



“LElvraria”

ESCOLA DE ENGENHARIA
LICENCIATURA EM ENGENHARIA
INFORMÁTICA
Bases de Dados



Ana Alves, Gonçalo Brandão & Simão Antunes (GRUPO 33)
Caso de Estudo de Sistemas de Bases de Dados
Universidade do Minho, Junho 2023

Estrutura do Trabalho

Definição do sistema

- Contextualização
- Fundamentação
- Motivação e Objetivos
- Viabilidade
- Equipa de Trabalho
- Plano de execução
- Diagrama de GANTT

Modelação Lógica

- Construção e validação
- Relacionamentos binários 1:N
- Relacionamentos binários 1:1
- Relacionamentos binários N:N
- Desenho do modelo lógico
- Normalização
- Apresentação e explicação
- Validação
- Revisão

Definição de Requisitos

- Levantamento
- Análise
- Organização
- Validação

Modelação Conceptual

- Identificação das entidades
- Identificação dos relacionamentos
- Caracterização das entidades e relacionamentos
- Elaboração do esquema conceptual
- validação



Estrutura do Trabalho

Implementação Física

- Seleção do sistema de gestão
- Tradução do esquema lógico
- Tradução das interrogações
- Definição e caracterização das vistas
- Cálculo do espaço da bases de dados
- Indexação
- Procedimentos Implementados
- Plano de segurança

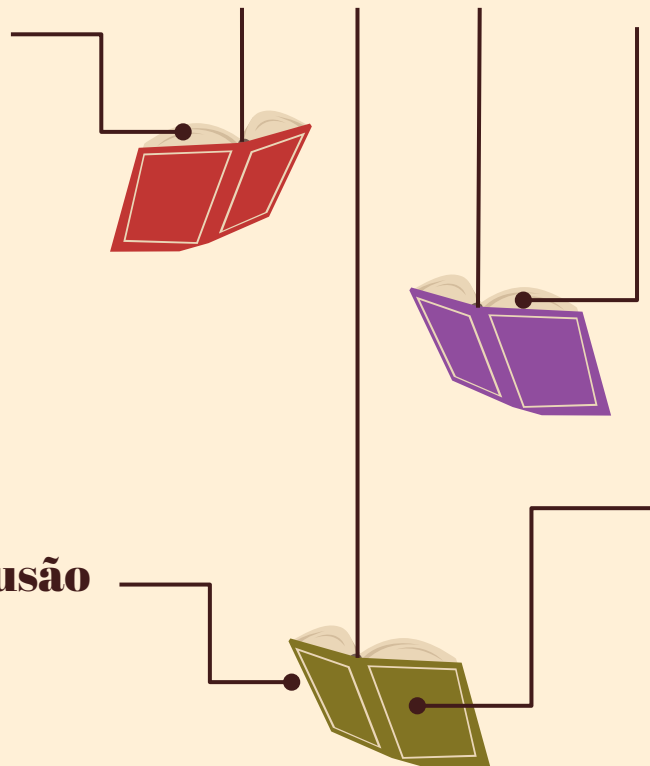
Implementação do Sistema de Recolha de Dados

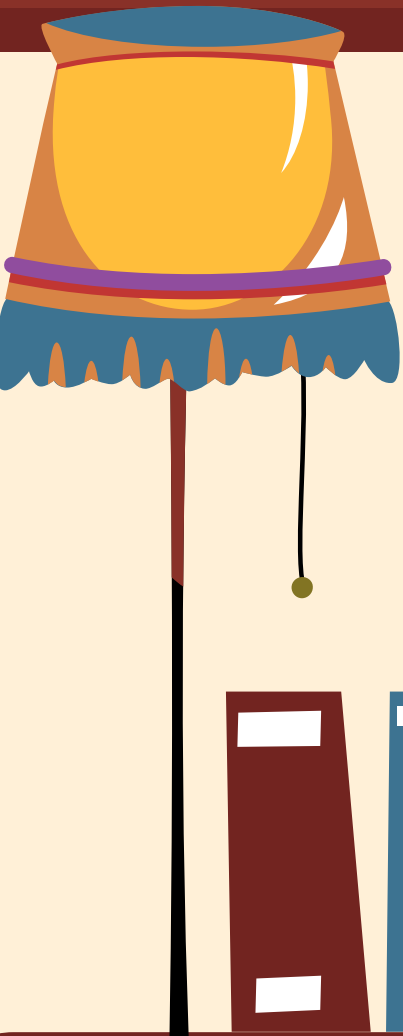
- Apresentação e modelo
- Implementação do sistema
- Funcionamento do sistema

Implementação do sistema de Painéis de Análise

- Definição e caracterização da vista de dados para análise
- Povoamento das estruturas de dados para análise
- Apresentação e caracterização dos dashboards implementados

Conclusão





Definição do Sistema



Contextualização



1965

Nascimento
do Sr. João



1987

**Abertura da
livraria**



1993

Contratação
do Zé Maria

AMPLA
ACOLHEDORA

ILUMINADA
CONVIDATIVA
INCLUSIVA



Fundamentação



Livros digitais

Diminuição da demanda
de livros físicos



Compras on-line

Menor número de visitas
na livraria

Diminuição do número de
receitas em lojas físicas



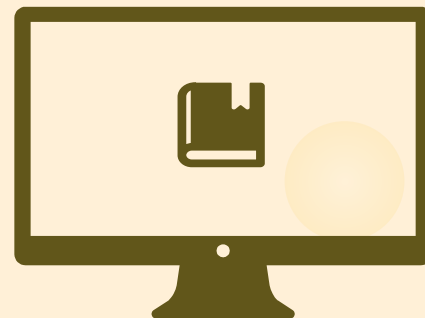
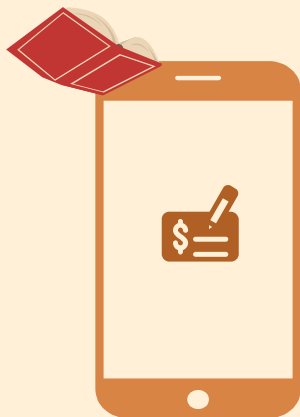
Livro de registos de vendas

Dificuldade em aceder a
dados

Perda de informação

Falta de segurança

Fundamentação





Motivação e Objetivos do Trabalho



Novos Produtos

Implementação da venda
de e-books



Aumento de receitas

Expansão para o mercado
digital



Suporte

Tratamento de dados



Viabilidade



**Tomada de
decisões**



Consistência



Eficiência



Organização





Equipa de Trabalho

Analista
Análise e tratamento de dados

Livreiro
Atendimento ao público e caixeiro



Entregador
Entrega de encomendas

Engenheiro Informático

Desenvolvimento do software e da Base de Dados

Especialista em marketing

Promoção e divulgação do software e do negócio



Plano de execução

O Sr. João reuniu-se com diversos membros da equipa de desenvolvimento da base de dados, para expor as suas ideias e discutirem-se certos aspetos, chegou-se a um consenso e estabeleceu-se um **diagrama de GANTT** de forma a organizar o trabalho já realizado e o trabalho a realizar no próximo mês para a realização da primeira parte do projeto.



