



Universidade do Minho
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2022/2023

AmazóniaTM

**Breno Marrão A97768, Jorge Silva A80931,
Tales Rovaris A96314, Tiago Rodrigues A96414**

01/23

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

AmazóniaTM

**Breno Marrão A97768, Jorge Silva A80931,
Tales Rovaris A96314, Tiago Rodrigues A96414**

Resumo

Este trabalho tem como o objetivo o planeamento, arquitetura e implementação de uma base de dados de Gestão de entregas a domicílio. O objetivo da mesma é proporcionar um meio de realizar entregas de um armazém da 'AmazoniaTM'. Dado o objetivo, nós iniciamos com a definição do sistema (contextualização, motivação, objetivo, viabilidade), depois disso passamos para a formação dos requisitos de Controlo, exploração e descrição através de reuniões com o dono, após isso definimos o modelo conceptual através dos requisitos de descrição e a partir do conceptual fizemos a definição do modelo lógico e posteriormente definimos o físico. Entre cada um desses processos é realizado uma validação e análise destes passos.

Tendo isto feito começamos a trabalhar com o modelo físico ao definir triggers, os utilizadores através dos requisitos de controlo, e as queries e os procedures de acordo com os requisitos de exploração.

Finalizamos o projeto tirando conclusões sobre o trabalho que realizamos e elaborando uma análise de todo o processo, o que poderíamos ter feito melhor, os problemas pelos quais passamos, o que tiramos de aprendizagem através da execução deste projeto.

Área de Aplicação: Desenho e arquitetura de Sistemas de Bases de Dados

Palavras-Chave: Bases de dados Relacional, Modelo Conceptual, Modelo Lógico, Modelo Físico, Normalização, Validação, brModelo, MySQL Workbench, SQL

Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	vi
1. Definição do Sistema	1
1.1. Contextualização	1
1.2. Motivação e Objetivos do Trabalho	1
1.3. Análise da viabilidade do processo	2
1.4. Recursos e Equipa de Trabalho	2
1.5. Plano de Execução do Projeto	3
2. Levantamento e Análise de Requisitos	4
2.1. Método de levantamento e de análise de requisitos adotado	4
2.2. Organização dos requisitos levantado	4
2.2.1 Requisitos de descrição	4
2.2.2 Requisitos de exploração	4
2.2.3 Requisitos de controlo	5
2.3. Análise e validação geral dos requisitos	6
3. Modelação Conceptual	7
3.1. Apresentação da abordagem de modelação realizada	7
3.2. Identificação e caracterização das entidades	7
3.3. Identificação e caracterização dos relacionamentos	7
3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos.	9
3.5. Apresentação e explicação do diagrama ER produzido	10
4. Modelação Lógica	11
5. Implementação Física	17
5.1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	17
5.2. Tradução das interrogações do utilizador para SQL	23
5.3. Definição e caracterização das vistas de utilização em SQL	27

5.4. Cálculo do espaço da bases de dados (inicial e taxa de crescimento anual)	28
5.5. Indexação do Sistema de Dados	33
5.6. Procedimentos Implementados	33
5.7. Plano de segurança e recuperação de dados	35
6. Conclusões e Trabalho Futuro	37
Referências	38
Lista de Siglas e Acrónimos	39

Índice de Figuras

Figura 1 - Diagrama de GANTT	3
Figura 2 - Modelo Conceptual	10
Figura 3 - Modelo Lógico	12
Figura 4 - Clientes	12
Figura 5 - Funcionários	13
Figura 6 - Pedidos	13
Figura 7 - Armazéns	13
Figura 8 - Itens	13
Figura 9 - Pedidos/Funcionários	14
Figura 10 - Itens/Pedidos	14
Figura 11 - Armazéns/Itens	14
Figura 12 - Álgebra Relacional RE7	15
Figura 13 - Álgebra Relacional RE11	16
Figura 14 - Álgebra Relacional RE10	16
Figura 15 - Create Armazém	17
Figura 16 - Create Cliente	18
Figura 17 - Create Funcionário	18
Figura 18 - Create Item	19
Figura 19 - Create Pedido	20
Figura 20 - Create contem	21
Figura 21 - Create possui	22
Figura 22 - Create tem	22
Figura 23 - Query RE2	23
Figura 24 - Query RE3	23
Figura 25 - Query RE5	23
Figura 26 - Query RE7	23
Figura 27 - Query RE8	24
Figura 28 - Query RE11	24
Figura 29 - Query RE13	24

Figura 30 - Query RE15	24
Figura 31 - Query RE16	24
Figura 32 - Query RE14	25
Figura 33 - Query Trigger	26
Figura 34 - Query Função	26
Figura 35 - Query Transação	27
Figura 36 - Query View RE6	27
Figura 37 - Query View	28
Figura 38 - Procedimento RE10	33
Figura 39 - Procedimento RE12	34
Figura 40 - Procedimento RE17	34
Figura 41 - Procedimento RE4	35
Figura 42 - Permissões	36

Índice de Tabelas

Tabela 1 - Relações	8
Tabela 2 - Atributos das Entidades	9
Tabela 3 - Relacionamento	10
Tabela 4 - Tamanho Tipos	29
Tabela 5 - Tamanho Cliente	29
Tabela 6 - Tamanho Armazém	29
Tabela 7 - Tamanho Contem	30
Tabela 8 - Tamanho Funcionário	30
Tabela 9 - Tamanho Item	31
Tabela 10 - Tamanho Pedido	31
Tabela 11 - Tamanho Possui	32
Tabela 12 - Tamanho Tem	32
Tabela 13 - Tamanho Completo	32

1. Definição do Sistema

1.1. Contextualização

A AmazoniaTM é uma empresa com sede em Lisboa. Foi fundada em janeiro de 2018 pelo Sr. Jefferson Bazos percebendo a inexistência de uma Amazon em Portugal, viu uma grande oportunidade de negócio, de modo a proporcionar uma entrega satisfatória. Dois anos após ter idealizado o projeto e procura dos fundos, finalmente inaugurou a “AmazoniaTM”. Nos primeiros meses o negócio estava a ter sucesso, esta evolução chegou ao ponto em que o número de clientes era excessiva para as condições verificadas, visto que só existia armazéns na zona de Lisboa. Com o intuito de proporcionar tempos de entrega menores em outras zonas do país, tais como, Coimbra, Porto e Braga, decidiu expandir para essas localizações. Em cada um dos armazéns existem 21 funcionários, sendo eles 1 gestor, 10 distribuidores e 10 repositores. O crescimento originou vários problemas ao nível da gestão, uma vez que conciliar vários armazéns em diferentes localidades tornou-se impossível. Esta circunstância, tem conduzido, a uma perda de possíveis lucros, ou pelo menos é o que o seu gestor Joaquim afirma. Assim, para resolver esta situação, por sugestão do Joaquim, recorreu aos nossos serviços para a elaboração de um sistema de base de dados capaz de reverter a situação em que se encontra a “AmazoniaTM”.

1.2. Motivação e Objetivos do Trabalho

Com a sua grande vontade de maximizar os seus lucros, o Sr. Jefferson definiu um conjunto de objetivos que o irá ajudar com o desenvolvimento da base de dados, nomeadamente:

- Melhoria da estruturação do seu modelo de negócio, bem como melhorar a capacidade de gerir cada armazém e o registo da movimentação da mercadoria.
- Facilitação na obtenção de informação sobre os clientes, através da implementação de mecanismos de análise de vendas e de profiling de clientes;
- Aperfeiçoar a gestão de todos os produtos disponíveis;
- Organizar as finanças de forma a diminuir os gastos;
- Reduzir o tempo de entrega;
- Implementação de um sistema de recompensas para os funcionários exemplares.
- Consulta dos dados dos funcionários da empresa;

- Tratar grandes quantidades de dados; O desenvolvimento do sistema proposto constitui uma grande responsabilidade, pois qualquer falha pode comprometer o futuro da empresa visto que podem perder muito dinheiro.

1.3. Análise da viabilidade do processo

O Sr. Jefferson tem a certeza que se tiver uma base de dados mais eficiente, conseguirá:

- Aumentar o lucro por cerca de 10% logo no primeiro mês, isto suportará o custo do sistema;
- Otimizar as entregas com a melhor organização do stock e consequentemente efetuar mais envios por dia, aumentando a eficiência;
- Traçar um perfil do cliente de forma a fazer recomendações e eventualmente descontos.

