQUE INDEX id_UNIQUE (id ASC) VISIBLE,
    INDEX fk_Reserva_Livro_Funcionário1_idx (Funcionário_id ASC) VISIBLE,
    INDEX fk_Reserva_Livro_Leitor1_idx (Leitor_id ASC) VISIBLE,
    CONSTRAINT fk_Reserva_Livro_Funcionário1
        FOREIGN KEY (Funcionário_id)
        REFERENCES Funcionário (id)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT fk_Reserva_Livro_Leitor1
        FOREIGN KEY (Leitor_id)
        REFERENCES Leitor (id)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table Livro

```

```

-----
CREATE TABLE IF NOT EXISTS Livro (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    género VARCHAR(20) NOT NULL,
    autor VARCHAR(25) NOT NULL,
    título VARCHAR(25) NOT NULL,
    sala INT UNSIGNED NOT NULL,
    corredor INT UNSIGNED NOT NULL,
    secção VARCHAR(15) NOT NULL,
    PRIMARY KEY (id),
    UNIQUE INDEX id_UNIQUE (id ASC) VISIBLE)
ENGINE = InnoDB;

-----

-- Table ReservaLivro_Livro
-----

CREATE TABLE IF NOT EXISTS ReservaLivro_Livro (
    Reserva_Livro_id INT UNSIGNED NOT NULL,
    Livro_id INT UNSIGNED NOT NULL,
    INDEX fk_ReservaLivro_Livro_Reserva_Livro1_idx (Reserva_Livro_id ASC)
VISIBLE,
    INDEX fk_ReservaLivro_Livro_Livro1_idx (Livro_id ASC) VISIBLE,
    CONSTRAINT fk_ReservaLivro_Livro_Reserva_Livro1
        FOREIGN KEY (Reserva_Livro_id)
        REFERENCES Reserva_Livro (id)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT fk_ReservaLivro_Livro_Livro1
        FOREIGN KEY (Livro_id)
        REFERENCES Livro (id)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- drop database Biblioteca_MariaJudite;

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

```


II. Anexo 2 – Script do Povoamento da Base de Dados

-- Povoamento--

```
INSERT                                INTO                                Funcionário
(nome,email,nº_telemóvel,rua,localidade,codigo_postal)
VALUES
('Francisca de Andrade','xica72@gmail.com','932432543','Rua da
Xica','Lousada','4522-55'),
('João Silva','joao.silva@gmail.com','912345678','Rua dos
Lírios','Porto','4000-001'),
('Maria Santos','maria.santos@gmail.com','919876543','Avenida das
Flores','Lisboa','1000-200'),
('Pedro Costa','pedro.costa@hotmail.com','925678943','Travessa das
Oliveiras','Braga','4700-400'),
('Sofia Ferreira','sofia.ferreira@outlook.com','933215477','Praça
Central','Coimbra','3000-100');
```

```
INSERT                                INTO                                Leitor
(nome,email,nº_do_CC,rua,localidade,codigo_postal,Funcionário_id)
VALUES
('José Silva','jose@gmail.com','75423311','Rua do
Largo','Lousada','4322-011','1'),
('Ana Santos','ana.santos@gmail.com','64567832','Avenida das
Rosas','Porto','4000-123','2'),
('Rui Costa','rui.costa@hotmail.com','84562378','Travessa das
Flores','Lisboa','1000-345','3'),
('Marta Pereira','marta.pereira@gmail.com','95678432','Rua dos
Pinheiros','Braga','4700-567','4'),
('Carlos Ferreira','carlos.ferreira@hotmail.com','73647892','Praça
Central','Coimbra','3000-678','1'),
('Sara Martins','sara.martins@gmail.com','84562312','Rua da
Fonte','Guimarães','4800-789','2'),
```

```

        ('André
Rodrigues','andre.rodriques@outlook.com','92567890','Avenida
Principal','Faro','8000-890','3');

```

```

INSERT INTO N°_Telemóvel(n°_telemóvel,Leitor_id)
VALUES
    ('912435617','5'),
    ('912431117','1'),
    ('944435617','3'),
    ('963377227','3'),
    ('919393937','6'),
    ('933892927','4');

```