Diagrama de GANTT

Implementação do Sistema de Bases de Dados

LEIvria do Sr. João
Sr. João

Project Start:

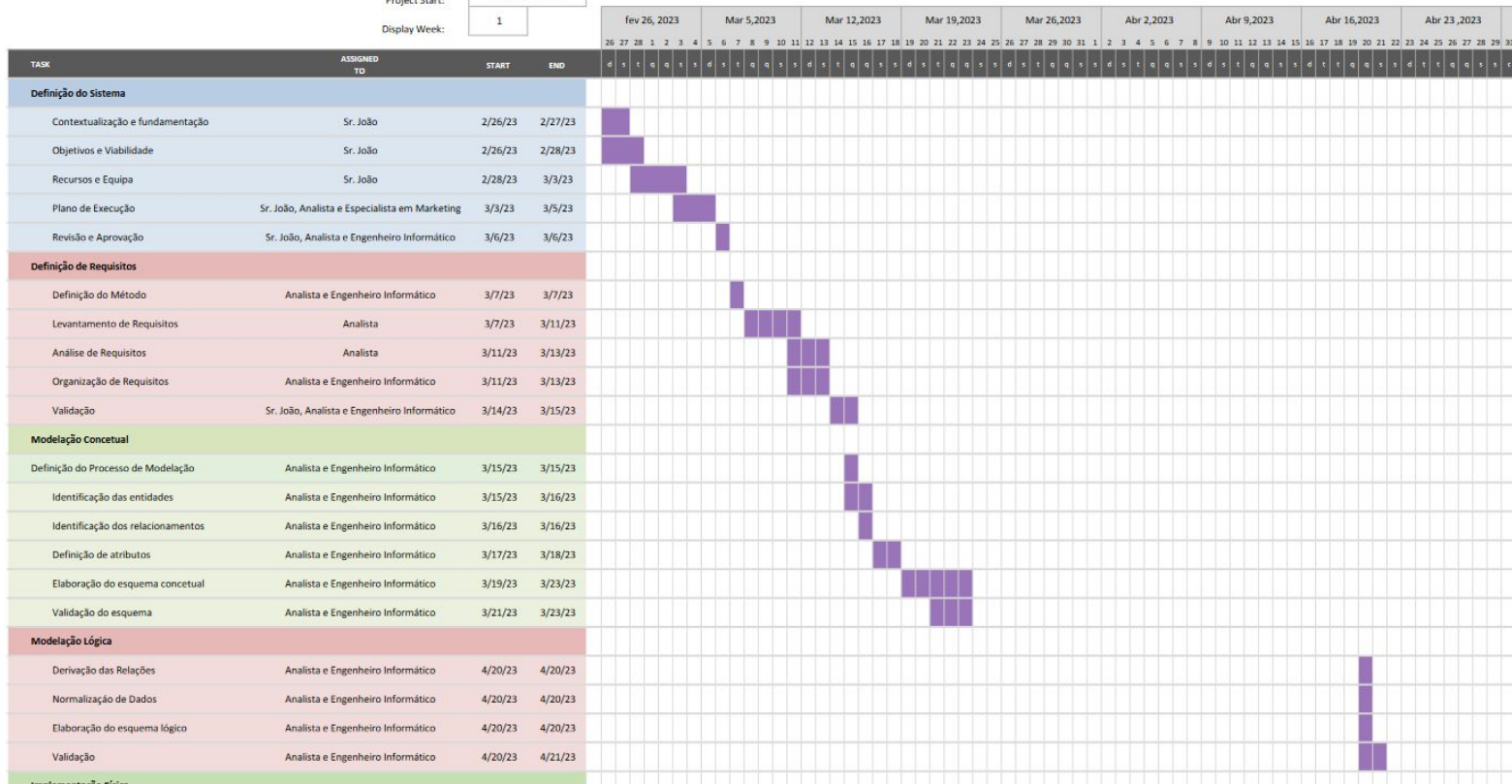
sáb,02/26/2023

Display Week:

1

SIMPLE GANTT CHART by Vertex42.com

<https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html>



Implementação Física

Diagrama de GANTT



Diagrama de Gantt



**Dependência entre
tarefas**



**Tarefas de longa
duração**



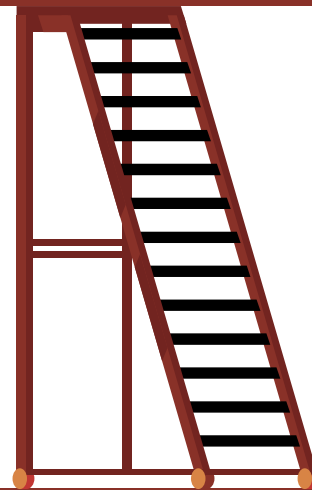
**Períodos
desajustados**



**Coordenação de
recursos**



Definição de Requisitos



Método de levantamento de requisitos



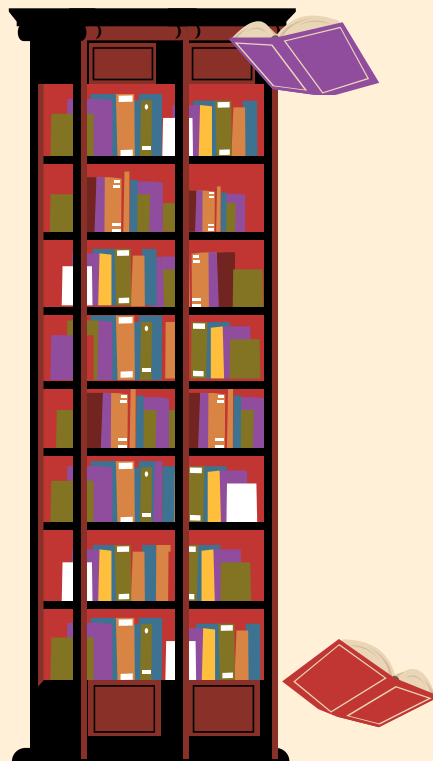
Reunião com o Sr. João

Reunião das exigências do Senhor João no software



Análise de Software

Estudo da estrutura do software



Consulta de faturas

Aprimoramento de material contábilístico



Questionário

Melhor compreensão do número de pessoas que preferem realizar compras online

Questionário-

Ata da reunião com o Sr. João

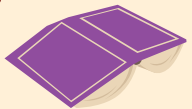
Fatura resultante

Requisitos de descrição

N.º	Data e Hora	Descrição	Fonte
RD01	3/07/2023 10:30	Os clientes devem ter um nome, um username único, uma password, um email, um número de telemóvel, um NIF, um método de pagamento e uma morada. A cada cliente está associado um identificador único (ID).	Senhor João
RD02	3/07/2023 10:32	Os livros devem ter diversas informações para tornar a sua procura mais fácil e acessível, tais como um título, autor, editora, edição, género, estado (disponível ou indisponível), formato (físico ou digital) e identificador próprio.	
RD03	3/07/2023 10:33	A compra é caracterizada por uma data, preço de custo, preço de consumidor, lucro e um identificador único.	
RD04	3/07/2023 10:35	Os funcionários devem ter todos os mesmos dados que um utilizador (excluindo o NIF, número de telemóvel, email e método de pagamento), com o acréscimo do IBAN, possuindo também um identificador.	