1.4. Recursos e Equipa de Trabalho

Recursos:

- Humanos
 - Funcionários dos armazéns e da empresa de desenvolvimento, clientes e fornecedores.
- Materiais
 - Hardware (1 servidor, 5 tablets por armazém)
 - Software (SGBD e plataforma digital de compra e venda)

Equipa de trabalho:

- Pessoal Interno
 - Sr. Jefferson Bazos, Brenin Marrão, Tata Rovaris, Tigas Rodrigues
 - Fundador da empresa e gestores dos armazéns.
- Pessoal Externo
 - Engenheiros de Bases de Dados da empresa a contratar e Desenvolvedores da plataforma digital
 - Levantamento de requisitos, modelação do sistema, implementação do sistema.
 - Outros oClientes e fornecedores voluntários
 - Inquéritos de opinião e validação dos serviços.

1.5. Plano de Execução do Projeto

No presente relatório, foi realizada a divisão em quatro capítulos:

- Definição do Sistema
- Levantamento e Análise de Requisitos
- Implementação da Base de Dados Relacional
 - Modelação Conceptual
 - Modelação Lógica
 - Implementação Física
- Conclusão

Na primeira fase, existiu uma contextualização do problema, a apresentação dos objetivos e motivação para a realização do projeto e ainda a análise da viabilidade do mesmo.

No segundo capítulo, é realizada uma recolha de dados para saber quais as necessidades do Sr. Jefferson Bazos para a sua aplicação, resultando num levantamento de requisitos aos quais a base de dados implementada terá de responder e satisfazer.

No terceiro capítulo, é apresentado toda a metodologia, desde a modelação conceptual até à modelação lógica e física. Na modelação conceptual são identificados as entidades e os relacionamentos entre estas, bem como os atributos das entidades e dos relacionamentos.

Na segunda fase deste capítulo é ainda apresentado o modelo lógico bem como a validação do mesmo através da normalização e verificação das restrições de integridade. Por último, é descrito o modelo físico em concreto a resposta a queries com o sistema SQL.

Para terminar o nosso relatório, é apresentado uma conclusão, incluindo uma análise crítica do que foi realizado e uma avaliação do projeto.

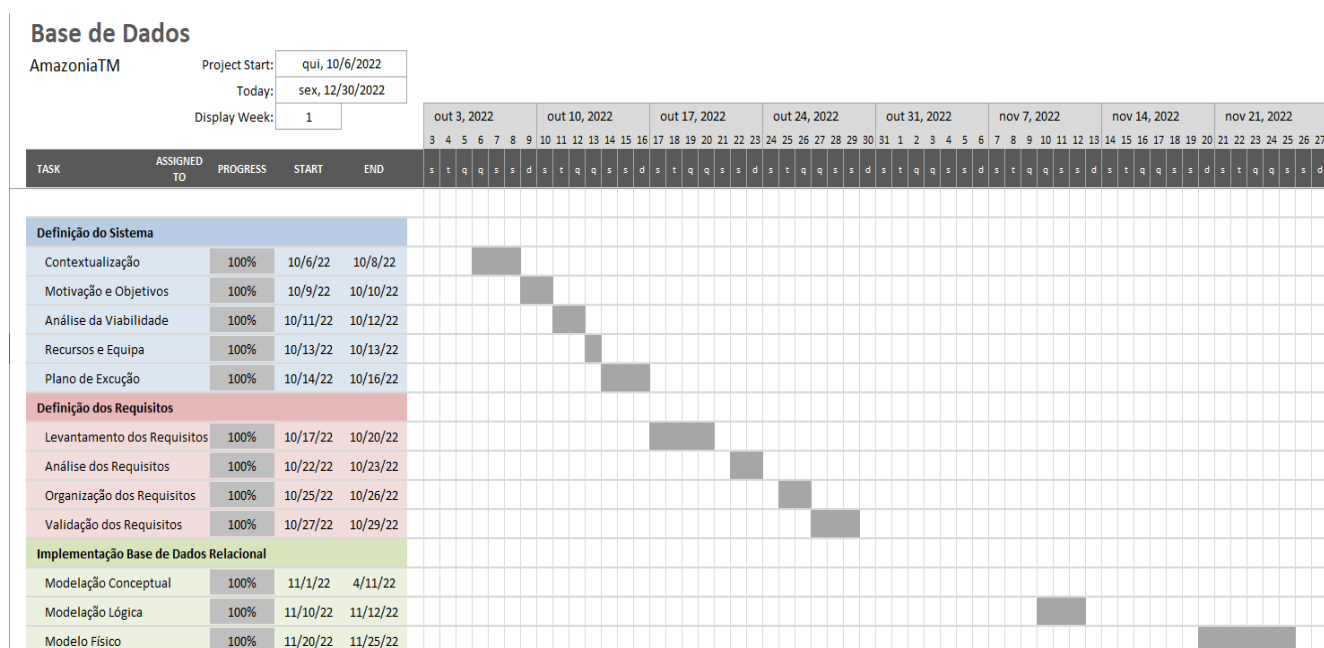


Figura 1 - Diagrama de GANTT

2. Levantamento e Análise de Requisitos

2.1. Método de levantamento e de análise de requisitos adotado

Foram feitos os seguintes métodos para obter o conhecimento necessário para o levantamento dos requisitos:

- Entrevistas/reuniões com Jefferson Bazos e com os gestores de cada armazém;
- A análise da documentação gerada por cada armazém;
- A observação dos processos em cada armazém;
- Questionários aos clientes e funcionários.

2.2. Organização dos requisitos levantado

2.2.1 Requisitos de descrição

1. Cada pedido deve conter o seu número de pedido, endereço de entrega, data do pedido/envio/entrega, cliente associado, o estado, possíveis cupões e por fim o total.
2. Cada item tem um id, preço, tipo, quantidade e descrição.
3. A relação entre pedido e item tem o seu valor, IVA e quantidade.
4. Cada cliente possui o seu número de identificação, nome, email, telemóvel, endereço de faturação.
5. Cada funcionário possui um id, nome, tipo, salário, endereço, armazém, data que começou a trabalhar na empresa, data de nascimento, género, armazém associado e desempenho.
6. O tipo de um funcionário pode ser, administrativo, distribuidor e repositor.
7. A cada pedido está associado um distribuidor e a cada distribuidor muitos pedidos.
8. Cada armazém tem um id, nome, localidade, conjunto de funcionários e itens.
9. A relação entre armazém e item tem o seu stock e posição.

2.2.2 Requisitos de exploração

1. Deve ser possível adicionar novos armazéns, itens, funcionários, clientes e pedidos.

2. Obter o número de clientes.
3. Verificar o número de funcionários.
4. Saber para cada pedido quais funcionários estão associados.
5. Conhecer o armazém em que o pedido está relacionado.
6. Identificar os clientes com maiores gastos.
7. Consultar os pedidos feitos por um cliente.
8. Consultar os funcionários com melhores desempenho.
9. Identificar os itens mais vendidos.
10. Saber para cada funcionário quais pedidos estão associados.
11. Obter quais funcionários trabalham em dado armazém.
12. Obter quais tipos de itens são mais populares.
13. Obter uma relação entre salário e desempenho de cada funcionário.
14. Obter os itens de um armazém.
15. Qual o valor total de vendas de cada armazém.
16. Qual o armazém com mais pedidos.
17. Quais os funcionários com mais pedidos.
18. Qual o item mais popular por número de pedidos.

2.2.3 Requisitos de controle

1. Somente o gestor do armazém pode alterar o valor do salário e desempenho de um funcionário.
2. Os dados dos clientes podem ser alterados/adicionados pelo mesmo.
3. Os clientes podem adicionar os itens que pretendem a um pedido.
4. As quantidades dos itens são controladas pelos repositores e administrativos.
5. Os administrativos podem atualizar o salário dos funcionários, alterar o estado dos pedidos e adicionar e alterar itens.
6. O funcionário pode alterar o estado do pedido.
7. Somente Jefferson Bazos tem controle sobre a informação completa da base de dados.
8. Somente Jefferson Bazos pode ter acesso a informação em relação aos gestores dos armazéns.
9. Somente o gestor do armazém e Jefferson Bazos pode acessar os dados dos armazéns (funcionários, itens, pedidos).
10. O funcionário relacionado a um pedido tem acesso a informação desse pedido.
11. O funcionário pode alterar os dados sobre o seu endereço.

2.3. Análise e validação geral dos requisitos

Após a elaboração final dos requisitos, reunimos novamente com o Sr. Jefferson Bazos para discutir os mesmos. O nosso cliente deu a aprovação pelo que prosseguimos para a elaboração do modelo conceptual a partir destes mesmos requisitos.

