



Universidade do Minho

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Bases de Dados

Ano Letivo de 2022/2023

Belos Transportes

Diogo Matos(100741) Inês Ferreira(97372) José Correia(100610)
Pedro Ferreira(97646)

5 de junho de 2023

BD

Data de Receção	
Responsável	
Avaliação	
Observações	

Belos Transportes

Diogo Matos(100741) Inês Ferreira(97372) José Correia(100610) Pedro
Ferreira(97646)

5 de junho de 2023

Índice

1	Definição do Sistema	1
1.1	Contextualização	1
1.2	Fundamentação	2
1.3	Motivação e Objetivos	2
1.4	Análise de Viabilidade	3
1.5	Recursos e Equipa de Trabalho	4
1.6	Plano de Execução	5
2	Levantamento e Análise de Requisitos	6
2.1	Método de levantamento e de análise de requisitos adotado	6
2.2	Organização dos requisitos levantados	7
2.2.1	Requisitos de Descrição	7
2.2.2	Requisitos de Manipulação	8
2.2.3	Requisitos de Controlo	9
2.3	Análise e validação geral dos requisitos	9
3	Modelação Conceptual	10
3.1	Apresentação da abordagem de modelação realizada	10
3.2	Identificação e caracterização das entidades	11
3.3	Identificação e caracterização dos relacionamentos	12
3.3.1	Relação Passageiro - Viagem	12
3.3.2	Relação Passageiro - Passe Mensal	13
3.3.3	Relação Motorista - Viagem	14
3.3.4	Relação Motorista - Autocarro	15
3.3.5	Relação Viagem - Percurso	16
3.3.6	Relação Viagem - Paragem (Inicial e Final)	17
3.3.7	Relação Percurso - Paragem	18
3.4	Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	19
3.4.1	Passageiro	19

3.4.2	Passe Mensal	19
3.4.3	Percurso	19
3.4.4	Viagem	20
3.4.5	Paragem	20
3.4.6	Motorista	20
3.4.7	Autocarro	21
3.4.8	Neologismo TIPOPASSE	21
3.5	Apresentação e explicação do diagrama ER produzido	22
3.6	Análise e validação do modelo concetual produzido	23
4	Modelação Lógica	31
4.1	Construção e validação do modelo de dados lógico	31
4.2	Normalização de Dados	35
4.3	Apresentação e explicação do modelo lógico produzido	36
4.4	Validação do modelo com interrogações do utilizador	37
5	Implementação Física	40
5.1	Tradução do esquema lógico para o sistema de gestão de bases de dados esco- lhido em SQL	40
5.2	Tradução das interrogações do utilizador para SQL (alguns exemplos)	41
5.3	Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	42
5.4	Cálculo do espaço da bases de dados (inicial e taxa de crescimento anual)	43
5.5	Análise da Implementação Física	47
6	Implementação do Sistema de Recolha de Dados	48
6.1	Apresentação e modelo do sistema	48
6.2	Implementação e funcionamento do sistema de recolha	48
7	Implementação do Sistema de Painéis de Análise	50
7.1	Definição e caracterização da vista de dados para análise	50
7.2	Povoamento das estruturas de dados para análise	50
7.3	Apresentação e caracterização dos dashboards implementados	51
8	Conclusões e Trabalho Futuro	53
	Lista de Siglas e Acrónimos	54
	Anexos	55

Lista de Figuras

1.1	Logótipo da empresa <i>Belos Transportes</i>	1
1.2	Diagrama de Gantt (NÃO FINAL)	5
3.1	Ilustração da relação Passageiro - Viagem	12
3.2	Ilustração da relação Passageiro - Passe Mensal	13
3.3	Ilustração da relação Motorista - Viagem	14
3.4	Ilustração da relação Motorista - Autocarro	15
3.5	Ilustração da relação Viagem - Percurso	16
3.6	Ilustração das relações Viagem - Paragem	17
3.7	Ilustração da relação Percurso - Paragem	18
3.8	Diagrama ER produzido	22
3.9	Ilustração da entidade Paragem	23
3.10	Ilustração da relação Percurso - Paragem	23
3.11	Ilustração da relação Viagem - Paragem	24
3.12	Ilustração da relação Viagem - Percurso	25
3.13	Ilustração da entidade Passe	25
3.14	Ilustração da entidade Passageiro	26
3.15	Ilustração da relação Passageiro - Passe Mensal	26
3.16	Ilustração da relação Passageiro - Viagem	27
3.17	Ilustração da entidade Autocarro	28
3.18	Ilustração da entidade Motorista	28
3.19	Ilustração da relação Motorista - Viagem	29
3.20	Ilustração da relação Motorista - Autocarro	29
4.1	Ilustração da tabela da entidade Viagem	31
4.2	Ilustração da tabela da entidade Passageiro	32
4.3	Ilustração da tabela da entidade Motorista	32
4.4	Ilustração da tabela da entidade Paragem	33
4.5	Ilustração da tabela da entidade Percurso	33
4.6	Ilustração da tabela da entidade Passe Mensal	33

4.7	Ilustração da tabela da entidade Autocarro	34
4.8	Ilustração da tabela do relacionamento PercursoParagem	34
4.9	Modelo Lógico	36
5.1	Código SQL relativo à interrogação 1	41
5.2	Código SQL relativo à interrogação 4	41
5.3	Código SQL relativo à interrogação 5	41
5.4	Código SQL relativo à interrogação 8	42
5.5	Código SQL relativo à interrogação 9	42
5.6	Código SQL relativo à vista de paragens	42
5.7	Código SQL relativo à vista de percursos	43
5.8	Código SQL relativo à vista de viagens	43
6.1	Exemplo de ficheiro de <i>CSV</i> do motorista	48
6.2	Função <code>insere_dados_motorista()</code>	49
7.1	Gráfico de passageiros por dia	51
7.2	Gráfico da percentagem de passageiros por cidade	52
7.3	Gráfico das paragens mais usadas	52

Lista de Tabelas

3.1	Tabela de atributos da entidade Passageiro	19
3.2	Tabela de atributos da entidade Passe Mensal	19
3.3	Tabela de atributos da entidade Percurso	19
3.4	Tabela de atributos da entidade Viagem	20
3.5	Tabela de atributos da entidade Paragem	20
3.6	Tabela de atributos da entidade Motorista	20
3.7	Tabela de atributos da entidade Autocarro	21
3.8	Tabela de atributos da entidade neologismo tipo-passe	21
5.1	Cálculo do tamanho da tabela Passageiro	43
5.2	Cálculo do tamanho da tabela PasseMensal	44
5.3	Cálculo do tamanho da tabela Paragem	44
5.4	Cálculo do tamanho da tabela Percurso	44
5.5	Cálculo do tamanho da tabela PercursoParagem	45
5.6	Cálculo do tamanho da tabela Motorista	45
5.7	Cálculo do tamanho da tabela Autocarro	45
5.8	Cálculo do tamanho da tabela Viagem	46
5.9	Cálculo do tamanho total da BD	46

1 Definição do Sistema

1.1 Contextualização

A empresa de transportes públicos *Belos Transportes* tem sede em Braga e foi fundada recentemente a 21/09/2021 por um jovem empreendedor. Opera sobre a região do Cávado e Ave oferecendo serviços urbanos e intermunicipais, com foco no conforto, acessibilidade e mobilidade verde.

Desde jovem, António Belo, fundador da *Belos Transportes*, assistiu em primeira mão ao aumento do congestionamento e sobrelotação das ruas e cidades por automóveis poluidores e barulhentos. Na criação da sua nova rede de transportes António espera ultimamente oferecer melhores soluções aos seus clientes, melhorando o acesso a transportes públicos acessíveis, confortáveis e verdes que permitam reduzir ao máximo o recurso aos veículos de transporte pessoal, que vêm com custos associados cada vez maiores.

Na empresa trabalham atualmente cerca de 40 trabalhadores que mantêm uma pequena frota de veículos que operam a energia elétrica e combustíveis renováveis. Com o aumento da exposição e crescimento da *startup*, António espera conseguir aumentar as suas frotas e incluir mais percursos na sua rede.



Figura 1.1: Logótipo da empresa *Belos Transportes*

1.2 Fundamentação

No decorrer dos dois primeiros anos de operação, a gestão de bilhetes, frotas, percursos, paragens, clientes, entre outros aspetos foram asseguradas de forma arcaica em lápis e papel, sendo claros os problemas de organização e a dificuldade de análise dos dados gerados. Esta falta de digitalização torna-se, assim, limitadora.

Com o objetivo de expansão em mente, António planeia superar estes percalços com a introdução de uma base de dados que se provará fundamental no controlo, gestão e organização da diversa gama de informações que precisam de ser manuseadas para um excecional e profissional funcionamento da sua rede de transporte, permitindo o crescimento da sua empresa.

Depois de alguma pesquisa e recomendação, foi selecionada a empresa *BaseTech* como responsável pela construção e implementação do novo sistema de bases de dados, assim como o estudo e análise da viabilidade da mesma.

1.3 Motivação e Objetivos

Para além de ultrapassar as limitações impostas por uma falta de digitalização, António espera que com a implementação do futuro sistema de bases de dados consiga alcançar um conjunto de objetivos principais que ultimamente conduzam a empresa aos padrões que esta defende, nomeadamente:

1. Melhorar a organização e controlo das vendas de bilhetes, lotação dos autocarros, percursos, paragens, horários, etc;
2. Melhorar a gestão das contas da empresa, nomeadamente lucros e despesas de manutenção, combustível, limpeza, etc;
3. Melhorar a distribuição das frotas e percursos para atender as áreas com mais necessidade;
4. Reduzir os erros humanos e minimizar os custos operacionais associados a processos manuais;

5. Identificar áreas de redução de custos, como percursos pouco utilizados ou horários com baixa demanda, de forma a otimizar as operações e maximizar o uso dos veículos;
6. Possibilitar a análise de dados em tempo real, contribuindo para uma eficiente identificação de problemas e rápida implementação de soluções;
7. Contabilizar o impacto ambiental positivo das frotas renováveis, nomeadamente a quantidade de emissões de CO2 poupadas;
8. Tomar decisões mais informadas e estratégicas;
9. Permitir a implementação de um sistema de passes e bilhetes online, para além dos bilhetes físicos atuais, que ajudem a melhorar os tempos de espera, a adesão dos clientes e diminuir o impacto ambiental;
10. Fornecer informação mais precisa, atualizada e útil aos passageiros, melhorando a acessibilidade da mesma;
11. Recolher informação acerca dos clientes de forma a permitir um desenvolvimento personalizado dos serviços tendo em conta os diversos públicos-alvo;
12. Aumentar a satisfação dos clientes e contribuir para uma imagem positiva da empresa no mercado.

1.4 Análise de Viabilidade

Tendo em conta os pontos apresentados anteriormente, provam-se vários os possíveis benefícios da implementação de um sistema de base de dados na *Belos Transportes*. No entanto, esta decisão deve ser viável operacional e financeiramente.

É importante destacar que o investimento na implementação da base de dados pode ser rapidamente recuperado. Os benefícios financeiros e operacionais mencionados acima, em conjunto com a melhoria do atendimento ao cliente, possibilitam um aumento significativo dos lucros da empresa. Adicionalmente, a implementação de uma base de dados é uma solução a longo prazo, que traz benefícios contínuos para a empresa ao longo do tempo.

De forma a analisar a viabilidade do projeto de forma concreta, foi realizado um estudo estatístico pela empresa *BaseTech*, responsável pela implementação da base de dados, do qual resultaram estatísticas aproximadas que prevêm uma recuperação de 20% do valor total

investido nos primeiros dois meses de funcionamento do novo sistema. Isto significa que a rentabilidade seria alcançada após apenas 10 meses de funcionamento, um valor bastante promissor para António.

Considerando estas variáveis, conclui-se que o investimento na implementação de uma base de dados para a *Belos Transportes* é altamente recomendado, uma vez que a mesma se prova adequada para responder às necessidades do projeto e compõe uma solução essencial para que a empresa se mantenha competitiva, maximize o seu lucro e melhore a qualidade dos seus serviços.

1.5 Recursos e Equipa de Trabalho

Para uma implementação bem sucedida da base de dados é necessário um conjunto de recursos humanos e materiais, nomeadamente:

- António Belo, funcionários, motoristas, passageiros e equipa de desenvolvimento;
- Software de gestão de bases de dados (SGBD);
- Máquinas de faturação digitais para os veículos.

Para além disso, é necessária uma equipa de trabalho para a implementação e utilização/atualização da base de dados, esta divide-se em pessoal **interno** e **externo**.

O pessoal interno é composto por todos os trabalhadores que garantem o funcionamento da empresa, desde motoristas a funcionários, assim como o fundador da empresa, António Belo. Estes são responsáveis pela operação dos veículos, venda de bilhetes, gestão de percursos e frotas, manutenção de veículos, secretariado, atendimento ao cliente, etc.