Requisitos de exploração

N.º	Data e Hora	Descrição	Fonte
RE01	3/08/2023 10:30	Consulta todos os livros de um dado autor	Sr.João, Analista e Engenheiro Informático
RE02	3/08/2023 10:32	Consultar todos os livros disponíveis em formato digital/físico	
RE03	3/08/2023 10:33	Consultar o lucro gerado por um cliente	
RE04	3/08/2023 10:35	Consulta as vendas realizadas num dado intervalo de tempo	
RE05	3/08/2023 10:37	Consultar os livros mais vendidos	
RE06	3/08/2023 10:39	Consultar o lucro gerado por todos os autores ordenadamente	



Requisitos de controlo

Nrº	Data e Hora	Entidade	Descrição	Fonte
RC01	3/09/2023 10:30	Cliente	Pode visualizar e alterar as suas próprias informações	Sr.João, Analista e Engenheiro Informático
RC02			Pode efetuar compras	
RC03			Pode verificar o seu histórico de compras	
RC04			Pode consultar os dados da livraria	
RC05			Pode visualizar informações sobre os livros	
RC06	3/09/2023 10:40	Funcionário	Pode consultar qualquer informação sobre os livros	
RC07			Pode consultar qualquer informação sobre os clientes	
RC08			Pode consultar e elaborar qualquer registo de venda	
RC09	3/09/2023 10:50	Administrador	Pode visualizar e modificar qualquer informação sobre dos livros	
RC10			Pode consultar, alterar e elaborar qualquer registo de venda	
RC11			Pode visualizar e modificar qualquer informação sobre os clientes	



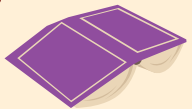
Validação geral dos requisitos

Portanto, após a recolha e análise de toda a informação, a **equipa de trabalho resolveu marcar outra reunião** para validar os mesmos de modo a garantir que os requisitos estão alinhados com as expectativas, **visando a satisfação com o produto final e procedendo com as respetivas tarefas**. A informação foi validada.

