



**Universidade do Minho**  
Licenciatura em Ciências da Computação

## **Unidade Curricular de Bases de Dados**

Ano Lectivo de 2021/2022

### **Stand de Automóveis**

**Bruna Araújo a84408**

**João Henriques a81705**

**José Mendes a81809**

**Marlon Ferreira a81735**

Dezembro, 2021

# **BD**

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Resumo

No âmbito da presente disciplina, e como método de aprendizagem e evolução enquanto alunos do curso de LCC, é nos inferido que elaborem um sistema de base de dados real e concreto, e com um grau de complexidade significativo. É do senso comum, entre a comunidade do nosso curso, que a disciplina de base de dados é e será uma ferramenta essencial para o futuro de cada aluno durante toda a sua carreira enquanto programador.

O desafio lançado pelo docente é um desafio que exige que o aluno explore e ganhe conhecimento acerca do universo de base de dados, em que após a concretização do mesmo, o aluno apresente competências necessárias para poder enfrentar o mercado de trabalho/a realidade em si e a exigências que futuramente se apresentem.

No contexto atual, o desafio é elaborar uma base de dados que incorpore o tema de “Comercialização de Bens e Serviços”, tal tema abrange uma enorme opção de escolha. O nosso grupo decidiu elaborar um sistema de base de dados de um Stand de automóveis, tema tal que já nos permite navegar pelas várias etapas de construção e desenvolvimento do sistema, abrigando todas as construções de esquemas conceptuais, lógicos e físicos, o levantamento e análise de requisitos necessários, e nos permite também trabalhar com os conceitos de Entidade, relacionamentos, palavras chave entre outros que posteriormente serão abordados.

Por fim, iremos ter a oportunidade de trabalhar com programas tais como o *workbench*, *brmodelo*, e com a linguagem *mysql* que é se não a mais importante ferramenta e motor de todas as base de dados.

**Área de Aplicação:** Desenho e arquitectura de Sistemas de Bases de Dados.

**Palavras-Chave:** Base de Dados, Modelo Conceptual, Modelo Lógico, Modelo Físico, *MySQL*, *MySQL Workbench*, *brModelo*, *Query*, Vistas e Índices.

# Índice

Resumo.....	i
1. Definição do Sistema.....	1
1.1.Contexto de aplicação do sistema.....	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.....	2
2.Levantamento e Análises de Requisitos.....	4
2.1 Método de levantamento e de análise de requisitos adotado.....	4
2.2. Organização dos requisitos levantados.....	4
2.2.1.Requisitos de descrição.....	5
2.2.2. Requisitos de exploração.....	6
2.2.3.Requisitos de controlo.....	7
2.3 Análise e validação geral dos requisitos.....	7
3. Modelação Conceptual .....	8
3.1 Apresentação da abordagem de modelação realizada.....	8
3.2 Identificação e caracterização das entidades.....	9
3.3 Identificação e caracterização dos relacionamentos.....	10
3.4 Identificação e caracterização da associação dos atributos com as entidades e relacionamentos.....	11
3.5 Apresentação e explicação do diagrama ER.....	14
4. Modelação Lógica.....	15
4.1 Construção e validação do modelo de dados lógico.....	15
4.2 Desenho do modelo lógico.....	18

4.3 Validação do modelo com interrogações do utilizador.....	19
5. Implementação Física.....	20
5.1 Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL.....	20
5.2 Tradução das interrogações do utilizador para SQL (alguns exemplos).....	24
5.3 Definição e caracterização das vistas de utilização em SQL (alguns exemplos).....	27
5.4 Cálculo do espaço da bases de dados (inicial e taxa de crescimento anual).....	28
6. Conclusões e Trabalho Futuro.....	34
7. Referências Bibliográficas (estilo APA).....	35

# Índice de Figuras

Figura 1-Modelo Conceptual	14
Figura 2- Modelo Lógico	18
Figura 3-Criação da Tabela Morada	18
Figura 4-- Criação da Tabela Contacto	20
Figura 5-Criação da Tabela Funcionário	20
Figura 6-Criação da Tabela Cliente	20
Figura 7- Criação da Tabela Seguro	26
Figura 8 - Criação da Tabela Veículo	21
Figura 9- Criação da Tabela Aluguer	21
Figura 10- Criação da Tabela Fatura	22
Figura 11- Query Funcionários	23
Figura 12-Query Veiculos Combustível	23
Figura 13-Query Veículos Disponíveis para Compra	24
Figura 14-Query Funcionário Com Mais Vendas	24
Figura 15- Query Mail	25
Figura 16-Query Localidade e Rua	25
Figura 17- View Veículos	26
Figura 18- View Facturas	26

## Índice de Tabelas

Tabela 1- Caracterização das Entidades	9
Tabela 2-- Tabela de atributos da Entidade Cliente	11
Tabela 3-Tabela de atributos da Entidade Veículo	12
Tabela 4-Tabela de Atributos da Entidade Aluguer	13
Tabela 5-Tabela de atributos da Entidade Fatura	13
Tabela 6-Tabela de atributos da Entidade Funcionario	14
Tabela 7-Tabela da Entidade Cliente	15
Tabela 8-Tabela da Entidade Aluguer	15
Tabela 9-Tabela da Entidade Fatura	15
Tabela 10-Tabela da Entidade Funcionário	16
Tabela 11-Tabela do atributo composto morada	16
Tabela 12-Tabela da Entidade Veículo	16
Tabela 13-Tabela do atributo composto contacto	16
Tabela 14-Tabela do atributo composto seguro	17
Tabela 15 -Espaço máximo dos atributos de Cliente	28
Tabela 16-Espaço máximo dos atributos de Fatura	29
Tabela 17-Espaço máximo dos atributos de Veiculo	30
Tabela 18-- Espaço máximo dos atributos de Funcionario	30
Tabela 19-Espaço máximo dos atributos de Aluguer	31
Tabela 20-Espaço máximo do atributo composto morada	31
Tabela 21-Espaço máximo do atributo composto contacto	32
Tabela 22-Espaço máximo do atributo composto seguro	32



# **1. Definição do Sistema**

## **1.1. Contexto de aplicação do sistema/ 1.2. Motivação e Objetivos do Trabalho**

Quer queiramos quer não, grande parte de nos enfrenta se com a necessidade de se movimentar de local para local, quer seja em trabalho, lazer, saúde entre outros, e um dos principais meios de transporte utilizados na atualidade é o carro. Nessa linha de pensamento surgiu no nosso grupo a ideia de perceber como funciona toda a logística do local onde nos é fornecido tal objeto precioso, o carro.

