



Universidade do Minho
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Letivo de 2022/2023

<<Yamazon Games>>

**<<José Miguel Ferreira Barbosa (95088),
Leonardo Lordello Fontes (96308),
Maria Filipa Veiga Luso Rodrigues (97536),
Tiago Adriano Pereira Feixa Gomes Moreira (92046) >>**

<<2022/2023>>

BD

<<Yamazon Games>>

**<<José Miguel Ferreira Barbosa (95088),
Leonardo Lordello Fontes (96308),
Maria Filipa Veiga Luso Rodrigues (97536),
Tiago Adriano Pereira Feixa Gomes Moreira (92046) >>**

<<2022/2023>>

Índice

Índice de Tabelas	2
1. Definição do sistema	4
1.1. Contexto de aplicação e fundamentação do sistema	4
1.2. Motivação e Objetivos do Trabalho	4
1.3. Análise da viabilidade do processo	5
1.4. Recursos e Equipa de Trabalho	5
1.5. Plano de Execução do Projeto	6
1.6. Revisão e Aprovação	8
2. Levantamento e Análise de Requisitos	9
2.1. Método de levantamento e de análise de requisitos adotado	9
2.2. Organização dos requisitos levantados	9
2.2.1 Requisitos de descrição	9
2.2.2 Requisitos de exploração	11
2.2.3 Requisitos de controlo	11
2.3. Análise e validação geral dos requisitos	12
3. Modelação Conceptual	13
3.1. Apresentação da abordagem de modelação realizada	13
3.2. Identificação e caracterização das entidades	13
3.3. Identificação e caracterização dos relacionamentos	16
3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	17
3.5. Apresentação do diagrama ER produzido	21
4. Modelação Lógica	23
4.1. Construção e validação do modelo de dados lógico	23
4.2. Desenho do modelo lógico	32
4.3. Validação do modelo com interrogações do utilizador	32
5. Implementação Física	34
5.1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	34
5.2. Tradução das interrogações do utilizador para SQL (alguns exemplos)	37
5.3. Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	39
5.4. Cálculo do espaço da bases de dados (inicial e taxa de crescimento anual)	39
5.5. Plano de segurança e recuperação de dados	41
6. Conclusão	42
Referências Bibliográficas	43

Índice de Tabelas

Tabela 1 – Caracterização das entidades.	15
Tabela 2 – Caracterização dos relacionamentos.	16
Tabela 3 – Caracterização da associação dos atributos com as entidades.	17
Tabela 4 - Caracterização da associação dos atributos com os relacionamentos.	20

Resumo

Neste projeto conhecemos uma empresa de entregas ao domicílio, a Yamazon. O seu dono, Jefferson, tenciona fazer um tratamento mais especializado à divisão de videojogos uma vez que é uma indústria em notória expansão e a demanda destes artigos têm aumentado no website da empresa. Esta será naturalmente chamada de Yamazon Games. Para isto ele contratou uma equipa especialista em bases de dados para desenvolvimento deste projeto. Esta equipa de 4 elementos, juntou-se com a empresa e discutiu aquilo que se pretendia obter com a implementação de um sistema de bases de dados. Após obter aprovação, a equipa, continuou com as fases do ciclo de vida de uma base de dados. Através de muitas reuniões, foram estabelecidos requisitos de descrição, exploração e controlo, muito importantes para a boa realização dos passos seguintes. Após estes serem finalizados, começaram a trabalhar e iniciaram o processo de modelação conceptual, um esquema bastante abstrato e de alto nível, que nos revela as várias entidades envolvidas no problema, os seus relacionamentos, e os diversos atributos que caracterizam as entidades e os relacionamentos. Após a sua conclusão, começaram o modelo lógico tendo sido necessário a validação do esquema e das suas tabelas. Após a sua revisão, começaram a implementação física, desde a implementação do sistema de dados baseado nos modelos anteriores, da criação de queries e pouparam algum trabalho aos funcionários da empresa, povoando-a.

1. Definição do sistema

1.1. Contexto de aplicação e fundamentação do sistema

Yamazon, é uma empresa multinacional de entregas ao domicílio portuguesa, com sede na cidade de Braga. Através do seu website Yamazon.pt, a companhia vende artigos de todos os tipos.

Esta empresa foi fundada, em 5 de julho de 1994 por Jefferson Beijos, um brasileiro, também conhecido por Adamastor, por causa do seu titânico impacto na economia internacional. A sua empresa foi referida como uma das forças económicas e culturais mais influentes do mundo e a marca mais valiosa do mundo ocidental.

Nos últimos anos, devida a notória expansão da indústria de jogos eletrónicos, a demanda destes artigos na Yamazon tem aumentado. O que levou Adamastor a considerar um tratamento mais personalizado a esta área.

A Yamazon pretende criar uma plataforma, a Yamazon Games, focada na venda de produtos desta indústria (mantendo as entregas a domicílio). Para tal, ela pretende efetuar testes no distrito de Braga. Em caso de sucesso, ele pretende efetivar a região de Braga e expandir o serviço para outras regiões.

Para efetuar estes testes, a empresa precisará de um sistema de base de dados local para gerir toda a informação necessária para a validação do teste.

1.2. Motivação e Objetivos do Trabalho

Como forma de avaliar a implementação da nova plataforma, Yamazon Games, Jefferson Beijos traçou alguns objetivos para facilitar a validação do teste:

1. Armazenar informações precisas e atualizadas sobre os funcionários, utilizadores, pedidos, itens, veículos, armazéns.
2. Gerir o processamento e a entrega de pedidos, garantindo que os pedidos sejam processados e entregues aos clientes de maneira rápida e precisa.

3. Otimizar o planejamento de rotas e a alocação de veículos, para garantir que os veículos sejam utilizados de maneira eficiente e econômica.
4. Personalizar a experiência do cliente, adaptando recomendações e ofertas de acordo com os interesses e comportamentos dos clientes.
5. Melhorar a experiência do utilizador na plataforma, permitindo-o associar dados pessoais à uma conta, de forma a não repetir processos manuais.
6. Gerir o stock de itens, garantindo que este esteja sempre atualizado e que os itens estejam disponíveis para venda quando necessário.
7. Melhorar o gerenciamento de finanças, armazenando e organizando informações sobre receitas dos artigos e despesas dos artigos, funcionários e veículos.
8. Gerir o acesso de funcionários a recursos da empresa, baseado nas suas funções.

1.3. Análise da viabilidade do processo

Para que o projeto que está a ser testado seja efetivado, Jefferson Beijos espera obter os seguintes resultados com a ajuda do sistema de bases de dados:

1. A partir de 12 meses da implementação do sistema, as receitas mensais provenientes da plataforma Yamazon Games deverão superar as despesas mensais da mesma.
2. Ao final de 36 meses da implementação do sistema, todo investimento inicial terá sido recuperado.
3. A partir de 12 meses da implementação do sistema, a plataforma deverá conter pelo menos 5000 utilizadores ativos, isto é, utilizadores que efetuem pelo menos uma compra por mês.
4. A partir de 18 meses da implementação do sistema, a plataforma Yamazon Games deverá vender pelo menos 20% mais artigos do que a quantidade de artigos relacionados a jogos que eram vendidos na plataforma principal Yamazon mensalmente na mesma região.
5. Personalizar o tratamento dos clientes, de forma a facilitar o encontro de artigos que o cliente possa estar interessado, aumentando as vendas.
6. Fácil acesso as despesas referentes ao salário dos funcionários, compra de veículos e manutenção dos mesmos, bem como a compra de itens, de forma a verificar gastos desnecessários, tornando o negócio mais lucrativo.

1.4. Recursos e Equipa de Trabalho

Recursos

-Humanos:

1. Equipa de desenvolvimento
2. Funcionários da empresa

-Materiais:

1. Hardware: 1 servidor e 5 máquinas (inicialmente, pois este valor pode vir a crescer).
2. Software: Sistema de gestão de bases de dados.

Equipa de Trabalho

-A empresa:

- Atendimento ao cliente, funcionamento do escritório e prestação de serviços.

-A equipa de desenvolvimento:

- Levantamento de requisitos, modelação do sistema, implementação do sistema.

1.5. Plano de Execução do Projeto

Para definir e planear a forma como o processo de desenvolvimento do sistema de bases de dados iria ser realizado, o líder da equipa de desenvolvimento traçou um plano de execução baseado nas fases do ciclo de vida de um sistema de bases de dados.

Expectativa:

Implementação de um sistema de base de dados: SIMPLE GANTT CHART by Vertex42.com

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

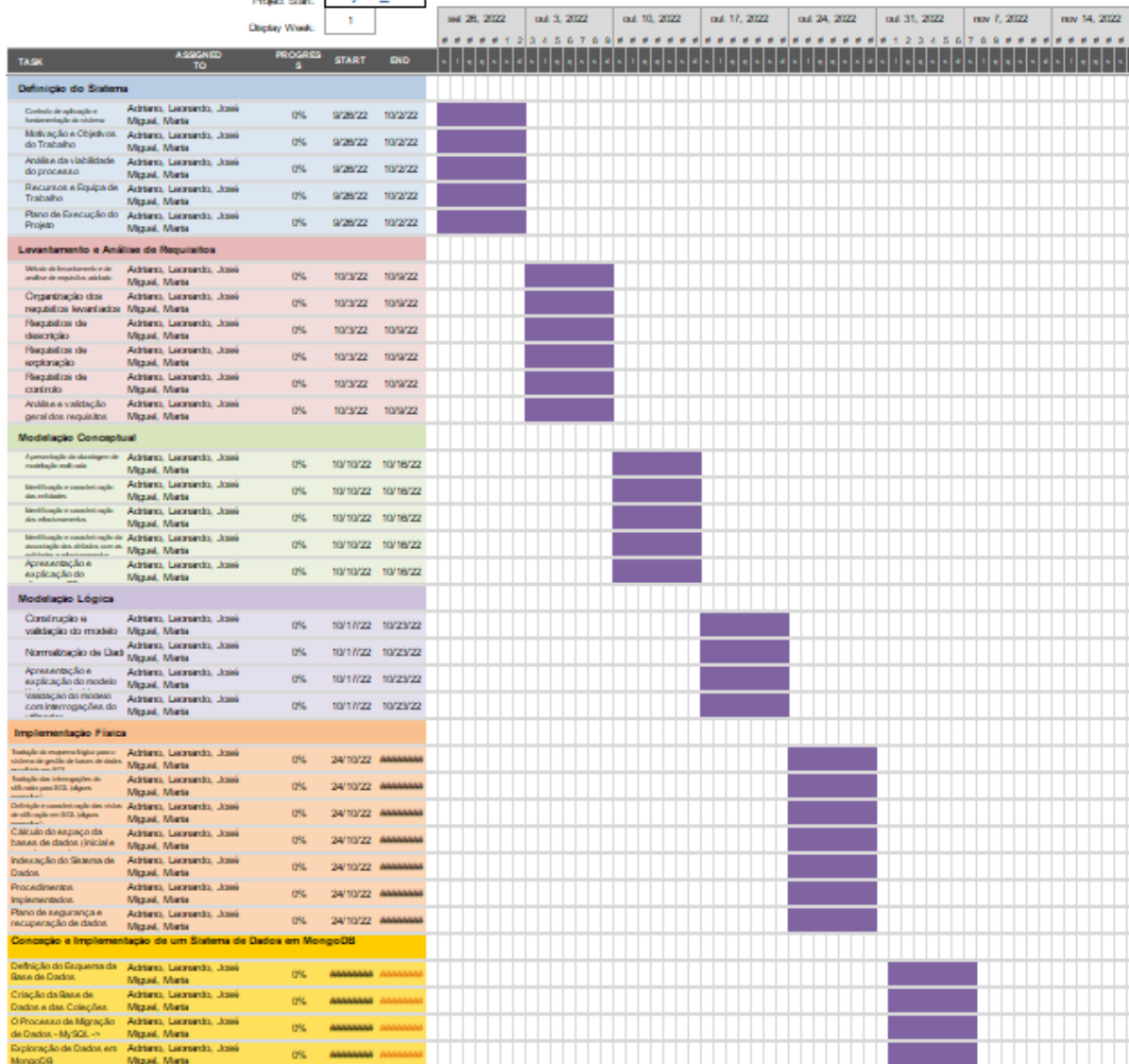
Yamazon

Grupo 6 LDA.

Project Start:

Project_Start

Classroom Walk:



Realidade:

[illegible]

1.6 Revisão e Aprovação

Depois de uma análise cuidadosa, numa reunião com toda a equipa de trabalho, dos fundamentos e de todos os objetivos bem como a sua viabilidade. O projeto foi validado e aprovado na sua totalidade. Permitindo deste modo avançar com o desenvolvimento do projeto.