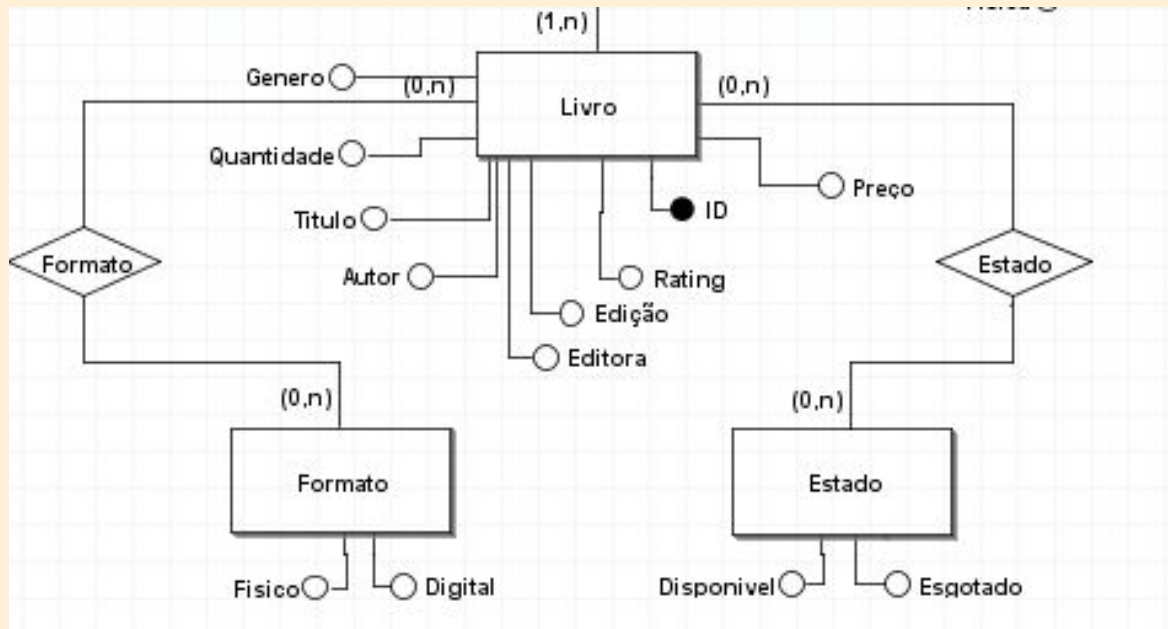


Modelação Concetual

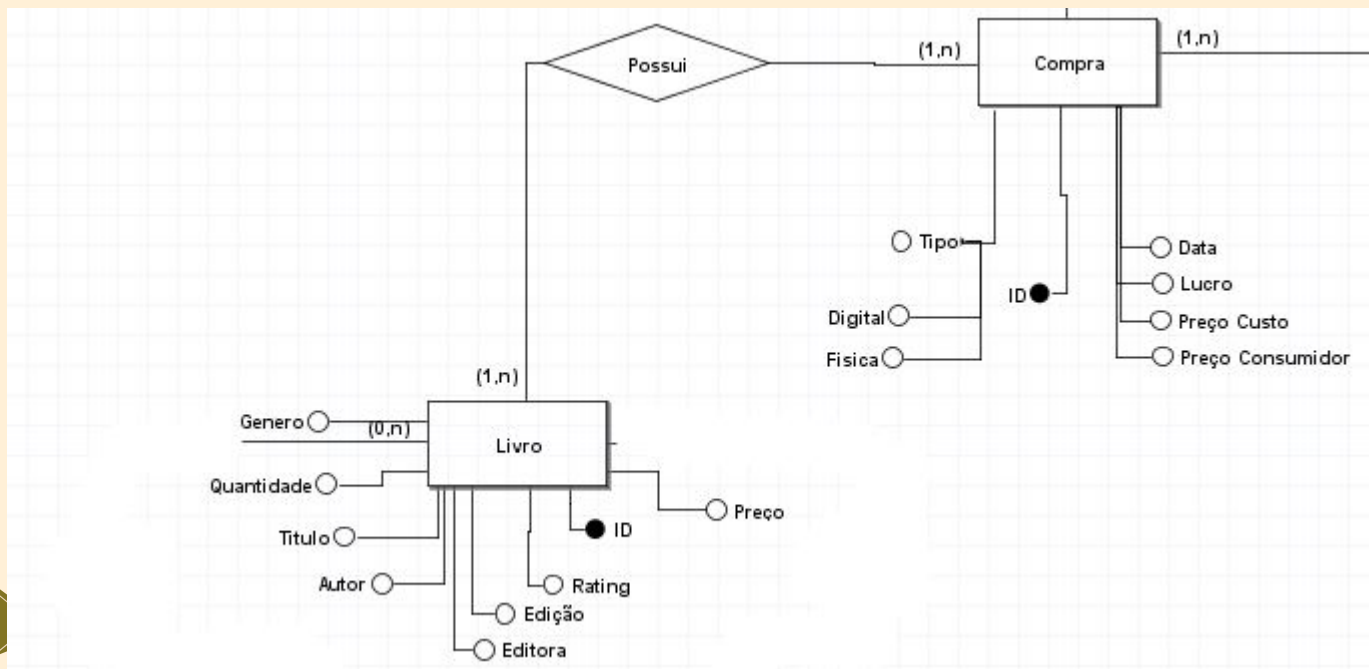


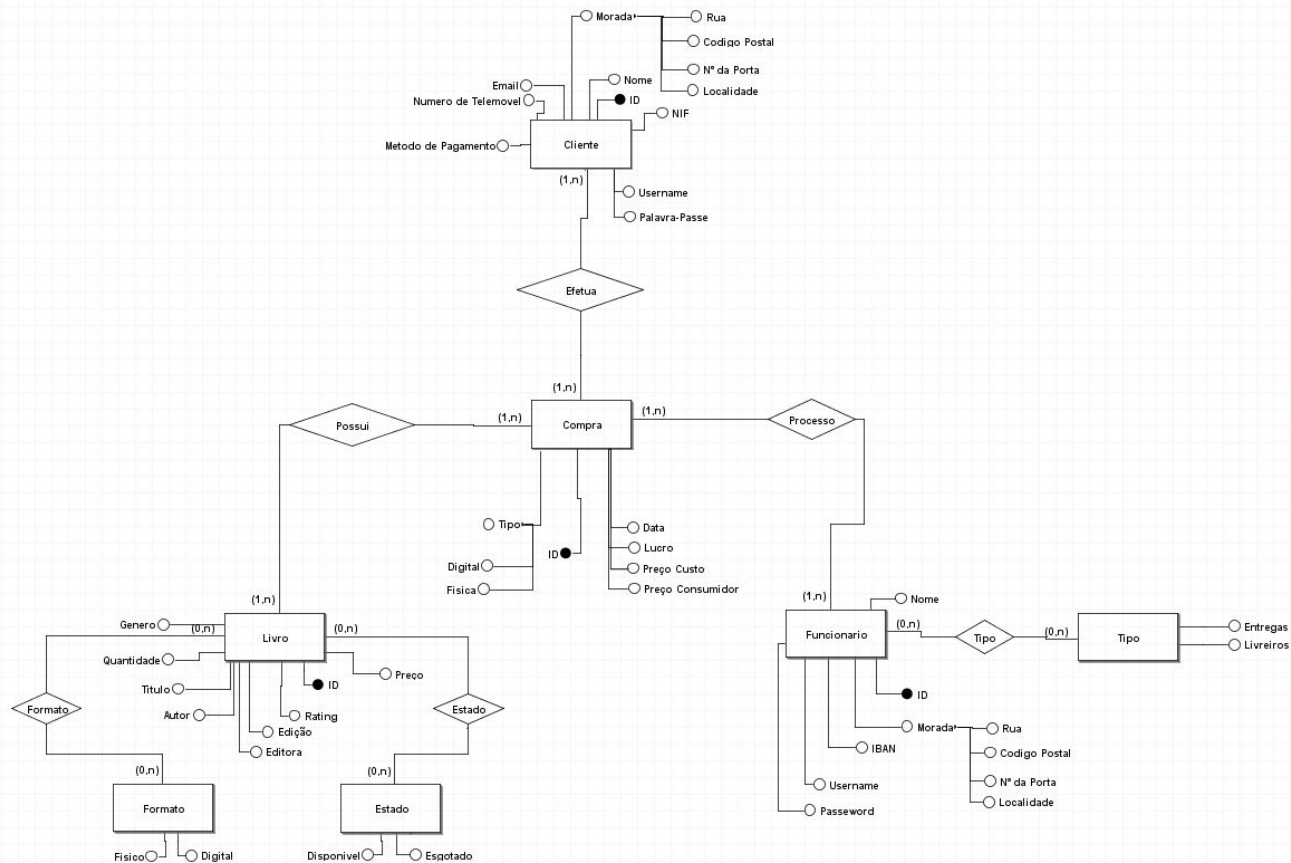


Identificação e caracterização das entidades e atributos



Identificação dos relacionamentos





Modelo Conceptual

Entidade	Atributo	Descrição	Tipo de Dados e Tamanho	Null	Tipo de atributo
Livro	Gênero	Gênero literário referente a cada livro	VARCHAR (50)	Não	Simples
	Quantidade	Nº de Livros disponíveis	INT	Não	Simples
	Título	Título do livro	VARCHAR (50)	Não	Simples
	Autor	Nome do autor do livro	VARCHAR (50)	Não	Simples
	Editora	Nome da editora	VARCHAR (50)	Não	Simples

Figura - Tabelas de Dados

Validação



A **equipa de trabalho resolveu marcar outra reunião** para validar o modelo concetual para prosseguir com o projeto. O modelo foi aprovado.



Modelação Lógica





Construção e validação do modelo



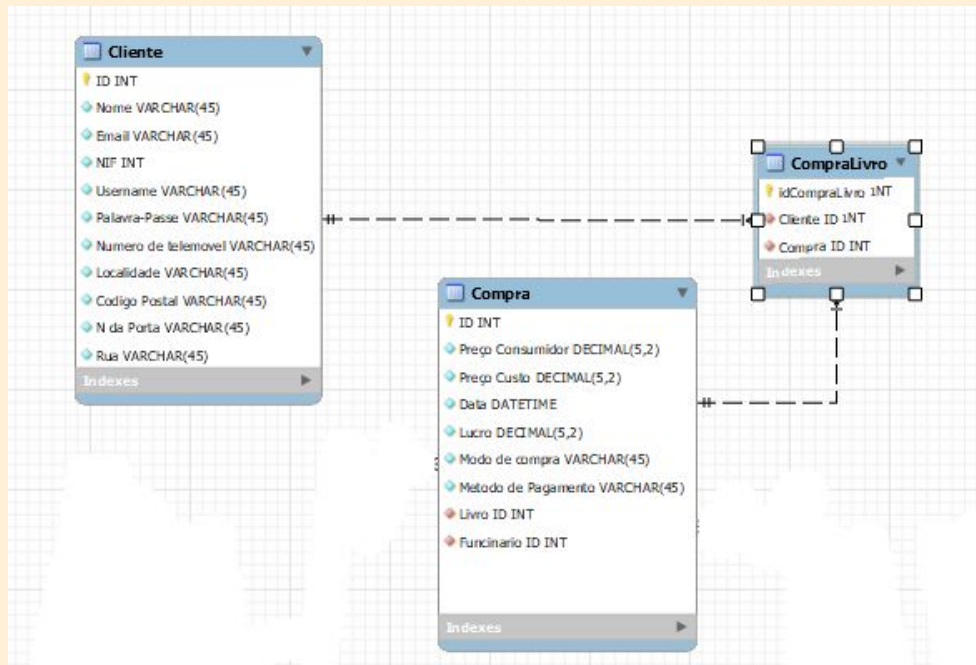
Objetivos;

- Realizar um modelo lógico que represente o nosso modelo conceptual
- Realizar um modelo lógico que melhore a implementação da nossa base de dados
- Realizar um modelo perceptível, prático e dinâmico



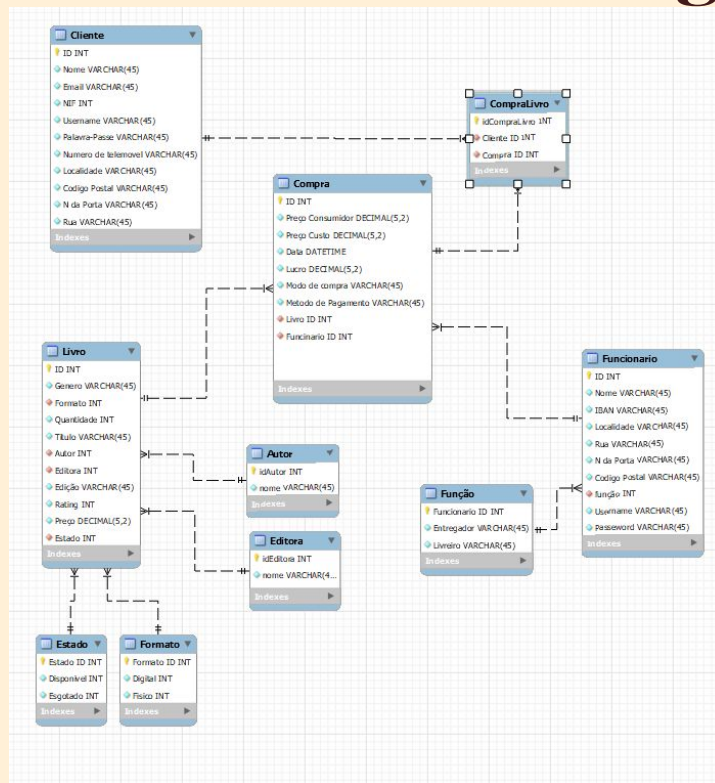


Relacionamentos binários





Desenho do modelo lógico



Normalização de dados



- 1FN - Toda relação deve ter uma chave primária e deve-se garantir que todo atributo seja atômico.
- 2FN - Uma relação está na 2ª Forma Normal se está na primeira e se todos os atributos não-chave dependerem da totalidade da chave-primária.
- 3FN - Uma relação está na 3ª Forma Normal se está na 2ª Forma Normal e nenhum atributo não-chave é transitivamente dependente da chave primária.