```

INSERT INTO Sala_de_Estudo (número_da_sala,tipo,capacidade,estado)
VALUES
    ('1','Individual','1','Livre'),
    ('7','Grupo','5','Ocupada'),
    ('4','Individual','1','Ocupada'),
    ('3','Grupo','3','Livre'),
    ('6','Grupo','8','Livre');

```

```

INSERT                                                                    INTO
Reserva_Sala_Estudo(tipo,dia,hora_de_entrada,hora_de_saída,Leitor_id,Funcionário_id,Sala_de_Estudo_número_da_sala)
VALUES
    ('Individual','2020-05-10','10:05','12:05','1','2','4'),
    ('Grupo','2022-05-10','10:10','12:10','2','3','3'),
    ('Individual','2021-04-05','13:35','15:35','4','5','1'),
    ('Grupo','2022-03-02','10:05','12:25','3','1','6'),
    ('Grupo','2023-01-02','12:05','14:45','5','3','7');

```

```

INSERT INTO Reserva_Livro (dia,Funcionário_id,Leitor_id)
VALUES
    ('2020-10-10','4','6'),
    ('2020-08-15','1','3'),
    ('2020-05-15','2','1'),

```

```

('2020-07-19','3','4'),
('2020-02-24','5','2');

```

```

INSERT INTO Livro(género,autor,título,sala,corredor,secção)
VALUES
    ('ficção científica','Francisca de Sousa','Perdido no
Espaço','1','2','4'),
    ('comedia','Francisca de Andrade','Perdido no
Shopping','2','5','5'),
    ('comedia','Francisco de Sousa','Caso perdido','4','1','5'),
    ('ficção científica','Jose Almeida','Sou humano?','3','3','3'),
    ('romance','Jose Amaro','Perdido em ti','5','4','1'),
    ('terror','Joao Afonso','Perdido na UM','2','1','2'),
    ('cientifico','Josefina Alberta','Perdido na
Mitocondria','3','6','3');

```

```

INSERT INTO ReservaLivro_Livro(Reserva_Livro_id,Livro_id)
VALUES
    ('1','2'),
    ('2','1'),
    ('3','7'),
    ('4','3'),
    ('5','6');

```


III. Anexo 3 – Script das Queries

```
-- QUERIES
-- Quantos livros estao disponieis
create view countLivrosDisponiveis as
    SELECT count(título)
        FROM Livro
        WHERE id NOT IN (

Expand
Biblioteca_MariaJudite_Queries.sql
11 KB
-- CRIAS AS RESTRICOES
create user 'funcionario'@'localhost' identified by 'password';
grant      select,insert,update,delete      on      EventsWorkbench.*      to
'funcionario'@'localhost';
grant execute on procedure getlivrosgenero to 'funcionario'@'localhost';
grant execute on procedure getlivrosautor to 'funcionario'@'localhost';
grant      execute      on      procedure      getLivrosDaSeccao      to
'funcionario'@'localhost';
Expand
Biblioteca_MariaJudite_Restricoes.sql
3 KB
Attachment file type: unknown
salas_de_estudo.pbit
8.04 KB
```

Ema Martins

Ema Martins#8549

```
-- QUERIES
-- Quantos livros estao disponieis
create view countLivrosDisponiveis as
    SELECT count(título)
        FROM Livro
```

```

        WHERE id NOT IN (
            SELECT Livro_id
            FROM ReservaLivro_Livro);

-- Quantos livros emprestados
CREATE view countLivrosEmprestados as
    SELECT count(título)
    FROM ReservaLivro_Livro
    JOIN Livro ON (ReservaLivro_Livro.Livro_id = Livro.id);

-- Quais livros estão emprestados
CREATE view getLivrosEmprestados as
    SELECT título
    FROM ReservaLivro_Livro
    JOIN Livro ON (ReservaLivro_Livro.Livro_id = Livro.id);