Dito isto, e como é de prever, toda uma organização é necessária para uma demanda enorme de veículos que satisfaça as necessidades dos cidadãos, tais como um carro mais veloz para os viciados em adrenalina, um carro mais confortável para quem quer desfrutar de uma viagem tranquila e cómoda, um carro mais acessível para quem tem menos meios financeiros, existe mil e uma razoes para a compra de um carro. Convém acrescentar também que se tem observado um crescimento elevado da procura de automóveis para aluguer, uma nova forma de negócio que esta a explodir tanto a nível nacional como internacional, que ajuda tanto os clientes como os “patrões”.

### **1.3. Análise da Viabilidade do Processo**

Como referido no ponto anterior, a quantidade de automóveis existentes no mundo(matriculados) é astronómica, dados referentes ao ano de 2010, aponta para o número de 1 bilião de automóveis. Tais números exigem toda uma logística de compra e venda e destruição e produção de automóveis bastante dinâmica, local que sofre diretamente com todo este universo são os stand de automóveis, por isso parece nos razoável dizer que é viável a criação de uma base de dados para suportar toda esta diversidade.

### **1.4. Recursos e Equipa de Trabalho**

A equipa pertencente a este trabalho é constituída por 4 alunos. Os recursos utilizados para a elaboração deste projeto foi essencialmente uma busca e exploração por parte de todos os membros de conteúdo digital.

Exploramos diversas lojas de stand de automóveis online para entender a logística da mesma, também explorarmos conteúdo de base de dados disponível na internet, para além de toda a matéria lecionada e conteúdo disponível na BB, para nos auxiliar em determinadas questões , e também o conhecimento geral de cada um de nós que, juntando tudo isto, culmina numa ampla base de informação para a realização do projeto.

### **1.5. Plano de Execução do Projeto**

O nosso grupo elaborou um plano de execução do projeto estruturado e, de forma gradual, completar e organizar todos as etapas necessárias a realização do mesmo.

O primeiro passo foi a recolha de informação necessária, por parte de cada membro, tanto individualmente como em conjunto, durante as aulas da disciplina como fora dela.

Começamos por elaborar em simultâneo o esquema conceptual e o levantamento e aplicação de requisitos adequadas ao esquema. Fazendo isto

em conjunto tornou o projeto mais sólido e estável no que toca a todas as funcionalidades do mesmo, e de futuras exigências da base de dados. Neste primeiro passo utilizamos o TerraER e o brmodelo para nos ajudar.

Após isso partimos para a elaboração do esquema lógico, utilizamos a ferramenta que é disponibilizada pelo brmodelo de transformar o esquema conceptual num esquema logic, e feito isto, criamos um esquema logic no workbench adaptado a ele. Verificamos todos os atributos e relacionamentos existentes, verificamos também todas as chaves primárias e estrangeiras existentes nas tabelas das entidades.

## **2. Levantamento e Análise de Requisitos**

### **2.1. Método de Levantamento e de Análise de Requisitos Adotado**

O método de levantamento de requisitos foi efetuado através de dialogo entre o grupo, após analisarmos o modelo conceptual, e na transformação do mesmo para o esquema lógico fomos colocando questões e verificando quais as necessidades que cada entidade tem em se relacionar com as outras, e quais as possibilidades e funcionalidades das mesmas.

Criamos um processo de início ao fim de um cliente a entrar e sair de um stand, efetuando ou uma compra ou aluguer de um veiculo, e durante todo esse processo analisamos quais os requisitos necessários tanto por parte do cliente como por parte do administrador do stand em responder as necessidades de ambos.

### **2.2. Organização dos requisitos levantados**

Após juntar todos os requisitos propostos, retiramos alguns que não fossem relevantes, e modificamos outros para se adaptarem melhor ao nosso projeto. Feito isto ficamos preparados para os categorizar da forma como nos foi proposto, Requisitos de Descrição, Exploração e de Controlo, como vamos ver na secção seguinte.

### **2.2.1. Requisitos de Descrição**

1. Cada Cliente é identificado com o seu identificador único(numero), com o seu nome, o seu contacto (telemóvel, E-Mail), a sua morada (Código Postal, Rua, Concelho, Localidade) e data de nascimento, todos estes campos são de preenchimento obrigatório.
2. Existem dois tipos de transações que um cliente pode realizar, ou compra ou aluguer de um veículo.
3. Tanto a compra como o aluguer de um veiculo é um ato único e isolado.
4. Qualquer Veiculo existente é caracterizado pela sua Matricula (identificador único), o seu tipo de combustível (diesel, gasolina,GPL, híbrido, elétrico), a sua Marca, o seu modelo, o seu preço para compra que é fixo, o seu preço de aluguer(diário), e o seu estado(em que representará se o veículo está disponível ou não)
5. Existem carros específicos para Venda e carros específicos para Aluguer .
- 6.Cada Aluguer é caracterizado pelo seu identificador único(numero), o seu seguro correspondente (standart e premium), a duração do aluguer pretendido, a data inicial do aluguer, e a matricula(fk) do veiculo correspondente.
7. Quer o seguro standart quer o premium podem ser adquiridos por qualquer cliente.
8. Ambos os seguros têm acréscimos monetários ao preço final do aluguer.
9. Toda a transação efetuada tem uma fatura associada.
10. Cada Fatura tem o seu identificador único(número), a data da compra do veiculo, a data final do aluguer, a matricula do veiculo a ser transacionado, o

funcionário responsável pela transação, o tipo de transação (aluguer ou compra) e o cliente em questão .

11. A data na fatura representa tanto a data de compra ou a data final do aluguer, consoante a fatura seja de compra ou aluguer respectivamente.

12. O cliente só pode efetuar uma e só uma compra de um carro de cada vez.

13. O funcionário tem a informação do seu identificador único (numero), o seu nome, e a quantidade de vendas que já realizou no stand.

14. O cliente que pretenda Alugar um veiculo tem um preço diária de algures que pode ser estendido ate quando desejar.

### **2.2.2. Requisitos de Exploração**

1. Visualizar Todos os veículos do stand.
2. Visualizar quais os veículos alugados no momento presente.
3. Visualizar Todos os veículos comprados.
4. Visualizar todos os alugueres efetuados.
5. Visualizar todas as características de um dado veiculo.
6. Visualizar toda a informação de um dado cliente.
7. Visualizar todas a informação de um dado aluguer.
8. Visualizar todas as transacções/fatura efetuadas no stand(compra e aluguer).
9. Visualizar todos os clientes existentes no stand.
10. Visualizar todos os funcionários do stand.
11. Visualizar todas as vendas efetuadas por dado funcionário.
12. Visualizar a quantidade e qual o seguro efetuado.

13. Visualizar veículos para compra.

14. Visualizar veículos para aluguer.

### **2.2.3. Requisitos de Controlo**

1. O Funcionário emite todas as futuras existentes
2. O funcionário é responsável pela remoção e introdução de veículos.
3. O Cliente pode aceder aos veículos existentes.
4. Só o funcionário pode aceder aos alugueres realizados.
5. O cliente tem acesso á sua própria fatura.
6. O funcionário é responsável pelo manegement do aluguer(mudar o seu estado).