Já o pessoal externo é constituído por todos os responsáveis pelo desenvolvimento do sistema relacional de armazenamento de dados, isto é, o Arquiteto e a equipa de Engenheiros Informáticos e Engenheiros de Bases de Dados. Estes devem garantir o correto funcionamento do sistema, realizando o levantamento de requisitos, modelação e implementação do sistema, entre outros requisitos.

Adicionalmente, os passageiros são uma componente essencial para se obter uma avaliação precisa do desempenho do serviço. Deste modo, foi selecionado um grupo de clientes para preenchimento de questionários de opinião e satisfação.

1.6 Plano de Execução

A correta implementação e integração da base de dados está dependente do estabelecimento e definição de um plano de execução a ser seguido por todos os envolvidos no processo.

Deste modo, António Belo, juntamente com a equipa de desenvolvimento da *BaseTech*, estabeleceu um plano de trabalhos a ser seguido, assim como a sua cronologia e datas a serem cumpridas. Este foi sendo construído ao longo de algumas semanas, na sequência de um conjunto de reuniões e entrevistas.

Este plano inclui todas as fases relevantes ao ciclo de vida de um sistema de base de dados, desde a definição do sistema e requisitos até à implementação e testagem. Em anexo encontra-se o ficheiro *diagrama_gantt.xlsx* que constitui uma representação gráfica da cronologia do plano de execução.

Segue-se um excerto desse mesmo ficheiro:

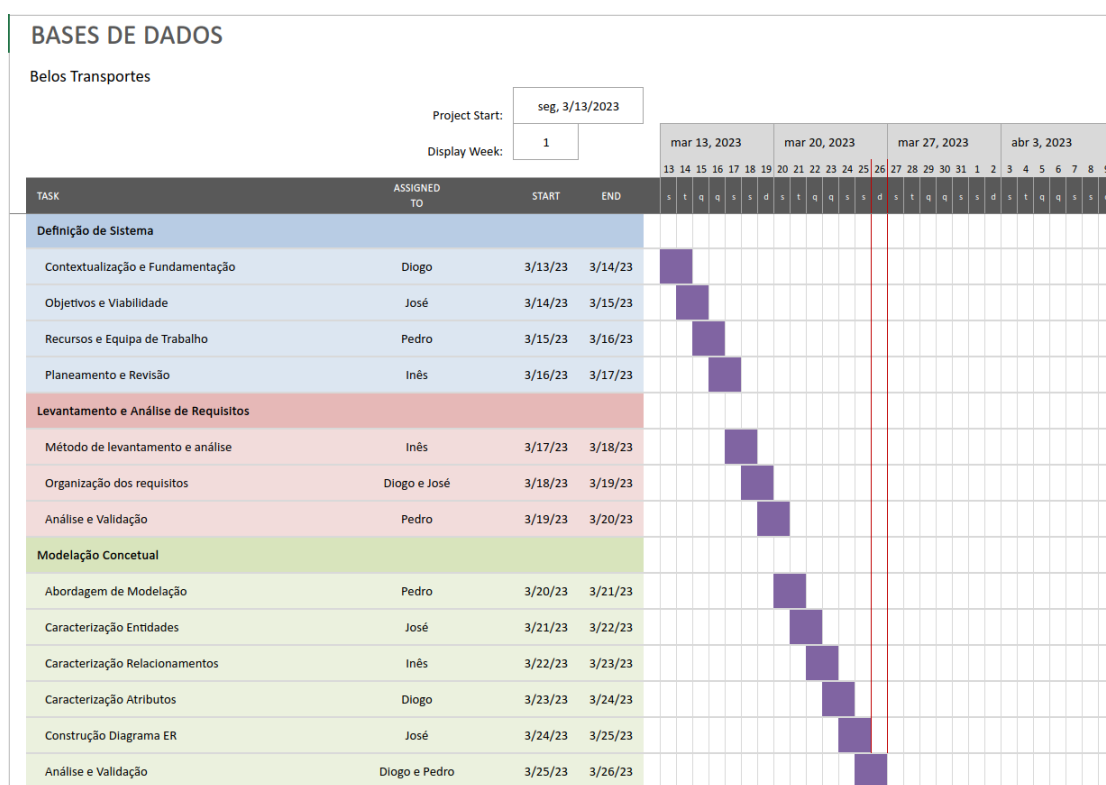


Figura 1.2: Diagrama de Gantt (**NÃO FINAL**)

2 Levantamento e Análise de Requisitos

2.1 Método de levantamento e de análise de requisitos adotado

No processo de implementação da base de dados, foi adotado um procedimento estruturado para o levantamento dos requisitos necessários, que consistiu em diversas etapas.

Numa abordagem inicial, foram realizadas diferentes reuniões com António, os membros da direção e a equipa de desenvolvimento, com o objetivo de identificar as necessidades da base de dados. Ao longo destas reuniões, foram discutidos os principais desafios da empresa, as áreas onde a implementação da base de dados pode trazer melhorias mais significativas, assim como os melhores métodos de obtenção de informação. Posteriormente, foi feita uma análise cuidadosa da documentação gerada pela empresa antes da implementação da base de dados. Foram examinados documentos como relatórios de vendas, dados financeiros, planos de percursos e paragens e horários dos autocarros. A análise destes documentos permitiu identificar as áreas mais críticas e as principais dificuldades que precisavam de ser endereçadas.

Para além disso, foi levada a cabo pela equipa de levantamento de requisitos uma observação dos processos e funcionamento da empresa. Foram realizadas visitas à central dos autocarros, às paragens e às bilheteiras espalhadas pelas cidades. A equipa acompanhou também os motoristas num conjunto de viagens pelos diferentes percursos existentes. Estas observações permitiram identificar os principais desafios enfrentados pelos trabalhadores no seu quotidiano e compreender como a base de dados pode ajudar na sua melhoria.

Por fim, foram realizadas entrevistas aos trabalhadores e clientes. Foram entrevistados motoristas, funcionários e clientes frequentes e não frequentes, portadores e não portadores de passe mensal. Estas entrevistas permitiram recolher informações adicionais acerca das necessidades da empresa e dos seus utilizadores, assim como as suas expectativas e opiniões.

Com base nestas etapas, foi elaborada uma lista de requisitos essenciais para a implementação da base de dados. Estes requisitos foram organizados em três distintas categorias: descrição, manipulação e controlo.

2.2 Organização dos requisitos levantados

Cada um dos requisitos apresentados de seguida encontra-se documentado no ficheiro anexado *requisitos.xlsx*.

2.2.1 Requisitos de Descrição

1. Cada **paragem** tem associado um id, um nome e a sua localização (endereço e coordenadas GPS);
2. Um **percurso** é composto por um id, um tipo (urbano ou intermunicipal), uma duração, um conjunto de paragens e a hora em que passa em cada paragem;
3. Uma paragem tem que estar incluída em um ou mais percursos;
4. Uma **viagem** é constituída por um número, uma data, uma hora, uma duração, um percurso, uma paragem inicial e final, uma distância e tem um custo associado;
5. Uma viagem é realizada como parte de um percurso;
6. Um **passe mensal** contém um id, um tipo de plano mensal (urbano ou intermunicipal) e um custo;
7. Um **passageiro** possui um id e pode possuir um nome, um NIF, uma morada e um contacto;
8. Um passageiro pode ou não possuir um passe mensal;
9. Um passageiro com passe mensal deve estar registado com as suas informações pessoais, para além do id;
10. Uma viagem deve ser realizada por um passageiro através de um passe mensal ou bilhete;
11. Um passageiro pode realizar diversas viagens;

12. Um **autocarro** é constituído por um id, uma marca, um modelo, uma matrícula e uma capacidade;
13. O registo de um **motorista** inclui o seu nome, contacto, data de nascimento, morada, NIF, IBAN e o seu salário;
14. Um motorista faz várias viagens, e pode conduzir diferentes autocarros.

2.2.2 Requisitos de Manipulação

1. Para cada viagem deve ser calculada uma estimativa da quantidade de emissões de CO2 poupadas, através da sua distância;
2. O total de CO2 poupado por dia deve ser contabilizado pelo sistema;
3. O total de CO2 poupado na operação da empresa deve ser contabilizado pelo sistema;
4. As paragens com mais e menos percursos devem ser contabilizadas pelo sistema;
5. O sistema deve registar a quantidade de viagens efetuadas por cada passageiro;
6. Deve ser registada a quantidade de viagens realizadas por cada motorista;
7. O total de viagens realizadas diariamente deve ser contabilizado pelo sistema;
8. Para cada trajeto (Paragem Inicial → Paragem Final), deverá ser contabilizada a quantidade de passageiros que o realizaram;
9. Para cada percurso deve ser calculado o lucro obtido através da venda de bilhetes;
10. Deve ser registado o número de autocarros que estiveram em operação numa determinada data;
11. O sistema deverá ser capaz de contabilizar o lucro total obtido pela venda de passes e bilhetes.

2.2.3 Requisitos de Controlo

1. Um passageiro pode ou não possuir um passe mensal;
2. Um passageiro com passe mensal deve estar registado com as suas informações pessoais, para além do id;
3. Uma viagem deve ser realizada por um passageiro através de um passe mensal ou bilhete;
4. Um passageiro pode realizar diversas viagens;
5. Um motorista faz várias viagens, e pode conduzir diferentes autocarros.
6. Cada viagem só pode ser realizada apenas por um motorista;
7. Cada viagem tem que estar associada a um percurso;
8. O plano mensal urbano apenas permite efetuar viagens de circuito urbano, isto é, que se efetuam em percursos urbanos;
9. O plano mensal intermunicipal permite efetuar viagens de circuito urbano e intermunicipal, isto é, que se efetuam em percursos urbanos ou intermunicipais;
10. Um passe mensal pode ter apenas um de dois tipos existentes (plano urbano ou plano intermunicipal).

2.3 Análise e validação geral dos requisitos

Depois da identificação, definição e formulação dos requisitos todo o trabalho realizado foi, naturalmente, analisado e revisto pela equipa de analistas em colaboração com António Belo. Toda a documentação foi estudada de forma a assegurar a integridade do projeto. Após algumas modificações e correções, os requisitos foram então aprovados e a equipa procedeu para a próxima fase de desenvolvimento.

3 Modelação Conceptual

3.1 Apresentação da abordagem de modelação realizada

Após a obtenção de toda a informação necessária e a definição dos requisitos, procedemos à elaboração do planeamento da estrutura e design da Base de Dados. Para melhorar a eficiência deste processo, optou-se por iniciar com um Diagrama ER, que consiste num tipo de fluxograma capaz de ilustrar e representar as diversas entidades interligadas por relacionamentos, assim como os seus atributos.

O desenvolvimento deste Diagrama ER baseou-se na identificação das distintas entidades implicadas na problemática, seguido das suas inter-relações e, por último, dos respetivos atributos, tanto das entidades como das suas conexões. Assim sendo, torna-se crucial que a nossa modelação conceptual represente de forma clara e precisa a estrutura das entidades, das suas relações e atributos, de acordo com as necessidades de gestão, para criar um modelo comportamental que reproduza, tanto quanto possível, a realidade e possa gerir um sistema de informação relativo a uma rede de transportes de forma eficaz.

Desta forma, concedeu-se uma atenção particular a esta modelação, com o objetivo de possibilitar uma aproximação cada vez mais exata a um sistema real, que possa satisfazer as necessidades dos utilizadores.

3.2 Identificação e caracterização das entidades

Atendendo aos requisitos levantados, foram identificadas as entidades apresentadas de seguida:

- **Viagem** Representa as viagens efetuadas pelos passageiros, acolhendo as suas informações essenciais como número, data, hora, custo, duração, percurso, etc.
- **Passageiro** Caracteriza os passageiros que efetuam as viagens. No caso deste ser possuidor de um Passe Mensal, inclui as suas informações básicas como nome, NIF, morada, etc. Caso contrário, possui apenas um número identificador.
- **Motorista** Informação acerca dos motoristas, responsáveis por operar os autocarros. Inclui o seu salário e outras informações pessoais.
- **Percurso** Representa uma rota na sua integridade, desde o local de partida até ao destino, durante a qual são realizadas diversas viagens. Inclui informações relativas ao seu tipo, duração e conjunto de paragens por onde passa.
- **Passe Mensal** Caracteriza os passes mensais que podem ser utilizados pelos passageiros para efetuar viagens. Contém todas as informações relevantes, como o seu custo e tipo. Este está sempre e obrigatoriamente relacionado com um passageiro.
- **Autocarro** Informação acerca dos autocarros operados pelos motoristas. Inclui as suas informações principais, como matrícula, marca, modelo, etc. e é identificado por um ID.
- **Paragem** Representa as diversas paragens que os percursos incluem e onde a *Belos Transportes* opera, acolhendo informações essenciais como um nome, um endereço e as suas coordenadas GPS. É identificada por um ID.