3. Modelação Conceptual

3.1. Apresentação da abordagem de modelação realizada

Para a construção da modelação conceptual primeiro foi preciso ser feito a análise dos requisitos e assim foram criadas as entidades Cliente, Pedido, Funcionário, Item e Armazém com as suas relações respectivas, além de seus atributos e características.

Por exemplo a Entidade Pedido tem os seus atributos definidos pelo requisito de descrição 1 e algumas de suas relações estão definidas nos requisitos 3 e 7.

3.2. Identificação e caracterização das entidades

Como foi abordado anteriormente as nossas entidades para esta base de dados são:

- Cliente, possui um Indentificador, o seu nome, gênero, Email, Rua, Código Postal e Telemóvel, além de poder alterar dados sobre a sua conta e realizar novos pedidos.
- Pedido, possui um Identificador, o Preço total, Datas de entrega, envio e pedido, o seu estado e o endereço de entrega, além de possuir os seus itens que será abordado no ponto seguinte.
- Funcionário, possui um Indentificador, seu salário, Gênero, Endereço, Nome, Tipo, Desempenho, sua data de nascimento e data de começo na empresa, e por último o armazém que trabalha.
- Item, possui um Indentificador, seu preço, Descrição e Tipo além de 2 relações com pedido e armazém.
- Armazém, possui um Identificador, Nome e Localização do mesmo.

3.3. Identificação e caracterização dos relacionamentos

Para completar o modelo conceptual foi preciso também definir as relações entre as entidades criadas sendo elas:

Relação Cliente Pedido:

Em que cada Pedido está relacionado somente 1 cliente e a cada cliente vários pedidos em uma relação 1:N. A partir desta relação podemos saber qual o cliente de cada Pedido e qual os pedidos de cada cliente.

Relação Pedido Item:

Como esta relação é de N para M existe uma tabela nova para esta relação onde esta possui a quantidade do item que foi feito naquele pedido o seu preço naquele instante e o IVA. A partir desta relação podemos saber quais os itens de cada Pedido e quais os pedidos de cada item.

Relação Item Armazém:

Esta relação também é N para M e seus atributos são a posição em dado armazém e o seu stock também naquele armazém. A partir desta relação podemos saber onde cada item pertence nos armazéns.

Relação Pedido Armazém:

É a relação de 1:N na qual um armazém está associado a um pedido e vários pedidos estão associados a um armazém. A partir desta relação podemos saber em qual Armazém foi feito um Pedido.

Relação Pedido Funcionário:

Esta é a última e mais simples relação N para M e para cada funcionário está relacionado vários pedidos e cada pedido está relacionado a vários funcionários. A partir desta relação podemos saber quais os funcionários de cada Pedido e quais os pedidos de cada funcionário.

Tabela 1 - Relações

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade
Cliente	1.1	feito	1.N	Pedido
Pedido	1.N	contem	1.N	Item
Armazem	1.N	possui	1.N	Item
Funcionario	1.N	tem	1.N	Pedido
Funcionario	1.N	trabalha	1.1	Armazem
Pedido	1.N	esta	1.1	Armazem

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

Tabela 2 - Atributos das Entidades

Entidade	Atributos	Descrição	Tipo do atributo	Tipo de dados
Cliente	ID_Cliente	Identificador do cliente	Chave primaria	INT
	Nome	Nome do Cliente	Simples	VARCHAR(45)
	Email	Email do Cliente	Simples	VARCHAR(45)
	Telemovel	Telemovel do Cliente	Simples	INT
	Rua	Endereço do Cliente	Simples	VARCHAR(45)
	Cod_Postal	Codigo postal da rua	Simples	INT
	Genero	Genero do Cliente	Simples	VARCHAR(1)
Pedido	ID_Pedido	Identificador do pedido	Chave Primaria	INT
	Endereco_entrega	Endereço de entrega do pedido	Simples	VARCHAR(45)
	Cupao	Cupao opcional	Simples	VARCHAR(45)
	Estado	Estado da entrega	Simples	VARCHAR(1)
	Total	Preço do pedido	Simples	FLOAT
	Data_Entrega	Data da entrega do pedido	Simples	DATE
	Data_Pedido	Data da criação do pedido	Simples	DATE
	Data_Envio	Data do envio do Pedido	Simples	DATE
	fk_Cliente_ID_Cliente	Cliente associado ao pedido	Foreign Key	INT
	fk_Armazem_Armazem_ID	Armazem associado ao Pedido	Foreign Key	INT
Item	Item_ID	Identificador do Item	Chave Primaria	INT
	Preco	Custo do Item	Simples	FLOAT
	Descrição	Descrição do produto	Simples	VARCHAR(45)
	Tipo	Tipo do Produto	Simples	VARCHAR(45)
Funcionario	Funcionario_ID	Identificador do Funcionario	Chave Primaria	INT
	Nome	Nome do Funcionario	Simples	VARCHAR(45)
	Tipo	Tipo do Funcionario	Simples	VARCHAR(45)
	Salario	Salario do Funcionario	Simples	FLOAT
	Endereco	Morada do Funcionario	Simples	VARCHAR(45)
	Desempenho	Classificação do desempenho	Simples	INT
	Genero	Genero do Funcionario	Simples	VARCHAR(1)
	data_comeco	Dia da Contratação	Simples	DATE
	data_nascimento	Dia do nascimento do Funcionario	Simples	DATE
	fk_Armazem_Armazem_ID	Qual Armazem o funcionario trabalha	Foreign Key	INT
Armazem	Armazem_ID	Identificador do Armazem	Chave Primaria	INT
	Nome	Nome do Armazem	Simples	VARCHAR(45)
	Localidade	Em que parte do pais se encontra	Simples	VARCHAR(45)

Tabela 3 - Relacionamento

Relacionamento	Atributos	Descrição	Tipo do Atributo	Tipo de Dados
contem	Quantidade	Numero de itens comprados	Simple	INT
	Valor	Soma dos preços dos itens naquele instante	Simple	FLOAT
	IVA	Iva do Pedido	Simple	INT

3.5. Apresentação e explicação do diagrama ER produzido

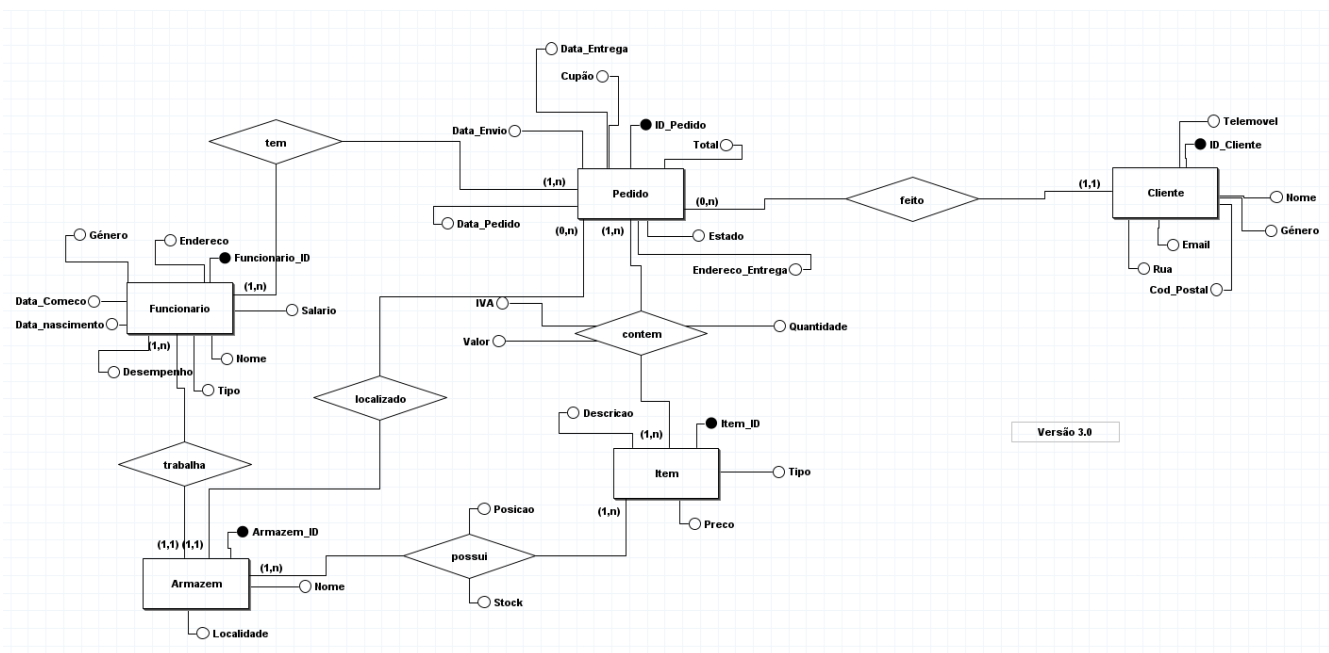


Figura 2 - Modelo Conceptual

4. Modelação Lógica

Depois de terminarmos a modelação conceptual passamos para a criação do modelo lógico através do modelo conceptual que criamos. A partir da validação deste esquema lógico podemos passar para a modelação física.