2. Levantamento e Análise de Requisitos

2.1. Método de levantamento e de análise de requisitos adotado

Devido à elevada experiência de Jefferson Beijos no mundo dos negócios, mais especificamente nas entregas ao domicílio, bastou realizar reuniões entre ele e a equipa de desenvolvimento de forma a traçar os requisitos necessários para um bom funcionamento da plataforma para que seja possível atingir os objetivos anteriormente estabelecidos.

2.2. Organização dos requisitos levantados

2.2.1 Requisitos de descrição

A plataforma Yamazon Games tem um website (yamazongames.pt) e um aplicativo (Yamazon Games).
As vendas dos artigos são feitas através do website ou através do aplicativo.
Para criar uma conta na plataforma, o utilizador precisa fornecer um nome completo, um endereço de correio eletrónico e uma senha.
Após fornecer os dados para a criação da conta, o utilizador receberá um email de confirmação para validar a sua conta.
Os dados inseridos pelo utilizador só são armazenados se forem válidos (não existem dois utilizadores com o mesmo email).
Um endereço é constituído por uma rua e um código postal.
Um utilizador pode ter mais do que um endereço.
Para proceder com uma compra o Utilizador pode escolher um endereço, entre os disponíveis na sua conta, ou pode colocar um outro endereço.
Para finalizar a criação da conta, o utilizador precisa seguir as instruções do email de confirmação.
O método de pagamento e os dados do método de pagamento não são armazenados.
A finalização de uma compra configura a criação de um pedido.
Um pedido é constituído por um identificador (um número inteiro gerado automaticamente), data de emissão, data limite de entrega, data que foi entregue, estado, rua, código-postal, itens que foram

comprados, o preço, o utilizador que fez a compra, um estafeta.
Existem 3 estados possíveis para o estado de um pedido: em espera, a caminho, entregue.
O estado inicia-se em "em espera" após a criação de um pedido.
O custo do pedido já tem incluído o custo de entrega determinada por um algoritmo da plataforma.
Cada utilizador pode fazer múltiplos pedidos.
Caso um pedido ultrapasse a data limite de entrega, na data de entrega o cliente recebe um vale de desconto, para a plataforma Yamazon, dado pelo estafeta no ato de entrega.
Um funcionário tem um identificador (chave primária gerada automaticamente), nome completo, salário e categoria.
Uma categoria tem um identificador (gerado automaticamente), um tipo (estafeta por exemplo) e uma descrição.
Um estafeta é um funcionário.
A aplicação garante que apenas os funcionários com categoria de estafetas podem ser atribuídos veículos e pedidos.
Um veículo tem um identificador (chave primária gerada automaticamente), uma matrícula única, um valor correspondente ao gasto da sua compra e uma data de aquisição.
A manutenção é constituída por uma data de manutenção, uma descrição, um gasto e um identificador (gerado automaticamente).
Um veículo pode efetuar várias manutenções.
Os gastos de um veículo correspondem a qualquer gasto realizado com este, seja no reparo, seja com gasolina, etc.
Um estafeta pode conduzir vários veículos e um veículo pode ser conduzido por vários estafetas.
Todos os dias, os estafetas são atribuídos a um veículo.
O estafeta só é atribuído no dia da entrega de um determinado pedido.
Todo item tem um identificador (chave primária gerada automaticamente), data de aquisição, custo ao fornecedor, um nome, um custo ao cliente, uma categoria, o género, as consolas e pode ter uma descrição.
Inicialmente os únicos itens disponíveis são jogos e consolas, mas pretende-se expandir as possibilidades.
Além dos atributos normais os jogos têm um atributo multivalorado de géneros, um multivalorado das consolas para qual aquele determinado item funciona.
Os jogos têm de ter um género como por exemplo: ação, aventura, quebra-cabeça, tiro, estratégia, simulação, desporto, luta, corrida, plataforma, MMO, RPG, arcade.
As consolas têm de ter um nome por exemplo Playstation 4, Playstation 5, Xbox One, Xbox Series X, Xbox Series S.
A aplicação garante que apenas os itens com categoria de jogo podem ter um array de géneros e consolas.
Um armazém tem um identificador (chave primária gerada automaticamente), um endereço, um código-postal.
Um armazém contém um ou mais itens.

2.2.2 Requisitos de exploração

A qualquer momento após a criação da conta, o utilizador pode fornecer um telemóvel e endereços.
A qualquer momento o utilizador pode alterar o seu nome, endereço eletrónico, a senha, os endereços e o telemóvel.
A plataforma Yamazon Games faz todas as validações necessárias dos dados inseridos pelo utilizador, seja na criação da conta, seja na alteração dos dados.
A plataforma notifica o utilizador quando seu input é inválido e o porquê.
A plataforma recomenda ao utilizador itens seguindo dois critérios: o primeiro é baseado nas suas compras passadas, o outro é baseado nas tendências dos utilizadores.
A data de emissão é gerada automaticamente baseada na data e hora do sistema no momento que o pedido foi criado.
A data limite de entrega é gerada automaticamente baseado num algoritmo interno da plataforma.
O custo de um pedido é calculado automaticamente pela soma do preço de todos os itens mais a taxa de entrega (+23% do custo total).
O estafeta muda o estado do pedido, com um certo identificador, quando este está respetivamente "a caminho" e "entregue".
A data de entrega é gerada baseado na data e hora do sistema no momento, quando o estado muda para "entregue".
Ao final de cada dia é gerado um relatório para saber quantos pedidos foram entregues no total.
Ao final de cada mês é gerado um relatório com as receitas dos jogos, das consolas e de todos juntos.
Ao final de cada mês é gerado um relatório com a quantidade de itens com o mesmo nome que foram vendidos.
Ao final de cada mês é gerado um relatório com a quantidade de jogos que foram vendidos por género.
Ao final de cada mês é gerado um relatório com as despesas dos itens adquiridos naquele mês.
Ao final de cada mês é gerado um relatório com as despesas de cada veículo e a despesa total deles juntos.
Ao final de cada mês é gerado um relatório com as despesas correspondente ao pagamento dos funcionários.

2.2.3 Requisitos de controlo

Não é necessário ter uma conta criada no website ou aplicativo para efetuar pesquisa de artigos de forma não personalizada.
Para um tratamento mais personalizado das pesquisas e para fazer compras é necessário ter uma conta criada na plataforma Yamazon Games.
Uma conta criada é associada à plataforma, isto é, a mesma conta (com os mesmos dados) pode ser utilizada no website ou no aplicativo.
Para proceder com uma compra o Utilizador precisa de ter antes definido um endereço e um código-postal.
A decisão de qual estafeta é atribuído a cada pedido é feita pelo CEO com o auxílio dos dados.
O utilizador não tem permissão de alterar nenhum dado do seu pedido após finalizar a compra.
O utilizador tem acesso a todo o histórico de pedidos realizados por ele.
O estafeta tem acesso a todos os dados em relação aos pedidos a qual ele foi atribuído, mas apenas pode alterar o estado de entrega do pedido.

Os únicos funcionários que têm permissão de adicionar, remover e alterar itens do sistema são o CEO, e os gestores de stock.
Quem atribuí os estafetas aos seus veículos é o CEO.
Os funcionários de apoio ao cliente têm acesso aos dados dos utilizadores.

2.3. Análise e validação geral dos requisitos

Após a realização de inúmeras reuniões com os elementos da equipa e o Jefferson Beijos e da execução dos requisitos, estes foram analisados aprovados na sua totalidade.

3. Modelação Conceptual

3.1. Apresentação da abordagem de modelação realizada

Através dos requisitos, previamente apresentados, conseguiu-se identificar as principais entidades, e os respetivos atributos e relacionamentos, necessárias para a base de dados da Yamason Games, de modo a armazenar a informação e melhor gerir o processo de entregas dos pedidos.

3.2. Identificação e caracterização das entidades

Da análise aprofundada dos requisitos foi possível identificar as entidades que vão estar presentes na futura base de dados. Esta identificação foi feita através dos próximos nove requisitos, estando o nome da entidade, referente a cada requisito, destacada a azul e os respetivos atributos a verde.

1. “Para criar uma conta na plataforma, o **utilizador** precisa fornecer um **nome completo**, um **endereço de correio eletrónico** e uma **senha**.”
2. “Um **endereço** é constituído por uma **rua** e um **código-postal**.”
3. “Um **pedido** é constituído por um **identificador** (um **número inteiro** gerado automaticamente), **data de emissão**, **data limite de entrega**, **data que foi entregue**, **estado**, **rua**, **código-postal**, itens que foram comprados, o **preço**, o utilizador que fez a compra, um estafeta.”
4. “Um **funcionário** tem um **identificador** (chave primária gerada automaticamente), **nome completo**, **salário** e categoria.”
5. “Uma **categoria** tem um **identificador** (gerado automaticamente), um **tipo** (estafeta por exemplo) e uma **descrição**.”
6. “Um **veículo** tem um **identificador** (chave primária gerada automaticamente), uma **matrícula** única, um valor correspondente ao **gasto** da sua compra e uma **data de aquisição**.”

7. “A **manutenção** é constituída por uma **data de manutenção**, uma **descrição**, um **gasto** e um **identificador** (gerado automaticamente).”
8. “Todo **item** tem um **identificador** (chave primária gerada automaticamente), **data de aquisição**, **custo ao fornecedor**, um **nome**, um **custo ao cliente**, uma **categoria**, o **género**, as **consolas** e pode ter uma **descrição**.”
9. “Um **armazém** tem um **identificador** (chave primária gerada automaticamente), um **endereço**, um **código-postal**.”

Após a execução da identificação das entidades, foi procedida a caracterização de cada uma delas. Para o efeito, a informação retirada foi colocada numa tabela.

Designação	Descrição	Sinónimos	Ocorrência
Utilizador	Informação relativa aos clientes e suas contas na plataforma Yamazon Games.	Cliente	Um utilizador tem um número único associado. Cada registo é atribuído automaticamente após a confirmação da criação de uma conta na plataforma.
Pedido	Informações de localização, datas, custo, bem como o estado de um pedido.	Encomenda	Um pedido tem um número único associado. Cada registo é atribuído automaticamente, imediatamente após a finalização de uma compra na plataforma.
Endereço	Localização de uma possível entrega de um pedido de um utilizador.	-	Um endereço tem um número único associado. Cada registo é atribuído automaticamente após o preenchimento de uma secção da plataforma relativa a localização por parte de um utilizador.
Item	Os itens que são vendidos na plataforma Yamazon	Produto, Artigo	Todos os itens estão registados de forma

	Games.		única na base de dados.
Armazéns	Informação relativa ao local onde os itens são armazenados.	Depósito	Todos os armazéns estão registados de forma única na base de dados.
Funcionário	Informação relativa aos funcionários responsáveis pelo setor da Yamazon Games.	-	Um funcionário tem um número único associado. Cada registo é atribuído manualmente após a contratação de um novo funcionário.
Categoria	O papel de cada funcionário na Yamazon Games.	Cargo	Todas as categorias estão registadas de forma única na base de dados.
Veículo	Os pedidos são entregues através dos veículos, temos aqui a informação destes veículos.	-	Um veículo tem um número único associado. Cada registo é atribuído manualmente após a aquisição de um novo veículo.
Manutenção	Informação relativa a manutenção de um veículo.	Reparação	Uma manutenção tem um número único associado. Cada registo é atribuído manualmente após a finalização da manutenção.

Tabela 1 – Caracterização das entidades.

3.3. Identificação e caracterização dos relacionamentos

À semelhança do que foi concretizado no ponto anterior, analisou-se os requisitos de forma a ser possível a identificação dos relacionamentos. Os requisitos dos quais se identificou os oito relacionamentos, onde as entidades que se relacionam se encontram destacadas a cor azul, são os seguintes.

1. “Um **utilizador** pode ter mais do que um **endereço**.”
2. “Cada **utilizador** pode fazer múltiplos **pedidos**.”
3. “Um **funcionário** tem um identificador (chave primária gerada automaticamente), nome completo, salário e **categoria**.”
4. “Todos os dias, os **estafetas** são atribuídos a um **veículo**.”
5. “Um **armazém** contém um ou mais **itens**.”
6. “Um **pedido** é constituído por um identificador (um número inteiro gerado automaticamente), data de emissão, data limite de entrega, data que foi entregue, estado, rua, código-postal, **itens** que foram comprados, o preço, o utilizador que fez a compra, um **estafeta**.”
7. “Um **veículo** pode efetuar várias **manutenções**.”

Depois de realizada a identificação dos relacionamentos, é necessário fazer a sua caracterização de modo a melhor entender a cardinalidade de cada um. Tal é apresentada por uma tabela.