3.3 Identificação e caracterização dos relacionamentos

3.3.1 Relação Passageiro - Viagem

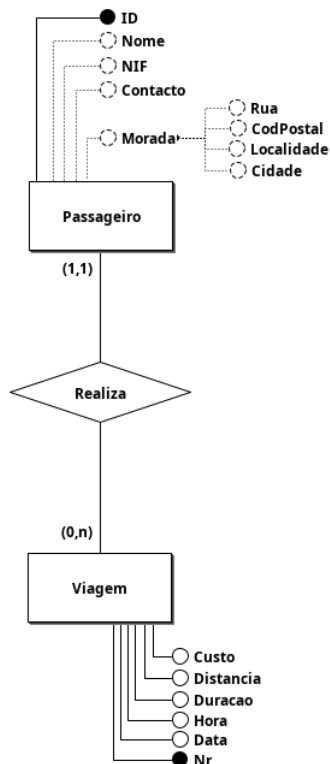


Figura 3.1: Ilustração da relação Passageiro - Viagem

Relacionamento: Passageiro realiza Viagem

Descrição: A relação Passageiro - Viagem é essencial para o registo da quantidade de passageiros que realizam viagens, assim como o dinheiro que gastam, a distância que percorrem, etc., pelo que consideramos importante estabelecer este relacionamento.

Multiplicidade: Passageiro (1, 1) -> Viagem (0, n)

Um passageiro pode realizar várias viagens, no entanto cada viagem é única e associada obrigatoriamente a 1 passageiro.

Atributos: Este relacionamento não tem atributos associados.

3.3.2 Relação Passageiro - Passe Mensal

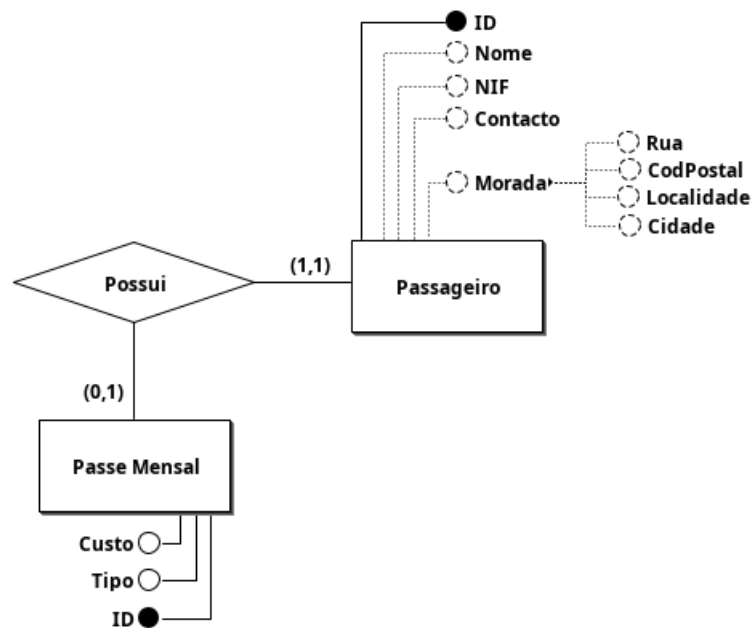


Figura 3.2: Ilustração da relação Passageiro - Passe Mensal

Relacionamento: Passageiro possui Passe Mensal

Descrição: A relação Passageiro - Passe Mensal assegura o registo dos passageiros que possuem um passe mensal e permite atribuir/associar um passe a um passageiro, assim como as suas informações pessoais. Estes registos são importantes do forma a gerir os pagamentos e a utilização de passes nas viagens.

Multiplicidade: Passageiro (1,1) -> Passe Mensal (0,1)

Um passageiro pode possuir 1 passe mensal e um passe mensal está obrigatoriamente associado a 1 passageiro.

Atributos: Este relacionamento não tem atributos associados.

3.3.3 Relação Motorista - Viagem

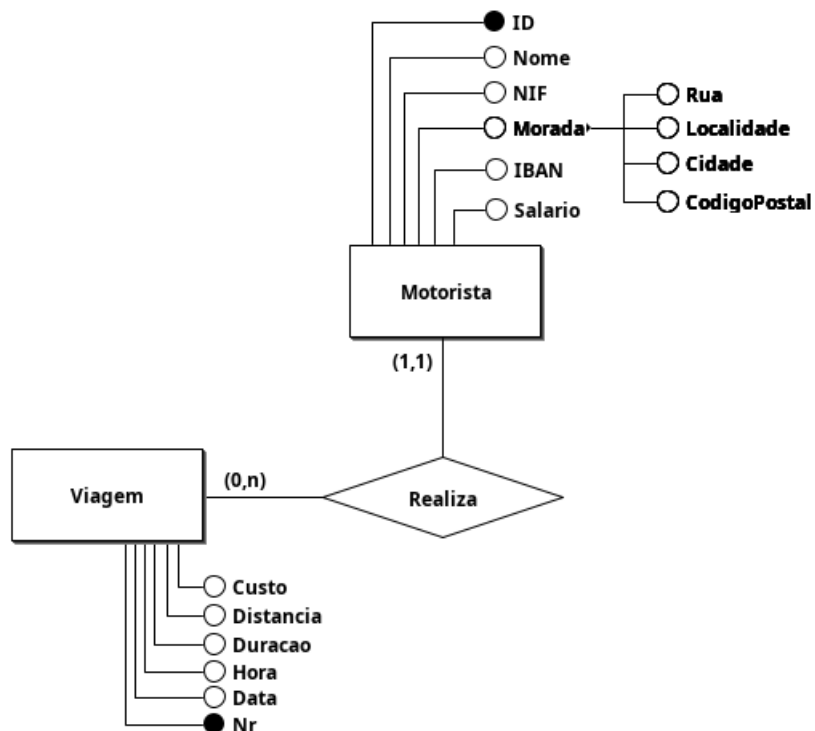


Figura 3.3: Ilustração da relação Motorista - Viagem

Relacionamento: Motorista realiza Viagem

Descrição: A relação Motorista - Viagem é importante de forma a identificar que motorista realizou cada viagem, o que permite registar as viagens realizadas por cada motorista.

Multiplicidade: Motorista (1,1) -> Viagem (0,n)

Um motorista pode realizar diversas viagens e cada viagem está obrigatoriamente associada a 1 motorista.

Atributos: Este relacionamento não tem atributos associados.

3.3.4 Relação Motorista - Autocarro

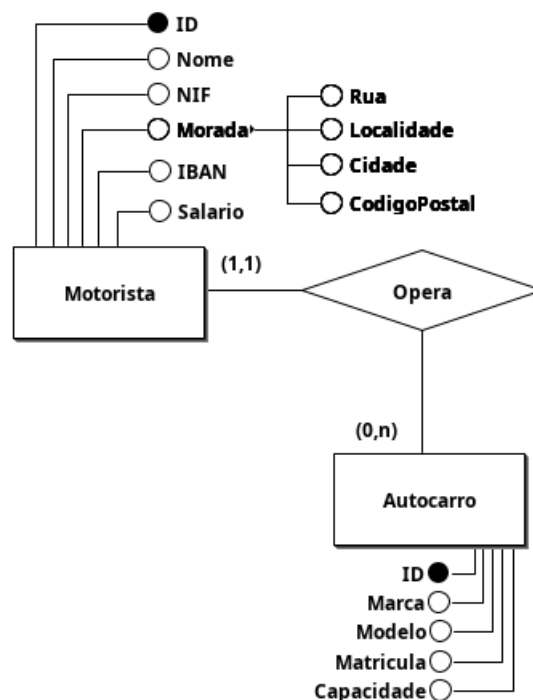


Figura 3.4: Ilustração da relação Motorista - Autocarro

Relacionamento: Motorista opera Autocarro

Descrição: A relação Motorista - Autocarro permite registar que motorista opera cada autocarro, assim como identificar que autocarros cada motorista opera e a quantidade de motoristas que operam cada autocarro.

Multiplicidade: Motorista (1,1) -> Autocarro (0,n)

Um motorista pode operar vários autocarros e cada autocarro é obrigatoriamente operado por 1 motorista.

Atributos: Este relacionamento não tem atributos associados.

3.3.5 Relação Viagem - Percurso

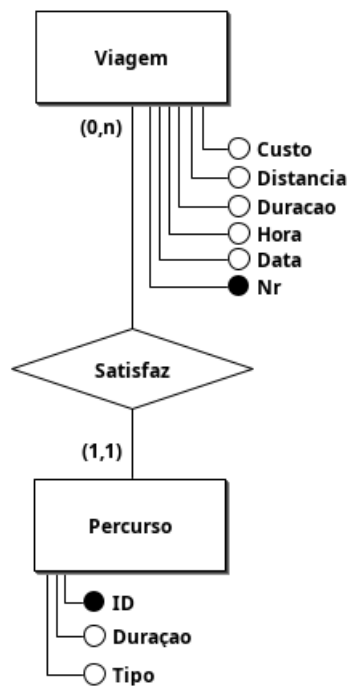


Figura 3.5: Ilustração da relação Viagem - Percurso

Relacionamento: Viagem satisfaz Percurso

Descrição: A relação Viagem - Percurso define em que percurso cada viagem é efetuada e permite identificar a quantidade de viagens realizadas em cada percurso. É importante registar estas informações de forma a identificar os percursos mais e menos populares.

Multiplicidade: Viagem (0,n) -> Percurso (1,1)

Uma viagem satisfaz obrigatoriamente 1 percurso e num percurso podem ser realizadas diversas viagens.

Atributos: Este relacionamento não tem atributos associados.

3.3.6 Relação Viagem - Paragem (Inicial e Final)

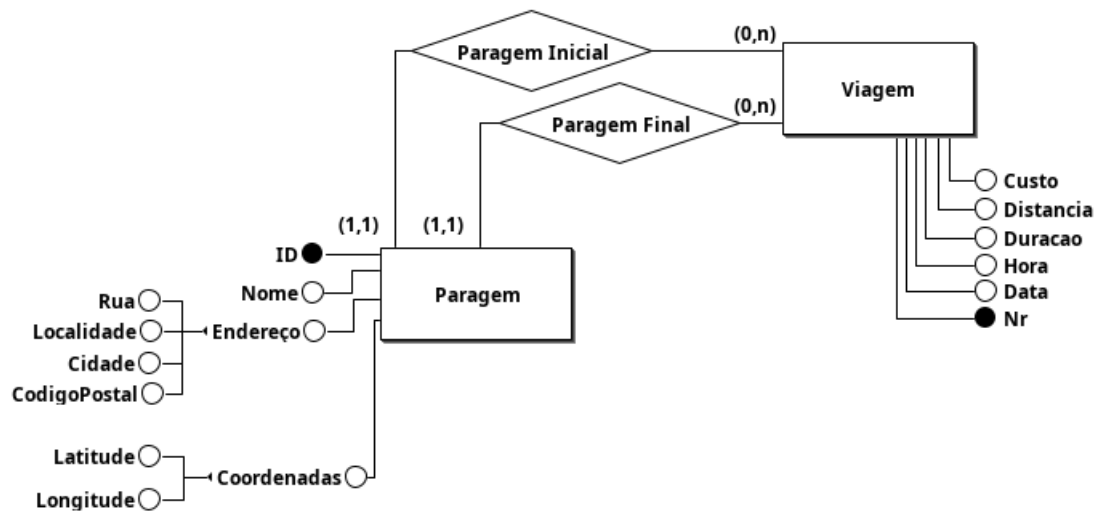


Figura 3.6: Ilustração das relações Viagem - Paragem

Descrição: As relações Viagem - Paragem asseguram que a cada viagem está associada uma paragem inicial e uma paragem final, isto é, a paragem onde o passageiro entrou e a paragem onde o passageiro saiu. Este relacionamento permite identificar a quantidade de viagens iniciadas e finalizadas em cada paragem e tirar conclusões acerca das paragens mais populares.

Multiplicidade: Viagem (0,n) -> Paragem (1,1)

Cada viagem está obrigatoriamente associada a 2 paragens, uma inicial (pela relação Viagem - Paragem (Inicial)) e uma final (pela relação Viagem - Paragem (Final)) e cada paragem pode estar associada a diversas viagens.

Atributos: Estes relacionamentos não tem atributos associados.