Revisão



A **equipa de trabalho resolveu marcar outra reunião** para validar o modelo lógico para prosseguir com o projeto. Sendo este aprovado.



Implementação Física





Seleção do SGBD

Lecionado

Fácil Manuseamento

**Suporte a linguagens de
Programação**

Escalabilidade

Compatibilidade



Alto Desempenho



Tradução do esquema lógico

01

Descrição das relações base

02

Definição de restrições gerais

03

Otimização e Desempenho

04

Configuração de parâmetros do SGBD

05

Implementação do esquema físico



Descrição das relações base

Relação Livro

Domínio `ID`: INT NOT NULL, (Inteiro)

Domínio `Gênero`: VARCHAR(45) NOT NULL,
(String com um tamanho máximo de 45 caracteres)

Domínio `Formato`: INT NOT NULL, (Inteiro)

Domínio `Quantidade`: INT NOT NULL, (Inteiro)

Domínio `Titulo`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Autor`: INT NOT NULL, (Inteiro) (Chave Estrangeira)

Domínio `Editora`: INT NOT NULL, (Inteiro) (Chave Estrangeira)

Domínio `Edição`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Rating`: INT NOT NULL, (Inteiro)

Domínio `Preço`: DECIMAL(5,2) NOT NULL, (Valor com 5 dígitos com duas casas decimais)

Domínio `Estado`: INT NOT NULL, (Inteiro) (Chave estrangeira)



Descrição das relações base

Relação Cliente

Domínio `ID`: INT NOT NULL, (Inteiro)

Domínio `Nome`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Email`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `NIF`: INT NOT NULL, (Inteiro)

Domínio `Username`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Palavra-Passe`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Numero de telemovel`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Localidade`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Codigo Postal`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `N da Porta`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Rua`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)



Descrição das relações base

Relação Compra

Domínio `ID`: INT NOT NULL, (Inteiro)

Domínio `Preco Consumidor`: DECIMAL(5,2) NOT NULL, (Valor com 5 dígitos com duas casas decimais)

Domínio `Preco Custo`: DECIMAL(5,2) NOT NULL, (Valor com 5 dígitos com duas casas decimais)

Domínio `Data`: DATETIME NOT NULL, (Data e tempo)

Domínio `Lucro`: DECIMAL(5,2) NOT NULL, (Valor com 5 dígitos com duas casas decimais)

Domínio `Modo de compra`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Metodo de Pagamento`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Livro ID`: INT NOT NULL, (Inteiro)(Chave Estrangeira)

Domínio `Funcionario ID`: INT NOT NULL, (Inteiro)(Chave Estrangeira)



Descrição das relações base

Relação Funcionário

Domínio `ID`: INT NOT NULL, (Inteiro)

Domínio `Nome`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Username`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Palavra-Passe`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `IBAN`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Localidade`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Rua`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `N da Porta`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `Codigo Postal`: VARCHAR(45) NOT NULL, (String com um tamanho máximo de 45 caracteres)

Domínio `função`: INT NOT NULL, (Inteiro)(Chave Estrangeira)



Desenho das restrições gerais



Livro

```
-- Table `Livraria`.`Livro`
```

```
CREATE TABLE IF NOT EXISTS `Livraria`.`Livro` (
```

```
  `ID` INT NOT NULL,
```

```
  `Genero` VARCHAR(45) NOT NULL,
```

```
  `Formato` INT NOT NULL,
```

```
  `Quantidade` INT NOT NULL,
```

```
  `Titulo` VARCHAR(45) NOT NULL,
```

```
  `Autor` INT NOT NULL,
```

```
  `Editora` INT NOT NULL,
```

```
  `Edição` VARCHAR(45) NOT NULL,
```

```
  `Rating` INT NOT NULL,
```

```
  `Preço` DECIMAL(5,2) NOT NULL,
```

```
  `Estado` INT NOT NULL,
```

```
  PRIMARY KEY (`ID`),
```

```
  INDEX `autor_idx` (`Autor` ASC) VISIBLE,
```

```
  INDEX `editora_idx` (`Editora` ASC) VISIBLE,
```

```
  INDEX `formato_idx` (`Formato` ASC) VISIBLE,
```

```
  INDEX `estado_idx` (`Estado` ASC) VISIBLE,
```

```
  INDEX `estado_idx` (`Estado` ASC) VISIBLE,
```

```
  CONSTRAINT `fk_autor`
```

```
    FOREIGN KEY (`Autor`)
```

```
    REFERENCES `Livraria`.`Autor` (`idAutor`)
```

```
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION,
```

```
  CONSTRAINT `fk_editora`
```

```
    FOREIGN KEY (`Editora`)
```

```
    REFERENCES `Livraria`.`Editora` (`idEditora`)
```

```
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION,
```

```
  CONSTRAINT `fk_formato`
```

```
    FOREIGN KEY (`Formato`)
```

```
    REFERENCES `Livraria`.`Formato` (`Formato ID`)
```

```
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION,
```

```
  CONSTRAINT `fk_estado`
```

```
    FOREIGN KEY (`Estado`)
```

```
    REFERENCES `Livraria`.`Estado` (`Estado ID`)
```

```
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION
```

```
)
```

```
ENGINE = InnoDB;
```



Desenho das restrições gerais



Cliente

```
-- Table `Livraria`.`Cliente`  
  
CREATE TABLE IF NOT EXISTS `Livraria`.`Cliente` (  
  `ID` INT NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `Email` VARCHAR(45) NOT NULL,  
  `NIF` INT NOT NULL,  
  `Username` VARCHAR(45) NOT NULL,  
  `Palavra-Passe` VARCHAR(45) NOT NULL,  
  `Numero de telemovel` VARCHAR(45) NOT NULL,  
  `Localidade` VARCHAR(45) NOT NULL,  
  `Codigo Postal` VARCHAR(45) NOT NULL,  
  `N da Porta` VARCHAR(45) NOT NULL,  
  `Rua` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`ID`)  
)  
  
ENGINE = InnoDB;
```



Desenho das restrições gerais



Funcionário

```
-- Table `Livraria`.`Funcionario`  
-----  
CREATE TABLE IF NOT EXISTS `Livraria`.`Funcionario` (  
  `ID` INT NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `Username` VARCHAR(45) NOT NULL,  
  `Palavra-Passe` VARCHAR(45) NOT NULL,  
  `IBAN` VARCHAR(45) NOT NULL,  
  `Localidade` VARCHAR(45) NOT NULL,  
  `Rua` VARCHAR(45) NOT NULL,  
  `N da Porta` VARCHAR(45) NOT NULL,  
  `Codigo Postal` VARCHAR(45) NOT NULL,  
  `função` INT NOT NULL,  
  PRIMARY KEY (`ID`),  
  INDEX `funcao_idx` (`função` ASC) VISIBLE,  
  CONSTRAINT `fk_funcao`  
    FOREIGN KEY (`função`)  
    REFERENCES `Livraria`.`Função` (`Funcionario ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
)  
ENGINE = InnoDB;
```