-- Quais livros estão disponíveis
CREATE view getLivrosDisponíveis as
    SELECT título
    FROM Livro
    WHERE id NOT IN (
        SELECT Livro_id
        FROM ReservaLivro_Livro);

CREATE view getLivrosAtrasados as
    SELECT Reserva_Livro.id, Livro.título, Reserva_Livro.dia
    FROM Reserva_Livro
    JOIN ReservaLivro_Livro ON (Reserva_Livro.id =
ReservaLivro_Livro.Reserva_Livro_id)
    JOIN Livro ON (ReservaLivro_Livro.Livro_id = Livro.id)
    WHERE DATEDIFF(CURDATE(), Reserva_Livro.dia) > 30; --
Considerando 30 dias como um mês

-- Que leitores estão a usar as salas de estudo desde sempre
CREATE view getSalasEstudoUsadasDesdeSempre as
    SELECT distinct Leitor.nome
    FROM Reserva_Sala_Estudo JOIN Leitor ON
(Reserva_Sala_Estudo.Leitor_id = Leitor.id);

```

```

-- Onde se encontra determinado livro
DELIMITER $$
CREATE FUNCTION    getlivrolocal (input VARCHAR(25))
    RETURNS VARCHAR(100)
    DETERMINISTIC
    BEGIN
        DECLARE output VARCHAR(100);

        SELECT CONCAT('Sala: ', sala, ' Corredor: ', corredor, ' Secção: ',
secção)
            INTO output
            FROM Livro
            WHERE título = input;

        RETURN output;
    END $$
DELIMITER ;

-- gêneros mais presente na biblioteca
CREATE view getGeneroMaisPresenteNaBiblio as
    SELECT género
        FROM Livro
        GROUP BY género
        ORDER BY count(*) DESC
        LIMIT 1;

-- horário onde a biblioteca é mais frequentada pelos leitores.
CREATE view getHorarioMaisFrequente as
    SELECT HOUR(hora_de_entrada)
        FROM Reserva_Sala_Estudo
        GROUP BY HOUR(hora_de_entrada)
        ORDER BY count(*) DESC
        LIMIT 1;

-- Quantidade de livros da biblioteca
CREATE view getQuantidadeLivros as
    SELECT count(*)
        FROM Livro;

```

```

-- Listagem dos livros de um determinado genero.
DELIMITER $$
CREATE PROCEDURE getlivrosgenero (IN input VARCHAR(25)) -- permite
executar uma ou mais instrucoes varias vezes
BEGIN
    SELECT título
    FROM Livro
    WHERE género = input;
END $$
DELIMITER ;

-- CALL getlivrosgenero('comedia');

-- Listagem dos livros de um autor à escolha.
DELIMITER $$
CREATE PROCEDURE getlivrosautor (IN input VARCHAR(25)) -- permite
executar uma ou mais instrucoes varias vezes
BEGIN
    SELECT título
    FROM Livro
    WHERE autor = input;
END $$
DELIMITER ;

-- CALL getlivrosautor('Francisco de Sousa');
-- DROP PROCEDURE getlivrosautor;

-- Listagem das salas de estudo disponíveis.
CREATE view getSalasEstudoLivres as
    SELECT número_da_sala, tipo
    FROM Sala_de_Estudo
    WHERE estado = 'Livre';

-- Quantas salas de estudo livres
CREATE view countSalasEstudoLivres as
    SELECT count(número_da_sala)
    FROM Sala_de_Estudo
    WHERE estado = 'Livre';

-- Quantas salas de estudo ocupadas

```

```

CREATE view countSalasEstudoOcupadas as
    SELECT count(número_da_sala)
    FROM Sala_de_Estudo
    WHERE estado = 'Ocupada';

-- gêneros mais requisitados pelos leitores.
CREATE view getMaisRequisitado as
    SELECT Livro.género
    FROM livro
    JOIN ReservaLivro_Livro ON ReservaLivro_Livro.Livro_id = Livro.id
    GROUP BY Livro.género
    ORDER BY COUNT(*) DESC
    LIMIT 1;

-- que livros diferentes existem ordenados alfabeticamente
CREATE view getLivrosDiferentes as
    select distinct título from Livro
    order by título;

