



universidade de aveiro

# TrainsDB

Universidade de Aveiro

Licenciatura em Engenharia Informática

Bases de Dados

**Docentes:**

Prof. Carlos Costa

Prof. Tiago Godinho

**Trabalho realizado por:**

Guilherme Antunes - 103600

# Índice

Introdução .....	3
Análise de Requisitos .....	4
Diagrama Entidade/Relacionamento .....	5
Esquema Relacional .....	6
Diagrama da Base de Dados .....	7
SQL DDL e Inserções .....	8
Triggers .....	9
Stored Procedures .....	9
User Defined Functions .....	10
Segurança .....	10
Implementação .....	10
Vídeo .....	10
Conclusão .....	11

# Introdução

No seguimento do projeto final proposto na unidade curricular de Bases de Dados, este relatório tem como objetivo expor a organização da base de dados construída bem como os conhecimentos adquiridos na unidade curricular que foram postos em prática na realização do projeto.

A escolha do tema recaiu sobre uma base de dados para uma empresa de transporte ferroviário de passageiros. A base de dados está planeada para permitir aos clientes realizar reservas de viagens mediante a sua estação de origem, destino, data e hora bem como aos funcionários validarem as reservas efetuadas por cada passageiro durante as viagens.

# Análise de Requisitos

Este trabalho partiu de uma simplificação daquilo que são bases de dados dos serviços relacionados ao tema escolhido

As entidades que estão previstas de interagir com o sistema são **Cliente** e **Funcionario** que podem ser abstraídos numa entidade geral **Pessoa** que conterà informações pessoais básicas como o Nome da pessoa, a Data de Nascimento e o número de Cartão de Cidadão que serve como identificador de cada **Pessoa**. Por sua vez a entidade **Cliente** contém mais informações como o seu e-mail e opcionalmente o número de telefone. Um **Funcionario** terá apenas extra a uma **Pessoa** um número de funcionário.

O **Funcionario** tem como função validar as reservas de um **Comboio**, sendo que este pode validar mais do que um. Um **Comboio** por sua vez pode ser validado por mais do que um **Funcionario** e é caracterizado por pelo seu tipo (Regional, Urbano, Alfa, etc), pela sua **Rota** e pelo seu identificador.

Um **Comboio** é constituído por entidades **Carruagem** que tem como atributos a sua classe (1ª, 2ª, única/nenhuma), o seu identificador, o seu número relativo ao comboio a que pertence, e o identificador do **Comboio** a que pertence.

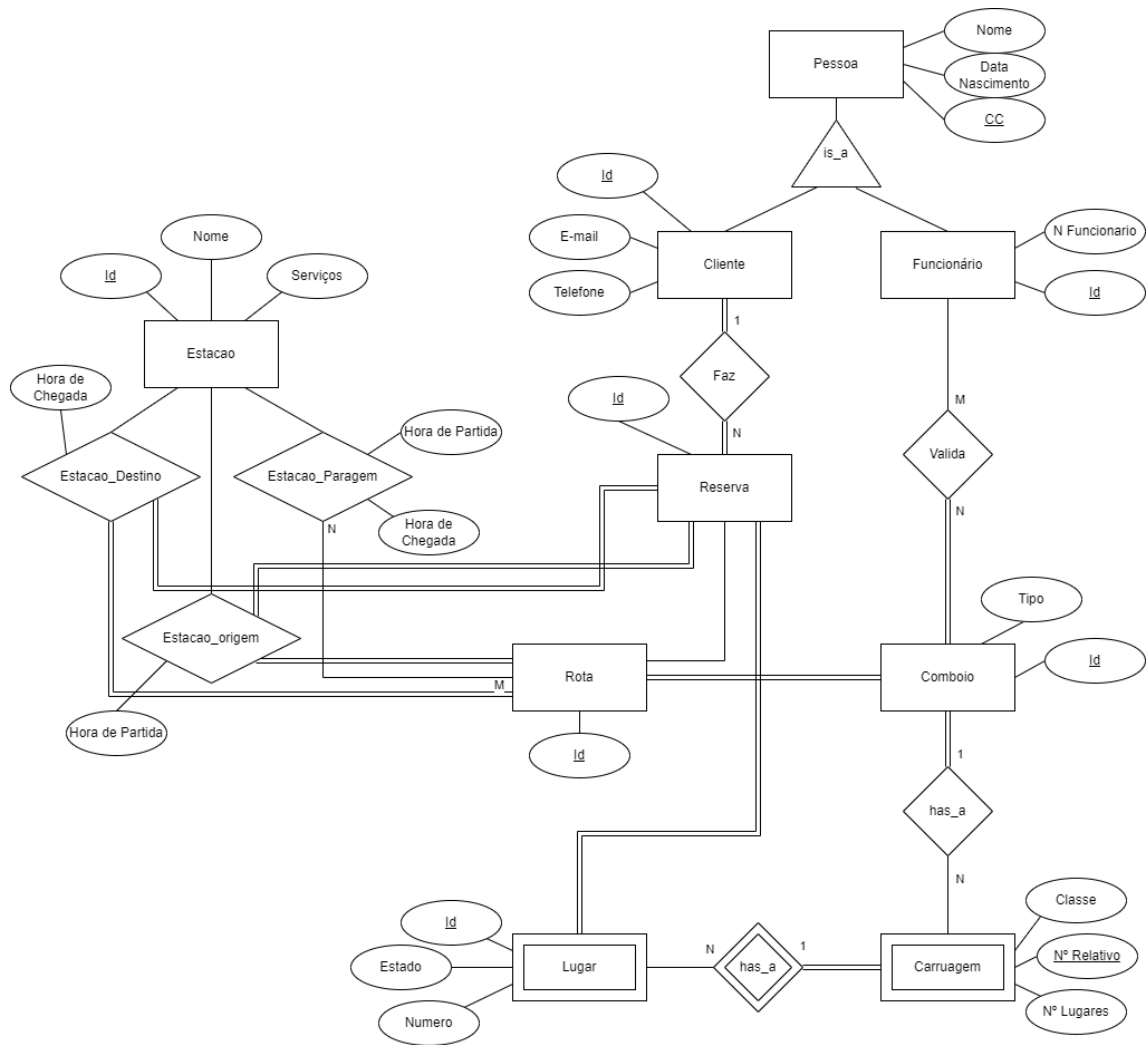
Por sua vez cada **Carruagem** é constituída por vários **Lugar**, sendo que pode ter diferentes quantidades de entidades deste tipo referenciadas a ela. **Lugar** é caracterizado pelo seu identificador, por Lugar que corresponde ao número relativo à carruagem a que pertence, o seu estado, que é alterado mediante a reserva ou cancelamento da reserva para esse mesmo **Lugar** e a referência ao identificador da carruagem a que pertence.