Desenho das restrições gerais



Compra

```
-- Table `Livraria`.`Compra`  
-----  
CREATE TABLE IF NOT EXISTS `Livraria`.`Compra` (  
  `ID` INT NOT NULL,  
  `Preço Consumidor` DECIMAL(5,2) NOT NULL,  
  `Preço Custo` DECIMAL(5,2) NOT NULL,  
  `Data` DATETIME NOT NULL,  
  `Lucro` DECIMAL(5,2) NOT NULL,  
  `Modo de compra` VARCHAR(45) NOT NULL,  
  `Metodo de Pagamento` VARCHAR(45) NOT NULL,  
  `Livro ID` INT NOT NULL,  
  `Funcionario ID` INT NOT NULL,  
  PRIMARY KEY (`ID`),  
  INDEX `Livro ID_idx` (`Livro ID` ASC) VISIBLE,  
  INDEX `Funcionario ID_idx` (`Funcionario ID` ASC) VISIBLE,  
  CONSTRAINT `fk_livro_id`  
    FOREIGN KEY (`Livro ID`)  
    REFERENCES `Livraria`.`Livro` (`ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_funcionario_id`  
    FOREIGN KEY (`Funcionario ID`)  
    REFERENCES `Livraria`.`Funcionario` (`ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
)  
ENGINE = InnoDB;
```



Tradução das interrogações



```
SELECT SUM(c.Lucro) AS LucroTotal  
FROM Livraria.Cliente cl  
JOIN Livraria.CompraLivro clv ON cl.ID = clv.`Cliente ID`  
JOIN Livraria.Compra c ON clv.`Compra ID` = c.ID  
WHERE cl.ID = 8;
```



Lucro gerado pelo cliente com id 8



Tradução das interrogações

```
SELECT Livro.Titulo, COUNT(*) AS TotalVendas
FROM Livraria.Livro
JOIN Livraria.Compra clv ON Livro.ID = clv.`Livro ID`
JOIN Livraria.CompraLivro c ON clv.ID = c.`Compra ID`
GROUP BY Livro.Titulo
ORDER BY TotalVendas DESC;
```





Tradução das interrogações

```
SELECT a.idAutor, a.nome, SUM(c.Lucro) AS LucroTotal
FROM Livraria.Autor a
JOIN Livraria.Livro l ON a.idAutor = l.`Autor`
JOIN Livraria.Compra c ON l.`ID` = c.`Livro ID`
JOIN Livraria.CompraLivro clv ON c.ID = clv.`Compra ID`
GROUP BY a.idAutor, a.nome
ORDER BY LucroTotal DESC;
```



Autores que geram maior lucro

Definição e caracterização das vistas



```
CREATE VIEW LivrosDisponiveis AS
SELECT L.ID, L.Titulo, L.Genero, L.Quantidade, L.Preço, A.nome AS Autor, E.nome AS Editora
FROM Livro L
JOIN Autor A ON L.Autor = A.idAutor
JOIN Editora E ON L.Editora = E.idEditora
WHERE L.Estado = 1;
```



Definição e caracterização das vistas



```
CREATE VIEW ComprasPorCliente AS
SELECT C.ID AS ClienteID, C.Nome AS ClienteNome, COUNT(CL.idCompraLivro) AS TotalCompras
FROM Cliente C
LEFT JOIN CompraLivro CL ON C.ID = CL.`Cliente ID`
GROUP BY C.ID, C.Nome;
```



Definição e caracterização das vistas



```
CREATE VIEW LivrosPorAutor AS  
SELECT A.nome AS Autor, COUNT(L.ID) AS TotalLivros  
FROM Autor A  
LEFT JOIN Livro L ON A.idAutor = L.Autor  
GROUP BY A.nome;
```



Cálculo do espaço da Base de Dados



Exemplo da tabela dos Autores

Tabela de ocupação após o povoamento inicial

Tabela	Atributo	Tipos de dados	Espaço Ocupado(bytes)
Autor	idAutor	INT	4
	nome	VARCHAR(45)	45
Total	Nº de entradas* Tamanho de uma entrada		$30 \times 49 = 1470$



Tabela de ocupação anual

Tabela	Atributo	Tipos de dados	Espaço Ocupado(bytes)
Autor	idAutor	INT	4
	nome	VARCHAR(45)	45
Total	Nº de entradas* Tamanho de uma entrada		$35 \times 49 = 1715$

Cálculo do espaço da Base de Dados



Tabela	Ocupação inicial(bytes)	Ocupação anual (bytes)
Cliente	12390	20650
Livro	5100	6800
Editora	980	1225
Autor	1470	1715
Compra	8460	11280
CompraLivro	360	480
Formato	24	24
Estado	24	24
Funcionário	1104	1840
Função	282	282





Indexação

```
CREATE INDEX IDAutor ON Autor(idAutor);  
CREATE INDEX NomeAutor ON Autor(nome);  
CREATE INDEX TituloLivro ON Livro(Titulo);  
CREATE INDEX IDCompra ON Compra(ID);  
CREATE INDEX IDCliente ON Cliente(ID);
```





Procedimentos Implementados



```
CREATE PROCEDURE InserirLivro(  
    IN id INT,  
    IN genero VARCHAR(45),  
    IN formato INT,  
    IN quantidade INT,  
    IN titulo VARCHAR(45),  
    IN autor INT,  
    IN editora INT,  
    IN edicao VARCHAR(45),  
    IN rating INT,  
    IN preco DECIMAL(5,2),  
    IN estado INT  
)  
BEGIN  
    INSERT INTO Livro (ID, Genero, Formato, Quantidade, Titulo, Autor, Editora, Edição, Rating, Preço, Estado)  
    VALUES (id, genero, formato, quantidade, titulo, autor, editora, edicao, rating, preco, estado);  
END //  
  
DELIMITER ;
```





Procedimentos Implementados

```
CREATE PROCEDURE AtualizarQuantidadeLivro(  
    IN livro_id INT,  
    IN nova_quantidade INT  
)  
  
BEGIN  
    UPDATE Livro  
    SET Quantidade = nova_quantidade  
    WHERE ID = livro_id;  
END //
```

DELIMITER ;



Plano de segurança e recuperação de dados



Permissões

Permissões de consulta SELECT nas tabelas 'Livro', 'Formato', 'Editora', 'Estado', 'Cliente' e 'CompraLivro'.



Revogações

Revogadas as permissões DROP, CREATE, DELETE, UPDATE e INSERT para o 'Funcionario' nas tabelas 'Livro', 'Formato', 'Editora', 'Estado', 'Cliente' e 'CompraLivro'.

Funcionário

```
-- Funcionario

CREATE USER 'Funcionario'@'localhost'
  identified by 'funcpassword';

GRANT SELECT ON Livraria.Livro TO 'Funcionario'@'localhost';
GRANT SELECT ON Livraria.Formato TO 'Funcionario'@'localhost';
GRANT SELECT ON Livraria.Editora TO 'Funcionario'@'localhost';
GRANT SELECT ON Livraria.Estado TO 'Funcionario'@'localhost';
GRANT SELECT ON Livraria.Cliente TO 'Funcionario'@'localhost';
GRANT SELECT ON Livraria.CompraLivro TO 'Funcionario'@'localhost';

REVOKE DROP,CREATE,DELETE,UPDATE,INSERT
ON Livraria.Livro
FROM 'Funcionario'@'localhost';

REVOKE DROP,CREATE,DELETE,UPDATE,INSERT
ON Livraria.Formato
FROM 'Funcionario'@'localhost';

REVOKE DROP,CREATE,DELETE,UPDATE,INSERT
ON Livraria.Editora
FROM 'Funcionario'@'localhost';

REVOKE DROP,CREATE,DELETE,UPDATE,INSERT
ON Livraria.Estado
FROM 'Funcionario'@'localhost';
```


Plano de segurança e recuperação de dados



Administrador



```
-- Administrador

CREATE USER 'Administrador' @ 'localhost'
    identified by 'adminpassword';
GRANT SELECT, INSERT, DELETE, UPDATE ON *.* TO 'Administrador' @ 'localhost';
REVOKE DROP, CREATE ON *.* FROM 'Administrador' @ 'localhost';
```