Entidade	Relacionamento	Cardinalidade	Entidade
Utilizador	Utilizador-Endereço	1: N	Endereço
Utilizador	Pedido-Utilizador	1: N	Pedido
Funcionário	Funcionário-Categoria	N: 1	Categoria
Funcionário	Estafetas-Veículo	M: N	Veículo
Armazéns	Armazéns-Item	1: N	Item
Pedido	Funcionário-Pedido	N: 1	Funcionário
Pedido	Item-Pedido	M: N	Item
Veículo	Veículo-Manutenção	1: N	Manutenção

Tabela 2 – Caracterização dos relacionamentos.

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

De forma a identificar e caracterizar a associação dos atributos com as entidades e relacionamentos, estes foram organizados numa tabela indicando a respetiva entidade ou relação, os seus atributos, descrição, domínio e tamanho, se é nulo ou não e um exemplo.

Entidade	Atributo	Descrição	Domínio e Tamanho	Nulo
Utilizador	ID	Identificador do utilizador/cliente	INT	N
	Nome	Nome do utilizador	VARCHAR(100)	N
	Correio_Eletrónico	O e-mail do utilizador	VARCHAR(100)	N
	Senha	Palavra passe que o utilizador usa para aceder à plataforma	VARCHAR(45)	N
	Telemovel	Número de telemóvel do utilizador	VARCHAR(15)	S
Pedido	ID	Identificador do pedido	INT	N
	Data_Limite	Data e hora máxima planeada para a entrega da encomenda	DATETIME	N
	Estado_de_Entrega	Qual das fases de processamento da entrega a encomenda está	VARCHAR(45)	N

		atualmente (Em espera, a caminho ou entregue)		
	Data_que_foi_Entregue	Data e hora em que a entrega foi realizada	DATETIME	N
	Rua	Rua do local de entrega	VARCHAR(100)	S
	Codigo_Postal	Código-posta do local de entrega	VARCHAR(45)	N
	Data_Emissão	Data e hora da realização do pedido	DATETIME	N
	Preço	Valor total do pedido	DECIMAL(5,2)	N
Endereço	ID	Identificador do endereço	INT	N
	Codigo_Postal	Código-postal do utilizador	VARCHAR(45)	S
	Rua	Nome da rua do utilizador	VARCHAR(100)	S
Item	ID	Identificador do produto	INT	N
	Custo_ao_Fornecedor	Custo da compra do artigo diretamente ao fornecedor	DECIMAL(5,2)	N
	Custo_ao_Cliente	Custo de compra do produto para o cliente	DECIMAL(5,2)	N

	Nome	Nome do produto	VARCHAR(100)	N
	Descrição	Informação adicional sobre o produto em formato de texto	TEXT	S
	Data_de_Aquisição	Data e hora da compra do artigo ao fornecedor	DATETIME	N
	Categoria	Tipo de artigo	VARCHAR(100)	N
Armazéns	idArmazens	Identificador do armazém	INT	N
	Endereço	Endereço do armazém	VARCHAR(100)	N
	Código-Postal	Código-postal do armazém	VARCHAR(45)	N
Funcionário	idFuncionário	Identificador do funcionário	INT	N
	Nome	Nome do funcionário	VARCHAR(100)	N
	Salário	Salário do funcionário	DECIMAL(10,2)	N
Categoria	IdCategoria	Identificador da categoria	INT	N
	Descrição	Informações sobre a função desempenhada do funcionário com a respetiva categoria	TEXT	N
	Tipo	Nome da função do	VARCHAR(45)	N

		funcionário		
Veículo	IdVeículo	Identificador do veículo	INT	N
	Matrícula	Matrícula do veículo	VARCHAR(45)	N
	Gastos	Custo na compra do veículo	DECIMAL(10,2)	N
	Data_de_Aquisição	Data em que o veículo foi comprado	DATE	N
Manutenção	IdManutenção	Identificador da manutenção de um veículo	INT	N
	Data_de_Manutenção	Data em que a manutenção foi realizada	DATE	N
	Descrição	Informações sobre a manutenção feita em certo veículo	TEXT	N
	Gasto	Custo da manutenção	DECIMAL(5,2)	N

Tabela 3 – Caracterização da associação dos atributos com as entidades.

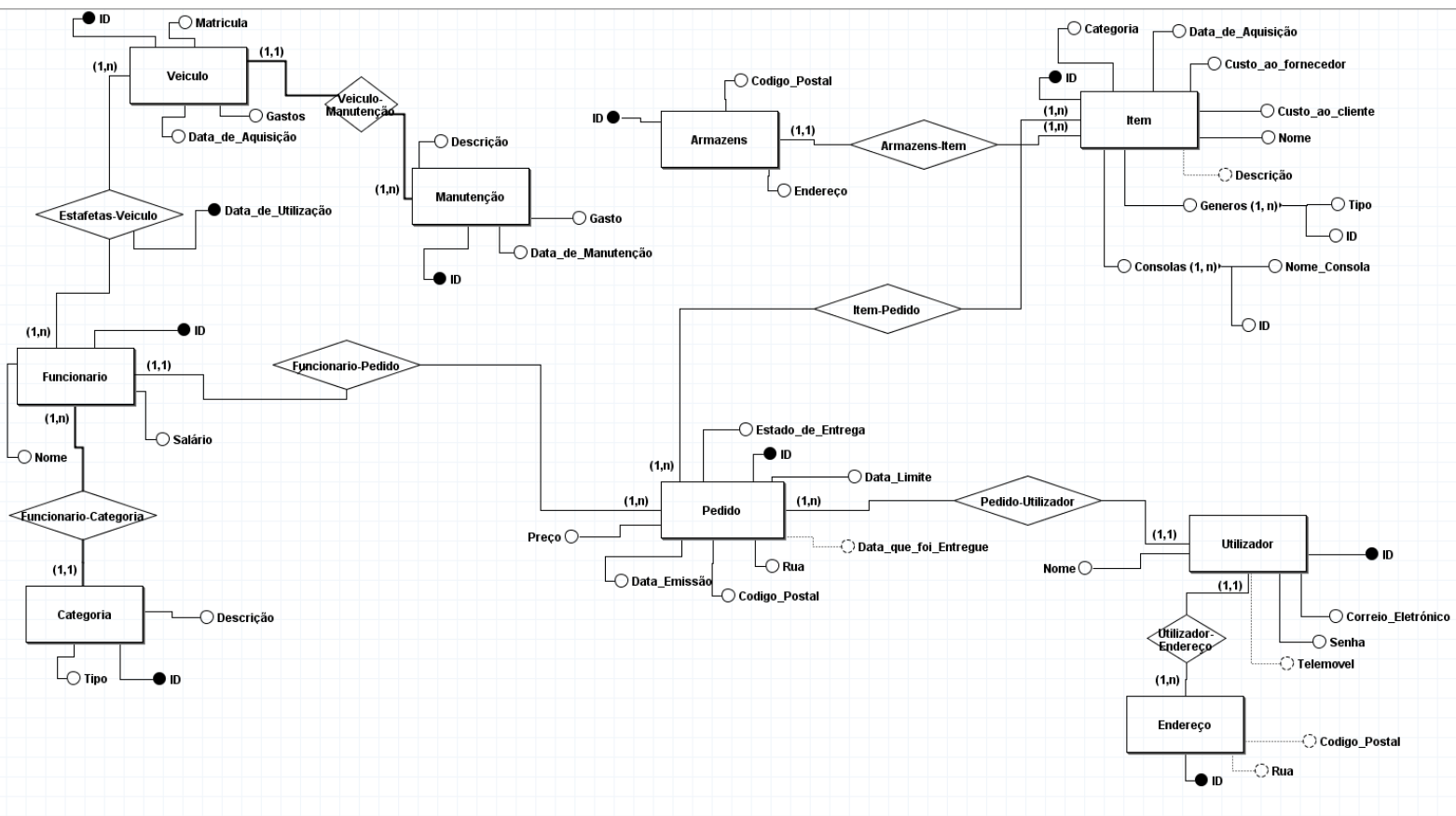
Relacionamento	Atributo	Descrição	Domínio e Tamanho	Nulo
Estafetas_Veiculo	Data_de_Utilização	Data em que o veículo foi utilizado por um determinado estafeta	DATE	N

Tabela 4 - Caracterização da associação dos atributos com os relacionamentos.

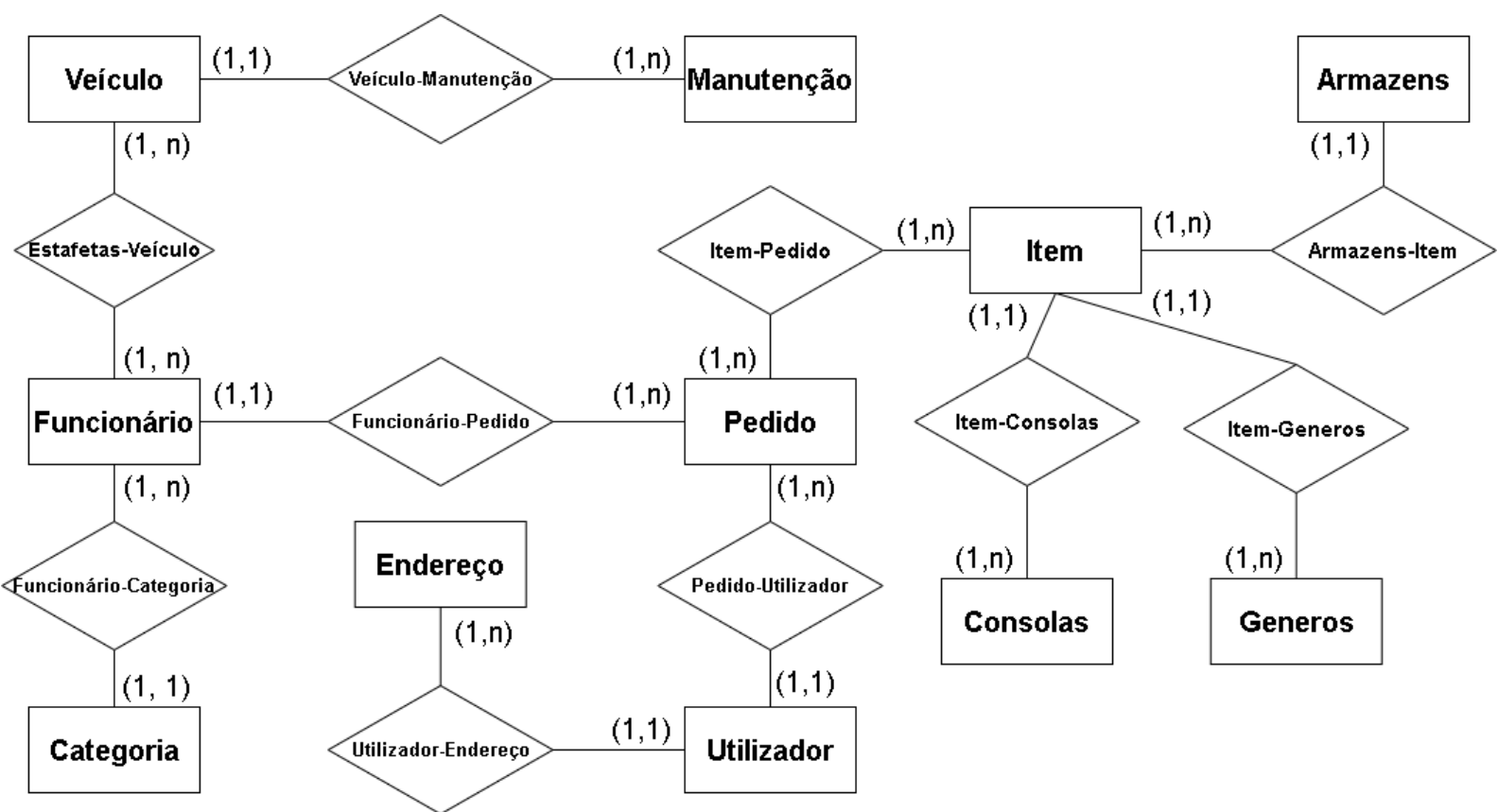
3.5. Apresentação do diagrama ER produzido

Consoante os pontos 3.2, 3.3, 3.4, o diagrama é gerado quase de imediato. Este foi construído tendo por base os requisitos obtidos e aprovados. Através desses requisitos foram adquiridas as diversas entidades (previamente identificadas e caracterizadas), as relações necessárias e os atributos a elas associadas para a futura base de dados.

O esquema conceptual foi realizado, baseando-se na notação de Chen, na ferramenta BR Modelo.