4.1 - Construção e validação do modelo de dados lógico

Para a construção do modelo lógico importamos para o 'MySQL Workbench' o modelo conceptual que criamos pelo brModelo e criamos o modelo lógico.

4.2 - Normalização dos dados

4.2.1 - Primeira Forma Normal (1FN)

A primeira forma normal diz para identificar cada conjunto de dados com uma chave primaria o que é feito , também diz para não usar vários campos de uma tabela para guardar dados similares, um caso não normalizado deste tipo que poderia ocorrer no nosso modelo lógico seria se nós tivéssemos na tabela pedido 3 chaves estrangeiras para os 3 funcionários relacionados a um pedido, porém nós criamos uma tabela 'tem' para representar os funcionários relacionados a um pedido, concluímos que o nosso modelo está de acordo com a primeira forma normal.

4.2.2 – Segunda Forma Normal (2FN)

A segunda forma normal diz que o que está dentro de uma tabela não deve depender em nada que não seja a chave primária ou seja tudo que está dentro de uma tabela não teria como estar em outra tabela, consideramos que a segunda forma normal é respeitada neste modelo lógico.

4.2.2 - Terceira Forma Normal (3FN)

Como a nossa modelação lógica não contém dependências transitivas, ou seja, se todos os atributos que não são a chave primária dependerem somente da chave primária e não dependem de outro atributo que por sua vez dependem da chave primária. Também como todos os atributos não chave dependem exclusivamente da chave primária concluímos que esta forma normal é respeitada neste modelo lógico.

4.3 - Apresentação e explicação do modelo lógico produzido

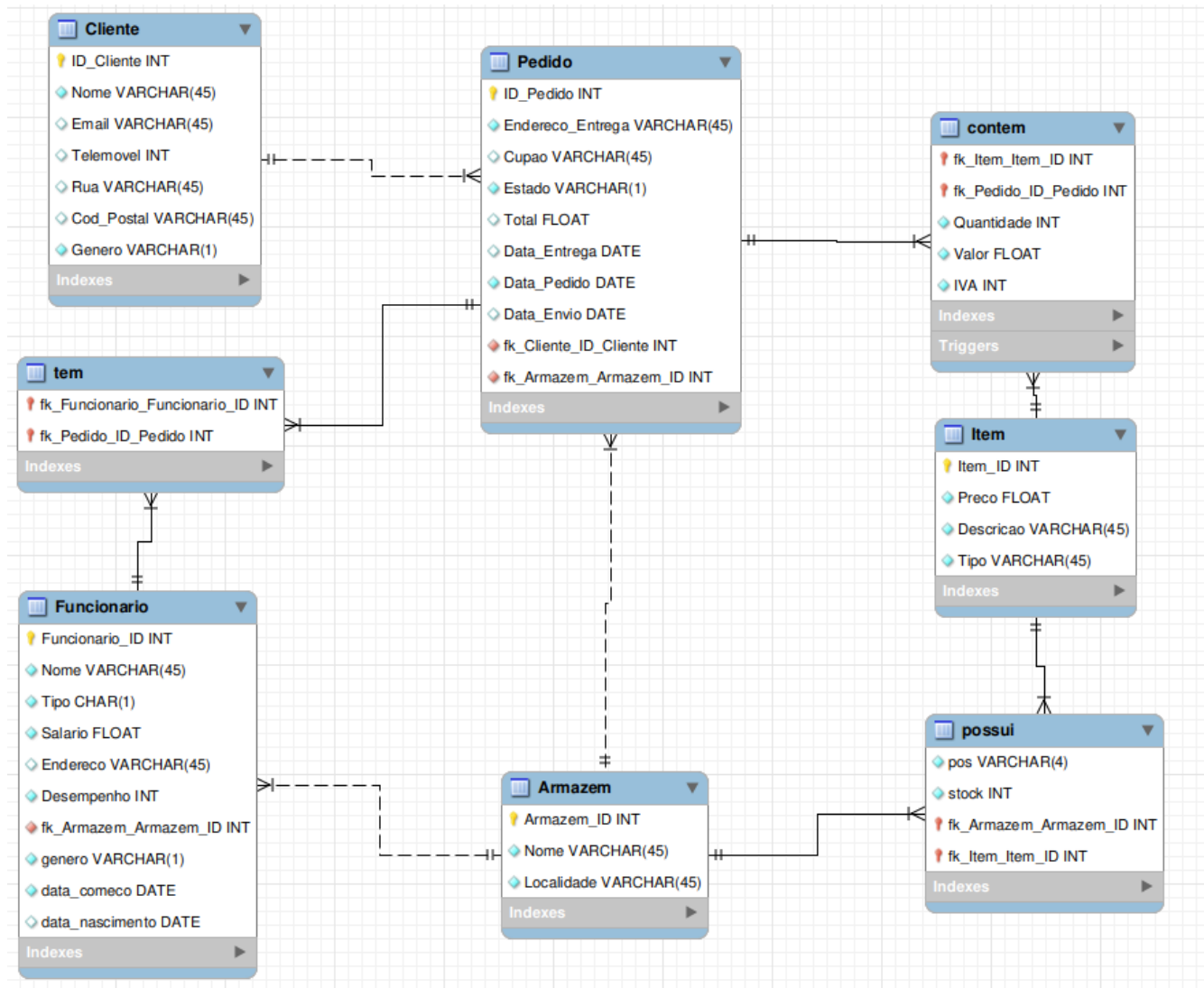


Figura 3 - Modelo Lógico

4.3.1 Entidades fortes

Entidades que possuem uma chave primária, que a identifica e não depende de outras entidades.

Cliente(ID_cliente, Nome, Email, Telemovel, Cod_Postal, Rua)

Chave primária : ID_cliente

#	ID_Cliente	Nome	Email	Telemovel	Rua	Cod_Posta	Genero
1	1	Ana Maria	Cliente1@gmail.com	911111110	Rua da Capela	4700-800	F
2	2	Zeca Afonso	Cliente2@gmail.com	911111119	Rua da Boavista	4701-800	M

Figura 4 - Clientes

Funcionario(Funcionario_ID, Nome, Tipo, Salario, Endereco, Desempenho, fk_Armazem_Armazem_ID, genero, data_comeco, data_nascimento)

Chave primária : Funcionario_ID

#	Funcionario_ID	Nome	Tipo	Salario	Endereco	Desempenho	fk_Armazem_Armazem_ID	data_nascimento	data_comeco	genero
	1	Tata	G	1000	Rua da mae nova	10	1	1900-02-02	2019-01-01	F
	2	Andre Costa	D	499	Rua do pai novo	7	1	1900-03-02	2019-01-02	M

Figura 5 - Funcionários

Pedido(ID_pedido,Endereco_Entrega,cupao,Estado>Total,Data_Entrega,Data_Pedido,Data_Envio,Desempenho, fk_Cliente_ID_Cliente, fk_Armazem_Armazem_ID)

Chave primária : Id_pedido

#	ID_Pedido	Endereco_Entrega	Cupao	Estado	Total	Data_Entrega	Data_Pedido	Data_Envio	fk_Cliente_ID_Cliente	fk_Armazem_Armazem_ID
	1	Rua Doutor Damasceno Ferreiro	NATAL2023	P	144	NULL	2023-01-01	NULL	1	1
	2	Rua da ponte	NULL	E	15	2023-01-03	2023-01-02	2023-01-03	2	1

Figura 6 - Pedidos

Armazem(Armazem_ID,Nome, Localidade)

Chave primária : Armazem_ID

#	Armazem_ID	Nome	Localidade
	1	Armazem de Braga	Norte
	2	Armazem de Lisboa	Sul

Figura 7 - Armazéns

Item(Item_ID,Preco,Descricao,Tipo)

Chave primária : Item_ID

#	Item_ID	Preco	Tipo	Descricao
	1	20	Informatica	Earbuds Salozne 3.12
	2	100	EletoDomesticos	Frigorifico Apar certificado C

Figura 8 - Itens

4.3.2 Relacionamentos (1:N)

No nosso modelo lógico encontram-se os seguintes relacionamentos (1:N) :

- 1 - Cada pedido tem somente um armazém enquanto cada armazém pode ter vários pedidos, por isto existe na tabela “Pedidos” um atributo fk_Armazem_Armazem_ID.
- 2 – Cada pedido tem somente um cliente enquanto cada cliente pode ter vários pedidos, por isto existe na tabela “Pedidos” um atributo fk_Cliente_ID_Cliente.