-- que autores diferentes existem ordenados alfabeticamente
CREATE view getAutoresDiferentes as
    select distinct autor from Livro
    order by autor;

-- os livros de uma dada secção
DELIMITER $$
create procedure getLivrosDaSeccao(in input varchar(15))
begin
    select título from Livro
    where secção = input;
end $$
DELIMITER ;

-- CALL getLivrosDaSeccao('5');
-- DROP PROCEDURE getLivrosDaSeccao;

-- quantos leitores existem
create view getLeitoresExistentes as
    select count(*) from Leitor;

```

```

-- quantos leitores existem nas salas de estudo neste momento
create view getLeitoresNasSalas as
    select count(*) from Reserva_Sala_Estudo where dia=curdate()
and hora_de_entrada> curtime() and hora_de_saída< curtime();

-- ----- CONTROLO -----
----

DELIMITER $$
CREATE PROCEDURE addSalaEstudoGrupo(IN numero INT, IN hora_inicio TIME,
IN hora_final TIME, IN id_funcionario INT, IN leitor INT)
BEGIN
    DECLARE sala INT UNSIGNED;
    DECLARE conta_funcionarios INT UNSIGNED;
    DECLARE conta_leitor INT UNSIGNED;

    -- Verificar se o leitor existe
    SELECT COUNT(*) INTO conta_leitor FROM Reserva_Sala_Estudo WHERE
Leitor_id = leitor and dia=curdate();

    IF conta_leitor > 0 THEN
        SELECT 'O leitor ja reserrvou hoje' AS mensagem;
    ELSE
        IF numero <3 or numero >8 THEN
            SELECT 'Número de pessoas errado' AS mensagem;
        ELSE
            IF TIMESTAMPDIFF(HOUR, hora_inicio, hora_final) > 2
THEN
                SELECT 'Deve-se alugar menos tempo para a sala'
AS mensagem;
            ELSE
                -- Obtém o número da sala livre
                SELECT número_da_sala INTO sala FROM
Sala_de_Estudo WHERE estado = 'Livre' AND tipo = 'Grupo' LIMIT 1;

                IF sala IS NOT NULL THEN
                    -- Sala está livre, atualiza o estado
para ocupado

```

```

UPDATE Sala_de_Estudo SET estado =
'Ocupada' WHERE número_da_sala = sala;
INSERT INTO Reserva_Sala_Estudo (tipo,
data_reserva, hora_inicio, hora_final, id_leitor, sala) VALUES ('Grupo',
CURRENT_DATE(), hora_inicio, hora_final, leitor, sala);
SELECT CONCAT('A sala ', sala, ' está
agora ocupada.') AS mensagem;
ELSE
-- Todas as salas estão ocupadas
SELECT 'Todas as salas de estudo em grupo
estão ocupadas.' AS mensagem;
END IF;
END IF;
END IF;
END IF;
END $$
DELIMITER ;

```

```

-- alugar sala de estudo de individual
DELIMITER $$
CREATE PROCEDURE addSalaEstudoIndividual(IN numero INT, IN hora_inicio
TIME, IN hora_final TIME, IN id_funcionario INT, IN leitor INT)
BEGIN
DECLARE sala INT UNSIGNED;
DECLARE conta_funcionarios INT UNSIGNED;
DECLARE conta_leitor INT UNSIGNED;

-- Verificar se o leitor existe
SELECT COUNT(*) INTO conta_leitor FROM Reserva_Sala_Estudo WHERE
Leitor_id = leitor and dia=curdate();

IF conta_leitor > 0 THEN
SELECT 'O leitor ja reserrvou hoje' AS mensagem;
ELSE
IF numero !=1 THEN
SELECT 'Número de pessoas errado' AS mensagem;
ELSE
IF TIMESTAMPDIFF(HOUR, hora_inicio, hora_final) > 2
THEN

```

```

SELECT 'Deve-se alugar menos tempo para a sala'
AS mensagem;

ELSE
    -- Obtém o número da sala livre
    SELECT    número_da_sala    INTO    sala    FROM
Sala_de_Estudo WHERE estado = 'Livre' AND tipo = 'Individual' LIMIT 1;