De modo a ser possível uma visualização mais perceptível e de fácil leitura, é apresentado de seguida as entidades e as respetivas relações, sem os atributos (tendo estes já sido indicados).

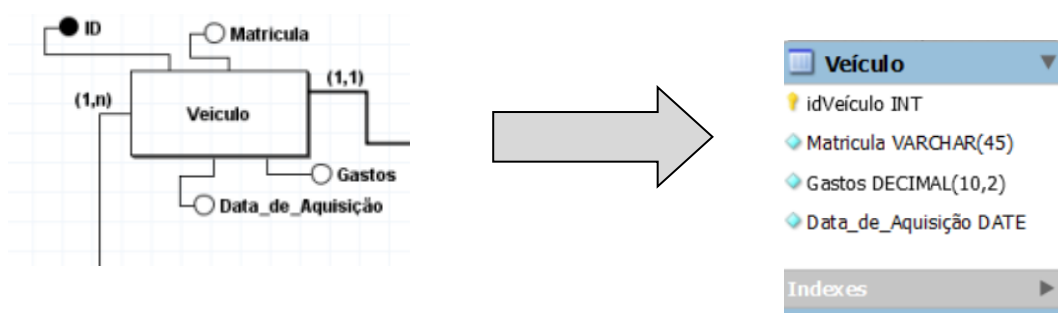


4. Modelação Lógica

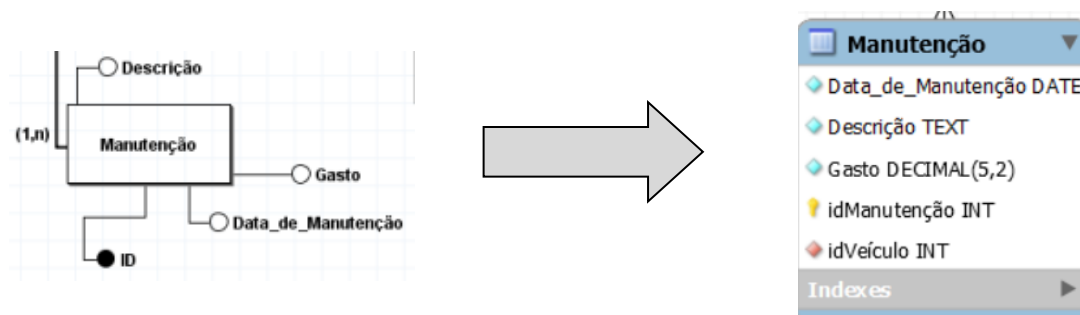
4.1. Construção e validação do modelo de dados lógico

A construção do modelo lógico foi feita com base no modelo conceptual apresentado anteriormente. Através das diversas entidades e relações muitos para muitos foi possível a construção das tabelas do modelo lógico.

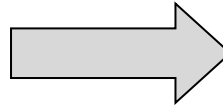
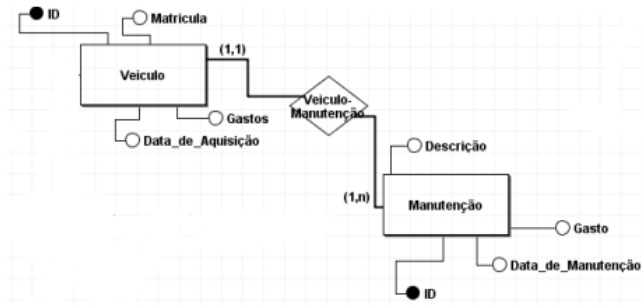
A entidade veículo deu origem a uma tabela com o mesmo nome.



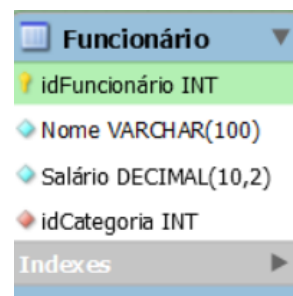
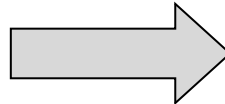
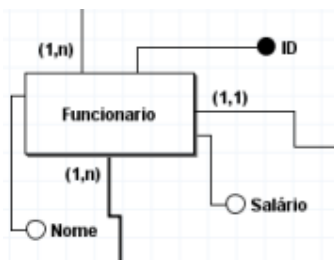
Manutenção é uma entidade fraca que gera uma tabela.



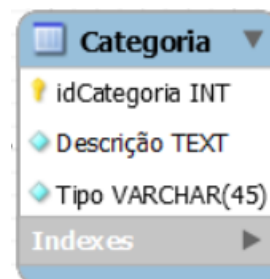
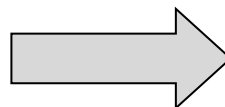
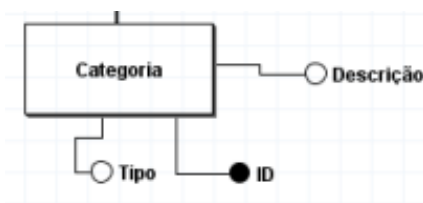
A relação de 1 (Veículo) para N (Manutenção) resultada na criação de uma chave estrangeira na tabela de cardinalidade N (Manutenção).



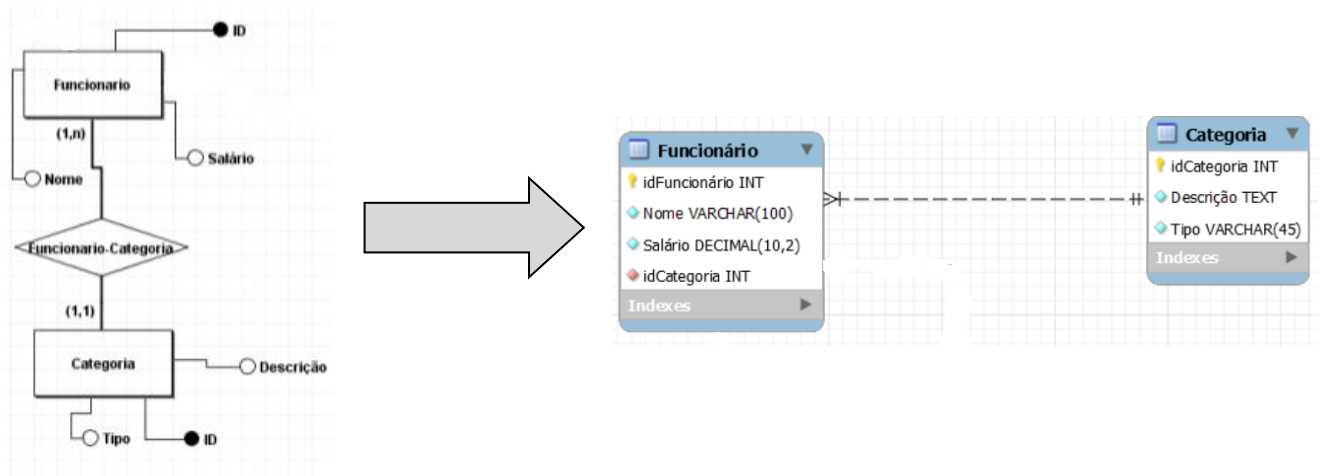
A entidade funcionário dá origem a uma tabela.



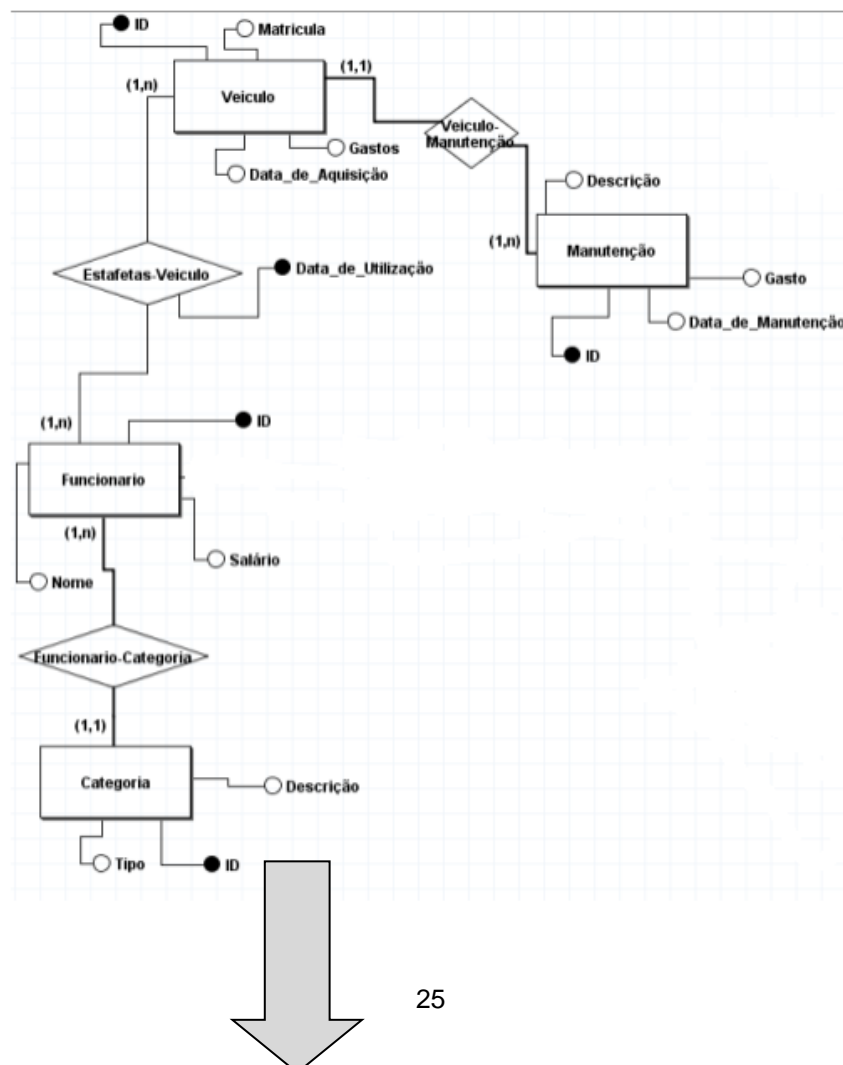
A Categoria é uma entidade fraca e gera uma tabela.

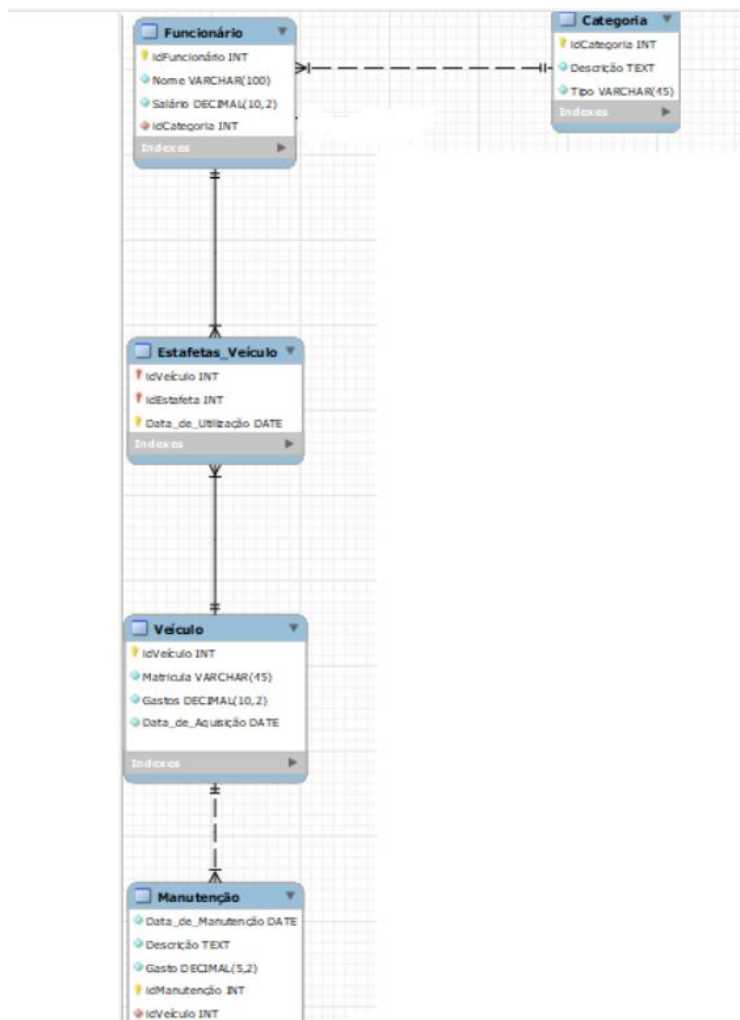


Da relação de 1 (Funcionário) para N (Categoria) resultada na criação de uma chave estrangeira na tabela de cardinalidade N (Categoria).