● ————— Permissões

● ————— Revogações



Plano de segurança e recuperação de dados



Cliente



```
-- Cliente
CREATE USER 'Cliente'@'localhost'
  identified by 'clientepassword';

GRANT SELECT ON Livraria.Livro TO 'Cliente'@'localhost';
GRANT SELECT ON Livraria.Formato TO 'Cliente'@'localhost';
GRANT SELECT ON Livraria.Editora TO 'Cliente'@'localhost';
GRANT SELECT ON Livraria.Estado TO 'Cliente'@'localhost';
GRANT SELECT ON Livraria.Autor TO 'Cliente'@'localhost';

GRANT SELECT, DELETE, UPDATE
(ID, Nome, Email, NIF, Username, 'Palavra-Passe', 'Numero de telemovel', Localidade, 'Codigo Postal', 'N da Porta', Rua) ON Livraria.Cliente
TO 'Cliente'@'localhost';
GRANT SELECT, DELETE, INSERT, UPDATE ON Livraria.Cliente TO 'Cliente'@'localhost';

REVOKE DROP, CREATE
ON Livraria.Cliente
FROM 'Cliente'@'localhost';
```

Permissões

Revogações

Revisão do sistema implementado



Analizando todos os parâmetros anteriormente abordados com o cliente, podemos afirmar que o nosso modelo foi **aprovado**, uma vez que é tido em conta uma solução que responde a todas as necessidades pretendidas.



Implementação do Sistema de Recolha de Dados





Apresentação e modelo do sistema



CLI

Interface que permite que os utilizadores possam inserir comandos SQL diretamente no programa para executar inserções, atualizações e outras operações na base de dados.



Metodologia

Foram usadas duas metodologias do povoamento: povoamento manual em SQL e o povoamento com CSV usando Python.

Implementação do sistema de recolha



Povoamento Manual

- **Inserção de dados** diretamente na base de dados
- Adequado para um número limitado de registos a serem inseridos e quando se está familiarizado com a estrutura da tabela e os tipos de dados que devem ser inseridos.
- O processo geralmente envolve a escrita de comandos **INSERT INTO** para cada registo e sua execução no ambiente SQL.



```
INSERT INTO `Livraria`.`Autor`  
(idAutor, nome)  
VALUES  
(1, 'José Saramago'),  
(2, 'Fernando Pessoa'),  
(3, 'Miguel Torga'),  
(4, 'José Luís Peixoto'),  
(5, 'Almeida Garrett'),  
(6, 'Sophia de Mello Breyner Andresen'),  
(7, 'António Lobo Antunes'),  
(8, 'Luís de Camões'),  
(9, 'Eça de Queirós'),  
(10, 'Agustina Bessa-Luís'),  
(11, 'Vergílio Ferreira'),  
(12, 'Camilo Castelo Branco'),  
(13, 'Fernando Namora'),  
(14, 'Jorge Amado'),  
(15, 'José Cardoso Pires'),  
(16, 'António Nobre'),  
(17, 'Vasco Graça Moura'),  
(18, 'Carlos Drummond de Andrade'),  
(19, 'Clarice Lispector'),  
(20, 'Gonçalo M. Tavares'),  
(21, 'Mário de Sá-Carneiro'),  
(22, 'Júlio Dinis'),  
(23, 'Miguel Sousa Tavares'),  
(24, 'Florbelá Espanca'),
```

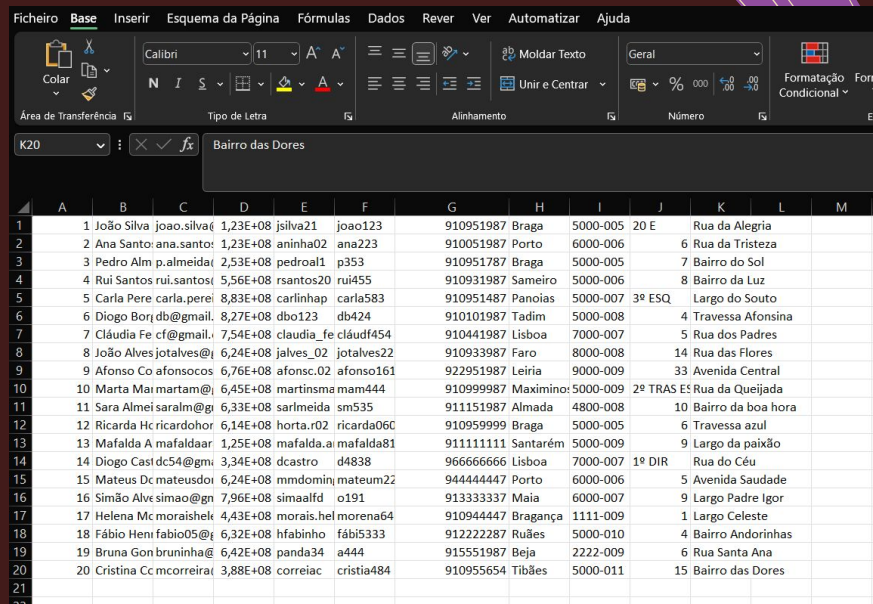


Implementação do sistema de recolha

Povoamento por CSV usando Python



- Oferece uma abordagem mais eficiente e escalável quando se lida com grandes volumes de dados.
- Os dados são armazenados em um arquivo CSV criado com informações que consideramos adequadas para o projeto.
- Formato simples e amplamente suportado para armazenar informações tabulares.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1		1	João Silva	joao.silva@	1,23E+08	jsilva21	joao123	910951987	Braga	5000-005	20 E		Rua da Alegria
2		2	Ana Santo: ana.santo:	1,23E+08	aninha02	ana223	910051987	Porto	6000-006				6 Rua da Tristeza
3		3	Pedro Alm p.almeida:	2,53E+08	pedroal1	p353	910951787	Braga	5000-005				7 Bairro do Sol
4		4	Rui Santos rui.santos:	5,56E+08	rsantos20	ruia455	910931987	Sameiro	5000-006				8 Bairro da Luz
5		5	Carla Pere carla.pere:	8,83E+08	carlinhap	carla583	910951487	Panoias	5000-007	3ª ESQ			Largo do Souto
6		6	Diogo Borj db@gmail.	8,27E+08	dbo123	db424	910101987	Tadim	5000-008				4 Travessa Afonsina
7		7	Cláudia Fe cf@gmail.	7,54E+08	claudia_fe	cláudf454	910441987	Lisboa	7000-007				5 Rua dos Padres
8		8	João Alves jotalves@j	6,24E+08	jvalves_02	jotalves22	910933987	Faro	8000-008				14 Rua das Flores
9		9	Afonso Co afonsocos	6,76E+08	afonsco.02	afonso161	922951987	Leiria	9000-009				33 Avenida Central
10		10	Marta Mai martam@j	6,45E+08	martinsma	mam444	910999987	Maximino:	5000-009	2ª TRAS	ES		Rua da Queijada
11		11	Sara Almei saralm@g	6,33E+08	sarlmeida	sm535	911151987	Almada	4800-008				10 Bairro da boa hora
12		12	Ricarda Hc ricardohor	6,14E+08	horta.r02	ricarda060	910959999	Braga	5000-005				6 Travessa azul
13		13	Mafalda A mafaldaar	1,25E+08	mafalda.a	mafalda81	911111111	Santarém	5000-009				9 Largo da paixão
14		14	Diogo Cast dc54@gm:	3,34E+08	dcastro	d4838	966666666	Lisboa	7000-007	1ª DIR			Rua do Céu
15		15	Mateus Dc mateusdoi	6,24E+08	mmdomin	mateum22	944444447	Porto	6000-006				5 Avenida Saudade
16		16	Simão Alc simao@gn	7,96E+08	simaalfd	o191	913333337	Maia	6000-007				9 Largo Padre Igor
17		17	Helena Mc moraishek	4,43E+08	morais.hel	morena64	910944447	Bragança	1111-009				1 Largo Celeste
18		18	Fábio Henri fabio05@j	6,32E+08	hfabinho	fábi5333	912222287	Ruões	5000-010				4 Bairro Andorinhas
19		19	Bruna Gon bruninha@	6,42E+08	panda34	a444	915551987	Beja	2222-009				6 Rua Santa Ana
20		20	Cristina Cc mcorreirai	3,88E+08	correiac	cristia484	910955654	Tibães	5000-011				15 Bairro das Dores
21													