### **2.3. Análise e Validação geral dos Requisitos**

A entidade fatura é a entidade com mais peso na globalidade do esquema, pois ela contém grande parte da informação que é mencionada nas outras entidades, mais especificamente ela contém os identificadores do cliente, do funcionário e do veiculo em questão.

Através da Entidade fatura é nos possível aceder a todas as transações efetuadas no Stand, bem como aceder a todos informações das mesmas. Com isto, torna se mais fácil para o administrador do stand aceder a qualquer informação que pretenda.

A atribuição dos atributos feita a todas as Entidades parece nos pertinente e necessária , como por exemplo a atribuição de um seguro ao veiculo na entidade aluguer, que vai ter um efeito monetário no valor final da transação, estas atribuições permitem guardar informações importantes do processo de aluguer.

A entidade veiculo também ela permite guardar informações relevantes tais como o seu estado atual no stand, se se encontra alugado ou não, o que

permite saber se se pode efectuar aluguer do mesmo, também os seus respectivos preços, quer seja para venda ou para aluguer.

Por fim na entidade Funcionário também nos é possível guardar informações interessantes, tais como o numero de vendas que cada um tem, e a partir daí fazer um gerenciamento dos mesmos por parte do administrador.

### **3.Modelação Conceptual**

#### **3.1 Apresentação da abordagem de modelação realizada**

A abordagem utilizada pelo nosso grupo foi, como referido anteriormente , fazer a modelação conceptual a par com o levantamento de requisitos. Esta abordagem permitiu existir flexibilidade tanto na criação de requisitos como na elaboração do modelo conceptual. Consoante surgissem novos requisitos, novas entidades e novos atributos eram atribuídos ao modelo. Após a conclusão dos requisitos já tínhamos o modelo conceptual bastante preenchido, no entanto fomos alterando alguns detalhes caso o projeto assim o exigisse.



### 3.2. Identificação e Caracterização das Entidades

Concluído o modelo conceptual e os requisitos necessários, obtivemos as seguintes Entidades:

ENTIDADE	DESCRIÇÃO
CLIENTE	Termo que representa e contem toda a informação da pessoa que realiza uma transação de um veículo
FATURA	Termo que exprime e contem informação de qualquer transação realizada no Stand
FUNCIONARIO	Termo que representa a pessoa que opera em todas as ações do Stand e contém informação adicional
VEICULO	Termo que representa e contém toda a informação de um veículo existente no Stand
ALUGUER	Termo que exprime e contém informação de qualquer aluguer realizado no Stand

Tabela 1- Caracterização das Entidades

### 3.3 Identificação e caracterização dos relacionamentos

Durante a realização do modelo conceptual foi necessário a atribuição de alguns relacionamentos para que as entidades se possam relacionar e trocarem informação entre elas, para que o modelo seja coeso e organizado.

Posto isto temos os seguintes relacionamentos:

**-> Cliente efetua Aluguer (N,1)**

Um cliente pode Realizar N alugueres;

Um aluguer é efetuado apenas por 1 Cliente;

**->Aluguer tem Veiculo (1,N)**

Um Veículo pode ser alugado N vezes;

Um Aluguer tem apenas 1 Veículo;

**->Cliente Compra Veiculo (1,1)**

Um Cliente pode comprar apenas 1 Veículo;

Um Veículo é comprado apenas por 1 Cliente;

**->Funcionario emite Fatura (N,1)**

Um Funcionário pode emitir N Facturas;

Uma Factura é emitida apenas por 1 Funcionário;

**->Aluguer possui Fatura (1,1)**

Um Aluguer possui apenas 1 Fatura;

Uma Fatura possui apenas 1 Aluguer;

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

Previsivelmente a atribuição de atributos a cada entidade é um tarefa obrigatória para a construção deste projeto. Durante os primeiros passos da construção do modelo conceptual todos os atributos foram sendo escolhidos, no entanto, em um estado mais avançado foi necessário um retificação dos mesmos, o que acaba nos seguintes atributos por cada Entidade:

#### Entidade: Cliente

Atributos	Descrição	NULL/NOTNULL	DataType
id_Cliente	Chave Primária	NN	INT
Nome	Nome do Cliente	NN	VARCHAR(45)
Data de Nascimento	Data de Nascimento do Cliente	NN	DATE
Morada	Código Postal, Rua Localidade e Concelho do Cliente	NN	VARCHAR(45)
Contacto	Telemóvel e E-Mail do Cliente	NN;N(opcional)	VARCHAR(45)

Tabela 2-- Tabela de atributos da Entidade Cliente

**Entidade: Veículo**

<b>Atributos</b>	<b>Descrição</b>	<b>NULL/NOTNULL</b>	<b>DataType</b>
matricula	Chave Primária	NN	VARCHAR(45)
Combustível	Tipo de Combustível do Veiculo	NN	VARCHAR(45)
Modelo	Modelo do veículo	NN	VARCHAR(45)
Marca	Marca do Veiculo	NN	VARCHAR(45)
Preço Venda	Preço de Venda do Veiculo(único)	N(opcional)	DECIMAL(10,2)
Preço Aluguer	Preço diário do aluguer do veiculo	N(opcional)	DECIMAL(10,2)
Estado	Disponibilidade do Veiculo	NN	VARCHAR(45)

Tabela 3-Tabela de atributos da Entidade Veículo

**Entidade : Aluguer**

<b>Atributos</b>	<b>Descrição</b>	<b>NULL/NOTNULL</b>	<b>DataType</b>
id_Aluguer	Chave primária	NN	INT
Seguro	Seguro do Veiculo pretendido: Tipo : Standart ou Premium Preço: preço respectivo	NN	VARCHAR(45);DECIMAL
Duração	Duração do Aluguer	NN	INT
matricula	Matricula do veiculo alugado;Chave Estrangeira	NN	VARCHAR(45)
Data Inicial	Data do inicio do aluguer	NN	DATE
id_Cliente	Cliente responsável pelo Aluguer; Chave Estrangeira	NN	INT

Tabela 4-Tabela de Atributos da Entidade Aluguer

**Entidade : Fatura**

<b>Atributos</b>	<b>Descrição</b>	<b>NULL/NOTNULL</b>	<b>DataType</b>
id_Fatura	Chave Primária	NN	INT
Garantia	Garantia do Veículo(Anos)	NN	INT
matricula	Matrícula do Veículo Transacionado; Chave Estrangeira	NN	VARCHAR(45)
id_Cliente	Cliente que efetuou compra ou venda;Chave Estrangeira	NN	INT
id_Funcionario	Funcionário responsável pela transação; Chave Estrangeira	NN	INT
Preço Final	Preço Final da transação	NN	DECIMAL
Tipo	Tipo de transação Compra/Aluguer; Chave Estrangeira se for Aluguer	N(opcional)	INT
Data	Data da Compra/Data final do aluguer	NN	DATE