Como referido cada **Lugar** pode ser reservado. Esta reserva pode ser feita pelo **Cliente** e é registada pela entidade **Reserva**. **Reserva** para além do seu identificador possui como atributos a referência ao **Lugar** reservado, referência à **Rota** efetuada pelo comboio respetivo e a referência ao **Cliente** que a efetuou.

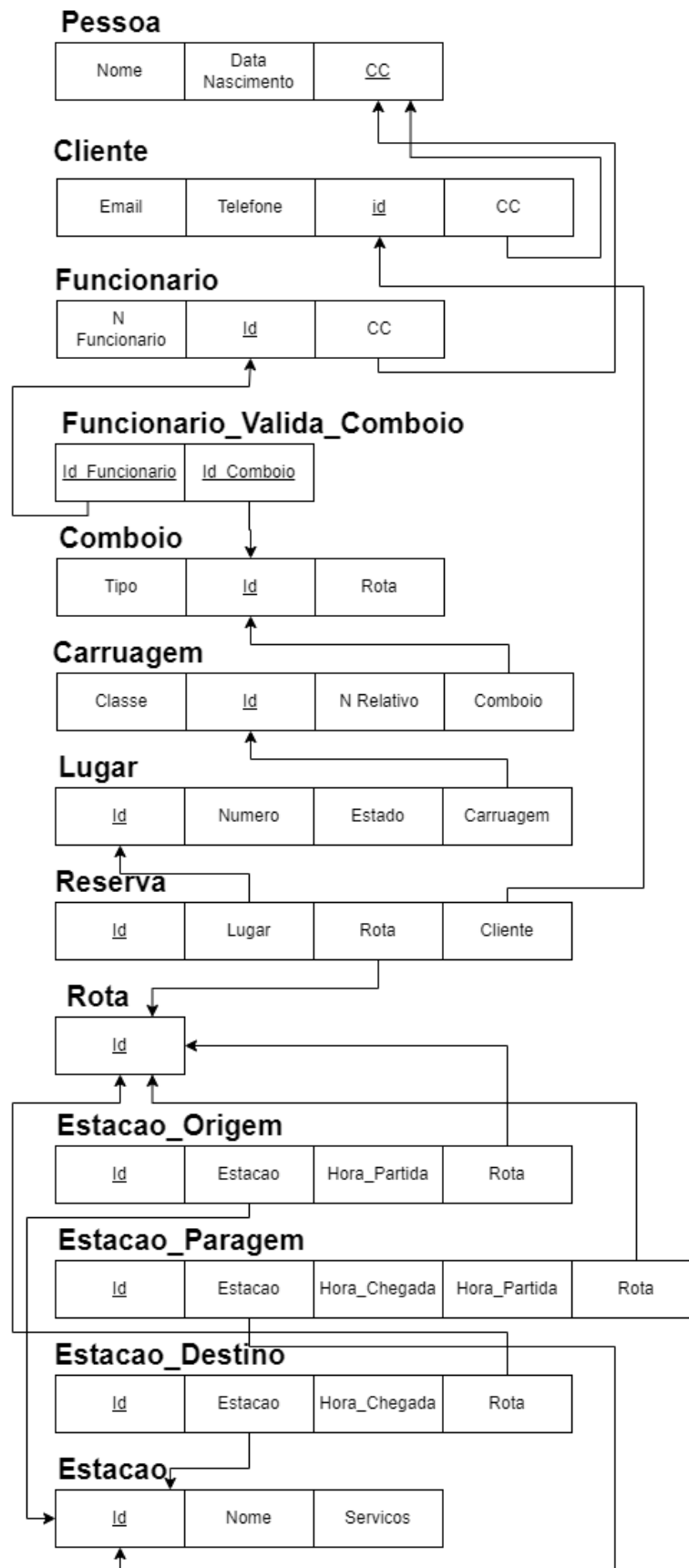
A já referida **Rota** é apenas caracterizada pelo seu identificador e a ela fazem referência as instâncias da entidade **Estacao**, nomeadamente as instâncias **Estacao\_Origem**, **Estacao\_Paragem**, e **Estacao\_Destino**. Como referido todas estas fazem referência à rota a que pertencem, bem como à entidade da qual são instanciadas e têm também o seu identificador. **Estacao\_Paragem** têm ainda os atributos hora\_chegada e hora\_partida enquanto **Estacao\_Destino** tem apenas hora\_chegada e **Estacao\_Origem** tem apenas hora\_partida.