3.3.7 Relação Percurso - Paragem

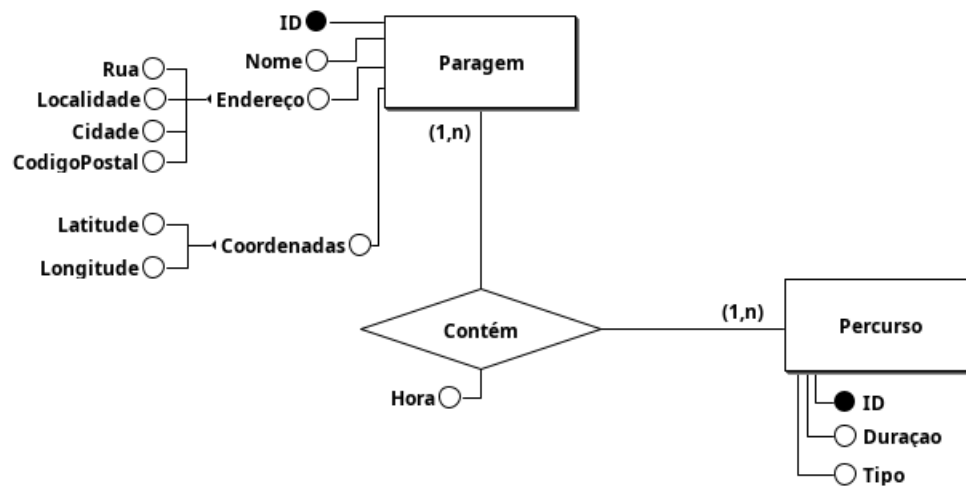


Figura 3.7: Ilustração da relação Percurso - Paragem

Relacionamento: Percurso contém Paragem

Descrição: A relação Percurso - Paragem permite registar por que paragens cada percurso passa, isto é, que paragens cada percurso inclui e é essencial para a definição de um percurso, uma vez que este é definido por um conjunto ordenado de paragens.

Multiplicidade: Percurso (1,n) -> Paragem (1,n)

Cada percurso tem que conter obrigatoriamente 1 ou mais paragens, assim como cada paragem tem que estar incluída obrigatoriamente em 1 ou mais percursos.

Atributos: O relacionamento possui o atributo "Hora" que permite identificar a que hora determinado percurso passa em cada uma das paragens que este inclui. Isto permite a construção de um horário para cada paragem, isto é, a que horas cada um dos percursos que a incluem lá passam.

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

3.4.1 Passageiro

Atributo	Domínio e Tamanho	Nulo	Composto	Multivalorado	Derivado	Candidata
ID	INT	Não	Não	Não	Não	Sim
Nome	VARCHAR(50)	Sim	Não	Não	Não	Não
NIF	INT	Sim	Não	Não	Não	Não
Contacto	VARCHAR(20)	Sim	Não	Não	Não	Não
Rua	VARCHAR(100)	Sim	Sim	Não	Não	Não
CodPostal	VARCHAR(8)	Sim	Sim	Não	Não	Não
Localidade	VARCHAR(50)	Sim	Sim	Não	Não	Não
Cidade	VARCHAR(30)	Sim	Sim	Não	Não	Não

Tabela 3.1: Tabela de atributos da entidade Passageiro

3.4.2 Passe Mensal

Atributo	Domínio e Tamanho	Nulo	Composto	Multivalorado	Derivado	Ch. Candidata
ID	INT	Não	Não	Não	Não	Sim
Custo	DECIMAL(3,2)	Não	Não	Não	Sim	Não
Tipo	TIPOPASSE	Não	Não	Não	Não	Não

Tabela 3.2: Tabela de atributos da entidade Passe Mensal

3.4.3 Percurso

Atributo	Domínio e Tamanho	Nulo	Composto	Multivalorado	Derivado	Ch. Candidata
ID	INT	Não	Não	Não	Não	Sim
Duração	TIME	Não	Não	Não	Não	Não
Tipo	TIPOPASSE	Não	Não	Não	Não	Não

Tabela 3.3: Tabela de atributos da entidade Percurso

3.4.4 Viagem

Atributo	Domínio e Tamanho	Nulo	Composto	Multivalorado	Derivado	Ch. Candidata
Nr	INT	Não	Não	Não	Não	Sim
Custo	NUMERIC(3,2)	Não	Não	Não	Não	Não
Distancia	NUMERIC(3,2)	Não	Não	Não	Não	Não
Duração	TIME	Não	Não	Não	Não	Não
DataHora	DATETIME	Não	Não	Não	Não	Não

Tabela 3.4: Tabela de atributos da entidade Viagem

3.4.5 Paragem

Atributo	Domínio e Tamanho	Nulo	Composto	Multivalorado	Derivado	Ch. Candidata
ID	INT	Não	Não	Não	Não	Sim
Nome	VARCHAR(50)	Não	Não	Não	Não	Não
Rua	VARCHAR(100)	Não	Sim	Não	Não	Não
Localidade	VARCHAR(50)	Não	Sim	Não	Não	Não
Cidade	VARCHAR(30)	Não	Sim	Não	Não	Não
CodPostal	VARCHAR(8)	Não	Sim	Não	Não	Não
Latitude	DECIMAL(9,6)	Não	Sim	Não	Não	Não
Longitude	DECIMAL(9,6)	Não	Sim	Não	Não	Não

Tabela 3.5: Tabela de atributos da entidade Paragem

3.4.6 Motorista

Atributo	Domínio e Tamanho	Nulo	Composto	Multivalorado	Derivado	Ch. Candidata
ID	INT	Não	Não	Não	Não	Sim
Nome	VARCHAR(50)	Não	Não	Não	Não	Não
NIF	INT	Não	Não	Não	Não	Não
Rua	VARCHAR(100)	Não	Sim	Não	Não	Não
Localidade	VARCHAR(50)	Não	Sim	Não	Não	Não
Cidade	VARCHAR(30)	Não	Sim	Não	Não	Não
CodPostal	VARCHAR(8)	Não	Sim	Não	Não	Não
IBAN	VARCHAR(34)	Não	Não	Não	Não	Não
Salário	DECIMAL(6,2)	Não	Não	Não	Não	Não

Tabela 3.6: Tabela de atributos da entidade Motorista

3.4.7 Autocarro

Atributo	Domínio e Tamanho	Nulo	Composto	Multivalorado	Derivado	Ch. Candidata
ID	INT	Não	Não	Não	Não	Sim
Marca	VARCHAR(30)	Não	Não	Não	Não	Não
Modelo	VARCHAR(30)	Não	Não	Não	Não	Não
Matrícula	VARCHAR(6)	Não	Não	Não	Não	Não
Capacidade	INT	Não	Não	Não	Não	Não

Tabela 3.7: Tabela de atributos da entidade Autocarro

3.4.8 Neologismo TIOPASSE

Devido ao facto deste *datatype* ser extenso, para que todo o conteúdo da tabela fosse legível, criamos o seguinte neologismo:

Nome	Domínio e tamanho
TIOPASSE	ENUM ('Urbano', 'Intermunicipal')

Tabela 3.8: Tabela de atributos da entidade neologismo tipo-passe

3.5 Apresentação e explicação do diagrama ER produzido

Depois de definidas de forma detalhada cada uma das entidades, as relações entre as mesmas e os respetivos atributos, segue-se o modelo concetual produzido na sua integridade:

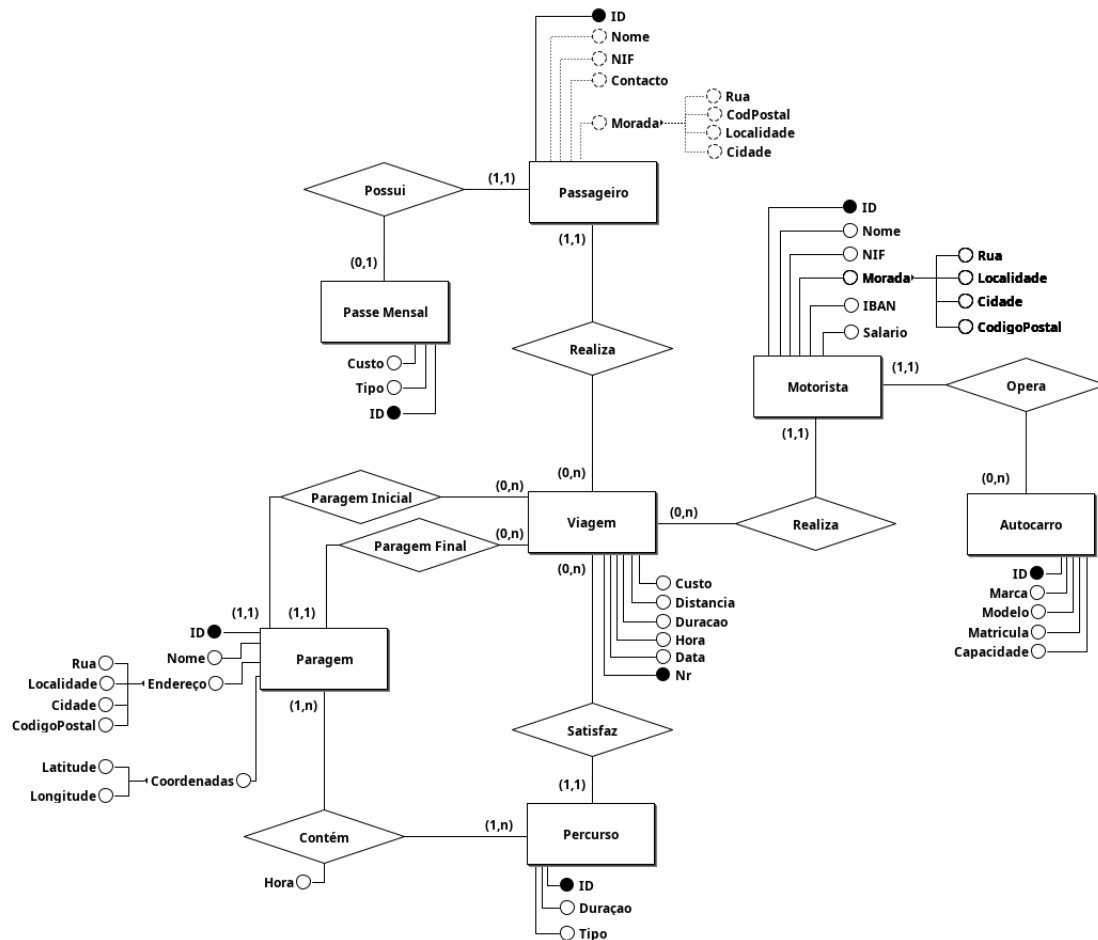


Figura 3.8: Diagrama ER produzido

Com o modelo concetual finalizado, temos uma melhor perceção da arquitetura que o sistema de base dados irá tomar e, quando devidamente validado, é possível prosseguir para o desenvolvimento do modelo lógico, uma nova modelação que nos irá conduzir à implementação física.

3.6 Análise e validação do modelo concetual produzido

De modo a validar o modelo concetual desenvolvido, a equipa de analistas confrontou a sua definição com os requisitos definidos anteriormente:

1. Cada **paragem** tem associado um id, um nome e a sua localização (endereço e coordenadas GPS).

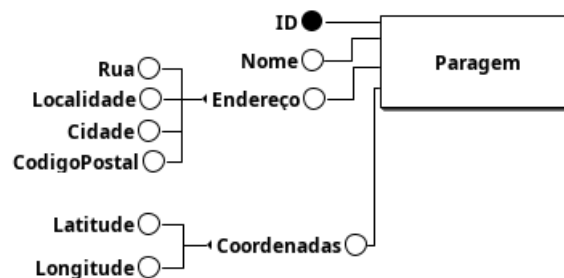


Figura 3.9: Ilustração da entidade Paragem

A entidade Paragem satisfaz o requisito, uma vez que inclui todos os parâmetros definidos.

2. Um **percurso** é composto por um id, um tipo (urbano ou intermunicipal), uma duração, um conjunto de paragens e a hora em que passa em cada paragem.

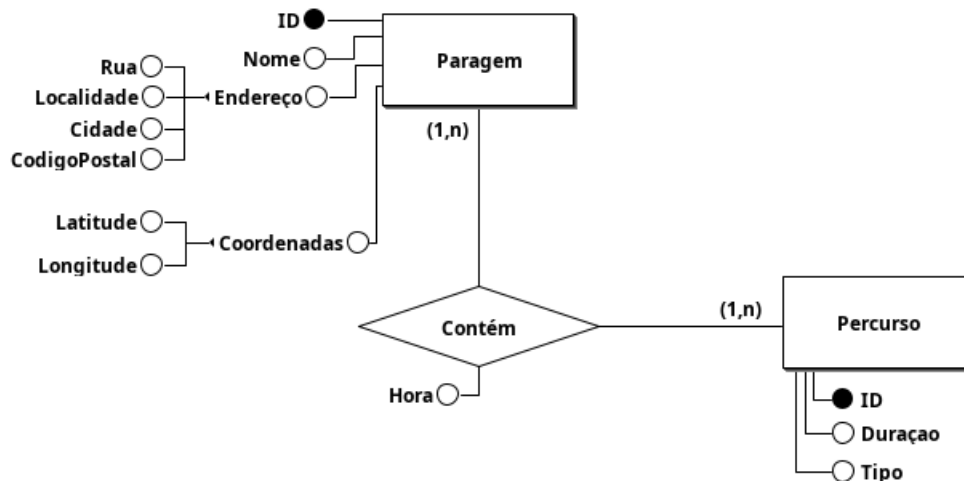


Figura 3.10: Ilustração da relação Percurso - Paragem

A entidade Percurso satisfaz também o requisito, uma vez que inclui todos os parâmetros definidos. O conjunto de paragens é assegurado pela relação Percurso - Paragem e a hora em que passa em cada paragem é assegurada pelo atributo "Hora" do relacionamento.