Tabela 5-Tabela de atributos da Entidade Fatura

**Entidade : Funcionario**

<b>Atributos</b>	<b>Descrição</b>	<b>NULL/NOTNULL</b>	<b>DataType</b>
id_Funcionario	Chave Primária	NN	INT
Nome	Nome do Funcionario do Stand	NN	VARCHAR(45)

Vendas	Número de vendas/alugueres do Funcionario	N(opcional)	INT
--------	---	-------------	-----

Tabela 6-Tabela de atributos da Entidade Funcionario

### 3.5 Apresentação e explicação do diagrama ER

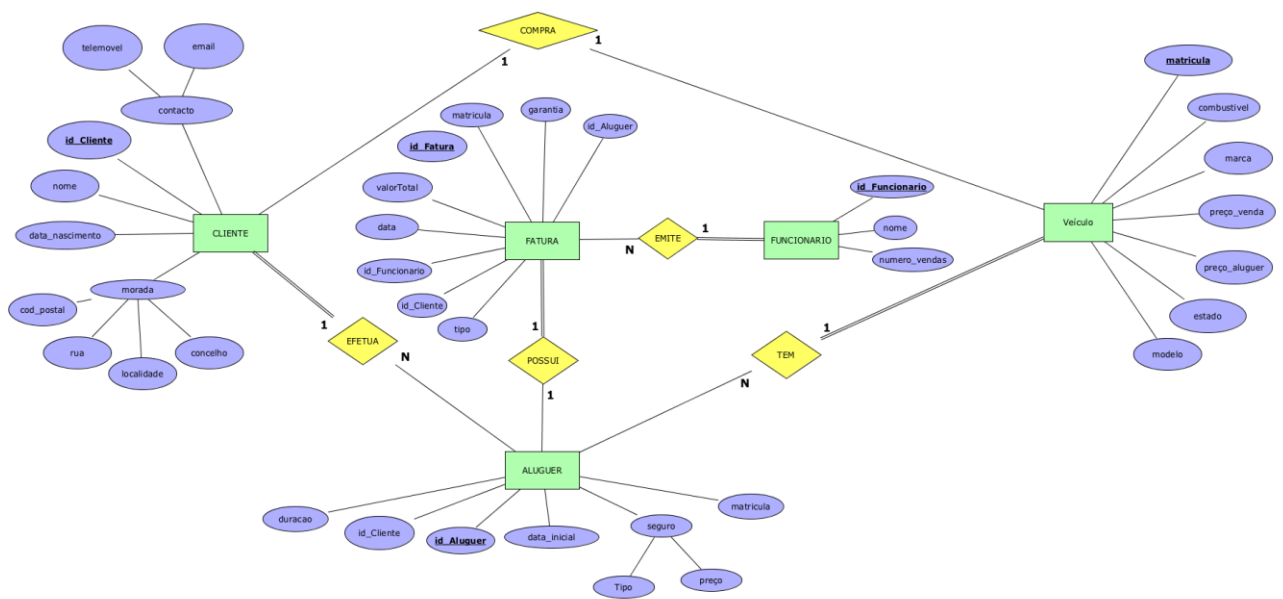


Figura 1-Modelo Conceptual

## 4. Modelação Lógica

### 4.1 Construção e validação do modelo de dados lógico

A Entidade Cliente produz a seguinte relação:

Cliente -> atributos -> (id\_Cliente, nome, morada, contacto, data\_nascimento)

Chave Primária -> id\_Cliente

Tabela 7-Tabela da Entidade Cliente

A Entidade Aluguer produz a seguinte relação:

Aluguer -> atributos -> (id\_Aluguer, seguro, duracao, matricula, id\_Cliente, data\_inicial)

Chave Primária -> id\_Aluguer

Chave Estrangeira -> matricula; seguro; id\_Cliente

Tabela 8-Tabela da Entidade Aluguer

A Entidade Fatura produz a seguinte relação:

Fatura -> atributos -> (id\_Fatura, valorTotal, garantia, data, idFuncionario\_emissor, id\_Cliente, matricula, tipo, id\_Aluguer)

Chave Primária -> id\_Fatura

Chave Estrangeira -> (idFuncionario\_emissor, id\_Cliente, matricula, id\_Aluguer)

Tabela 9-Tabela da Entidade Fatura

A Entidade Funcionario produz a seguinte relação:

Funcionario -> atributos -> (id_Funcionario, nome, numero_vendas)
Chave Primária -> id_Funcionario

Tabela 10-Tabela da Entidade Funcionário

O atributo morada da Entidade Cliente é composto, logo produz a seguinte relação:

Morada -> atributos -> (cod_postal, rua, localidade, concelho)
Chave Primária -> cod_postal

Tabela 11-Tabela do atributo composto morada

A Entidade Veiculo produz a seguinte relação:

Veiculo -> atributos -> (matricula, combustivel, marca, modelo, preço_venda, preço_aluguer, estado)
Chave Primária -> matricula

Tabela 12-Tabela da Entidade Veículo

O atributo contacto da Entidade Cliente é composto, logo produz a seguinte relação:

contacto -> atributos -> (telemovel, email)
Chave Primária -> telemovel

Tabela 13-Tabela do atributo composto contacto



O atributo seguro da Entidade Aluguer é composto, logo produz a seguinte relação:

seguro -> atributos -> (tipo, preco)
Chave Primária -> tipo

Tabela 14-Tabela do atributo composto seguro

Concluída esta fase de construção das tabelas para o modelo lógico, fizemos uma análise geral de todas as atribuições de chaves primárias e estrangeiras, e também das especificações de cada atributo, e asseguramos a validação e a robustez de todo o modelo lógico com sucesso. Posto isto imprimimos no capítulo seguinte o Desenho do Modelo Lógico.