A última entidade é a também já referida **Estacao** que é caracterizada pelo seu identificador, nome e serviços de que dispõe.

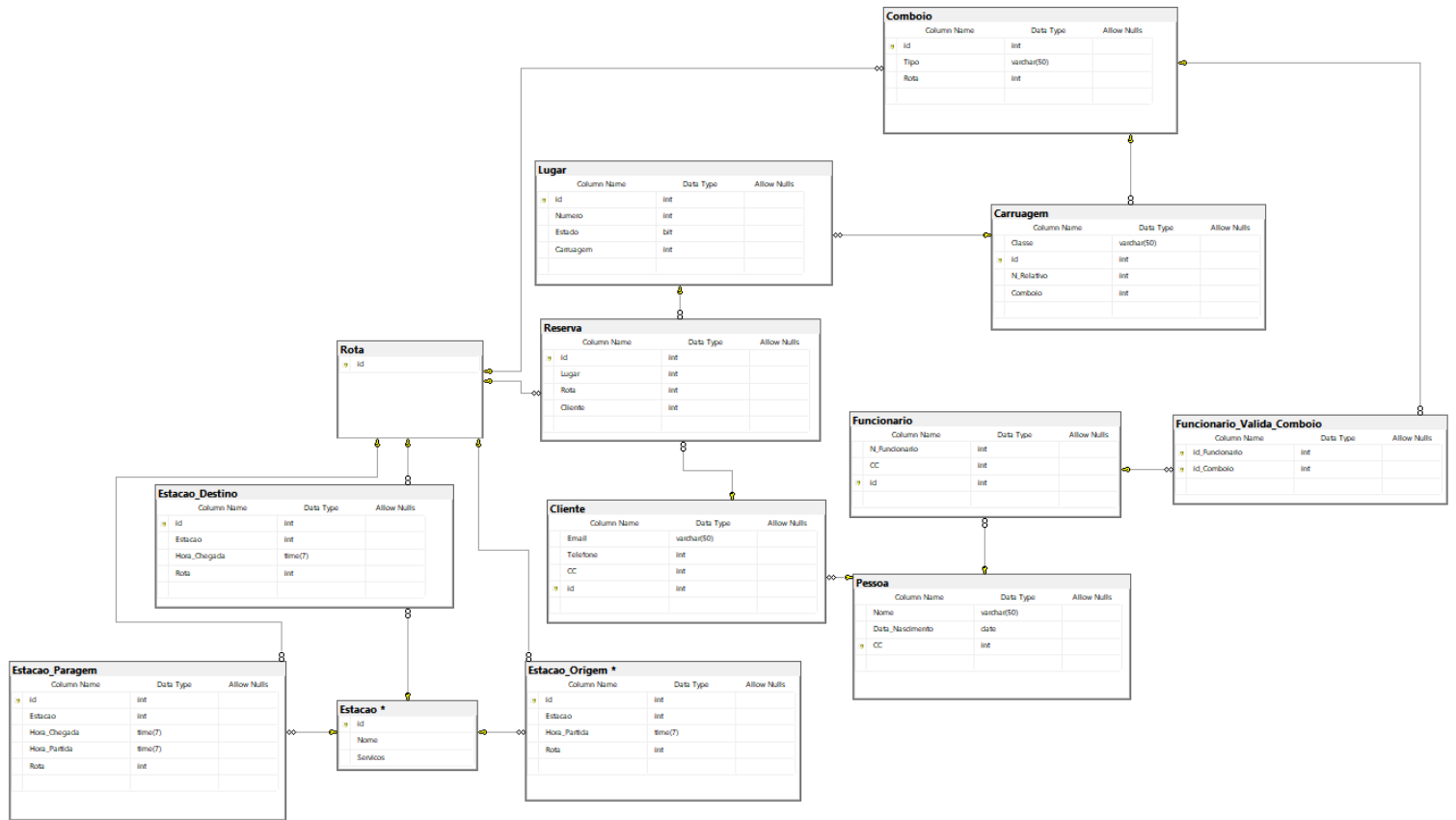
## Diagrama Entidade/Relacionamento



# Esquema Relacional



## Diagrama da Base de Dados



## SQL DDL e Inserções

De forma a criar as entidades referidas na Análise de Requisitos foram definidas as tabelas correspondentes usando DDL e executando na base de dados onde também foram definidas as chaves primárias de cada entidade e as chaves estrangeiras para as outras entidades. Também foram definidos **CHECKS** para garantir regras básicas das tabelas assim como foram limitados os dados que podiam ter o valor **NULL**. Este código pode ser encontrado no ficheiro *SQL DDL.sql*.

Foram também populares as tabelas com DML de maneira a poder realizar todas as funções a que a Base de Dados se propunha. Este código pode ser encontrado no ficheiro *Inserções.sql*.

Referir também alguns exemplos de dados a inserir no formulário para ter acesso às restantes funcionalidades desenvolvidas:

- Cliente:

- *lmosdall0@people.com.cn*
- *ogapper1@wordpress.org*
- *gmacgahy2@privacy.gov.au*

- Funcionario:

- *57723596*
- *79293723*
- *22186001*



## Triggers

Apenas foram utilizados triggers **ON UPDATE** e **ON DELETE CASCADE** convenientemente na declaração das tabelas pois não surgiu a necessidade de criar outros durante a implementação das restantes funcionalidades conseguidas.

## Stored Procedures

Foram criados SP's para validação de formulários e para inserção de novos dados na base de dados tendo sido criados 4 e implementados 3. O código referente a estes pode ser encontrado no ficheiro *SP.sql*.

Os Procedures desenvolvidos foram:

- CheckEmail

Procedure que verifica se o email de um cliente é válido para Log In, devolve 0 em caso negativo e devolve 1 em caso afirmativo, mediante a receção de um valor de tipo Text correspondente ao email inserido.