    IF sala IS NOT NULL THEN
        -- Sala está livre, atualiza o estado
para ocupado

        UPDATE    Sala_de_Estudo    SET    estado    =
'Ocupada' WHERE número_da_sala = sala;
        INSERT INTO Reserva_Sala_Estudo (tipo,
data_reserva, hora_inicio, hora_final, id_leitor, sala) VALUES
('Individual', CURRENT_DATE(), hora_inicio, hora_final, leitor, sala);
        SELECT CONCAT('A sala ', sala, ' está
agora ocupada.') AS mensagem;
    ELSE
        -- Todas as salas estão ocupadas
        SELECT    'Todas as salas de estudo
individuais estão ocupadas.' AS mensagem;
    END IF;
END IF;
END IF;
END IF;
END $$
DELIMITER ;

-- call addSalaEstudoGrupo('4','08:20','09:30',1,1);
-- drop procedure addSalaEstudoGrupo;

-- call addSalaEstudoIndividual('1','08:20','09:30',1,1);
-- drop procedure addSalaEstudoIndividual;

-- alugar livro
DELIMITER $$
CREATE PROCEDURE addLivroReserva(IN nome_livro VARCHAR(25), IN id_leitor
INT, IN id_funcionario INT)

```



```

BEGIN
    DECLARE id_livro_procuramos INT UNSIGNED;
    DECLARE conta_reservas INT UNSIGNED;
    DECLARE conta_funcionarios INT UNSIGNED;

    -- Verificar se o funcionário existe
    SELECT COUNT(*) INTO conta_funcionarios FROM Funcionário WHERE
id_funcionario = id;

    -- Verificar se o leitor tem menos de 5 livros reservados
    SELECT COUNT(*) INTO conta_reservas FROM Reserva_Livro WHERE
Leitor_id = id_leitor;

    IF conta_reservas > 4 THEN
        SELECT 'Tem demasiados livros reservados' AS mensagem;
    ELSE
        -- Obter o livro com o nome dado
        SELECT id INTO id_livro_procuramos FROM Livro
        WHERE título = nome_livro AND id NOT IN (SELECT Livro_id
FROM ReservaLivro_Livro);

        IF id_livro_procuramos IS NOT NULL THEN
            INSERT INTO ReservaLivro_Livro (Livro_id)
            VALUES (id_livro_procuramos);

            INSERT INTO Reserva_Livro (data_reserva,
id_funcionario, id_leitor)
            VALUES (CURRENT_DATE(), id_funcionario, id_leitor);

            SELECT CONCAT('Reserva com o id ',
id_livro_procuramos, ' feita com sucesso.') AS mensagem;
        ELSE
            -- Todas as cópias do livro estão ocupadas
            SELECT 'Não temos esse livro disponível.' AS mensagem;
        END IF;
    END IF;
END $$
DELIMITER ;

-- call addLivro('Perdido na UM',1,1);

```

```

-- drop procedure addLivro;

DELIMITER $$
CREATE PROCEDURE addLeitor(
    IN p_nome VARCHAR(30),
    IN p_email VARCHAR(30),
    IN p_n_do_CC INT,
    IN p_rua VARCHAR(45),
    IN p_localidade VARCHAR(15),
    IN p_codigo_postal VARCHAR(15),
    IN p_funcionário_id INT
)
BEGIN
    INSERT INTO Leitor (nome, email, n°_do_CC, rua, localidade,
        codigo_postal, funcionário_id)
        VALUES (p_nome, p_email, p_n_do_CC, p_rua, p_localidade,
            p_codigo_postal, p_funcionário_id);
END $$
DELIMITER ;

--          call          inserirLeitor('Joao
Figueiredo','fig@hotmail.com','256482923','Rua
direita','Ermesinde','4343-234',1);
-- drop procedure inserirLeitor;