## 4.2 Desenho do modelo lógico

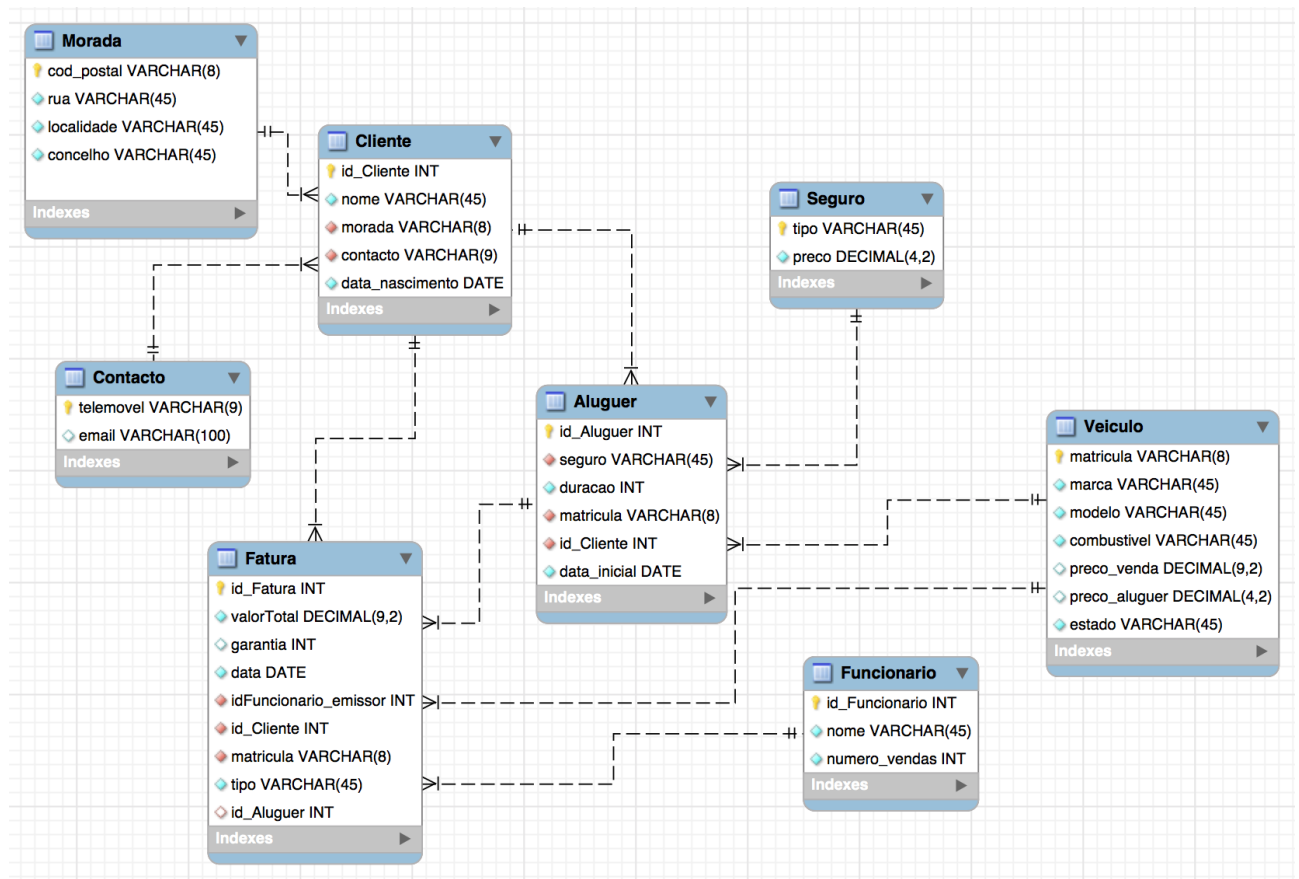


Figura 2- Modelo Lógico

### **4.3 Validação do modelo com interrogações do utilizador**

O Modelo Lógico elaborado é válido para as necessidades do próprio utilizador, com isto pretendemos dizer que é possível ao utilizador usufruir de uma forma dinâmica da sua base de dados, como vamos demonstrar a seguir.

O modelo permite guardar toda a informação constituinte de qualquer Fatura, incluído o preço final da transação, o veículo em questão, o cliente e o funcionário envolvidos no mesmo, entre outras funcionalidades. É permitido também remover ou adicionar faturas caso o utilizador assim o pretenda, ou caso uma nova transação surja.

O Modelo permite, como previsto, guardar toda a informação das restantes Entidades, bem como outras funcionalidades tais como adicionar novos veículos, novos clientes , novos funcionários.

Confirmamos deste modo o dinamismo pretendido para o utilizador.

## 5. Implementação Física

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

Nesta fase final da transformação do modelo lógico para o modelo físico, utilizamos o programa proposto pelo professor, o `mySQLWorkbench`. Fizemo-lo de forma bastante simples, utilizando o comando desse mesmo programa, o “forward Engineer”, ele cria toda a implementação física necessária, como podemos ver nas próximas figuras.

```
CREATE TABLE IF NOT EXISTS `Standinavia`.`Morada` (  
  `cod_postal` VARCHAR(8) NOT NULL,  
  `rua` VARCHAR(45) NOT NULL,  
  `localidade` VARCHAR(45) NOT NULL,  
  `concelho` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`cod_postal`))  
ENGINE = InnoDB;
```

Figura 3-Criação da Tabela Morada

```
CREATE TABLE IF NOT EXISTS `Standinavia`.`Contacto` (  
  `telemovel` VARCHAR(9) NOT NULL,  
  `email` VARCHAR(100) NULL,  
  PRIMARY KEY (`telemovel`))  
ENGINE = InnoDB;
```

Figura 4-- Criação da Tabela Contacto

```
CREATE TABLE IF NOT EXISTS `Standinavia`.`Funcionario` (  
  `id_Funcionario` INT NOT NULL,  
  `nome` VARCHAR(45) NOT NULL,  
  `numero_vendas` INT NOT NULL,  
  PRIMARY KEY (`id_Funcionario`))  
ENGINE = InnoDB;
```

Figura 5-Criação da Tabela Funcionário

```

CREATE TABLE IF NOT EXISTS `Standinavia`.`Cliente` (
  `id_Cliente` INT NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `morada` VARCHAR(8) NOT NULL,
  `contacto` VARCHAR(9) NOT NULL,
  `data_nascimento` DATE NOT NULL,
  PRIMARY KEY (`id_Cliente`),
  INDEX `morada1_idx` (`morada` ASC) VISIBLE,
  INDEX `contacto1_idx` (`contacto` ASC) VISIBLE,
  CONSTRAINT `morada1`
    FOREIGN KEY (`morada`)
      REFERENCES `Standinavia`.`Morada` (`cod_postal`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `contacto111`
    FOREIGN KEY (`contacto`)
      REFERENCES `Standinavia`.`Contacto` (`telemovel`)
      ON DELETE NO ACTION

```

Figura 6-Criação da Tabela Cliente

```

CREATE TABLE IF NOT EXISTS `Standinavia`.`Seguro` (
  `tipo` VARCHAR(45) NOT NULL,
  `preco` DECIMAL(4,2) NOT NULL,
  PRIMARY KEY (`tipo`))
ENGINE = InnoDB;

```

Figura 7- Criação da Tabela Seguro

```
CREATE TABLE IF NOT EXISTS `Standinavia`.`Veiculo` (
  `matricula` VARCHAR(8) NOT NULL,
  `marca` VARCHAR(45) NOT NULL,
  `modelo` VARCHAR(45) NOT NULL,
  `combustivel` VARCHAR(45) NOT NULL,
  `preco_venda` DECIMAL(9,2) NULL,
  `preco_aluguer` DECIMAL(4,2) NULL,
  `estado` VARCHAR(45) NULL,
  PRIMARY KEY (`matricula`))
ENGINE = InnoDB;
```