3. Uma paragem tem que estar incluída em um ou mais percursos.

Este requisito é também satisfeito pelo relacionamento Percurso - Paragem, observável na figura acima, onde a cardinalidade Percurso(1,n) assegura que uma paragem tem de estar associada a um ou mais percursos.

4. Uma viagem é constituída por um número, uma data, uma hora, uma duração, um percurso, uma paragem inicial e final, uma distância e tem um custo associado.

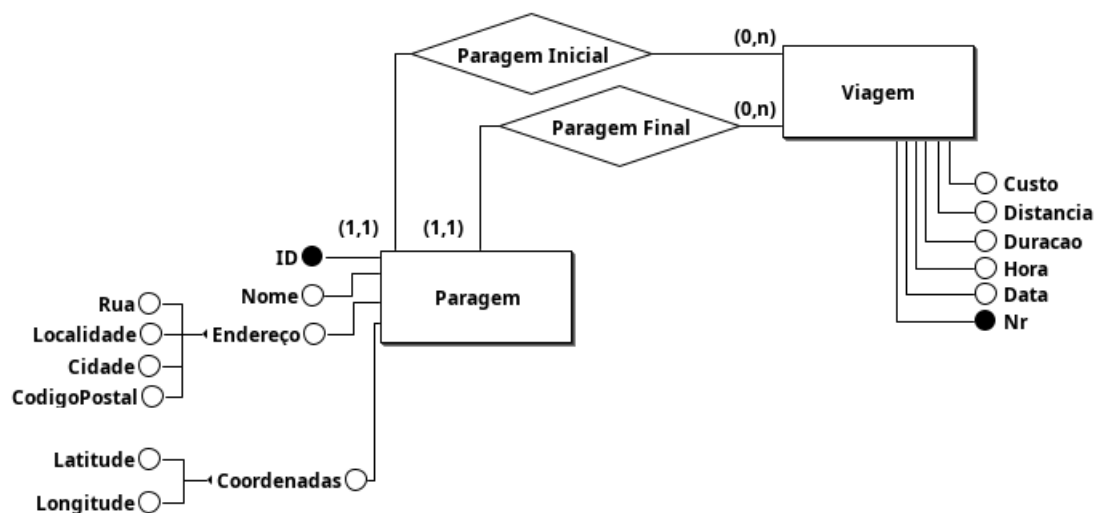


Figura 3.11: Ilustração da relação Viagem - Paragem

A entidade Viagem satisfaz o requisito, uma vez que inclui todos os parâmetros definidos. A paragem inicial e final associada a uma viagem é assegurada pelas relações ParagemInicial e ParagemFinal.

5. Uma viagem é realizada como parte de um percurso.

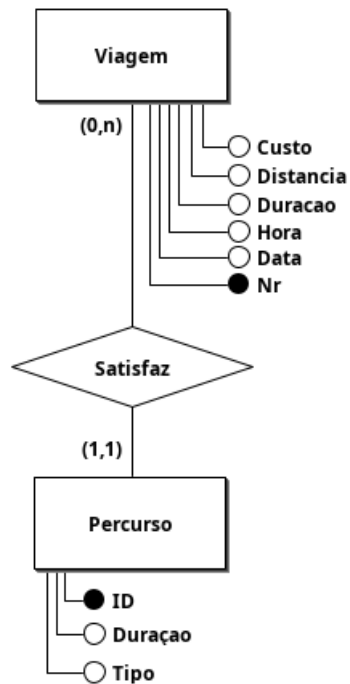


Figura 3.12: Ilustração da relação Viagem - Percurso

Este requisito é satisfeito pela relação Viagem - Percurso, a cardinalidade Percurso(1,1) assegura que a cada Viagem está obrigatoriamente associado um percurso.

6. Um passe mensal contém um id, um tipo de plano mensal (urbano ou intermunicipal) e um custo.

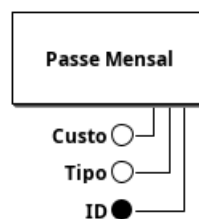


Figura 3.13: Ilustração da entidade Passe

A entidade Passe satisfaz o requisito, uma vez que inclui todos os parâmetros definidos.

7. Um passageiro possui um id e pode possuir um nome, um NIF, uma morada e um contacto.

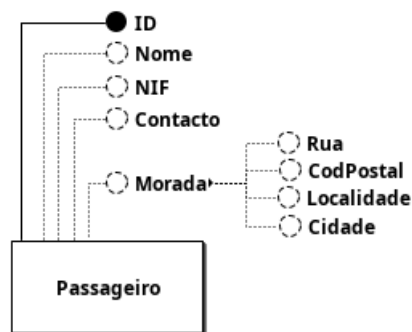


Figura 3.14: Ilustração da entidade Passageiro

A entidade Passageiro satisfaz o requisito, uma vez que inclui todos os parâmetros definidos. Os campos a tracejado asseguram a opcionalidade dos campos nome, NIF, morada e contacto.

8. Um passageiro pode ou não possuir um passe mensal.

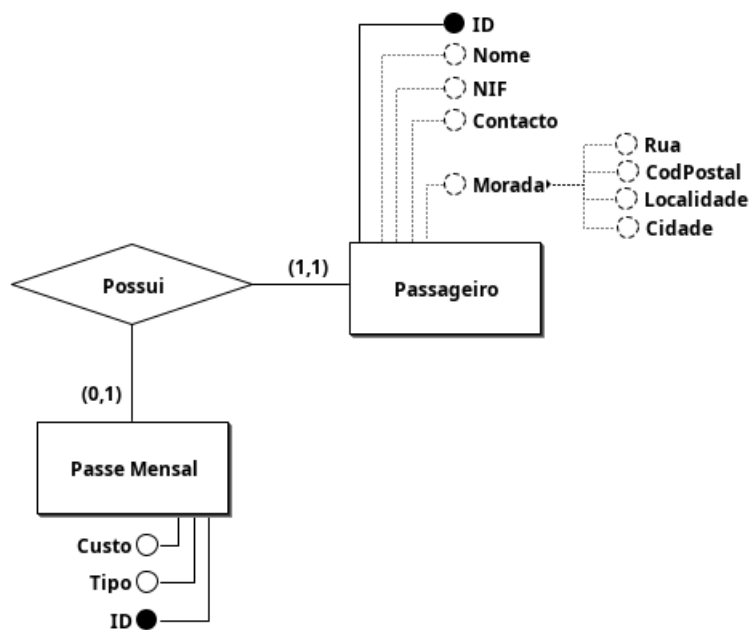


Figura 3.15: Ilustração da relação Passageiro - Passe Mensal

Este requisito é satisfeito pela relação Passageiro - Passe Mensal, a cardinalidade Passe-Mensal(0,1) assegura que um passageiro pode ou não possuir um passe e que pode possuir no máximo 1.

9. Um passageiro com passe mensal deve estar registado com as suas informações pessoais, para além do id.

A entidade passageiro inclui os atributos mencionados neste requisito como valores opcionais, precisamente pela natureza condicional dos mesmos. Sendo impossível representar num diagrama ER o facto de estes serem necessários apenas quando o passageiro possui um passe mensal, considera-se que o modelo satisfaz este requisito.

10. Uma viagem deve ser realizada por um passageiro através de um passe mensal ou bilhete.

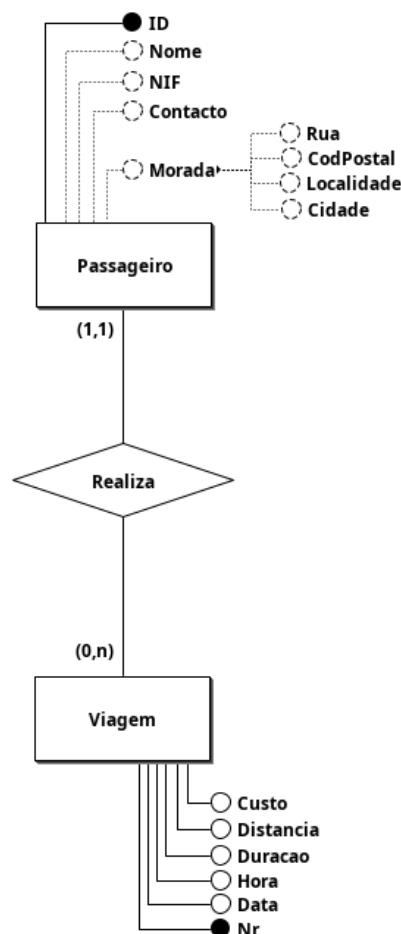


Figura 3.16: Ilustração da relação Passageiro - Viagem

A relação Passageiro - Viagem assegura que um passageiro pode realizar uma viagem. A forma como o passageiro tem acesso à viagem é já um requisito de controlo, que não é englobado pelo modelo concetual.

11. Um passageiro pode realizar diversas viagens.

A cardinalidade Viagem(0,n) da relação acima assegura que um passageiro pode, ou não, realizar uma ou mais viagens, pelo que o requisito é satisfeito.

12. Um autocarro é constituído por um id, uma marca, um modelo, uma matrícula e uma capacidade.

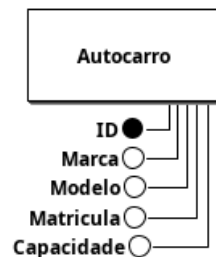


Figura 3.17: Ilustração da entidade Autocarro

A entidade Autocarro satisfaz o requisito, uma vez que inclui todos os parâmetros definidos.

13. O registo de um motorista inclui o seu nome, contacto, data de nascimento, morada, NIF, IBAN e o seu salário.

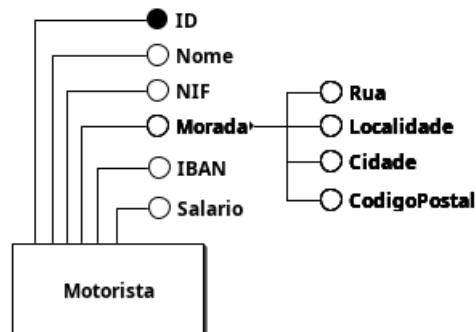


Figura 3.18: Ilustração da entidade Motorista

A entidade Motorista satisfaz o requisito, uma vez que inclui todos os parâmetros definidos.

14. Um motorista faz várias viagens, e pode conduzir diferentes autocarros.

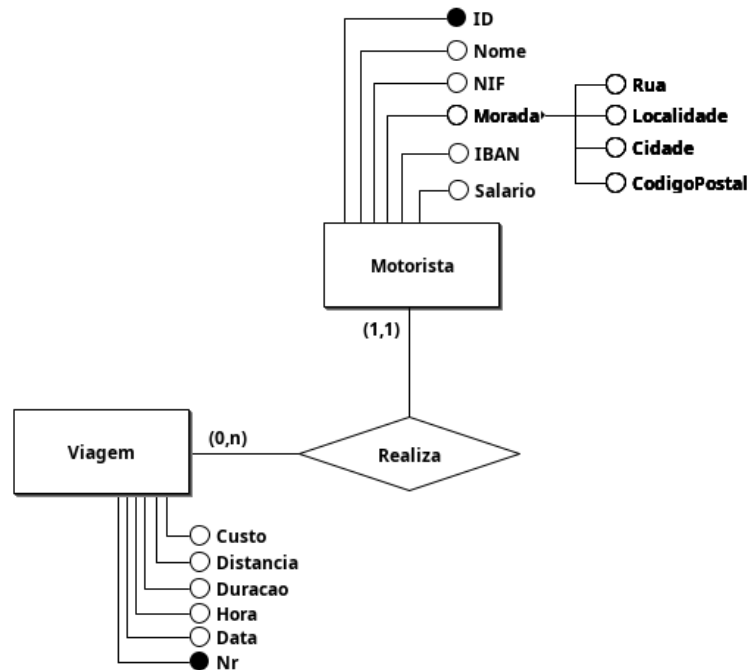


Figura 3.19: Ilustração da relação Motorista - Viagem

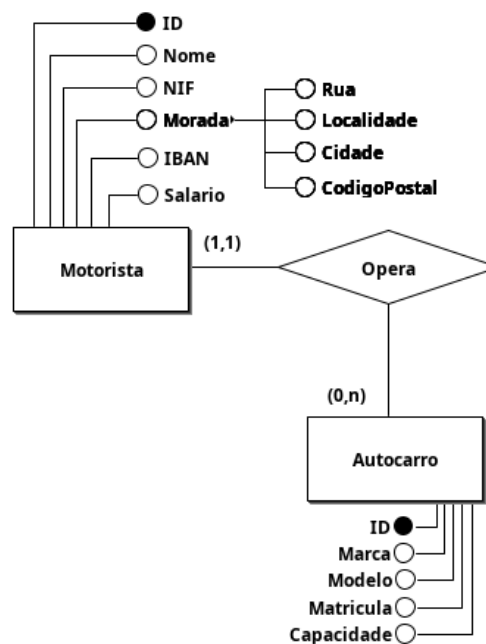


Figura 3.20: Ilustração da relação Motorista - Autocarro

O requisito é satisfeito pelas relações Motorista - Viagem e Motorista - Autocarro.