- Registrar

Procedure responsável por registar na base de dados novas entidades do tipo Cliente e consequentemente Pessoa.

- CheckCC

Procedure responsável por verificar se o número de CC inserido já está registado para outro utilizador ou se pode ser usado no registo atual.

- Search

Procedure responsável por verificar se as estações inseridas para efetuar uma nova reserva são válidas, dado assim autorização para prosseguir no processo de reserva.

## User Defined Functions

Não foram implementadas UDF's pois todas as verificações feitas foram aplicadas via Stored Procedures.

## Segurança

De maneira a minimizar vulnerabilidades da base de dados foram tidos em conta:

- Verificação dos dados introduzidos pelos utilizadores, apresentando mensagens de erro em caso de inserção de dados incorretos
- Foram utilizados ao máximo Stored Procedures em vez de SQL Dinâmico de maneira a tornar as operações mais fiáveis

## Implementação

De maneira a tentar colocar em prática as funcionalidades desenvolvidas para a base de dados foi desenvolvido um conjunto de formulários Windows na linguagem C# usando o IDE Visual Studio 2022.

O formulário funciona principalmente à base de TextBoxes e Buttons que verificam a conexão ao servidor e chamam os stored procedures criados para o efeito. Na existência de informação mal inserida são chamadas MessageBoxes a fim de anunciar o erro.

## Vídeo

O vídeo demonstrativo da execução do formulário e das funcionalidades implementadas segue no seguinte link:

<https://1drv.ms/u/s!AkletAPR59npiKUgKf0CdQXl65WmBg?e=ylzGpk>

## Conclusão

Após a “conclusão” do trabalho e entrega do mesmo desfecho que o trabalho carece de implementações importantes e relevantes que não foi possível implementar atempadamente e que isso impede o funcionamento total de um sistema que teria todas as condições para realizar um trabalho totalmente bem-sucedido.