Figura 8- Criação Tabela Veículo

```
CREATE TABLE IF NOT EXISTS `Standinavia`.`Aluguer` (
  `id_Aluguer` INT NOT NULL,
  `seguro` VARCHAR(45) NOT NULL,
  `duracao` INT NOT NULL,
  `matricula` VARCHAR(8) NOT NULL,
  `id_Cliente` INT NOT NULL,
  `data_inicial` DATE NOT NULL,
  PRIMARY KEY (`id_Aluguer`),
  INDEX `fk_idVeiculo_idx` (`matricula` ASC) VISIBLE,
  INDEX `fk_idCliente_idx` (`id_Cliente` ASC) VISIBLE,
  INDEX `fk_seguro1_idx` (`seguro` ASC) VISIBLE,
  CONSTRAINT `fk_Matricula1`
    FOREIGN KEY (`matricula`)
    REFERENCES `Standinavia`.`Veiculo` (`matricula`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_idCliente1`
    FOREIGN KEY (`id_Cliente`)
    REFERENCES `Standinavia`.`Cliente` (`id_Cliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_seguro1`
    FOREIGN KEY (`seguro`)
    REFERENCES `Standinavia`.`Seguro` (`tipo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Figura 9-Criação Tabela Aluguer

```

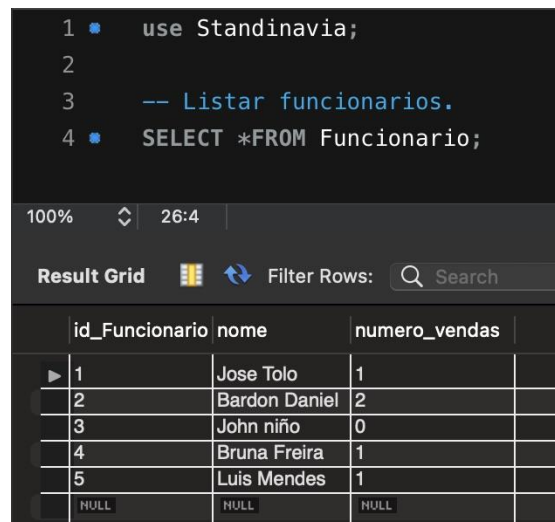
CREATE TABLE IF NOT EXISTS `Standinavia`.`Fatura` (
  `id_Fatura` INT NOT NULL,
  `valorTotal` DECIMAL(9,2) NOT NULL,
  `garantia` INT NULL,
  `data` DATE NOT NULL,
  `idFuncionario_emissor` INT NOT NULL,
  `id_Cliente` INT NOT NULL,
  `matricula` VARCHAR(8) NOT NULL,
  `tipo` VARCHAR(45) NOT NULL,
  `id_Aluguer` INT NULL,
  PRIMARY KEY (`id_Fatura`),
  INDEX `fk_idVeiculo_idx` (`matricula` ASC) VISIBLE,
  INDEX `fk_idCliente_idx` (`id_Cliente` ASC) VISIBLE,
  INDEX `fk_idFuncionario_emissor_idx` (`idFuncionario_emissor` ASC) VISIBLE,
  INDEX `fk_idaluguer_idx` (`id_Aluguer` ASC) VISIBLE,
  CONSTRAINT `fk_idFuncionario_emissor`
    FOREIGN KEY (`idFuncionario_emissor`)
      REFERENCES `Standinavia`.`Funcionario` (`id_Funcionario`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_idCliente`
    FOREIGN KEY (`id_Cliente`)
      REFERENCES `Standinavia`.`Cliente` (`id_Cliente`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_Matricula`
    FOREIGN KEY (`matricula`)
      REFERENCES `Standinavia`.`Veiculo` (`matricula`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_idaluguer`
    FOREIGN KEY (`id_Aluguer`)
      REFERENCES `Standinavia`.`Aluguer` (`id_Aluguer`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 10- Criação Tabela Fatura

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

->Visualizar Todos os Funcionários do Stand



```
1 • use Standinavia;
2
3 -- Listar funcionarios.
4 • SELECT *FROM Funcionario;
```

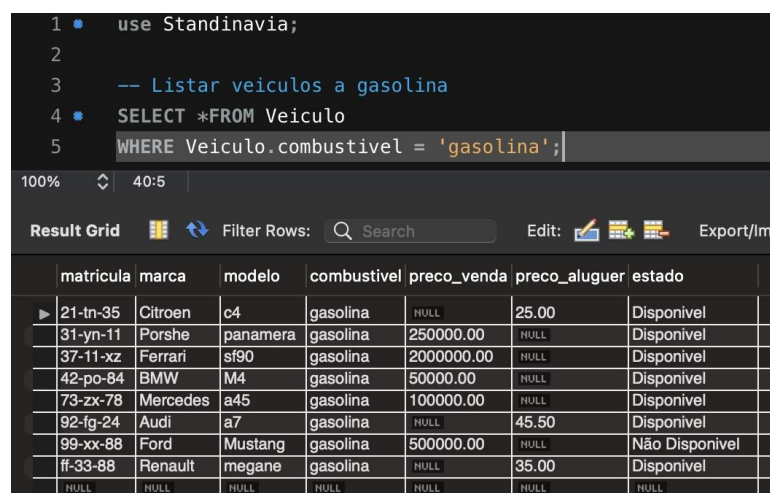
100% 26:4

Result Grid Filter Rows: Search

	id_Funcionario	nome	numero_vendas
▶	1	Jose Tolo	1
	2	Bardon Daniel	2
	3	John niño	0
	4	Bruna Freira	1
	5	Luis Mendes	1
	NULL	NULL	NULL

Figura 11-Query Funcionários

->Visualizar Todos os Veículos com a combustível gasolina



```
1 • use Standinavia;
2
3 -- Listar veiculos a gasolina
4 • SELECT *FROM Veiculo
5 WHERE Veiculo.combustivel = 'gasolina';
```

100% 40:5

Result Grid Filter Rows: Search Edit: Export/Im

	matricula	marca	modelo	combustivel	preco_venda	preco_aluguer	estado
▶	21-tn-35	Citroen	c4	gasolina	NULL	25.00	Disponivel
	31-yn-11	Porsche	panamera	gasolina	250000.00	NULL	Disponivel
	37-11-xz	Ferrari	sf90	gasolina	2000000.00	NULL	Disponivel
	42-po-84	BMW	M4	gasolina	50000.00	NULL	Disponivel
	73-zx-78	Mercedes	a45	gasolina	100000.00	NULL	Disponivel
	92-fg-24	Audi	a7	gasolina	NULL	45.50	Disponivel
	99-xx-88	Ford	Mustang	gasolina	500000.00	NULL	Não Disponivel
	ff-33-88	Renault	megane	gasolina	NULL	35.00	Disponivel
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 12-Query Veículos Combustível