4.3.3 Relacionamentos (N:M)

Estão presente no nosso modelo lógico os seguintes relacionamentos (N:M) :

1. **tem**(fk_Funcionario_Funcionario_ID,fk_Pedido_ID_Pedido): Relação entre funcionários e pedidos , essa tabela tem contém os seguintes atributos : a chave primária de um pedido e a chave primaria de cada funcionário, cada pedido precisa ter pelo menos 3 funcionários , um distribuidor , um gestor e repositor.

#	fk_Funcionario_Funcionario	fk_Pedido_ID_Pedid
1		1
2		1

Figura 9 - Pedidos/Funcionários

2. **contem**(fk_Pedido_ID_Pedido, fk_Item_Item_ID, Valor, IVA, Quantidade): relação entre pedido e item, além das chaves primárias fk_Pedido_ID_Pedido, fk_Item_Item_ID, Valor, o contem também possui a quantidade do item no pedido e também o seu valor e iva no momento, pois assim podemos saber qual era o preço do item no momento do pedido.

#	fk_Item_Item_ID	fk_Pedido_ID_Pedid	Quantidade	Valor	IVA
1	1	2	22	10	
2	16	1	25	13	

Figura 10 - Itens/Pedidos

3. **possui**(pos, stock, Armazem_Armazem_ID, Item_Item_ID): relação entre armazém e item além das chaves primárias Armazem_Armazem_ID, Item_Item_ID, possui o atributo pos que serve para sabermos a localização do produto dentro do armazém e assim encontrar mais fácil os itens e evitar perda de produto, e o stock do item no armazém

#	pos	stock	fk_Armazem_Armazem_ID	fk_Item_Item_ID
2A	100	1		1
2B	1000	1		2

Figura 11 - Armazéns/Itens

4.4 Validação do modelo com interrogações do utilizador

Fizemos a resolução de algumas interrogações que correspondem a alguns requisitos de exploração para podermos validar o modelo.

Requisitos de exploração 7:

- Consultar os pedidos feitos por um cliente.



Π c.Nome, p.ID_Pedido, p.Endereco_Entrega, p.Cupao, p.Estado, p.Total, p.Data_Entrega, p.Data_Pedido, p.Data_Envio, p.fk_Cliente_ID_Cliente, p.fk_Armazem_Armazem_ID σ c.Nome = 'Ana Maria' (ρ_p Pedido \bowtie ρ_c Cliente)

Figura 12 - Álgebra Relacional RE7

Requisitos de exploração 11:

- Obter quais funcionários trabalham em dado armazém.

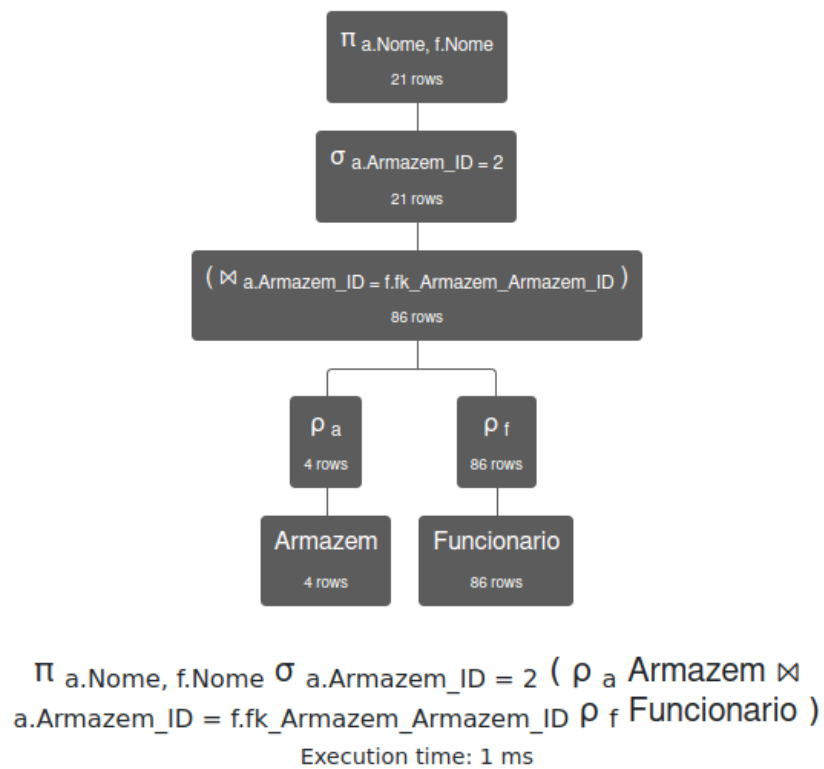
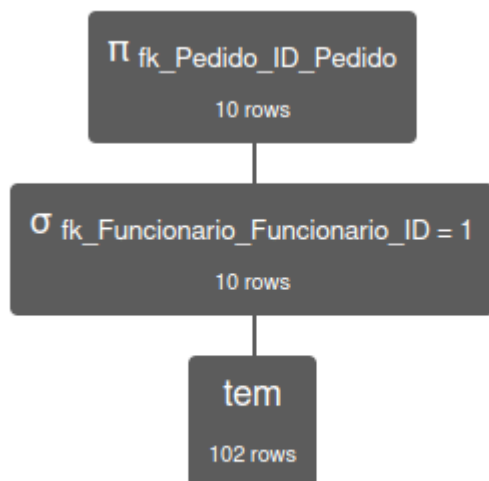


Figura 13 - Álgebra Relacional RE11

Requisitos de exploração 10:

- Saber para cada funcionário quais pedidos estão associados.



$\pi_{fk_Pedido_ID_Pedido} \sigma_{fk_Funcionario_Funcionario_ID = 1} \pi_{tem}$

Figura 14 - Álgebra Relacional RE10

5. Implementação Física

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

Finalizado o Modelo Lógico passamos à elaboração do Modelo Físico, para que as relações e as suas restrições sejam suportadas pelo sistema de gestão de bases de dados. Para isso é necessário realizar o desenho das relações e das suas restrições. Para a implementação física, optamos por utilizar o MySQL por se tratar de um Sistema de Gestão de fácil compreensão e também por ser o que utilizamos nas aulas da Unidade Curricular. Para gerar o Modelo Físico através do Modelo Lógico, recorremos à ferramenta do MySQL Workbench, Forward Engineer e obtivemos as seguintes tabelas :

```
-- -----  
-- Table `Amazonia`.`Armazem`  
-- -----  
● CREATE TABLE IF NOT EXISTS `Amazonia`.`Armazem` (  
  `Armazem_ID` INT NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `Localidade` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Armazem_ID`),  
  UNIQUE INDEX `Armazem_ID` (`Armazem_ID` ASC) VISIBLE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_0900_ai_ci;
```

Figura 15 - Create Armazém

```

-----
-- Table `Amazonia`.`Cliente`
-----

CREATE TABLE IF NOT EXISTS `Amazonia`.`Cliente` (
  `ID_Cliente` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Email` VARCHAR(45) NULL DEFAULT NULL,
  `Telemovel` INT NULL DEFAULT NULL,
  `Rua` VARCHAR(45) NULL DEFAULT NULL,
  `Cod_Postal` VARCHAR(45) NULL DEFAULT NULL,
  `Genero` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID_Cliente`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

Figura 16 - Create Cliente

```

-----
-- Table `Amazonia`.`Funcionario`
-----

CREATE TABLE IF NOT EXISTS `Amazonia`.`Funcionario` (
  `Funcionario_ID` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Tipo` CHAR(1) NOT NULL,
  `Salario` FLOAT NOT NULL,
  `Endereco` VARCHAR(45) NULL DEFAULT NULL,
  `Desempenho` INT NOT NULL,
  `fk_Armazem_Armazem_ID` INT NOT NULL,
  `genero` VARCHAR(1) NOT NULL,
  `data_comeco` DATE NOT NULL,
  `data_nascimento` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`Funcionario_ID`),
  UNIQUE INDEX `Funcionario_ID` (`Funcionario_ID` ASC, `fk_Armazem_Armazem_ID` ASC) VISIBLE,
  INDEX `FK_Funcionario_2` (`fk_Armazem_Armazem_ID` ASC) VISIBLE,
  CONSTRAINT `FK_Funcionario_2`
    FOREIGN KEY (`fk_Armazem_Armazem_ID`)
      REFERENCES `Amazonia`.`Armazem` (`Armazem_ID`)
    ON DELETE RESTRICT)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