A cardinalidade Viagem(0,n) na relação Motorista - Viagem assegura que um motorista pode fazer diversas viagens.

A cardinalidade Autocarro(0,n) na relação Motorista - Autocarro assegura que um motorista pode conduzir diferentes autocarros.

-

Considerando que todos os requisitos estão de acordo com o modelo concetual desenvolvido, a equipa aprovou o modelo como final e avançou para a próxima fase de desenvolvimento, a modelação lógica.

4 Modelação Lógica

4.1 Construção e validação do modelo de dados lógico

A modelação lógica permite ilustrar de forma mais técnica e próxima do modelo físico a arquitetura da nossa base de dados. Assim, o nosso modelo lógico foi construído com base no modelo concetual, onde cada uma das entidades é traduzida numa tabela com todos os seus atributos, tal como os relacionamentos que tenham requisitos associados. Os atributos que no modelo concetual funcionavam como identificadores de uma entidade, são traduzidos no modelo lógico para *Primary Keys* ou *Foreign Keys*. *Foreign Keys* são identificadores de uma entidade que estão presentes na tabela de uma outra.

Assim, foi feita a tradução do modelo concetual para o modelo lógico, obtendo-se as seguintes tabelas, onde o ícone de chave preta representa uma chave primária e o ícone de chave verde uma chave estrangeira.

Viagem:








Viagem	
	Nr: INT
	DataHora: DATETIME
	Duracao: TIME
	Distancia: DECIMAL(3,2)
	Custo: DECIMAL(3,2)
	fk_Percurso: INT
	fk_Passageiro: INT
	fk_Motorista: INT
	fk_ParagemInicial: INT
	fk_ParagemFinal: INT
	

Figura 4.1: Ilustração da tabela da entidade Viagem

Passageiro:



Figura 4.2: Ilustração da tabela da entidade Passageiro

Motorista:



Figura 4.3: Ilustração da tabela da entidade Motorista

Paragem:

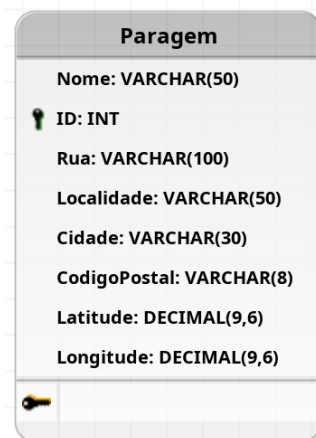


Figura 4.4: Ilustração da tabela da entidade Paragem

Percurso:

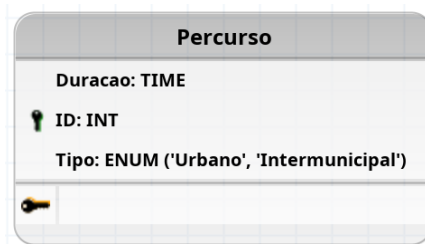


Figura 4.5: Ilustração da tabela da entidade Percurso

Passe Mensal:

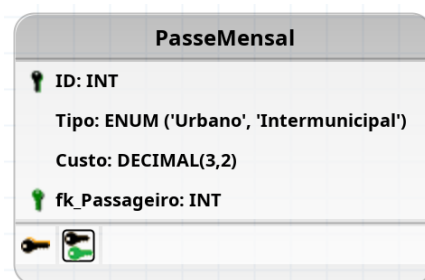


Figura 4.6: Ilustração da tabela da entidade Passe Mensal

Autocarro:



Figura 4.7: Ilustração da tabela da entidade Autocarro

Relacionamento Percurso 'Contem' Paragem:



Figura 4.8: Ilustração da tabela do relacionamento PercursoParagem

4.2 Normalização de Dados

A normalização de dados é um processo que visa organizar e estruturar uma base de dados para reduzir a redundância e aumentar a integridade dos dados. Este processo envolve a criação de tabelas e estabelecimento de relações entre elas, seguindo regras específicas chamadas "fórmulas normais". A normalização permite um armazenamento consistente e eficiente acesso aos dados, tornando a base de dados mais flexível e evitando inconsistências.

De modo a proceder com a normalização do nosso modelo, é essencial verificar se o mesmo satisfaz 3 regras, isto é, as 3 fórmulas normais:

A primeira fórmula normal (1FN) estabelece que todos os atributos de uma tabela devem ser atômicos, ou seja, a tabela não deve conter atributos repetidos nem atributos multivalorados. Para além disso, todas as tabelas devem conter uma chave primária.

Analisando o nosso modelo lógico, é possível confirmar que não existem atributos repetidos e que aqueles que eram multivalorados no Modelo Concetual foram decompostos nos seus constituintes, pelo que se valida 1FN.

A segunda fórmula normal (2FN) estabelece que uma tabela deve cumprir 1FN e que todos os atributos não-chave (atributos normais) devem depender unicamente da chave primária. Se uma tabela tiver uma chave primária composta, cada atributo normal deve depender de todas as partes da chave primária.

Tomando como exemplo a tabela do PasseMensal, por exemplo, verifica-se que o atributo *Tipo* está unicamente dependente da chave primária, *ID*. Isto acontece todos os restantes atributos do modelo. Tendo em conta que não temos qualquer chave primária composta, valida-se 2FN.

A terceira fórmula normal (3FN) estabelece que uma tabela deve cumprir 2FN e que não devem existir dependências transitivas entre os atributos normais. Isto significa que um atributo normal não deve depender de outro atributo normal.

No nosso modelo lógico, não existe qualquer atributo dependente ou proporcional a outro, pelo que se valida 3FN.

Assim, conclui-se que o nosso modelo lógico respeita as 3 fórmulas normais e se encontra, por isso, normalizado.

4.3 Apresentação e explicação do modelo lógico produzido

Criadas as tabelas e normalizados os dados, foi construído o Modelo Lógico final, que inclui as relações entre as diferentes tabelas, que são análogas às descritas no Modelo Concetual.

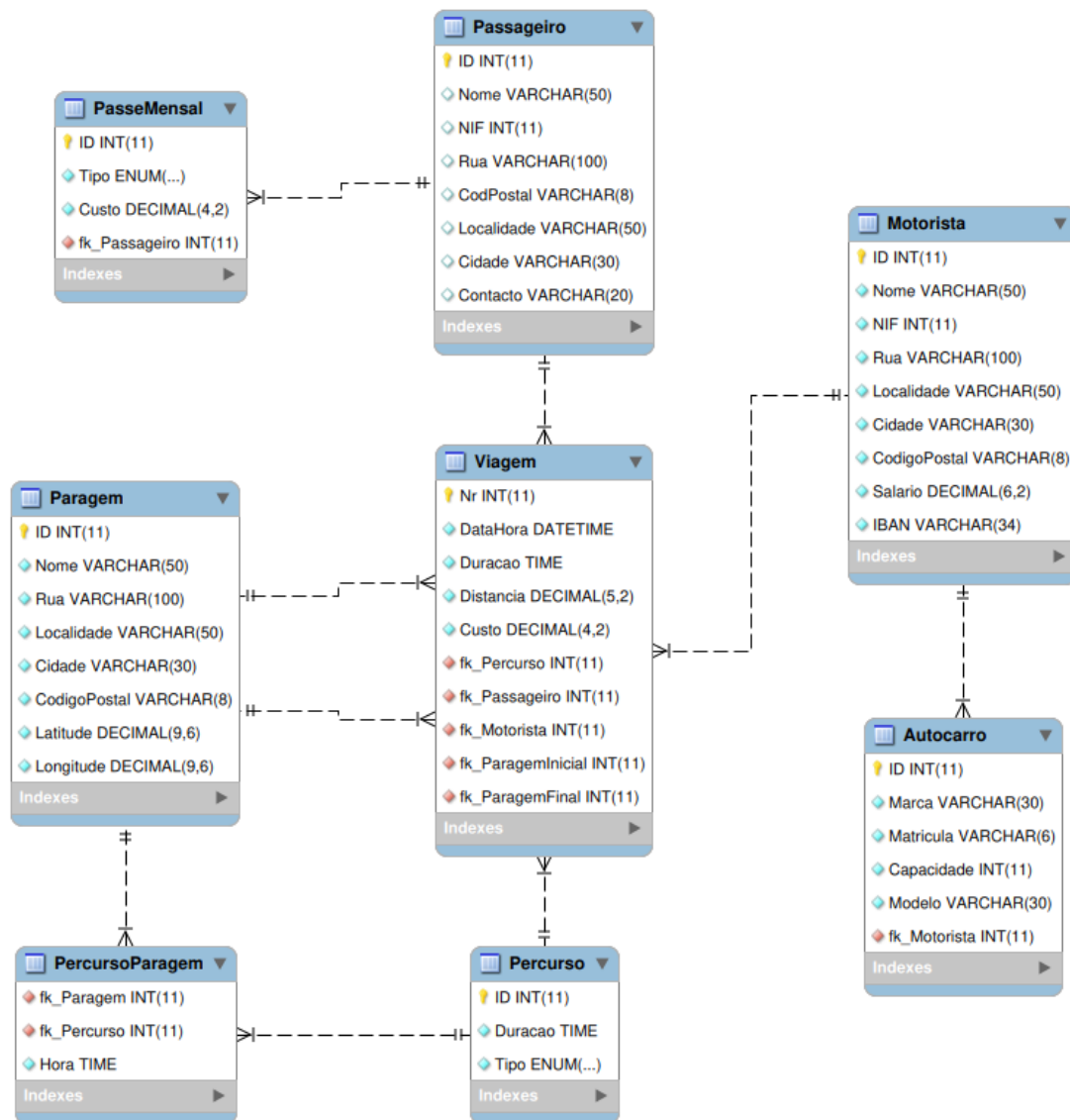


Figura 4.9: Modelo Lógico

4.4 Validação do modelo com interrogações do utilizador

De modo a validar o modelo lógico produzido, foi necessário comparar cada um dos requisitos de manipulação/exploração, definidos numa fase inicial do desenvolvimento, e verificar se o modelo lógico consegue dar resposta a cada um deles.

1. Para cada viagem deve ser calculada uma estimativa da quantidade de emissões de CO2 poupadas, através da sua distância:

Para cada viagem da tabela Viagem pode ser seleccionada a sua Distancia e calculada uma aproximação das emissões de CO2 que teria num autocarro a Diesel comum, através do produto de uma taxa de emissão de CO2 por quilómetro com a distância percorrida na viagem.

2. O total de CO2 poupado por dia deve ser contabilizado pelo sistema:

De forma análoga a 1., podemos calcular a emissão de CO2 de cada uma das viagens, efetuando adicionalmente um somatório das mesmas e agrupando os dados pela data incluída no atributo DataHora.

3. O total de CO2 poupado na operação da empresa deve ser contabilizado pelo sistema:

De forma análoga a 2., é calculado um somatório das emissões de CO2 de cada viagem. No entanto, os dados não são agrupados de modo a obter a poupança total de CO2.

4. As paragens com mais e menos percursos devem ser contabilizadas pelo sistema:

Através da tabela PercursoParagem podemos analisar todas as combinações de Percurso e Paragem existentes. Assim, realizando uma contagem da quantidade de percursos existentes e agrupando os dados por paragem, obtemos o número de percursos que passam em cada uma das paragens.

5. O sistema deve registar a quantidade de viagens efetuadas por cada passageiro:

Na tabela Viagem são registados todos os dados de cada uma das viagens efetuadas, podemos utilizar esses dados de modo a identificar quantas vezes um passageiro efetua viagens. Efetuando uma contagem das viagens existentes e agrupando os dados por passageiro, obtemos a quantidade de viagens realizadas por cada um dos passageiros.

6. Deve ser registada a quantidade de viagens realizadas por cada motorista:

Também aqui a tabela Viagem oferece todos os dados necessários. De forma análoga a 5., efetuando uma contagem das viagens existentes e agrupando os dados por motorista, obtemos a quantidade de viagens realizadas por cada motorista.

7. O total de viagens realizadas diariamente deve ser contabilizado pelo sistema:

Efetuando uma contagem das viagens existentes em Viagem e agrupando os dados pela data incluída no atributo DataHora, obtemos o número de viagens realizadas em cada um dos dias em que a empresa realizou pelo menos 1 viagem.