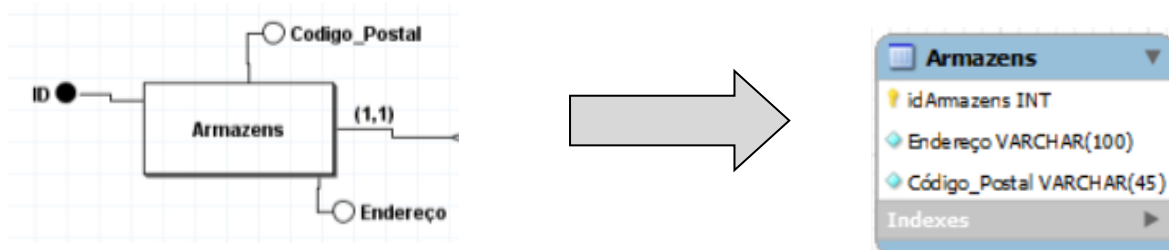


Da relação de M (Funcionários) para N (Veículos) resulta uma nova tabela intermédia a Estafetas_Veículos e duas relação 1:N onde a tabela da cardinalidade N é a tabela Estafetas_Veículos e as tabelas de cardinalidade 1 são a Funcionários e a Veículos. Como na tabela intermédia tem cardinalidade N, esta vai ter duas chaves estrangeiras.

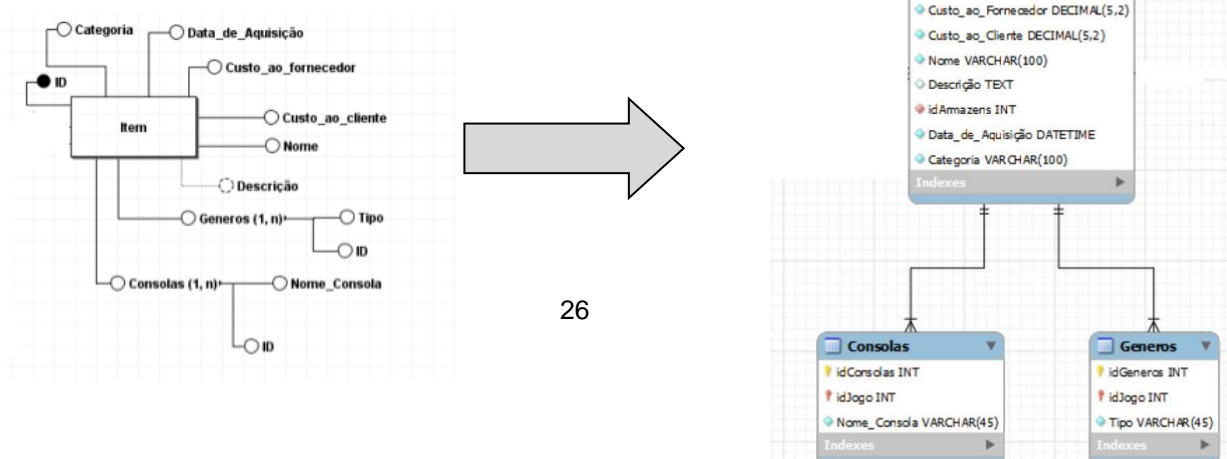




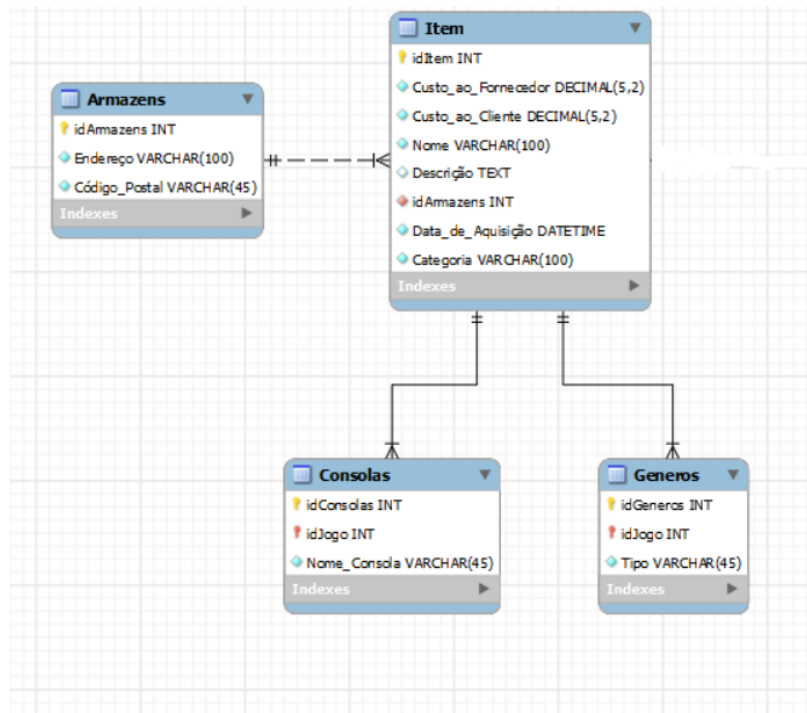
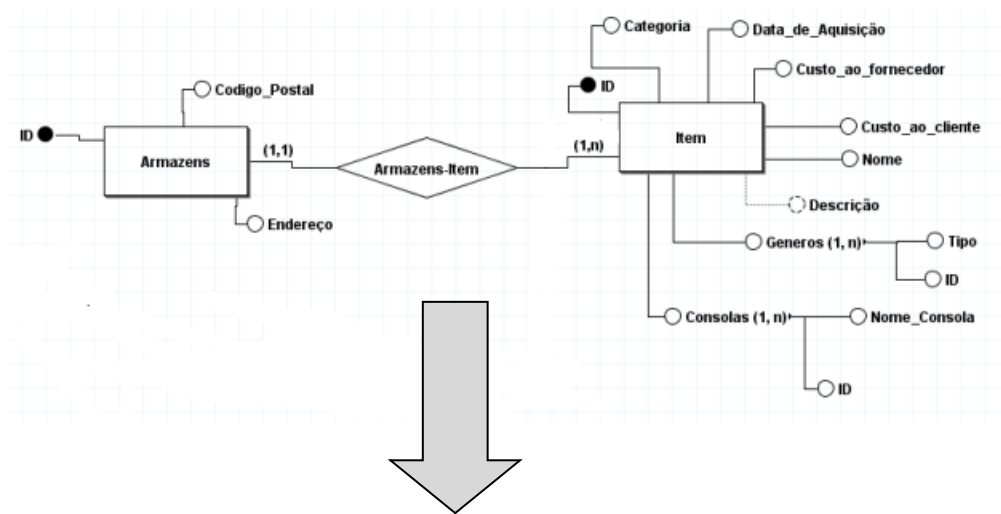
A entidade armazéns dá origem a uma tabela.



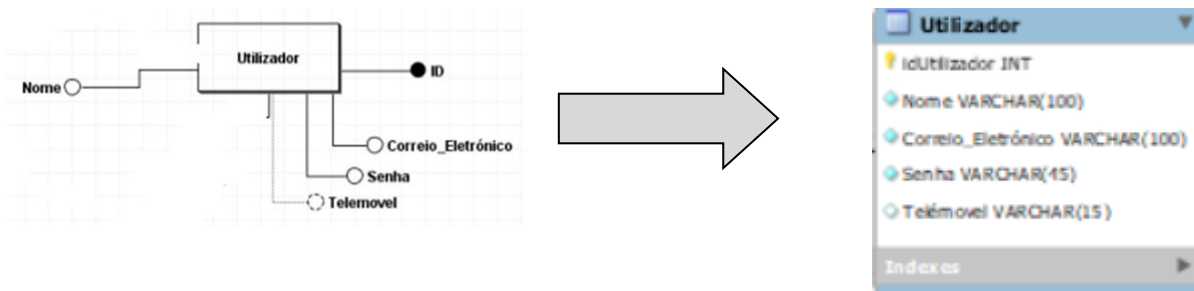
A entidade item gera uma tabela e os seus atributos compostos multivariados (consolas e géneros) vão gerar uma tabela, bem como uma relação de N:1 para a tabela item onde as tabelas consolas e géneros vão ter cardinalidade N (daí que cada uma vai ter uma chave estrangeira).



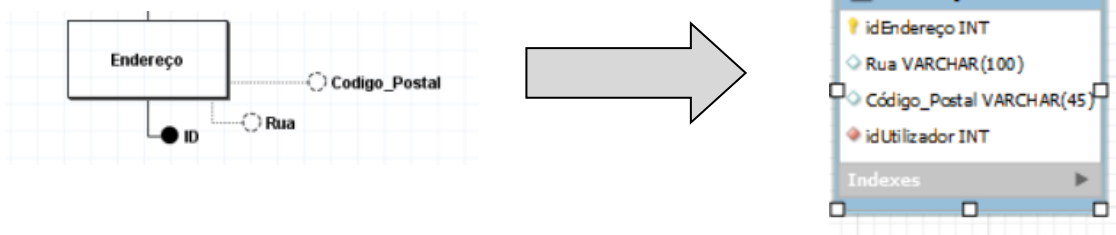
A relação 1 (Armazéns) para N (Item) gera uma chave estrangeira na tabela cuja cardinalidade é N (Itens).



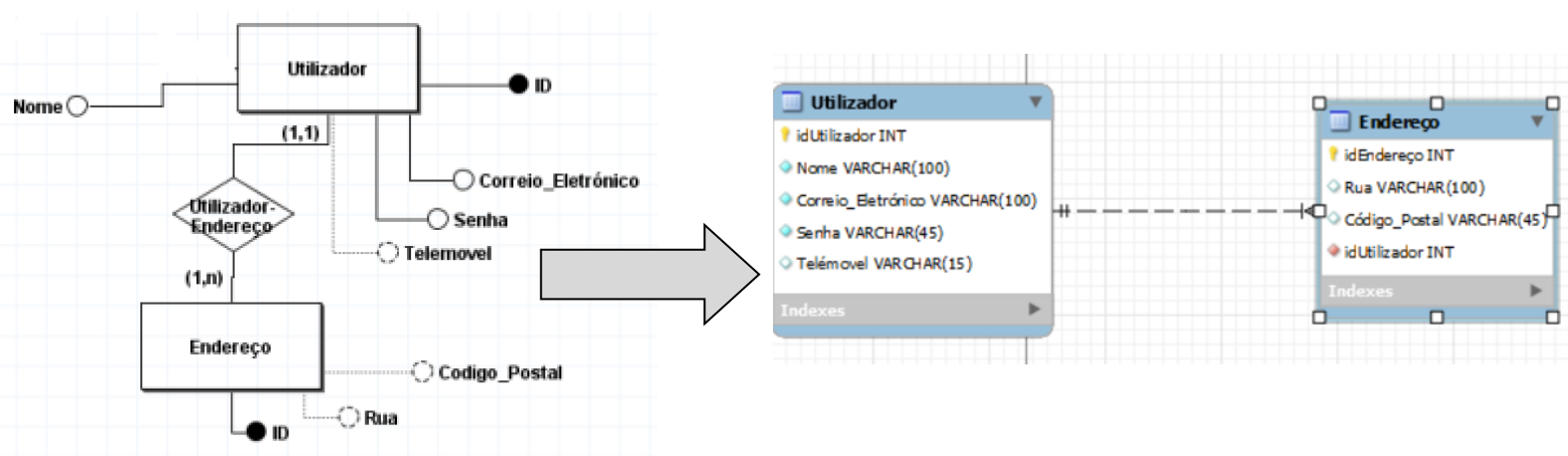
A entidade Utilizador dá origem a uma tabela.



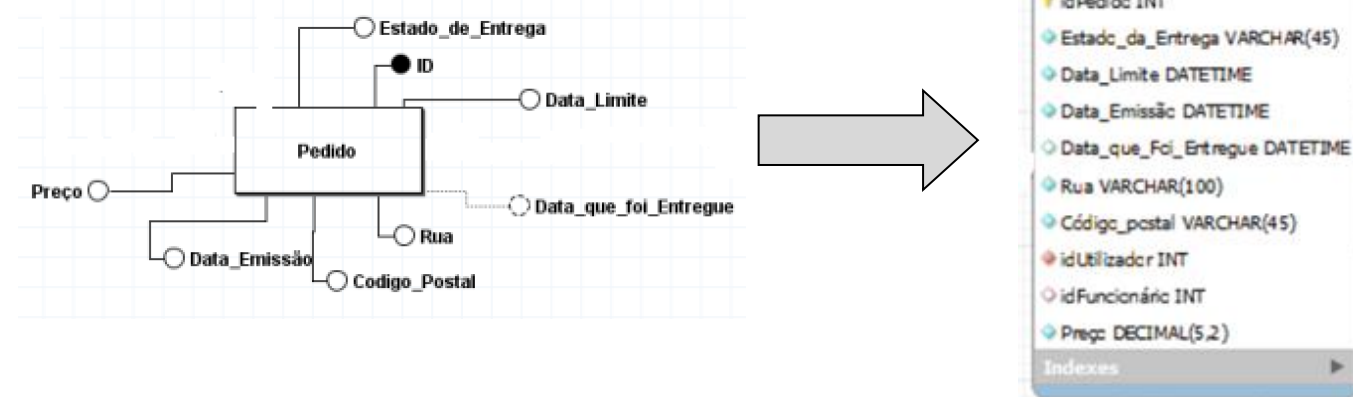
Endereço é uma entidade fraca que gera uma tabela.