Figura 17 - Create Funcionário

```

-----
-- Table `Amazonia`.`Item`
-----
) CREATE TABLE IF NOT EXISTS `Amazonia`.`Item` (
    `Item_ID` INT NOT NULL,
    `Preco` FLOAT NOT NULL,
    `Descricao` VARCHAR(45) NOT NULL,
    `Tipo` VARCHAR(45) NOT NULL,
    PRIMARY KEY (`Item_ID`),
    UNIQUE INDEX `Item_ID` (`Item_ID` ASC) VISIBLE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

Figura 18 - Create Item

```
-- Table `Amazonia`.`Pedido`
```

```
CREATE TABLE IF NOT EXISTS `Amazonia`.`Pedido` (  
  `ID_Pedido` INT NOT NULL,  
  `Endereco_Entrega` VARCHAR(45) NOT NULL,  
  `Cupao` VARCHAR(45) NULL DEFAULT NULL,  
  `Estado` VARCHAR(1) NOT NULL,  
  `Total` FLOAT ,  
  `Data_Entrega` DATE NULL,  
  `Data_Pedido` DATE NOT NULL,  
  `Data_Envio` DATE NULL,  
  `fk_Cliente_ID_Cliente` INT NOT NULL,  
  `fk_Armazem_Armazem_ID` INT NOT NULL,  
  PRIMARY KEY (`ID_Pedido`),  
  UNIQUE INDEX `ID_Pedido` (`ID_Pedido` ASC, `fk_Cliente_ID_Cliente` ASC) VISIBLE,  
  INDEX `FK_Pedido_2` (`fk_Cliente_ID_Cliente` ASC) VISIBLE,  
  CONSTRAINT `FK_Pedido_1`  
    FOREIGN KEY (`fk_Armazem_Armazem_ID`)  
    REFERENCES `Amazonia`.`Armazem` (`Armazem_ID`)  
    ON DELETE CASCADE,  
  CONSTRAINT `FK_Pedido_2`  
    FOREIGN KEY (`fk_Cliente_ID_Cliente`)  
    REFERENCES `Amazonia`.`Cliente` (`ID_Cliente`)  
    ON DELETE CASCADE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_0900_ai_ci;
```

Figura 19 - Create Pedido

```

-----
-- Table `Amazonia`.`contem`
-----

CREATE TABLE IF NOT EXISTS `Amazonia`.`contem` (
  `fk_Item_Item_ID` INT NOT NULL,
  `fk_Pedido_ID_Pedido` INT NOT NULL,
  `Quantidade` INT NOT NULL,
  `Valor` FLOAT NOT NULL,
  `IVA` INT NOT NULL,
  PRIMARY KEY (`fk_Item_Item_ID`, `fk_Pedido_ID_Pedido`),
  UNIQUE INDEX `fk_Item_Item_ID` (`fk_Item_Item_ID` ASC, `fk_Pedido_ID_Pedido` ASC) VISIBLE,
  INDEX `FK_contem_2` (`fk_Pedido_ID_Pedido` ASC) VISIBLE,
  CONSTRAINT `FK_contem_1`
    FOREIGN KEY (`fk_Item_Item_ID`)
      REFERENCES `Amazonia`.`Item` (`Item_ID`)
      ON DELETE RESTRICT,
  CONSTRAINT `FK_contem_2`
    FOREIGN KEY (`fk_Pedido_ID_Pedido`)
      REFERENCES `Amazonia`.`Pedido` (`ID_Pedido`)
      ON DELETE RESTRICT)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

Figura 20 - Create contem

```

-----
-- Table `Amazonia`.`possui`
-----
) CREATE TABLE IF NOT EXISTS `Amazonia`.`possui` (
  `pos` VARCHAR(4) NOT NULL,
  `stock` INT NOT NULL,
  `fk_Armazem_Armazem_ID` INT NOT NULL,
  `fk_Item_Item_ID` INT NOT NULL,
  `Armazem_Armazem_ID` INT NOT NULL,
  `Item_Item_ID` INT NOT NULL,
  PRIMARY KEY (`fk_Armazem_Armazem_ID`, `fk_Item_Item_ID`),
  UNIQUE INDEX `fk_Armazem_Armazem_ID` (`fk_Armazem_Armazem_ID` ASC, `fk_Item_Item_ID` ASC) VISIBLE,
  INDEX `fk_possui_Armazem1_idx` (`Armazem_Armazem_ID` ASC) VISIBLE,
  INDEX `fk_possui_Item1_idx` (`Item_Item_ID` ASC) VISIBLE,
  CONSTRAINT `fk_possui_Armazem1`
    FOREIGN KEY (`Armazem_Armazem_ID`)
    REFERENCES `Amazonia`.`Armazem` (`Armazem_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_possui_Item1`
    FOREIGN KEY (`Item_Item_ID`)
    REFERENCES `Amazonia`.`Item` (`Item_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

Figura 21 - Create possui

```

-----
-- Table `Amazonia`.`tem`
-----
CREATE TABLE IF NOT EXISTS `Amazonia`.`tem` (
  `fk_Funcionario_Funcionario_ID` INT NOT NULL,
  `fk_Pedido_ID_Pedido` INT NOT NULL,
  PRIMARY KEY (`fk_Funcionario_Funcionario_ID`, `fk_Pedido_ID_Pedido`),
  UNIQUE INDEX `fk_Funcionario_Funcionario_ID` (`fk_Funcionario_Funcionario_ID` ASC, `fk_Pedido_ID_Pedido` ASC) VISIBLE,
  INDEX `FK_tem_2` (`fk_Pedido_ID_Pedido` ASC) VISIBLE,
  CONSTRAINT `FK_tem_1`
    FOREIGN KEY (`fk_Funcionario_Funcionario_ID`)
    REFERENCES `Amazonia`.`Funcionario` (`Funcionario_ID`)
    ON DELETE RESTRICT,
  CONSTRAINT `FK_tem_2`
    FOREIGN KEY (`fk_Pedido_ID_Pedido`)
    REFERENCES `Amazonia`.`Pedido` (`ID_Pedido`)
    ON DELETE RESTRICT)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

Figura 22 - Create tem

5.2. Tradução das interrogações do utilizador para SQL

- 1- Quantos Clientes possui a AmazoniaTM?(RE 2)

```
#Numero de Clientes da AmazoniaTM
SELECT COUNT(*)
FROM Cliente;
```

Figura 23 - Query RE2

- 2- Quantos Funcionários tem a AmazoniaTM?(RE 3)

```
#Conta quantos funcionarios Trabalham na AmazoniaTM
Select Count(Funcionario_ID)
From Funcionario;
```

Figura 24 - Query RE3

- 3- Em que armazém está a ser tratado um pedido? (RE 5)

```
# Conhecer o armazém em que o pedido está relacionado
DELIMITER &&
CREATE FUNCTION armazen_do_pedido ( id_ped INT )
RETURNS INT
DETERMINISTIC
) BEGIN
    return (select fk_Armazem_Armazem_ID from Pedido where ID_Pedido = id_ped);
- END; &&
delimiter ;
```

Figura 25 - Query RE5

- 4- Que pedidos fez um determinado cliente? (RE 7)

```
#Seleciona os Pedidos de um Cliente especifico
SELECT c.nome , p.*
) FROM ( Pedido AS p
- INNER JOIN Cliente AS c ON p.fk_Cliente_ID_Cliente = c.ID_Cliente)
WHERE c.nome= 'Ana Maria';
```

Figura 26 - Query RE7

- 5- Quais os funcionários com melhor desempenho? (RE 8)

```
#ordenação dos funcionarios por desempenho
select Nome
from Funcionario
order by desempenho desc;
```

Figura 27 - Query RE8

- 6- Que funcionários trabalham num determinado armazém? (RE 11)

```
#Seleciona os funcionarios de um determinado Armazem
SELECT a.Nome, f.Nome
FROM ( Armazem AS a
INNER JOIN Funcionario AS f ON a.Armazem_ID = f.fk_Armazem_Armazem_ID )
WHERE a.Armazem_ID = '2';
```