8. Para cada trajeto (Paragem Inicial → Paragem Final), deverá ser contabilizada a quantidade de passageiros que o realizaram:

A partir da tabela Viagem podemos efetuar uma contagem única de passageiros que efetuaram viagens. Agrupando estes dados pela combinação (ParagemInicial, ParagemFinal) obtemos a quantidade de passageiros que realizou cada um dos trajetos registados.

9. Para cada percurso deve ser calculado o lucro obtido através da venda de bilhetes:

Efetuando um somatório sobre o Custo de cada viagem registada na tabela Viagem e agrupando os dados por Percurso, obtemos o lucro que cada percurso obteve com a venda de bilhetes.

10. Deve ser registado o número de autocarros que estiveram em operação numa determinada data:

Embora os dados de uma viagem, na nossa arquitetura, não incluam nenhuma informação acerca do autocarro que a realizou, o nosso modelo lógico consegue responder a esta interrogação. Sabendo que cada Autocarro é alocado a um e só um Motorista e que cada Viagem é realizada por um e só um Motorista, efetuando uma contagem dos motoristas distintos que realizaram viagens e agrupando os dados obtidos pela data incluída no atributo DataHora, obtemos o número de autocarros em operação para cada dia, que é equivalente ao número de motoristas em operação em cada dia.

11. O sistema deverá ser capaz de contabilizar o lucro total obtido pela venda de passes e bilhetes:

Um simples somatório do Custo de cada viagem da tabela Viagem permite obter o lucro total obtido pela venda de bilhetes. Já para os passes mensais, um simples somatório do Custo

de cada passe da tabela `PasseMensal` permite obter o lucro total obtido pela venda de passes mensais, em cada mês.

—

Concluimos, assim, que o modelo lógico desenvolvido é válido e avançamos para a fase seguinte, a implementação física do modelo.

5 Implementação Física

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

A tradução do esquema lógico produzido para a implementação física em *MySQL Workbench*, SGBD escolhido, passou numa fase inicial pela utilização da opção "Converter para físico" do programa *brModelo*, utilizado no desenvolvimento do modelo concetual e do modelo lógico. Esta opção é capaz de gerar o código SQL necessário para a criação das tabelas da base de dados, conforme o esquema lógico produzido. Assim, copiamos este código para um *script* `.sql`.

O código SQL gerado pelo *brModelo* apresentava algumas inconsistências com aquilo a que fomos habituados ao longo das aulas teóricas de Base de Dados, pelo que foram feitos ajustes ao mesmo, principalmente no que toca à declaração de chaves estrangeiras, declaração de variáveis como não nulas e o adicionar da opção de incrementação automática nas chaves primárias (`auto_increment`). Para além disso, adicionamos os comandos `CREATE SCHEMA BelosTransportes` e `USE BelosTransportes` para a criação e seleção da base de dados, respetivamente.

Deste modo, conseguimos traduzir com sucesso o nosso esquema lógico para o *MySQL Workbench*.

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

Estando definida a forma como cada uma das interrogações do utilizador pode ser atendida pela base de dados (ponto 4.4), o processo de tradução destas para *queries* SQL foi relativamente simples. Seguem-se alguns exemplos:

a) Para cada viagem deve ser calculada uma estimativa da quantidade de emissões de CO2 poupadas, através da sua distância:

```
delimiter $$
CREATE PROCEDURE emissoesPoupadasPorViagem()
BEGIN
    SELECT Nr AS "Nº Viagem", fk_ParagemInicial AS "Origem", fk_ParagemFinal AS "Destino",
        ROUND(Distancia * 0.822, 2) AS "CO2 Poupado (Kg)"
    FROM Viagem
    ORDER BY ROUND(Distancia * 0.822, 2) DESC;
END $$
```

Figura 5.1: Código SQL relativo à interrogação 1

b) As paragens com mais e menos percursos devem ser contabilizadas pelo sistema:

```
delimiter $$
CREATE PROCEDURE nrPercursosPorParagem()
BEGIN
    SELECT fk_Paragem AS "ID Paragem", Paragem.Nome AS "Nome", Paragem.Cidade AS "Cidade",
        COUNT(fk_Percurso) AS "Nº Percursos"
    FROM PercursoParagem
    INNER JOIN Paragem ON fk_Paragem = Paragem.ID
    GROUP BY fk_Paragem
    ORDER BY "Nº Percursos" DESC;
END $$
```

Figura 5.2: Código SQL relativo à interrogação 4

c) O sistema deve registar a quantidade de viagens efetuadas por cada passageiro:

```
delimiter $$
CREATE PROCEDURE nrViagensPorPassageiro()
BEGIN
    SELECT fk_Passageiro AS "Passageiro", Passageiro.Nome AS "Nome", COUNT(Nr) AS "Viagens"
    FROM Viagem
    INNER JOIN Passageiro ON fk_Passageiro = Passageiro.ID
    GROUP BY fk_Passageiro
    ORDER BY fk_Passageiro;
END $$
```

Figura 5.3: Código SQL relativo à interrogação 5

d) Para cada trajeto (Paragem Inicial → Paragem Final), deverá ser contabilizada a quantidade de passageiros que o realizaram:

```
delimiter $$
CREATE PROCEDURE nrPassageirosPorTrajeto()
BEGIN
    SELECT fk_ParagemInicial AS "Origem", fk_ParagemFinal AS "Destino",
        COUNT(DISTINCT fk_Passageiro) AS "Nº Passageiros"
    FROM Viagem
    GROUP BY CONCAT(fk_ParagemInicial, fk_ParagemFinal)
    ORDER BY "Nº Passageiros" DESC;
END $$
```

Figura 5.4: Código SQL relativo à interrogação 8

e) Para cada percurso deve ser calculado o lucro obtido através da venda de bilhetes:

```
delimiter $$
CREATE PROCEDURE lucroObtidoPorPercurso()
BEGIN
    SELECT Percurso.ID AS "Percurso", Percurso.Duracao AS "Duração", SUM(Custo) AS "Lucro Total (€)"
    FROM Viagem
    INNER JOIN Percurso ON fk_Percurso = Percurso.ID
    GROUP BY Percurso
    ORDER BY Percurso;
END $$
```

Figura 5.5: Código SQL relativo à interrogação 9

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

As vistas de utilização são essenciais para uma rápida e prática visualização de dados relevantes. Seguem-se alguns exemplos das vistas criadas:

a) Visualização das paragens existentes, assim como os percursos que passam pelas mesmas e a que horas:

```
CREATE VIEW verParagens AS
SELECT Paragem.ID AS "Paragem", Nome, Cidade,
    Percurso.ID AS "Percurso", Hora
FROM PercursoParagem
INNER JOIN Paragem ON fk_Paragem = Paragem.ID
INNER JOIN Percurso ON fk_Percurso = Percurso.ID;
```

Figura 5.6: Código SQL relativo à vista de paragens

b) Visualização dos percursos existentes, assim como as paragens por quais estes passam e a que horas:

```
CREATE VIEW verPercursos AS
SELECT Percurso.ID AS "Percurso", Duracao, Tipo,
       Nome AS "Paragem", Hora
FROM PercursoParagem
INNER JOIN Percurso ON fk_Percurso = Percurso.ID
INNER JOIN Paragem ON fk_Paragem = Paragem.ID;
```

Figura 5.7: Código SQL relativo à vista de percursos

c) Visualização das viagens existentes, assim como o nome da paragem de origem, paragem de destino e do motorista:

```
CREATE VIEW verViagens AS
SELECT Nr, DataHora, Duracao, Distancia AS "Distancia (Km)", Custo AS "Custo (€)",
       fk_Percurso AS "Percurso", fk_Passageiro AS "Passageiro",
       PI.Nome AS "Origem", PF.Nome AS "Destino", Motorista.Nome AS "Motorista"
FROM Viagem
INNER JOIN Paragem AS PI ON fk_ParagemInicial = PI.ID
INNER JOIN Paragem AS PF ON fk_ParagemFinal = PF.ID
INNER JOIN Motorista ON fk_Motorista = Motorista.ID;
```

Figura 5.8: Código SQL relativo à vista de viagens

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

Sabendo o tamanho ocupado pelos diferentes tipos de dados utilizados nas tabelas da nossa base de dados, é possível realizar uma estimativa da sua ocupação em disco:

Passageiro:

Atributo	Tipo de Dados	Tamanho (bytes)
ID	INT	4
Nome	VARCHAR(50)	52
NIF	INT	4
Rua	VARCHAR(100)	102
CodPostal	VARCHAR(8)	10
Localidade	VARCHAR(50)	52
Contacto	VARCHAR(20)	22
Total		246

Tabela 5.1: Cálculo do tamanho da tabela Passageiro

PasseMensal:

Atributo	Tipo de Dados	Tamanho (bytes)
ID	INT	4
Tipo	ENUM ('Urbano', 'Intermunicipal')	1
Custo	DECIMAL(4,2)	3
fk_Passageiro	INT	4
Total		12

Tabela 5.2: Cálculo do tamanho da tabela PasseMensal

Paragem:

Atributo	Tipo de Dados	Tamanho (bytes)
ID	INT	4
Nome	VARCHAR(50)	52
Rua	VARCHAR(100)	102
Localidade	VARCHAR(50)	52
Cidade	VARCHAR(30)	32
CodigoPostal	VARCHAR(8)	10
Latitude	DECIMAL(9,6)	6
Longitude	DECIMAL(9,6)	6
Total		264

Tabela 5.3: Cálculo do tamanho da tabela Paragem

Percurso:

Atributo	Tipo de Dados	Tamanho (bytes)
ID	INT	4
Duracao	TIME	3
Tipo	ENUM ('Urbano', 'Intermunicipal')	1
Total		8

Tabela 5.4: Cálculo do tamanho da tabela Percurso

PercursoParagem:

Atributo	Tipo de Dados	Tamanho (bytes)
fk_Paragem	INT	4
fk_Percurso	INT	4
Hora	TIME	3
Total		11

Tabela 5.5: Cálculo do tamanho da tabela PercursoParagem

Motorista:

Atributo	Tipo de Dados	Tamanho (bytes)
ID	INT	4
Nome	VARCHAR(50)	52
NIF	INT	4
Rua	VARCHAR(100)	102
Localidade	VARCHAR(50)	52
Cidade	VARCHAR(30)	32
CodigoPostal	VARCHAR(8)	10
Salario	DECIMAL(6,2)	4
IBAN	VARCHAR(34)	36
Total		296

Tabela 5.6: Cálculo do tamanho da tabela Motorista

Autocarro:

Atributo	Tipo de Dados	Tamanho (bytes)
ID	INT	4
Marca	VARCHAR(30)	32
Matricula	VARCHAR(6)	8
Capacidade	INT	4
Modelo	VARCHAR(30)	32
fk_Motorista	INT	4
Total		84

Tabela 5.7: Cálculo do tamanho da tabela Autocarro

Viagem:

Atributo	Tipo de Dados	Tamanho (bytes)
Nr	INT	4
DataHora	DATETIME	8
Duracao	TIME	3
Distancia	DECIMAL(5,2)	4
Custo	DECIMAL(4,2)	3
fk_Percurso	INT	4
fk_Passageiro	INT	4
fk_Motorista	INT	4
fk_ParagemInicial	INT	4
fk_ParagemFinal	INT	4
Total		42

Tabela 5.8: Cálculo do tamanho da tabela Viagem

Total:

Tabela	Tamanho (bytes)
Passageiro	246
PasseMensal	12
Paragem	264
Percurso	8
PercursoParagem	11
Motorista	296
Autocarro	84
Viagem	42
Total	963

Tabela 5.9: Cálculo do tamanho total da BD

Deste modo, estima-se que o tamanho da nossa base de dados, quando não povoada, seria de aproximadamente **963 bytes**.

Através da análise e consulta de dados de funcionamento prévios da *Belos Transportes* poderíamos realizar uma estimativa do tamanho inicial da base de dados e estudos estatísticos acerca da previsão de crescimento da empresa dariam os dados necessários para estimar o tamanho futuro da BD. Segue-se um exemplo do cálculo do tamanho inicial da base de dados, tendo em conta os dados descritos no script `dados.sql`, desenvolvido como parte da implementação física:

$$10 * 246 + 10 * 12 + 10 * 264 + 10 * 8 + 20 * 11 + 10 * 296 + 10 * 84 + 20 * 42 = 10160 \text{ bytes} = 10,16 \text{ KB}$$

Esta mesma fórmula seria aplicada para o número atual de passageiros, autocarros, motoristas, paragens, etc. que a *BelosTransportes* contém.

5.5 Análise da Implementação Física

Finalizada a implementação física do nosso sistema de base de dados, podemos constatar que este consegue responder a todos os requisitos definidos e que ocupa um espaço razoável em disco, estando por isso preparado para o eventual crescimento da empresa.