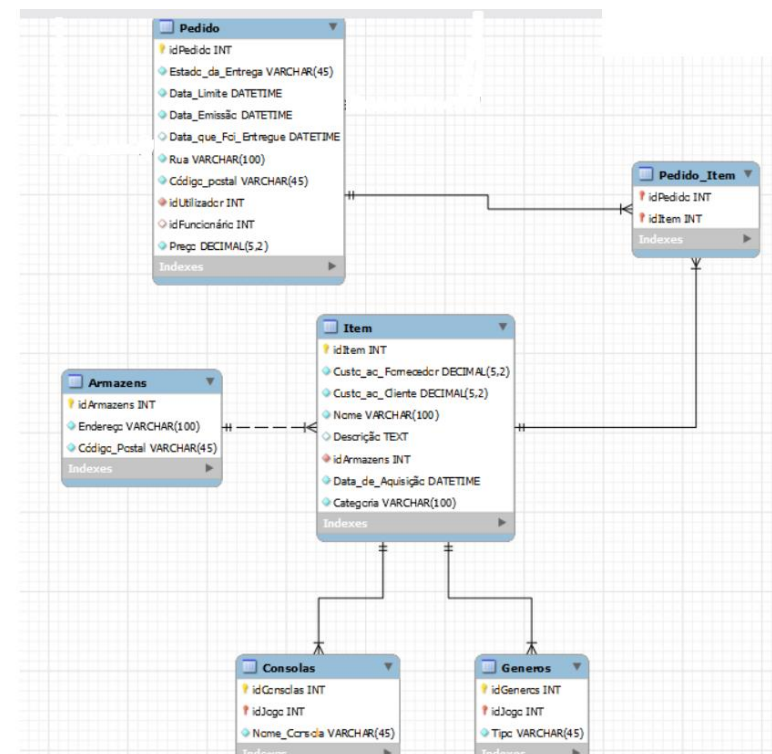
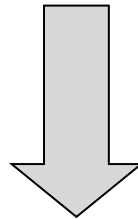
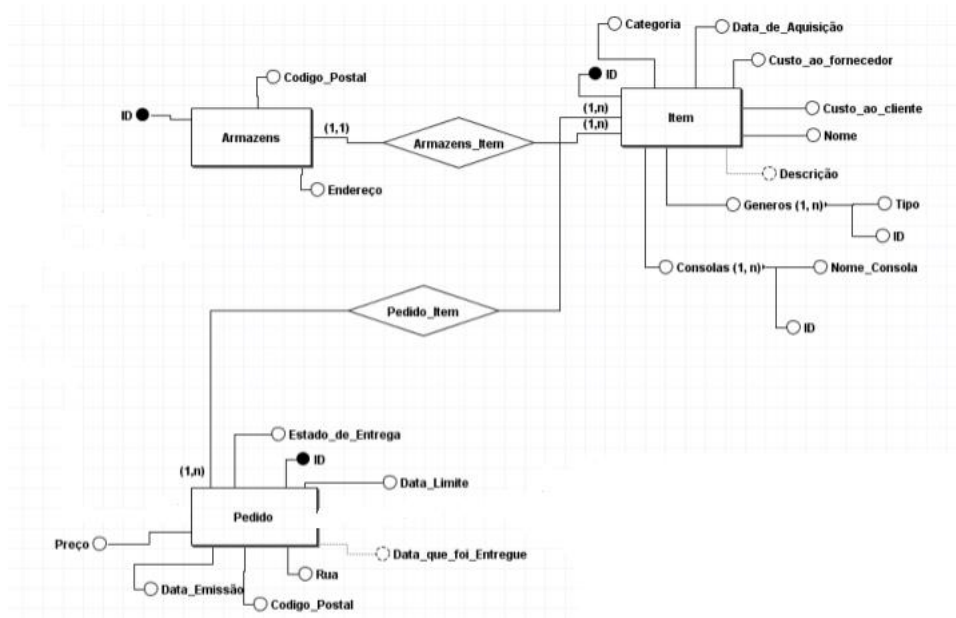
A relação 1(Utilizador) para N (Endereço) resultada na criação de uma chave estrangeira na tabela de cardinalidade N (Endereço).



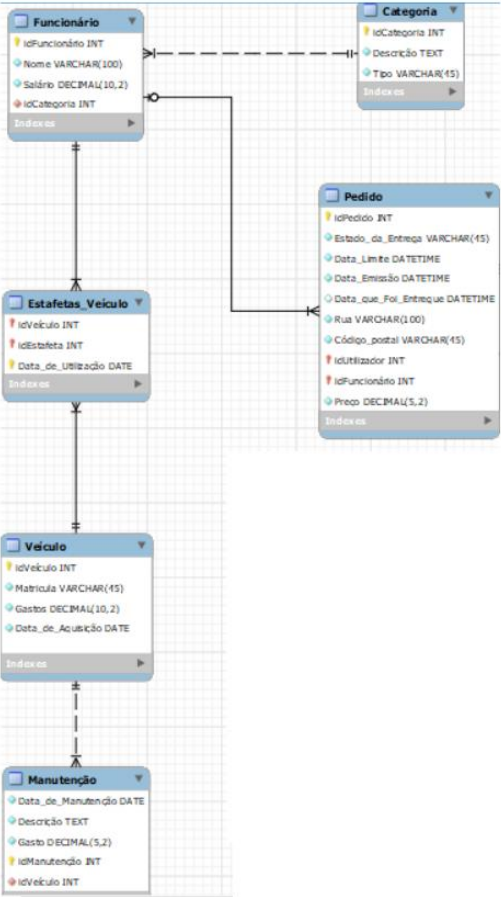
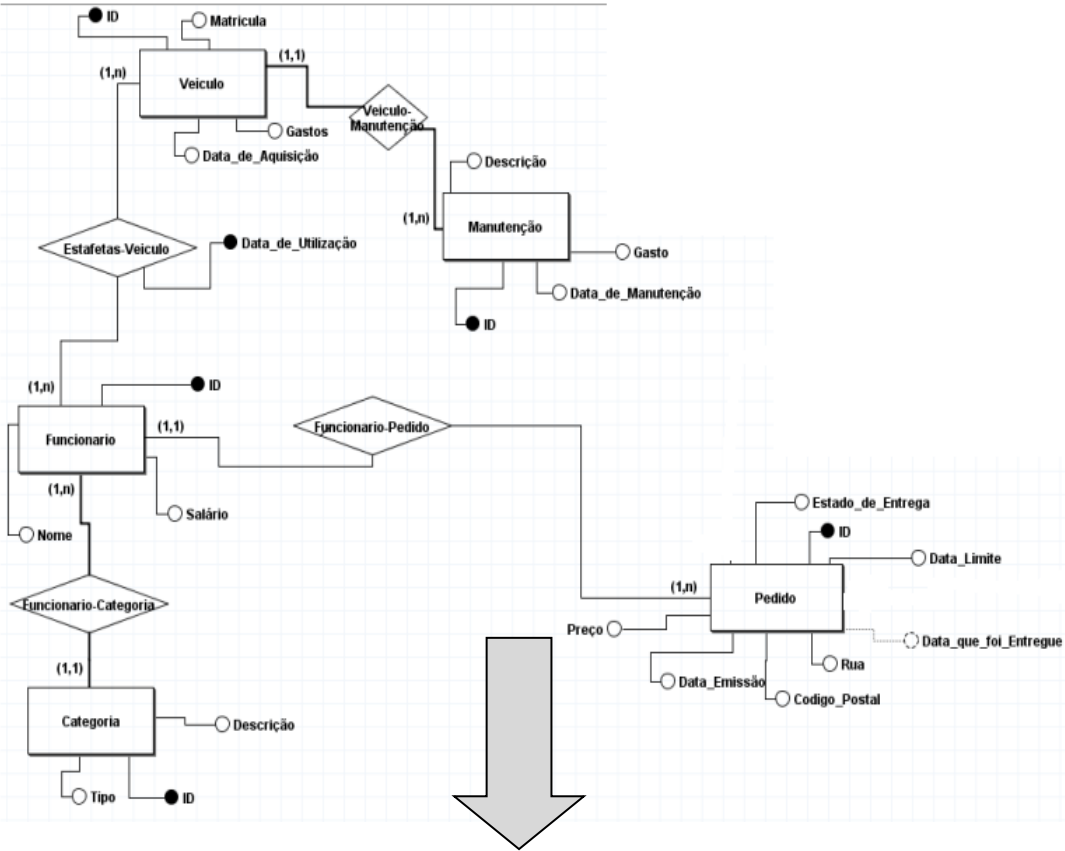
A entidade Pedido dá origem a uma tabela.



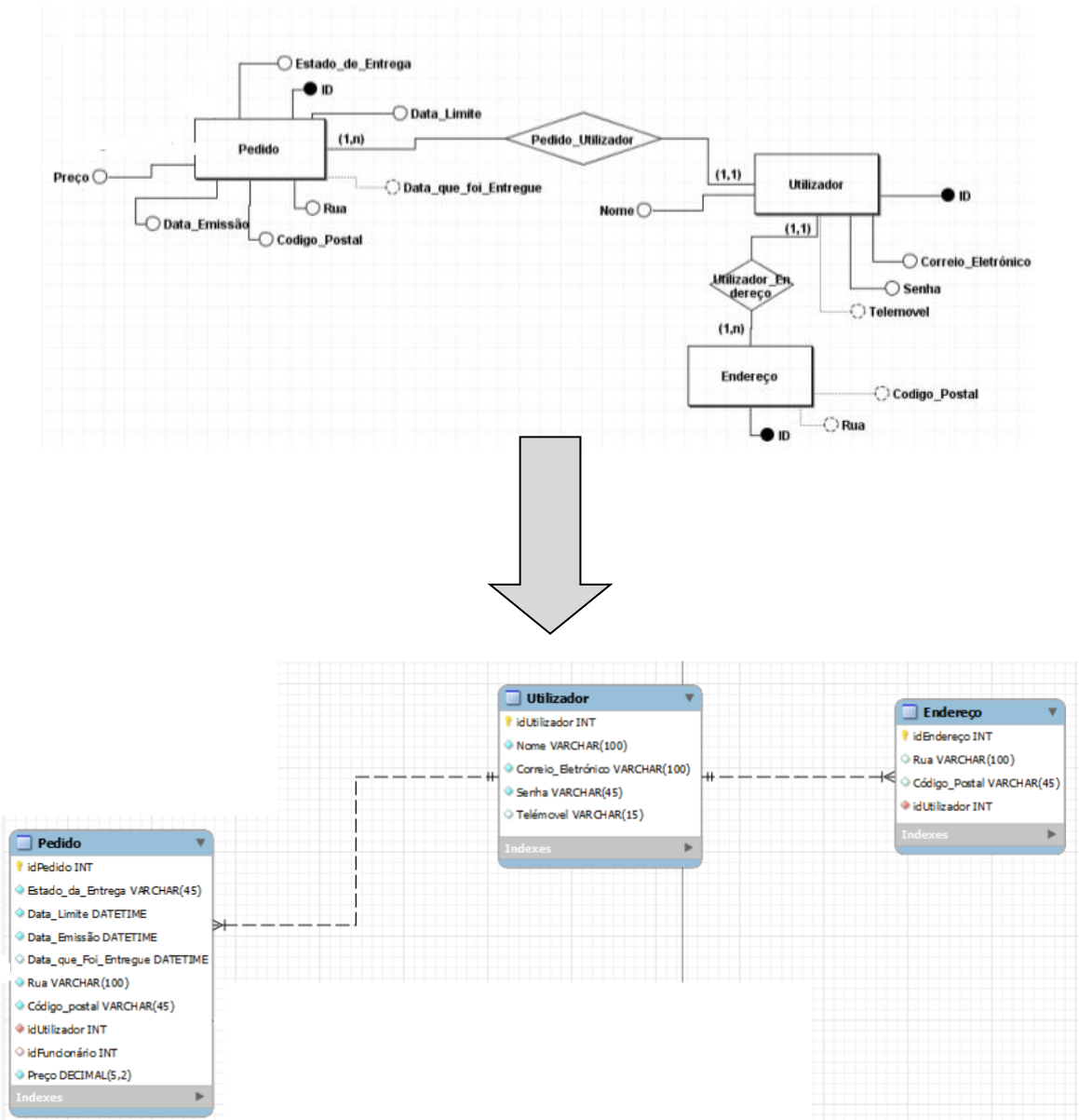
A relação M (Pedido) para N (Itens) gera uma tabela intermédia Pedido_Item e duas relações 1 (Pedido e Item) : N (Pedido_Item) com duas chaves estrangeiras do lado da tabela Pedido_Item cuja cardinalidade é N.