Figura 28 - Query RE11

- 7- Qual a relação de funcionário/desempenho? (RE 13)

```
# Obter uma relação entre salário e desempenho de cada funcionario
select Funcionario_ID, Salario/Desempenho as Quota
from Funcionario;
```

Figura 29 - Query RE13

- 8- Qual o valor total de vendas de cada armazém? (RE 15)

```
#Total ganho por cada Armazem
select sum(total), Armazem_ID
from Pedido
inner join Armazem on Pedido.fk_Armazem_Armazem_ID = Armazem.Armazem_ID
group by Armazem.Armazem_ID
order by sum(total) desc;
```

Figura 30 - Query RE15

- 9- Qual o armazém com mais pedidos? (RE 16)

```
#Numero de Pedidos por Armazem
select sum(total), Armazem_ID
from Pedido
inner join Armazem on Pedido.fk_Armazem_Armazem_ID = Armazem.Armazem_ID
group by Armazem.Armazem_ID
order by sum(total) desc;
```

Figura 31 - Query RE16

10- Que itens possui um armazém ? (RE 14)

```
# Obter itens de um armazem
SELECT i.Item_ID,i.preco,i.descricao,i.tipo,p.pos,p.stock
FROM( Item as i
INNER JOIN possui as p on p.fk_Item_Item_ID = i.Item_ID)
WHERE p.fk_Armazem_Armazem_ID = 2;
```

Figura 32 - Query RE14

11- Qual o item mais popular por número de pedidos? (RE 18)

```
#Itens em maior numero de pedidos
SELECT Descricao, count(Item_ID)
FROM Item
INNER JOIN contem ON contem.fk_Item_Item_ID = Item.Item_ID
group by Item_ID
order by count(Item_ID) desc;
```

Figura 33 - Query RE18

- Trigger para atualizar os valores de cada pedido

```

delimiter &&
CREATE TRIGGER Amazonia.total
AFTER
INSERT ON contem for each row
begin
declare total int;
declare cupao1 varchar(9);
select Total into total
from Pedido
where NEW.fk_Pedido_ID_Pedido = id_Pedido;
select Cupao into cupao1
from Pedido
where new.fk_Pedido_ID_Pedido = id_Pedido;
if cupao1 = null then
update Pedido set Total = (select sum(valor * quantidade)
from contem where fk_Pedido_ID_Pedido = NEW.fk_Pedido_ID_Pedido)
where ID_Pedido = NEW.fk_Pedido_ID_Pedido;
else
update Pedido set Total = (select sum(valor * quantidade)*0.8
from contem where fk_Pedido_ID_Pedido = NEW.fk_Pedido_ID_Pedido)
where ID_Pedido = NEW.fk_Pedido_ID_Pedido;
end if;
END &&
delimiter ;

```

Figura 34 - Query Trigger

- Função para identificar em que armazém está a ser tratado um pedido

```

# Conhecer o armazém em que o pedido está relacionado
DELIMITER &&
CREATE FUNCTION armazem_do_pedido ( id_ped INT )
RETURNS INT
DETERMINISTIC
BEGIN
    return (select fk_Armazem_Armazem_ID from Pedido where ID_Pedido = id_ped);
END; &&
delimiter ;

```

Figura 35 - Query Função

- Transação para inserir na tabela contem que atualiza o stock conforme o número de itens vendidos

```
DELIMITER $$
CREATE PROCEDURE InsereContem(IN item INT, IN pedido INT, IN Quantidade INT, IN IVA INT, IN Valor FLOAT)

) BEGIN
declare armazem int;
declare vstock int;
START TRANSACTION;

set armazem = (select fk_Armazem_Armazem_ID from pedido where ID_Pedido = pedido);
set vstock = (select stock from possui where fk_Item_Item_ID = pedido and fk_Armazem_Armazem_ID = armazem);
if vstock - Quantidade <= 0 then
    rollback;
end if;
update possui
SET stock = vstock - Quantidade
WHERE fk_Item_Item_ID = item;
INSERT INTO contem
(fk_Item_Item_ID, fk_Pedido_ID_Pedido, Quantidade, Valor, IVA)
VALUES(item,pedido,Quantidade,IVA,Valor);
END $$
DELIMITER ;
```

Figura 36 - Query Transação

5.3. Definição e caracterização das vistas de utilização em SQL

- 1- Quais os clientes que mais gastam? (RE 6)

```
#View que seleciona os clientes que mais gastaram
drop view if exists top_gastos;
create view top_gastos as
select fk_Cliente_ID_Cliente, sum(total) as gasto
from Pedido
group by fk_Cliente_ID_Cliente
Limit 10;
```

Figura 37 - Query View RE6

- 2- View dos piores funcionários

```

#view dos piores funcionarios
Drop View if exists view_pioresfunc;
CREATE VIEW view_pioresfunc
AS
SELECT Nome,Desempenho,data_comeco,fK_Armazem_Armazem_ID
From Funcionario
where Desempenho=0 or Desempenho=1
Order by Desempenho;

```

Figura 38 - Query View piores funcionários

- 3- View de quais os itens mais valiosos.

```

#view dos itens mais caros
Drop View if exists view_itensmaiscaros;
CREATE VIEW view_itensmaiscaros
AS
Select preco,descricao
From Item
Order by preco Desc
Limit 5;

```

Figura 39 - Query View itens mais valiosos

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

Na criação da base de dados tornou-se crucial obter o espaço que ocupará em disco. De seguida, vamos apresentar uma tabela com os tipos de dados e o tamanho que ocupam e depois vamos a cada entidade calcular o valor ocupado em disco.

Tabela 4 - Tamanho Tipos

Tipos de Dados	Tamanho(bytes)
INT	4
VARCHAR(N)	N+1
DATE	3
FLOAT	4
DOUBLE	8

Tabela 5 - Tamanho Cliente

Cliente		
Atributos	Tipos de Dados	Espaço no Disco
ID_Cliente	INT	4
Nome	VARCHAR(45)	46
Email	VARCHAR(45)	46
Telemovel	INT	4
Rua	VARCHAR(45)	46
Cod_Postal	INT	4
Genero	VARCHAR(1)	2
Total		152

Tabela 6 - Tamanho Armazém

Armazem		
Atributos	Tipos de Dados	Espaço no Disco
Armazem_ID	INT	4
Nome	VARCHAR(45)	46
Localidade	VARCHAR(45)	46
Total		96

Tabela 7 - Tamanho Contem

Contem		
Atributos	Tipos de Dados	Espaço no Disco
Quantidade	INT	4
Valor	FLOAT	4
IVA	INT	4
fk_Item_Item_ID	INT	4
fk_Pedido_ID_Pedido	INT	4
Total		20

Tabela 8 - Tamanho Funcionário

Funcionario		
Atributos	Tipos de Dados	Espaço no Disco
Funcionario_ID	INT	4
Nome	VARCHAR(45)	46
Tipo	VARCHAR(45)	46
Salario	FLOAT	4
Endereco	VARCHAR(45)	46
Desempenho	INT	4
Genero	VARCHAR(1)	2
Data_Comeco	DATE	3
Data_nascimento	DATE	3
fk_Armazem_Armazem_ID	4	4
Total		162

Tabela 9 - Tamanho Item

Item		
Atributos	Tipos de Dados	Espaço no Disco
Item_ID	INT	4
Preco	FLOAT	4
Descricao	VARCHAR(45)	46
Tipo	VARCHAR(45)	46
Total		100

Tabela 10 - Tamanho Pedido

Pedido		
Atributos	Tipos de Dados	Espaço no Disco
ID_Pedido	INT	4
Endereco_Entrega	VARCHAR(45)	46
Cupao	VARCHAR(45)	46
Estado	VARCHAR(1)	2
Total	FLOAT	4
Data_Entrega	DATE	3
Data_Pedido	DATE	3
Data_Envio	DATE	3
fk_Cliente_ID_Cliente	4	4
fk_Armazem_Armazem_ID	4	4
Total		119

Tabela 11 - Tamanho Possui

Possui		
Atributos	Tipos de Dados	Espaço no Disco
Pos	VARCHAR(4)	5
stock	INT	4
fk_Item_Item_ID	INT	4
fk_Armazem_Armazem_ID	INT	4
Total		17

Tabela 12 - Tamanho Tem

Tem		
Atributos	Tipos de Dados	Espaço no Disco
fk_Pedido_ID_Pedido	INT	4
fk_Funcionario_Funcionario_ID	INT	4
Total		8

Tabela 13 - Tamanho Completo

Tabela	Espaço no Disco
Cliente	$152 * 40 = 6080$
Pedido	$119 * 37 = 4403$
Funcionario	$162 * 86 = 13932$
Item	$100 * 20 = 2000$
Armazem	$96 * 4 = 384$
Contem	$20 * 51 = 1020$
Possui	$17 * 80 = 1360$
Tem	$8 * 102 = 816$
Total	29995

Como observamos nesta última tabela, a estimativa para o tamanho da nossa base de dados com o povoamento final é de 29995 bytes, mas é essencial verificar como se comportará com uma quantidade

de dados significante! Com o crescimento da AmazoniaTM , é bastante provável que o número de clientes aumente exponencialmente , por consequência também teríamos de aumentar o número de funcionários e possuir mais armazéns o que faria com que o número de bytes subisse consideravelmente.