6 Implementação do Sistema de Recolha de Dados

6.1 Apresentação e modelo do sistema

Nesta fase foi necessária a criação de um *parser*, onde o nosso objetivo é, através de ficheiros CSV, conseguir inserir os registos capturados nas tabelas da base de dados.

O sistema de recolha de dados desempenha um papel importante no projeto, permitindo a organização e armazenamento dos dados de forma estruturada e prática. O sistema utiliza *Python* na sua implementação, realizando a leitura dos arquivos CSV e inserindo os dados nas tabelas correspondentes na base de dados. Deste modo, é facilitada a inserção de dados no SGBD, sem que seja necessária a codificação manual dos mesmos.

```
Nome,NIF,Rua,Localidade,Cidade,CodigoPostal,Salario,IBAN
André Silva,123456789,Rua dos Pinheiros,Barcelos,Barcelos,4750-123,1500.00,PT50001000012345678901234
Marta Ferreira,987654321,Avenida Central,Braga,Braga,4700-456,1200.50,PT50001000098765432101234
Pedro Santos,654321987,Rua das Flores,Braga,Braga,4700-678,1400.75,PT50001000065432198701234
Sofia Costa,789456123,Rua do Lago,Braga,Braga,4700-890,1100.25,PT50001000078945612301234
Rui Oliveira,159753468,Rua dos Cedros,Barcelos,Barcelos,4750-456,1300.50,PT50001000015975346801234
Carla Pereira,753159852,Avenida da Liberdade,Braga,Braga,4700-123,1250.00,PT50001000075315985201234
João Fernandes,852963741,Rua do Sol,Barcelos,Barcelos,4750-789,1350.25,PT50001000085296374101234
Ana Rodrigues,369852147,Rua dos Castanheiros,Braga,Braga,4700-234,1150.75,PT50001000036985214701234
Manuel Gomes,214365879,Avenida das Flores,Braga,Braga,4700-567,1450.50,PT50001000021436587901234
Lúcia Almeida,741258963,Rua da Praia,Barcelos,Barcelos,4750-012,1050.25,PT50001000074125896301234
```

Figura 6.1: Exemplo de ficheiro de CSV do motorista

6.2 Implementação e funcionamento do sistema de recolha

A implementação do nosso sistema de recolha de dados passou por utilizar a biblioteca `csv`, que lê ficheiros CSV, e a biblioteca `mysql.connector`, que realiza a conexão ao *MySQL*.

O código estabelece a conexão com a base de dados, fornecidas as informações do *host*, *user* e *password*. De seguida, são definidas diversas funções, responsáveis por inserir os dados obtidos em cada uma das tabelas da base de dados.

Cada função de inserção é utilizada para lidar com um tipo específico de dados. Por exemplo, a função `inserir_dados_viagem()` insere informações sobre uma viagem na tabela "Viagem", enquanto que a função `inserir_dados_motorista()` insere dados sobre um motorista na tabela "Motorista". Essas funções recebem os valores de cada atributo como parâmetros e executam consultas SQL para inserir os dados nas tabelas correspondentes.

Cada arquivo CSV é aberto e é lida cada uma das suas linhas utilizando o leitor CSV. Dependendo do nome do ficheiro, os dados são processados e inseridos nas tabelas correspondentes, utilizando as funções de inserção respetivas. Após cada inserção, a função `db.commit()` é chamada para confirmar as alterações na base de dados.

Deste modo, o nosso sistema de recolha automatiza o processo de importação de dados de ficheiros CSV para a base de dados, permitindo assim que os dados sejam armazenados e posteriormente consultados ou analisados através das diferentes *queries* implementadas.

De seguida apresentamos uma das funções referidas acima, responsável por inserir dados na tabela Motorista:

```
def inserir_dados_motorista(nome,nif,rua,localidade,cidade,codPostal,salario,iban):
    query = "INSERT INTO Motorista (Nome, NIF, Rua, Localidade, Cidade, CodigoPostal, Salario, IBAN) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
    values = (nome,nif,rua,localidade,cidade,codPostal,salario,iban)
    cursor.execute("SET FOREIGN_KEY_CHECKS=0")
    cursor.execute(query, values)
    cursor.execute("SET FOREIGN_KEY_CHECKS=1")
```

Figura 6.2: Função `insere_dados_motorista()`

7 Implementação do Sistema de Painéis de Análise

7.1 Definição e caracterização da vista de dados para análise

O método utilizado para obtermos uma melhor vista de dados para análise foi o uso da ferramenta *PowerBI* da *Microsoft*. Esta ferramenta permite explorar, transformar e visualizar todos os dados da base de dados de forma simples e eficaz. Ou seja, conseguimos escolher as tabelas e definir as relações entre as tabelas que desejamos analisar. A ferramenta oferece uma variedade de visualizações, como gráficos, tabelas, mapas, etc. que podem ser personalizadas de acordo com as necessidades do utilizador.

7.2 Povoamento das estruturas de dados para análise

Em geral, o processo de povoamento das estruturas de dados envolve a conexão com a base de dados e, de seguida, a importação das tabelas da base de dados *MySQL* para os modelos de dados do *PowerBI*.

A conexão é realizada de forma direta com o servidor, referindo o endereço do servidor e o *SCHEMA* que queremos utilizar, de seguida introduzimos as nossas credenciais de autenticação (*user* e *password*), estabelecendo uma conexão privada e segura com a base de dados. Assim que a conexão é estabelecida, o *PowerBI* permite seleccionar as tabelas que desejamos utilizar para a análise.

7.3 Apresentação e caracterização dos dashboards implementados

O *PowerBI* permite ilustrar estatísticas sobre os registos da nossa base de dados de uma forma mais abstrata e simples, isto é, no formato de *dashboards*. De seguida apresentam-se alguns exemplos dos *dashboards* implementados:

a) Visualização do número de passageiros a utilizar a *BelosTransportes* por dia

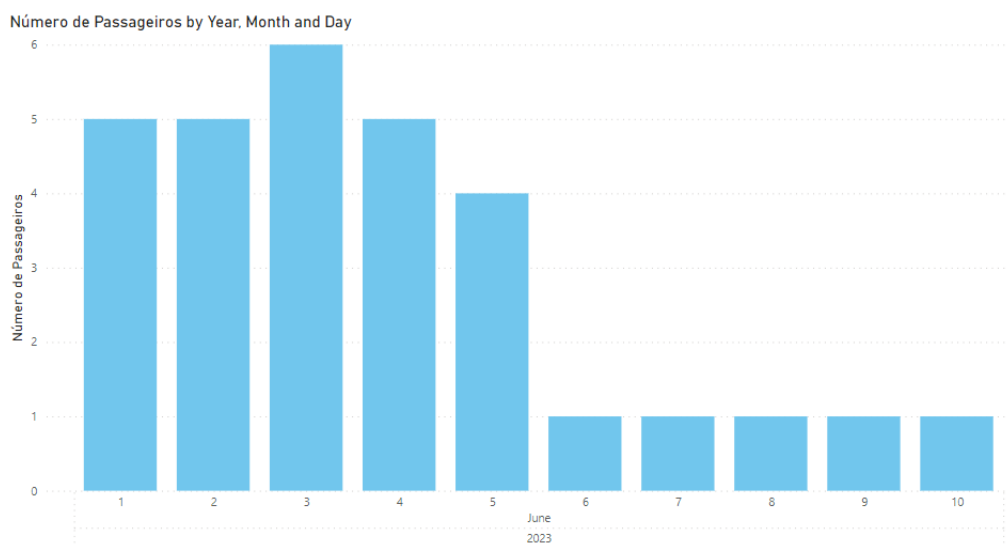


Figura 7.1: Gráfico de passageiros por dia

b) Visualização da percentagem de passageiros pertencentes a cada uma das cidades

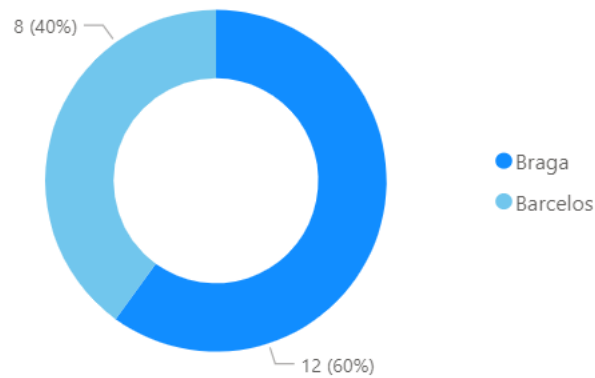


Figura 7.2: Gráfico da percentagem de passageiros por cidade

c) Visualização das paragens com mais utilização

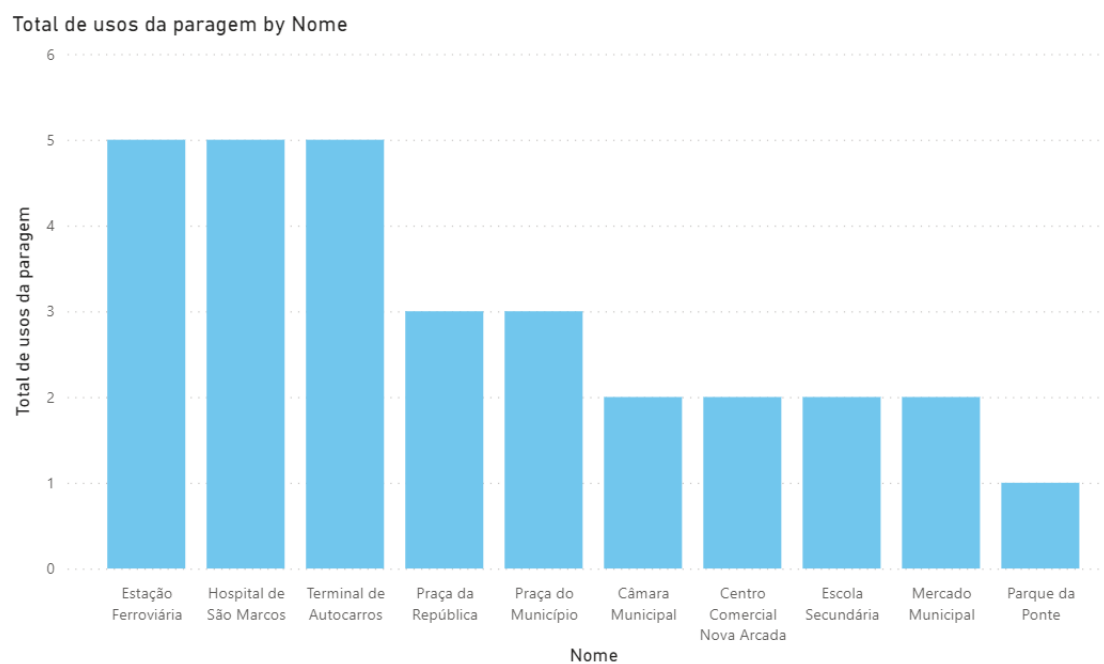


Figura 7.3: Gráfico das paragens mais usadas

8 Conclusões e Trabalho Futuro

Numa vista global, consideramos o trabalho bem conseguido, com uma complexidade suficientemente desafiante perante os objetivos pretendidos de se atingir na UC.

Num trabalho futuro, poderíamos melhorar alguns aspetos do trabalho desenvolvido que, na nossa opinião, poderiam ter sido mais explorados, tais como:

- A escolha do tema foi realizada de um modo algo vago, o que levou a retificações aos requisitos bem já adentro do desenvolvimento do esquema concetual, algo possivelmente indesejável numa situação profissional de arquitetura duma BD e que deveria ser melhorado.
- Como possível expansão da base de dados, poder-se-ia abranger outros tipos de funcionários da empresa que não os motoristas, podendo assim obter também um registo de informações de qualquer empregado num único sistema mais generalista. Não se prevê muita dificuldade em executar isto - a cada motorista já se associa o seu salário, por exemplo, o que induz um ponto de partida para uma definição duma entidade mais genérica de "Funcionário" que não ficaria propriamente descabida.
- Adicionalmente, e, de novo usando como ponto de partida a inclusão de salários na base de dados, conseguir-se-ia uma expansão na vertente da contabilidade, anuais, etc. Tal implicaria a inclusão de vários outros gastos, tais como aquisição/venda/manutenção de novos veículos na frota, seguros e combustível ou até a inclusão de diferentes frotas de veículos.

Tal como estes exemplos, também outras entidades poderiam ser adicionadas de modo a aumentar a complexidade e utilidade do SGBD.

Lista de Siglas e Acrónimos

BD Base de Dados

SGBD Sistema de Gestão de Bases de Dados

ID Identificador

GPS Global Positioning System

NIF Número de Identificação Fiscal

IBAN International Bank Account Number

ER Entidade Relação

Anexos

Diagrama de Gantt

Modelo Concetual

Modelo Lógico

Sistema de Recolha de Dados