->Visualizar Todos os Veículos disponíveis para Compra

```
1 use Standinavia;
2
3 -- Listar veiculos disponiveis para compra
4 SELECT V.matricula, V.marca, V.modelo, V.combustivel, V.preco_venda
5 FROM Veiculo AS V
6 WHERE V.preco_venda is NOT NULL AND V.estado = 'Disponivel';
7
```

100% 1:7

Result Grid Filter Rows: Search Edit: Export/Import:

	matricula	marca	modelo	combustivel	preco_venda
▶	10-11-pp	Toyota	corolla	gasoleo	30000.00
	31-gg-34	Ford	mustang	hibrido	75000.00
	31-yn-11	Porsche	panamera	gasolina	250000.00
	37-11-xz	Ferrari	sf90	gasolina	2000000.00
	42-po-84	BMW	M4	gasolina	50000.00
	73-zx-78	Mercedes	a45	gasolina	100000.00
	90-m-34	BMW	i8	eletrico	123000.00
	ch-22-33	Mercedes	EQ	eletrico	500000.00
	NULL	NULL	NULL	NULL	NULL

Figura 13-Query Veículos Disponíveis para Compra

->Visualizar o Funcionário do Stand com mais Vendas

```
1 use Standinavia;
2
3 -- Funcionario com mais vendas
4 SELECT F.nome AS MelhorVendedor
5 FROM Funcionario AS F
6 ORDER BY numero_vendas DESC
7 LIMIT 1;
8
```

100% 1:8

Result Grid Filter Rows: Search

	MelhorVended...
▶	Bardon Daniel

Figura 14- Query Funcionario Com Mais Vendas

-> Visualizar todos os Clientes com mail gmail

```

1 use Standinavia;
2
3 -- Listar os emails dos Clientes que pertencem ao gmail
4 SELECT Cl.nome AS Nome, C.email AS Email
5 FROM Contacto AS C
6     INNER JOIN Cliente AS Cl
7     ON C.telemovel = Cl.contacto
8     WHERE C.email like '%@gmail.com';

```

100% 38:8

Result Grid Filter Rows: Search Export:

Nome	Email
Tone Felgueiras	felgas@gmail.com
Sonia Barreiros	setemaravilhas@gmail.com
Xico Truques	truquesmagia@gmail.com
Albino Costa	albinacostinha@gmail.com

Figura 15-Query Mail

-> Visualizar a localidade e a rua de todos os Clientes que pertencem ao concelho de Fafe

```

1 use Standinavia;
2
3 -- Listar a Localidade, rua dos Clientes que pertencem ao concelho de Fafe
4 SELECT Cl.nome AS Nome, M.localidade AS Localidade, M.rua AS Rua
5 FROM Morada AS M
6     INNER JOIN Cliente AS Cl
7     ON M.cod_postal = Cl.morada
8     where M.concelho = 'Fafe';
9

```

100% 1:9

Result Grid Filter Rows: Search Export:

Nome	Email
Tone Felgueiras	felgas@gmail.com
Sonia Barreiros	setemaravilhas@gmail.com
Xico Truques	truquesmagia@gmail.com
Albino Costa	albinacostinha@gmail.com

Figura 16-Query localidade e rua

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

->Veículos Disponíveis para Aluguer

```
1 Use Standinavia;
2
3 CREATE VIEW vwVeiculosDisponiveisAluguer AS
4     SELECT V.matricula AS Matricula, V.marca AS Marca, V.modelo AS Modelo, V.combustivel AS Combustivel, V.preco_aluguer AS PreçoAluguer
5     FROM Veiculo AS V
6     WHERE V.preco_aluguer is NOT NULL AND V.estado = 'Disponivel';
7
8
9
10 SELECT *FROM vwVeiculosDisponiveisAluguer;
```

Result Grid

	Matricula	Marca	Modelo	Combustivel	PreçoAluguer
▶	12-mn-34	BMW	330	gasoleo	50.00
	21-m-35	Citroen	c4	gasolina	25.00
	92-ig-24	Audi	a7	gasolina	45.50
	#-33-88	Renault	megane	gasolina	35.00
	k-21-32	Ford	fiesta	gasoleo	40.00
	k-44-55	BMW	118	gasoleo	55.00

Figura 17-View Veículos

->Todas as Facturas emitidas no Stand

```
11 Use Standinavia;
12
13 CREATE VIEW vwFaturas AS
14     SELECT F.id_Fatura AS Fatura, F.valorTotal AS Preço, F.garantia AS Garantia,
15     F.data AS Data, Fu.nome AS Funcionario, Cl.nome AS Cliente, F.matricula AS Veiculo, F.tipo AS Tipo, F.id_Aluguer AS Nº_Aluguer
16     FROM Fatura AS F, Funcionario AS Fu, Cliente AS Cl
17     WHERE Fu.id_Funcionario = F.idFuncionario_emissor
18     AND Cl.id_Cliente = F.id_Cliente
19     GROUP BY id_Fatura;
20
21
22 SELECT *FROM vwFaturas;
```

Result Grid

	Fatura	Preço	Garantia	Data	Funcionario	Cliente	Veiculo	Tipo	Nº_Aluguer
▶	0	1260.00	NULL	2021-06-12	Jose Tolo	Albino Costa	88-nm-54	Aluguer	1
	1	2000.00	NULL	2021-07-03	Bardon Daniel	Cristiano Ronaldo	27-wm-35	Aluguer	2
	2	1260.00	NULL	2021-06-09	Luis Mendes	Albino Costa	45-23-yz	Aluguer	3
	3	500000.00	2	2021-04-15	Bruna Freira	Inacio Enes	99-xx-88	Compra	NULL
	4	480.00	NULL	2021-06-01	Bardon Daniel	Sonia Barreiros	20-mn-30	Aluguer	4

Figura 18-View Facturas

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

Com base no nosso projeto e no nosso povoamento, é nos possível desenvolver uma estimativa bem precisa do espaço que irá ser necessário para suportar toda a Base de Dados. É do senso comum entre DataBase Developers que a quantidade de espaço necessária tanto na estrutura inicial como no estimado crescimento é uma preocupação fundamental para a criação da mesma, pois é consoante a mesma que iremos fornecer mais ou menos memória dos nossos aparelhos, e qual a quantidade necessária a ser reservada.

De seguida iremos demonstrar o espaço ocupado pelo esqueleto do modelo, posteriormente e com base nos nossos dados, iremos fazer uma estimativa inicial e futura.