5.5. Indexação do Sistema de Dados

Decidimos criar 3 index diferentes, pois percebemos ao desenvolver as queries , procedures, transações e triggers que termos a indexação desses atributos ajudaria a nossa base de dados.

```
CREATE INDEX fk_funcionario
ON tem (fk_Funcionario_Funcionario_ID);
CREATE INDEX fk_pedido
ON tem (fk_Pedido_ID_Pedido);
CREATE INDEX fk_item_pedido
ON contem (fk_Item_Item_ID);
```

Figura 40 - Indexação

5.6. Procedimentos Implementados

- 1- Que pedidos tem associado cada funcionário? (RE 10)

```
# este procedure diz a quais pedidos um funcionario esta relacionado
delimiter &&
CREATE PROCEDURE `PedFunc` (IN id_func INT)
BEGIN
SELECT fk_Pedido_ID_Pedido FROM tem where fk_Funcionario_Funcionario_ID = id_func;
END &&
delimiter ;
```

Figura 41 - Procedimento RE10

- 2- Quais foram os itens mais vendidos por quantidade? (RE 12)

```

# Identificar os itens mais vendidos por quantidade
DELIMITER &&
CREATE PROCEDURE items_mais_vendidos ()
# τ TotalItems desc π I.Item_ID, TotalPedidos γ I.Item_ID; SUM(C.Quantidade)→TotalItems
# ( ρ I item left outer join C.fk_Item_Item_ID = I.Item_ID ρ C contem )
BEGIN
select
    I.Item_ID,
    SUM(C.Quantidade) AS TotalItems
from
    Item I
left join
    contem C
on
    C.fk_Item_Item_ID = I.Item_ID
group by
    I.Item_ID
order by
    TotalItems DESC;
END &&
delimiter ;

```

Figura 42 - Procedimento RE12

3- Quais os funcionários com mais pedidos? (RE 17)

```

# Funcionarios com mais pedidos associados
DELIMITER &&
CREATE PROCEDURE funcionarios_com_mais_pedidos ()
BEGIN
# τ TotalPedidos desc π F.Funcionario_ID, TotalPedidos γ F.Funcionario_ID; COUNT(T.fk_Pedido_ID_Pedido)
# ( ρ F funcionario left outer join T.fk_Funcionario_Funcionario_ID = F.Funcionario_ID ρ T tem )
select
    F.Funcionario_ID,
    COUNT(T.fk_Pedido_ID_Pedido) AS TotalPedidos
from
    Funcionario F
left join
    tem T
on
    T.fk_Funcionario_Funcionario_ID = F.Funcionario_ID
group by
    F.Funcionario_ID
order by
    TotalPedidos DESC;
END &&
delimiter ;

```

Figura 43 - Procedimento RE17

4- Que Funcionários estão associados a um pedido ? (RE 4)

```

# este procedure diz os funcionarios que estao associados a um pedido
delimiter &&
CREATE PROCEDURE `FuncPed` (IN id_ped INT)
) BEGIN
SELECT fk_Funcionario_Funcionario_ID FROM tem where fk_Pedido_ID_Pedido= id_ped;
END &&
delimiter ;

```

Figura 44 - Procedimento RE4

5.7. Plano de segurança e recuperação de dados

De acordo com o que foi definido nos requisitos de controlo, os diferentes tipos de utilizadores vão ter diferentes permissões na base de dados. O Sr. Jefferson Bazos tem controlo sobre toda a informação da base de dados, os gestores de armazém podem alterar o salário e desempenho de um funcionário. Os clientes podem adicionar itens aos seus pedidos e podem alterar a sua própria informação. Os funcionários podem alterar o seu endereço.

```

CREATE USER 'gestor'@'localhost';
SET PASSWORD FOR 'gestor'@'localhost' = 'B@naana1234';
GRANT SELECT, INSERT ON Amazonia.funcionario TO 'gestor'@'localhost'; # RC01
GRANT UPDATE (Salario, Desempenho)
    ON Amazonia.funcionario TO 'gestor'@'localhost';
GRANT SELECT, INSERT, UPDATE ON Amazonia.armazem TO 'gestor'@'localhost'; # RC09
GRANT SELECT ON Amazonia.Pedido TO 'gestor'@'localhost'; # RC05
GRANT UPDATE (Estado, Data_Entrega, Data_Envio)
    ON Amazonia.Pedido TO 'gestor'@'localhost';
GRANT SELECT, INSERT, UPDATE ON Amazonia.Item TO 'gestor'@'localhost'; # RC05
GRANT SELECT, INSERT, UPDATE ON Amazonia.possui TO 'gestor'@'localhost'; # RC05
CREATE USER 'cliente'@'localhost';
SET PASSWORD FOR 'cliente'@'localhost' = 'cliente1234';
GRANT UPDATE, INSERT ON Amazonia.Cliente TO 'cliente'@'localhost'; # RC02
CREATE USER 'JeffersonBazos'@'localhost';
SET PASSWORD FOR 'JeffersonBazos'@'localhost' = 'JeffersonBazos1234';
GRANT ALL ON *.* TO 'JeffersonBazos'@'localhost'; #RC 07
CREATE USER 'funcionario'@'localhost';
SET PASSWORD FOR 'funcionario'@'localhost' = 'funcionario1234';
GRANT UPDATE (Endereco)
    ON Amazonia.funcionario TO 'funcionario'@'localhost';
CREATE USER 'distribuidor'@'localhost';
SET PASSWORD FOR 'distribuidor'@'localhost' = 'distribuidor1234';
GRANT UPDATE (Estado, Data_Entrega, Data_Pedido, Data_Envio)
    ON Amazonia.Pedido TO 'distribuidor'@'localhost';
grant select on Amazonia.Pedido to 'funcionario'@'localhost';

```

Figura 45 - Permissões

6. Conclusões e Trabalho Futuro

Neste projeto foi-nos pedido para realizar a criação de uma base de dados. No âmbito de entregas ao domicílio começamos pela definição do sistema onde definimos a história de Jefferson Bazos e da empresa AmazoniaTM, acreditamos que fizemos uma boa contextualização, bem detalhada e lógica. Contudo acreditamos que a nossa análise da viabilidade poderia ser mais detalhada.

Realizamos o levantamento e análise dos requisitos, acreditamos que os métodos escolhidos para o levantamento foram corretos e expansivos. Inicialmente os requisitos criados não eram suficientes, visto essa carência, estabelecemos novos. Após isso, consideramos que tínhamos um bom número de requisitos que tornaram a nossa transição para a modelação conceptual mais fácil.

Na modelação conceptual julgamos conseguir ter feito uma boa tradução a partir dos requisitos criados. Estando esta estabelecida passamos para a modelação lógica que foi facilitada pelas ferramentas fornecidas de transição entre os modelos. O mesmo para a criação da base de dados no modelo físico.

Acreditamos que no modelo físico definimos bons triggers e procedures, porém temos poucas transações, poderíamos ter mais queries de maior complexidade e uma indexação melhor. Consideramos que criamos uma base de dados de boa dimensão e variada através de um bom povoamento.

Reconhecemos que a base de dados que definimos tem espaço para melhorar. Poderíamos futuramente adicionar uma nova entidade Fornecedor que nos permitiria fazer pedidos aos fornecedores de maneira a controlar melhor o stock e o investimento da empresa. Um outro aspeto a implementar seria adicionar o método de pagamento de os clientes.

Concluindo, consideramos que este trabalho permitiu-nos consolidar o nosso conhecimento sobre bases de dados, como criá-las e as desenvolver. Apesar de todos os aspetos referidos acima, consideramos que conseguimos cumprir os objetivos e um bom trabalho.

Referências

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

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

Lista de Siglas e Acrónimos

BD	Base de Dados
SGBD	Sistema de Gestão de Base de Dados
SQL	Structured Query Language
RE	Requisitos de Exploração