Implementação do sistema de recolha

Povoamento por CSV usando Python

- **Conexão com a base de dados**, criando um objecto cursor e lê os dados do arquivo CSV.
- **O cursor é responsável por enviar comandos SQL para a base de dados e receber os resultados, inserindo os dados.**
- **Executa um loop para processar cada linha do arquivo CSV, extrair os valores e inseri-los na tabela por meio de comandos SQL.**



```
import csv
import mysql.connector
from mysql.connector import errorcode

# Connect to the database
cnx = mysql.connector.connect(
    host='localhost',
    user='root',
    password='Anali012002.',
    database='Livraria'
)
cursor = cnx.cursor()

# Migrar para clientes

# Read data from CSV and insert into the table
with open('../datasets/clientes.csv', 'r') as csvfile:
    csvreader = csv.reader(csvfile, delimiter=',')
    next(csvreader) # Skip header row

    for row in csvreader:
        ID = int(row[0])
        Nome = row[1]
        Email = row[2]
        NIF = row[3]
        Username = row[4]
        PalavraPasse = row[5]
        Numerodetelemovei = row[6]
        Localidade = row[7]
        Codigopostal = row[8]
        NdaPorta = row[9]
        Rua = row[10]

        insert_query = "INSERT INTO 'Livraria'. 'Cliente' (ID, Nome, Email, NIF, Username, 'Palavra-Passe', 'Número de telemovel', Localidade, 'Codigo Postal', 'N da Porta', Rua) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"
        values = (ID, Nome, Email, NIF, Username, PalavraPasse, Numerodetelemovei, Localidade, Codigopostal, NdaPorta, Rua)
        cursor.execute(insert_query, values)

    cnx.commit()
    cursor.close()
    cnx.close()

print("SUCESSO!")
```


Funcionamento do Sistema



Após o preenchimento dos campos, os dados são enviados para o sistema, onde são armazenados na base de dados. O sistema pode realizar validações e verificações para garantir que os dados inseridos estejam corretos e em conformidade com as regras de negócio estabelecidas.



Result Grid		Filter Rows:	Edit:
	idAutor	nome	
▶	1	José Saramago	
	2	Fernando Pessoa	
	3	Miguel Torga	
	4	José Luís Peixoto	
	5	Almeida Garrett	
	6	Sophia de Mello Breyner Andresen	
	7	António Lobo Antunes	
	8	Luís de Camões	
	9	Eça de Queirós	
	10	Agustina Bessa-Luís	
	11	Vergílio Ferreira	
	12	Camilo Castelo Branco	
	13	Fernando Namora	
	14	Jorge Amado	
	15	José Cardoso Pires	
	16	António Nobre	
	17	Vasco Graça Moura	
	18	Carlos Drummond de Andrade	
	19	Clarice Lispector	
	20	Gonçalo M. Tavares	
	21	Mário de Sá-Carneiro	
	22	Júlio Dinis	
	23	Miguel Sousa Tavares	
	24	Florbela Espanca	
	25	Manuel Alegre	

Tabela dos autores povoada

Implementação do Sistema de Painéis de Análise





Definição e caracterização da vista de dados

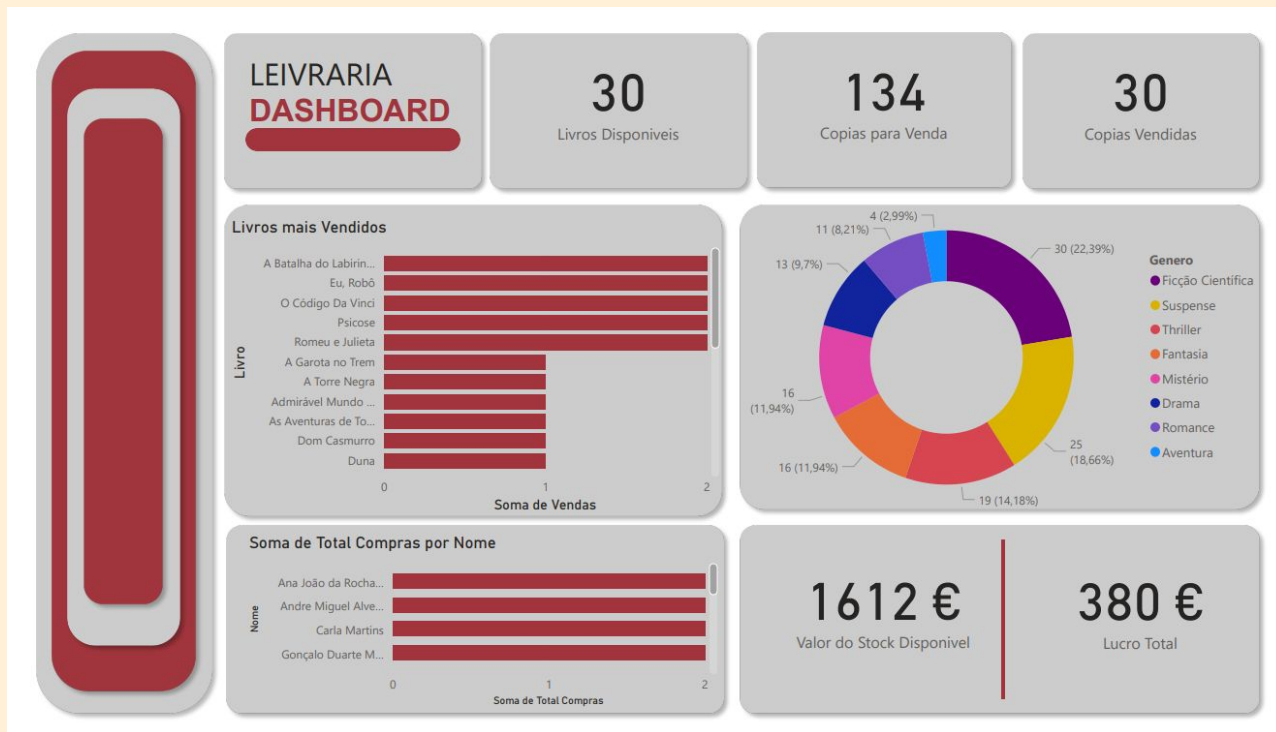
Objetivos:

- Ponto atual da Livraria
- Stock da Livraria
- Preferências dos clientes
- Resumo económico





Apresentação e caracterização dos dashboards



Conclusão



DIFICULDADES

Necessidade de
reestruturação

Organização dos dados



CONSEQUÊNCIA

Enfatização dos processos
de validação e modelação

Planeamento da estrutura
e dados





“LElvraria”

ESCOLA DE ENGENHARIA
LICENCIATURA EM ENGENHARIA
INFORMÁTICA
Base de Dados



Ana Alves, Gonçalo Brandão & Simão Antunes (GRUPO 33)
Caso de Estudo de Sistemas de Bases de Dados
Universidade do Minho, Junho 2023