A Entidade Cliente tem como espaço previsto calculado:

Atributo	Espaço máximo(bytes)
id_Cliente	4
nome	45
data_nascimento	4
morada	143
contacto	109
TOTAL	305

Tabela 15 -Espaço máximo dos atributos de Cliente

A Entidade Fatura tem como espaço previsto calculado:

<b>Atributo</b>	<b>Espaço máximo(bytes)</b>
id_Fatura	4
valorTotal	9.2
garantia	4
data	4
tipo	45
id_Aluguer	4
idFuncionario_emissor	4
id_Cliente	4
matricula	8
<b>TOTAL</b>	<b>86.2</b>

Tabela 16-Espaço máximo dos atributos de Fatura

A Entidade Veiculo tem como espaço previsto calculado:

<b>Atributos</b>	<b>Espaço máximo(bytes)</b>
matricula	8
combustivel	45
marca	45
modelo	45
estado	45
preco_venda	9.2
preco_aluguer	4.2
<b>TOTAL</b>	<b>200.4</b>

Tabela 17-Espaço máximo dos atributos de Veiculo

A Entidade Funcionario tem como espaço previsto calculado:

<b>Atributos</b>	<b>Espaço máximo(bytes)</b>
id_Funcionario	4
nome	45
numero_vendas	4
<b>TOTAL</b>	<b>54</b>

Tabela 18-- Espaço máximo dos atributos de Funcionario

A Entidade Aluguer tem como espaço previsto calculado:

<b>Atributos</b>	<b>Espaço máximo(bytes)</b>
id_Aluguer	4
seguro	49.2
id_Cliente	4
duracao	4
matricula	8
data_inicial	4
<b>TOTAL</b>	<b>73.2</b>

Tabela 19-Espaço máximo dos atributos de Aluguer

O atributo composto morada tem como espaço previsto calculado:

<b>Atributo</b>	<b>Espaço máximo(bytes)</b>
cod_postal	8
localidade	45
rua	45
concelho	45
<b>TOTAL</b>	<b>143</b>

Tabela 20-Espaço máximo do atributo composto morada

O atributo composto contacto tem como espaço previsto calculado:

Atributo	Espaço máximo(bytes)
telemovel	9
email	100
TOTAL	109

Tabela 21-Espaço máximo do atributo composto contacto

O atributo composto seguro tem como espaço previsto calculado:

Atributo	Espaço máximo(bytes)
tipo	45
preço	4.2
TOTAL	49.2

Tabela 22-Espaço máximo do atributo composto seguro

De acordo com o povoamento realizado pelo grupo, o espaço total inicial utilizado é dado por:

-> 19 Veículos =  $19 \times 200.4 = 3807.6$  bytes

-> 5 Facturas =  $5 \times 86.2 = 431$  bytes

-> 5 Funcionarios =  $5 \times 54 = 270$  bytes

-> 4 Alugueres =  $4 \times 73.2 = 292,8$  bytes

-> 7 Clientes =  $7 \times 305 = 2135$  bytes

**TOTAL = 6936,4 bytes**

No presente estado do projeto a memória utilizada é de 6936.4 bytes.



### **Taxa de Crescimento Anual:**

Para realizarmos uma estimativa de crescimento anual, propusemos um crescimento de 5 novas facturas por mês, em que 2 são de compra e 3 de aluguer, de 5 novos clientes por mês, com um acréscimo de 5 veículos por mês, e também a adição de 1 funcionário por ano.

$$\rightarrow 5 \text{ Facturas} = 5 \cdot 86.2 = 431 \cdot 12 = 5172$$

$$\rightarrow 3 \text{ Alugueres} = 3 \cdot 73.2 = 219.6 \cdot 12 = 2635,2$$

$$\rightarrow 5 \text{ Clientes} = 5 \cdot 305 = 1525 \cdot 12 = 18\,300$$

$$\rightarrow 5 \text{ Veículos} = 5 \cdot 200.4 = 1002 \cdot 12 = 12\,024$$

$$\rightarrow 1 \text{ Funcionario} = 1 \cdot 54 = 54$$

**TOTAL = 38185.2**

Somando todo este espaço necessário por um ano de negocio do Stand, deparamos nos com uma taxa de crescimento anual de memória de 38185.2 bytes.

## 6. Conclusões e Trabalho Futuro

Terminado o projeto, o nosso grupo sente que adquiriu bastantes competências relativas à Unidade Curricular de Base de Dados. Durante todo o processo, cada etapa do projeto provocou em nós um espírito de curiosidade e de entusiasmo em querer aprender e aumentar a nossa perceção do que realmente é uma Base de Dados. Tendo a noção de que a dimensão da mesma é bastante reduzida em comparação com as maiores e as mais diversificadas base de dados existentes, reconhecemos que os conceitos que aprendemos, os programas que manuseamos são parte da base de começo para um conhecimento mais maduro e abrangente de Base de Dados.

As etapas dos modelos lógicos, conceptuais e físicos foram bastante interessantes de realizar, visualizar toda a arquitectura permitiu ter uma percepção real do que uma base de dados se trata. A ponte existente entre o Mundo Real(no nosso caso, o Stand) e a base de suporte digital para o mesmo, após a realização este projeto, provou ser mais uma vez, já o tínhamos referido no início do projeto, como uma ferramenta essencial nos dias de hoje e garantidamente para o futuro também. A organização de Dados é e será uma ferramenta crucial para a quantidade aparentemente infinita de informação no mundo em que vivemos. Dito isto, admitimos também que poderíamos tornar o nosso projecto, em especifico os modelos, mais complexos, com mais tabelas, mais atributos entre outros, mas optamos por um caminho mais simples e que nos garantisse a solidez do projeto, o que pensamos ter concretizado com sucesso.

Para o trabalho futuro esperamos aprender tanto como neste trabalho, e também explorarmos mais as nossas próprias competências e se possível tornar lo mais complexo.

## 7. Referências Bibliográficas (estilo APA)

Connolly, T. M. & Begg, C. E., 2005. Database Systems, A Practical Approach to Design, Implementation, and Management. 4th Edition ed. Reino Unido: Harlow : Addison-Wesley.

Europcar.[Online]

Available at: [https://www.europcar.pt/pt-](https://www.europcar.pt/pt-pt?gclid=Cj0KCQiAnaenBhCUARIsABEee8VZBD18H1ljY9mQEzL7bfr3FBrF-zLf4VqpGH9i8Y8me_y5cu7QwK0aAuyOEALw_wcB)

[pt?gclid=Cj0KCQiAnaenBhCUARIsABEee8VZBD18H1ljY9mQEzL7bfr3FBrF-zLf4VqpGH9i8Y8me\\_y5cu7QwK0aAuyOEALw\\_wcB](https://www.europcar.pt/pt-pt?gclid=Cj0KCQiAnaenBhCUARIsABEee8VZBD18H1ljY9mQEzL7bfr3FBrF-zLf4VqpGH9i8Y8me_y5cu7QwK0aAuyOEALw_wcB)