Na relação 1 (Funcionário) para N (pedidos) é gerada uma chave estrangeira do lado da tabela com cardinalidade N (pedidos).

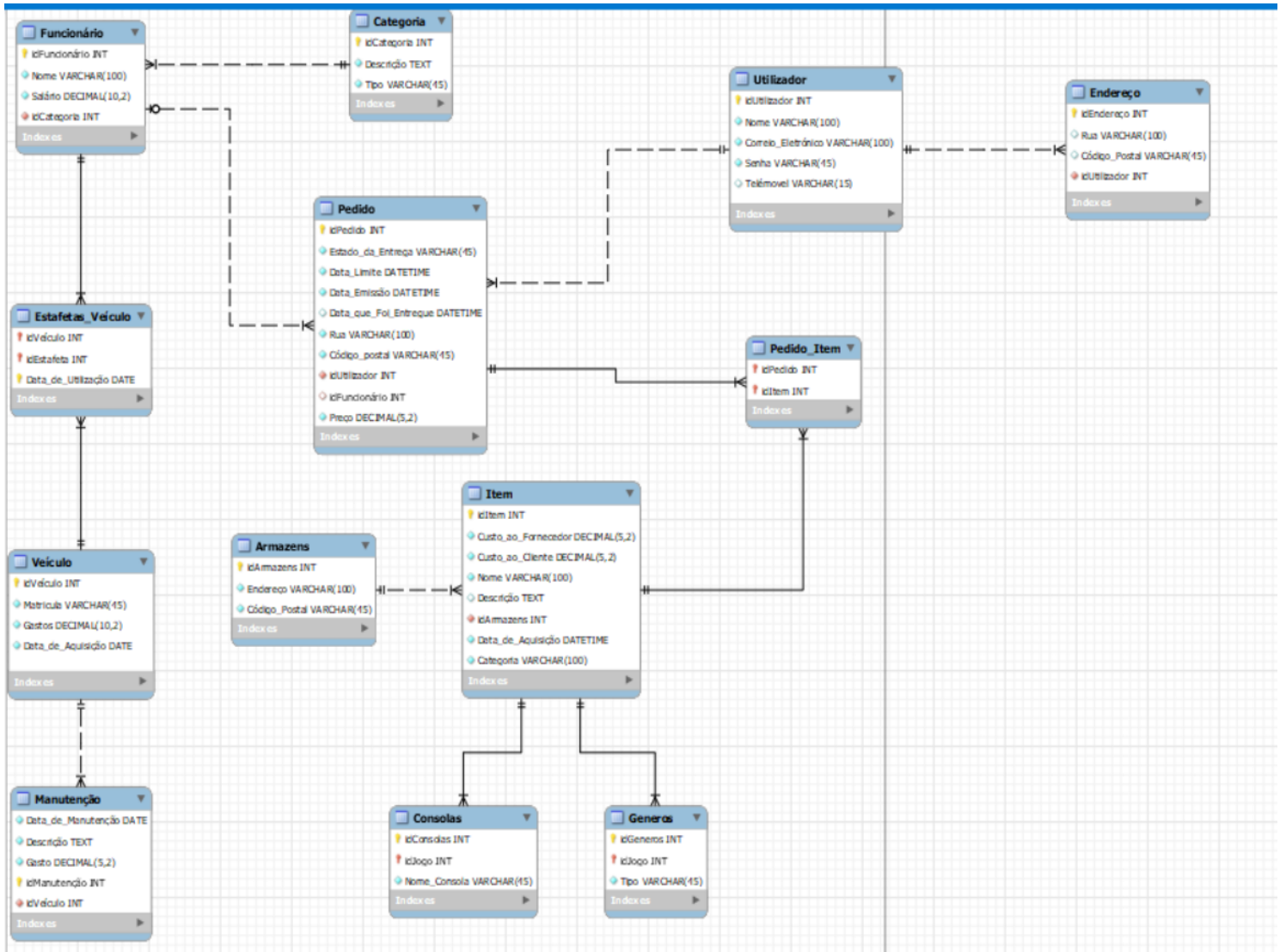


Na relação 1 (Utilizador) para N (pedidos) é gerada uma chave estrangeira do lado da tabela com cardinalidade N (pedidos).



4.2. Desenho do modelo lógico

O modelo lógico desta base de dados é o seguinte.



4.3. Validação do modelo com interrogações do utilizador

Para validar o nosso modelo conceptual constituímos algumas interrogações para garantir que todos os nossos requisitos foram cobertos por este modelo.

Em baixo tem as interrogações bem como quais tabelas consultar e como devemos filtrar essas nossas consultas.

Como foi concluído com as nossas interrogações, o nosso modelo conceptual é capaz de responder a tudo consoante os nossos requisitos.

Interrogações:

Quantos pedidos foram entregues no total?

R: Como temos a entidade pedido conseguimos executar uma query que nos conte quantos pedidos fizemos no total.

Qual é a receita dos jogos mensalmente?

R: Ao consultar as tabelas de Pedidos, Itens e Pedido_Items a resposta é dada.

Qual é o género de jogos mais vendido em um mês?

R: Ao consultar as tabelas de Pedidos, Itens, Pedido_Items e Géneros a resposta é dada.

Quantos God of war foram vendidos no mês 10 do ano de 2021?

R: Teríamos que consultar as tabelas Pedidos, Itens, Pedido_Items, e filtrar por nome = God of War, mês = 10 E ano =2021

Qual a despesa dos itens no mês de janeiro do ano de 2010?

R: Teríamos que consultar a tabela itens, e filtrar por mês = 01 E ano =2010

Qual a despesa dos veículos no mês de fevereiro de 2009?

R: Teríamos que consultar as tabelas Veículos e Manutenção, e filtrar por mês = 02 E ano =2009

Qual a despesa dos funcionários no mês de fevereiro de 2009?

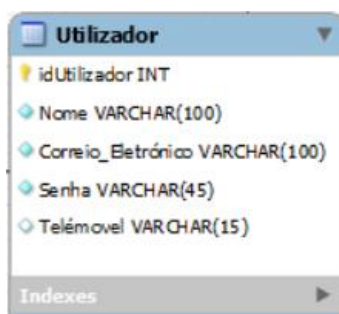
R: Teríamos que consultar as tabelas Funcionários, e filtrar por mês = 02 E ano =2009

5. Implementação Física

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

Para a criação do sistema de dados realizámos no ambiente do MySQL a instrução “create schema if not exists `Yamazon` ; “ e depois escrevemos, baseado nos nossos modelos anteriores, um script SQL a implementação da nossa base de dados.

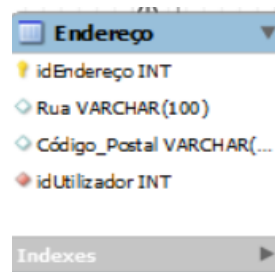
Para a tabela referente à entidade Utilizador apresentada a baixo, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Utilizador`”.



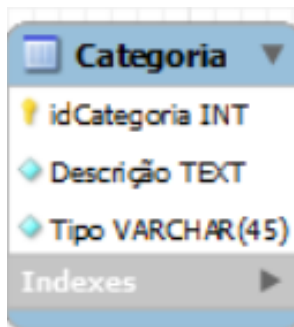
```
create table if not exists `Yamazon`.`Utilizador`(  
  `idUtilizador` int not null auto_increment,  
  `Nome` varchar(100) not null,  
  `Correio_Eletrónico` varchar(100) not null unique,  
  `Senha` varchar(45) not null,  
  `Telémovei` varchar(15) null,  
  primary key (`idUtilizador`)  
);
```

Para a tabela referente à entidade Endereço apresentada ao lado, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Endereço`”.

```
create table if not exists `Yamazon`.`Endereço`(  
  `idUtilizador` int not null,  
  `idEndereço` int not null auto_increment,  
  `Rua` varchar(100) null,  
  `Código_Postal` varchar(45) null,  
  primary key (`idEndereço`),  
  constraint `fk_Utilizador_Endereço` foreign key(`idUtilizador`) references `Yamazon`.`Utilizador`(`idUtilizador`)  
);
```



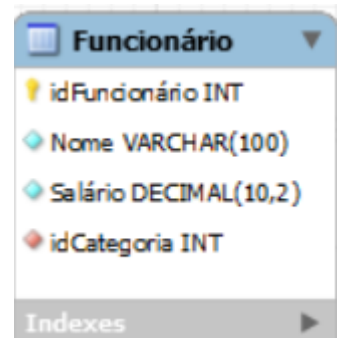
Para a tabela referente à entidade Categoria apresentada a baixo, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Categoria`”.



```
create table if not exists `Yamazon`.`Categoria` (
  `idCategoria` int not null,
  `Descrição` text not null,
  `Tipo` varchar(45) not null unique,
  primary key (`idCategoria`)
);
```

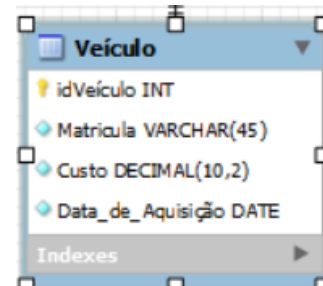
Para a tabela referente à entidade Funcionário apresentada ao lado, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Funcionário`”.

```
create table if not exists `Yamazon`.`Funcionário` (
  `idFuncionário` int not null auto_increment,
  `Nome` varchar(100) not null,
  `Salário` decimal(10,2) not null default 0.00,
  `idCategoria` int not null,
  primary key(`idFuncionário`),
  constraint `fk_Funcionário_Categoria` foreign key(`idCategoria`) references `Yamazon`.`Categoria`(`idCategoria`)
);
```



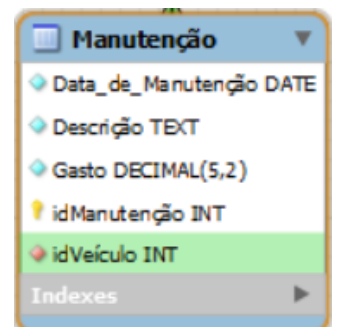
Para a tabela referente à entidade Veículo apresentada a baixo, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Veículo`”.

```
create table if not exists `Yamazon`.`Veículo` (
  `idVeículo` int not null auto_increment,
  `Matricula` varchar(45) not null unique,
  `Gastos` decimal(10,2) not null default 0.00,
  `Data_de_Aquisição` date not null,
  primary key(`idVeículo`)
);
```



Para a tabela referente à entidade Manutenção apresentada ao lado, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Manutenção`”.

```
create table if not exists `Yamazon`.`Manutenção` (
  `idManutenção` int not null auto_increment,
  `idVeículo` int not null,
  `Data_de_Manutenção` date not null,
  `Descrição` text not null,
  `Gasto` decimal(5,2) not null default 0.00,
  primary key(`idManutenção`),
  constraint `Veículo_Manutenção` foreign key(`idVeículo`) references `Yamazon`.`Veículo`(`idVeículo`)
);
```



Para a tabela referente à entidade Estafetas_Veículo apresentada a baixo, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Estafetas_Veículo`”.

```
create table if not exists `Yamazon`.`Estafetas_Veículo` (
`idVeículo` int not null,
`idEstafetas` int not null,
`Data_de_Utilização` date not null,
primary key(`idVeículo`,`idEstafetas`,`Data_de_Utilização`),
constraint `fk_Funcionário_Veículo_Veículo` foreign key(`idVeículo`) references `Yamazon`.`Veículo`(`idVeículo`),
constraint `fk_Estafetas_Veículo_Funcionário` foreign key(`idEstafetas`) references `Yamazon`.`Funcionário`(`idFuncionário`)
);
```