DELIMITER $$
CREATE PROCEDURE addLivro(
    IN género_p VARCHAR(20),
    IN autor_p VARCHAR(25),
    IN título_P VARCHAR(25),
    IN sala_p INT,
    IN corredor_p INT,
    IN secção_p VARCHAR(15)
)
BEGIN
    INSERT INTO Livro (género, autor, título, sala, corredor, secção)
        VALUES (género_p, autor_p, título_P, sala_p, corredor_p, secção_p);
END $$
DELIMITER ;

```

```

-- call addLivro ('comedia','autor','Achei o meu rumo','2','3','4');
-- drop procedure addLivro;

DELIMITER $$
CREATE PROCEDURE addFuncionario(
    IN nome_p VARCHAR(30),
    IN email_p VARCHAR(35),
    IN n_telemovei_p INT,
    IN rua_p VARCHAR(45),
    IN localidade_p VARCHAR(15),
    IN codigo_postal_p VARCHAR(15)
)
BEGIN
    INSERT INTO Funcionário (nome, email, n°_telemóvel, rua, localidade,
codigo_postal)
    VALUES (nome_p, email_p, n_telemovei_p, rua_p, localidade_p,
codigo_postal_p);
END $$
DELIMITER ;

-- call addFuncionario('Helder Pereira','hp@gmail.com','986251282','rua
do sol','Braga','3421-234');
-- drop procedure addFuncionario;

```

IV. Anexo 4 – Script das Restrições

```
-- CRIAS AS RESTRICOES
create user 'funcionario'@'localhost' identified by 'password';
grant      select,insert,update,delete      on      EventsWorkbench.*      to
'funcionario'@'localhost';
grant execute on procedure getlivrosgenero to 'funcionario'@'localhost';
grant execute on procedure getlivrosautor to 'funcionario'@'localhost';
grant      execute      on      procedure      getLivrosDaSeccao      to
'funcionario'@'localhost';
grant      execute      on      procedure      addSalaEstudoGrupo      to
'funcionario'@'localhost';
grant      execute      on      procedure      addSalaEstudoIndividual      to
'funcionario'@'localhost';
grant execute on procedure addLivro to 'funcionario'@'localhost';
grant execute on procedure addLeitor to 'funcionario'@'localhost';
grant execute on procedure addFuncionario to 'funcionario'@'localhost';
grant select on getLivrosEmprestados to 'funcionario'@'localhost';
grant select on getLivrosDisponiveis to 'funcionario'@'localhost';
grant select on getLivrosAtrasados to 'funcionario'@'localhost';
grant      select      on      getSalasEstudoUsadasDesdeSempre      to
'funcionario'@'localhost';
grant      select      on      getGeneroMaisPresenteNaBiblio      to
'funcionario'@'localhost';
grant select on getHorarioMaisFrequente to 'funcionario'@'localhost';
grant select on getQuantidadeLivros to 'funcionario'@'localhost';
grant select on getSalasEstudoLivres to 'funcionario'@'localhost';
grant select on getMaisRequisitado to 'funcionario'@'localhost';
grant select on getLivrosDiferentes to 'funcionario'@'localhost';
grant select on getAutoresDiferentes to 'funcionario'@'localhost';
grant select on getLeitoresExistentes to 'funcionario'@'localhost';
grant select on getLeitoresNasSalas to 'funcionario'@'localhost';
grant select on countLivrosDisponiveis to 'funcionario'@'localhost';
```

```

grant select on countLivrosEmprestados to 'funcionario'@'localhost';
grant select on countSalasEstudoOcupadas to 'funcionario'@'localhost';
grant select on countSalasEstudoLivres to 'funcionario'@'localhost';

create user 'leitor'@'localhost' identified by 'password';
grant      select,insert,update,delete      on      EventsWorkbench.*      to
'leitor'@'localhost';
grant      select      on      getGeneroMaisPresenteNaBiblio      to
'funcionario'@'localhost';
grant select on getSalasEstudoLivres to 'funcionario'@'localhost';
grant select on getMaisRequisitado to 'funcionario'@'localhost';
grant select on getLivrosDiferentes to 'funcionario'@'localhost';
grant select on getAutoresDiferentes to 'funcionario'@'localhost';

```