Estafetas_Veículo	
idVeículo	INT
idEstafeta	INT
Data_de_Utilização	DATE
Indexes	

Para a tabela referente à entidade Armazens apresentada a baixo, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Armazens`”

```
create table if not exists `Yamazon`.`Armazens` (
`idArmazens` int not null auto_increment,
`Endereço` varchar(100) not null unique,
`Código_Postal` varchar(45) not null unique,
primary key(`idArmazens`)
);
```

Armazens	
idArmazens	INT
Endereço	VARCHAR(100)
Código_Postal	VARCHAR(45)
Indexes	

Para a tabela referente à entidade Item apresentada a baixo, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Item`”

```
create table if not exists `Yamazon`.`Item` (
`idItem` int not null auto_increment,
`Custo_ao_Fornecedor` decimal(5,2) not null,
`Custo_ao_Cliente` decimal(5,2) not null,
`Nome` varchar(100) not null,
`Categoria` varchar(100) not null,
`Descrição` text null,
`Data_de_Aquisição` datetime not null,
`idArmazens` int not null,
primary key(`idItem`),
constraint `fk_Item_Armazens` foreign key(`idArmazens`) references `Yamazon`.`Armazens`(`idArmazens`)
);
```

Item	
idItem	INT
Custo_ao_Fornecedor	DECIMAL(5,2)
Custo_ao_Cliente	DECIMAL(5,2)
Nome	VARCHAR(100)
Descrição	TEXT
idArmazens	INT
Data_de_Aquisição	DATETIME
Categoria	VARCHAR(100)
Indexes	

Para a tabela referente à entidade Consolas apresentada a baixo, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Consolas`”

```
create table if not exists `Yamazon`.`Consolas` (
`idConsolas` int not null auto_increment,
`idJogo` int not null,
`Nome_Console` varchar(45) not null,
primary key(`idConsolas`),
constraint `fk_Consolas_Jogo` foreign key(`idJogo`) references `Yamazon`.`Item`(`idItem`)
);
```

Consolas	
idConsolas	INT
idJogo	INT
Nome_Console	VARCHAR(45)
Indexes	

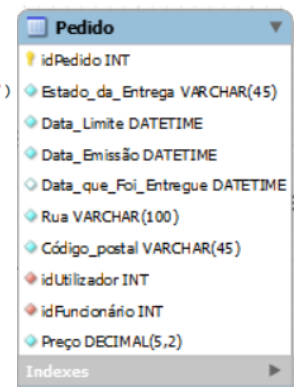
Para a tabela referente à entidade Generos apresentada ao lado, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Generos`”

```
create table if not exists `Yamazon`.`Generos` (
  `idGeneros` int not null auto_increment,
  `idJogo` int not null,
  `Tipo` varchar(45) not null,
  primary key(`idGeneros`),
  constraint `fk_Generos_Jogo` foreign key(`idJogo`) references `Yamazon`.`Item`(`idItem`)
);
```



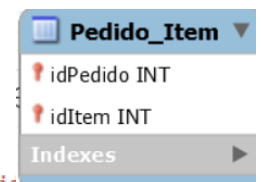
Para a tabela referente à entidade Pedido apresentada a baixo, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Pedido`”

```
create table if not exists `Yamazon`.`Pedido` (
  `idPedido` int not null auto_increment,
  `Estado_da_Entrega` varchar(45) null default 'Em_espera' check(`Estado_da_entrega` in('Em_espera', 'A_caminho', 'Entregue')),
  `Data_Limite` datetime not null,
  `Data_Emissão` datetime not null,
  `Data_que_foi_Entregue` datetime null,
  `Preço` decimal(5,2) not null default 0.00,
  `Rua` varchar(100) not null,
  `Código_Postal` varchar(45) not null,
  `idUtilizador` int not null,
  `idFuncionário` int null,
  primary key(`idPedido`),
  constraint `fk_Pedido_Utilizador` foreign key(`idUtilizador`) references `Yamazon`.`Utilizador`(`idUtilizador`),
  constraint `fk_Pedido_Funcionário` foreign key(`idFuncionário`) references `Yamazon`.`Funcionário`(`idFuncionário`)
);
```



Para a tabela referente à entidade Pedido_Item apresentada ao lado, obtemos a sua implementação inicializada por “create table if not exists `Yamazon`.`Pedido_Item`”

```
create table if not exists `Yamazon`.`Pedido_Item` (
  `idPedido` int not null,
  `idItem` int not null,
  primary key(`idPedido`,`idItem`),
  constraint `fk_Pedido_Item_Pedido` foreign key(`idPedido`) references `Yamazon`.`Pedido`(`idPedido`),
  constraint `fk_Pedido_Item_Item` foreign key(`idItem`) references `Yamazon`.`Item`(`idItem`)
);
```



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

As análises das interrogações no nosso projeto têm base em vários requisitos que foram definidos na segunda fase deste trabalho. É bastante importante pois é necessário verificarmos se as principais ações de exploração da nossa base de dados, traduzidas por procedimentos dos sistemas, podem ser satisfeitas. Relembrando as interrogações colocadas em cima, vamos agora traduzi-las para SQL.

- Quantos pedidos foram entregues no total?

```
DELIMITER $$
CREATE PROCEDURE Total_de_pedidos_entreges()
BEGIN
SELECT COUNT(idPedido) AS entregues, DATE(Data_que_foi_Entregue) AS 'data' FROM pedido
WHERE Estado_da_Entrega = 'Entregue' AND DATE (Data_que_foi_Entregue) = DATE ( Curdate() );

END $$
```

- Qual é a receita dos jogos mensalmente?

```
DELIMITER $$
CREATE PROCEDURE Receitas_Mensais(IN datas date)
BEGIN
SELECT SUM(Custo_ao_Cliente) AS Custo FROM Item AS I, Pedido_Item AS P_I, Pedido AS P
WHERE P_I.idPedido = P.idPedido AND I.idItem = P_I.idItem AND
MONTH(P.Data_Emissão) = MONTH (datas) AND YEAR(P.Data_Emissão) = YEAR (datas);

END $$
```

- Qual é o gênero de jogos mais vendido num mês?

```
DELIMITER $$
CREATE PROCEDURE jogos_por_genero(datas DATE)
) BEGIN
SELECT COUNT(G.idGeneros) AS "Quantidade", G.Tipo AS "gênero"
FROM Pedido P, Pedido_Item P_I, Item I, Generos G
where P_I.idPedido = P.idPedido AND P_I.idItem = I.idItem AND I.Categoria = "Jogo"
AND G.idJogo = I.idItem AND MONTH(P.Data_Emissão) = MONTH(datas) AND YEAR(P.Data_Emissão) = YEAR(datas)
GROUP BY G.Tipo
ORDER BY COUNT(G.idGeneros) DESC
Limit 1;

END $$
```

- Quantos God of war foram vendidos no mês 10 do ano de 2021?

```
DELIMITER $$
CREATE PROCEDURE itemscount (in D date)
BEGIN
Select I.Nome, count(I.idItem) as Quantidade
FROM Pedido as P, Item as I, Pedido_Item as P_I
Where P_I.idItem = I.idItem AND P_I.idPedido = P.idPedido AND Month(D) = Month(P.Data_Emissão)
AND Year(D) = Year(P.Data_Emissão) AND I.name = "God of War"
Group by I.Nome
Order by count(I.idItem) DESC;

END $$
```

- Qual a despesa com itens no mês de janeiro do ano de 2010?

```
DELIMITER $$
CREATE PROCEDURE Despesa_items_mensal()
) BEGIN
SELECT SUM(Custo_ao_Fornecedor) AS Custo FROM item
WHERE MONTH(Data_de_Aquisição) = MONTH ('01') AND YEAR(Data_de_Aquisição) = YEAR ('2010');

END $$
```

- Qual a despesa dos veículos no mês de fevereiro de 2009?

```
DELIMITER $$
CREATE PROCEDURE Despesa_Mensal_Veiculos(dt date)
BEGIN
SELECT sum(G.Gasto) FROM (
SELECT M.Gasto FROM Manutenção M, Veículo V WHERE MONTH(dt) = MONTH(M.Data_de_Manutenção)
AND YEAR(dt) = YEAR(M.Data_de_Manutenção) AND M.idVeículo = V.idVeículo
UNION
SELECT Gastos FROM Veículo WHERE MONTH(dt) = MONTH(Data_de_Aquisição) AND YEAR(dt) = YEAR(Data_de_Aquisição) ) AS G;

END $$

CALL Despesa Mensal Veiculos('2009-02-28');
```


- Qual a despesa dos funcionários no mês de fevereiro de 2009?

```
CREATE PROCEDURE Despesa_funcionários_mensal ()

BEGIN
    SELECT SUM(Salário) FROM funcionários;
END $$
```

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

Na nossa base de dados, não implementamos qualquer vista.

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

O Cálculo do espaço da nossa base de dados de dados foi feito através de algumas informações que encontramos na internet, quando pesquisamos sobre o tópico, sendo que os sites estarão devidamente referenciados na bibliografia. Não temos a certeza de que tudo esteja 100% correto principalmente por causa do Dominio *Text*. Este tem um tamanho máximo de 2GB, mas como é obvio nenhum registo na nossa base de dados justifica esse tamanho. Classificamos o espaço da base de dados inicial, como a definição das tabelas, sem qualquer povoamento. O primeiro calculamos o espaço de cada tabela e depois somamos todos.

Usamos os seguintes valores para calculo:

Cabeçalho- 4 bytes

Int – 4 bytes

Varchar[x] – x bytes

Decimal – Variável, depende dos valores, seguimos regras no site javapoint.com/mysql-decimal

Date/Datetime – 8 bytes

Text – Variável

Depois acrescentamos 1 ou 2 bytes extra dependendo do domínio de campo(2 para cada campo com dados variáveis, por exemplo VARCHAR) e bytes por null blocks, que depende da quantidade de campos numa tabela. Baseado nisto o espaço da base de dados inicial é dado por (valores em bytes):

Utilizador = 277
Endereço = 164
Categoria = 109
Funcionário = 123
Veículo = 72
Manutenção = 107
Estafetas_Veiculo = 24
Armazens = 154
Item = 338
Consolas = 62
Generos = 58
Pedido = 248
Pedido_Item = 11
TOTAL == 1747

Realizamos depois uma estimativa de crescimento, da nossa base de dados. Para fazer isto, nos baseamo-nos mais num ficheiro de teste que evoluímos de modo a ter uma aproximação do que nós queremos. Tomamos por base o ponto de viabilidade que espera que em 12 meses, existam 5000 utilizadores ativos, ou seja que façam pelo menos uma compra por mês. Criamos uma estimativa com esses dados, sendo que também foram adicionados 120 itens, centenas de inserções em Pedido_Item, 6 generos e 6 consolas e algumas dezenas de funcionario. O valor que obtemos foi 264kb. Como é obvio isto é só uma estimativa, que provavelmente está um bocado abaixo da realidade, sendo que o número de funcionários poderia ser mais alto e as manutenções são extremamente difíceis de prever, pois não sabemos quantas vezes cada veiculo terá de pôr gasolina. Ao fim de uns anos estimamos que a base de dados esteja ainda maior.

5.5. Plano de segurança e recuperação de dados

A plataforma envia logs a três discos externos quando uma informação é processada pelo sistema. Os logs ficam até 48h do momento que foram enviados. Ao final de todos os dias, é criada uma cópia de segurança que é armazenada nos 3 discos externos. Dessa maneira, qualquer problema que ocorra com os dados ou a base de dados em si durante qualquer período do dia, podemos recuperar um estado anterior através da cópia de segurança e completar os dados que faltam através dos logs do sistema.

6. Conclusão

Concluindo, este foi possivelmente um dos trabalhos mais desafiantes que nos foi proposto desde que ingressamos neste curso. Não só pelo grau de complexidade e da quantidade de matéria que tivemos de incorporar no projeto, como também tivemos de o inicializar desde o início do ano. Apesar disto, tivemos de voltar atrás várias vezes para corrigir e melhorar certas coisas que tínhamos feito, pois adquiríamos mais informação e fomos aprendendo mais ao longo da UC sendo que notamos que algumas mudanças eram necessárias para o nosso sistema de base de dados ficasse o melhor possível. Achamos que por estes desafios que enfrentamos no desenvolver do trabalho nos expandiram o conhecimento de sistemas de bases de dados e deram-nos uma visão um bocado diferente da que tínhamos tido até agora.

Referências Bibliográficas

[https://learn.microsoft.com/pt-br/previous-versions/cc668499\(v=technet.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/pt-br/previous-versions/cc668499(v=technet.10)?redirectedfrom=MSDN)

<https://www.hardware.com.br/comunidade/banco-calcular/1263184/>

<https://www.javatpoint.com/mysql-decimal>

<https://learnsql.com/blog/ms-sql-server-data-types/>