

## ***OnTrack* - Sistema de Gestão e Registo da Localização de Automóveis e Camiões**

47147

André Gonçalves Jardim

Orientadores    Nuno Datia

Relatório de primeira fase realizado no âmbito de Sistemas de Informação,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2021/2022

Maio de 2022



# Resumo

Este projeto tem como objetivo criar um sistema para gerir e registar a localização de automóveis e camiões para a empresa *OnTrack* aplicando os conceitos estudados em aula.

O sistema deve guardar informação de clientes e veículos. Cada veículo deve ter associado um conjunto de zonas verdes. Quando um veículo sai das suas zonas verdes deve ser gerado um alarme.

**Palavras-chave:** base de dados; consulta; funções; procedimentos armazenados; *triggers*.



# Abstract

The goal of this project is to create a system to manage and save the location of cars and trucks for the company *OnTrack*.

The system should save information about clients and vehicles. Each vehicle should be associated with a group of green zones. When a vehicle leaves its green zones an alarm should be generated.

**Keywords:** databases; functions; query; stored procedures; triggers.



# Índice

<b>RESUMO .....</b>	<b>III</b>
<b>ABSTRACT .....</b>	<b>V</b>
<b>LISTA DE FIGURAS .....</b>	<b>VIII</b>
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1 ORGANIZAÇÃO DO DOCUMENTO .....	1
<b>2. FORMULAÇÃO DO PROBLEMA .....</b>	<b>2</b>
2.1 REQUISITOS DA SOLUÇÃO .....	2
<b>3. SOLUÇÃO PROPOSTA.....</b>	<b>3</b>
3.1 MODELO DE DADOS .....	3
3.2 MODELO ER .....	5
3.3 RESTRIÇÕES DE INTEGRIDADE.....	6
<b>4. AVALIAÇÃO EXPERIMENTAL .....</b>	<b>8</b>
4.1 CRIAÇÃO DO MODELO FÍSICO .....	8
4.2 EXECUÇÃO DOS TESTES.....	8
4.3 INTEGRIDADE DOS DADOS .....	9

# Lista de Figuras

Figura 1 - Modelo ER da solução apresentada.....	5
--	---





# **1. Introdução**

Este projeto visa criar um sistema de informação para a empresa *OnTrack*, que pretende gerir e registar a localização de automóveis.

## **1.1 Organização do documento**

Nos próximos capítulos o problema é analisado e é proposta uma solução.

## **2. Formulação do Problema**

Neste capítulo é feita uma análise dos requisitos da solução apresentada.

### **2.1 Requisitos da Solução**

O sistema desenvolvido deve guardar dados de clientes, que podem ser particulares ou institucionais e veículos. Cada cliente tem uma frota de veículos.

Cada veículo tem um equipamento de GPS (que tem um estado associado) e poderá ter associado um conjunto de zonas verdes, onde pode circular livremente.

Os veículos enviam informação acerca da sua localização a cada 10 segundos; esta informação deve ser guardada em registos não processados.

Os registos não processados devem ser processados a cada 5 minutos. Se o registo for inválido deve ser movido para uma tabela de registos inválidos, onde deve ficar durante 15 dias. Se o registo for válido deve ser movido para uma tabela de registos processados.

Quando um registo é movido para a tabela de registos processados, deve ser gerado um alarme se o veículo estiver fora das suas zonas verdes e se o estado do equipamento de GPS não for “PausaDeAlarme”.

## 3. Solução Proposta

Neste capítulo é apresentada a nossa solução para o problema.

### 3.1 Modelo de Dados

O grupo começou por desenvolver um modelo de dados adequado ao problema em questão, que é apresentado de seguida:

CLIENTE(nif: integer, nome: varchar(60), morada: varchar(60), telefone: integer, ref\_cliente: integer, removido: bit, tipo: char)

CLIENTE\_PARTICULAR(nif\_cliente: integer, cc: integer)

CLIENTE\_INSTITUCIONAL(nif\_cliente: integer, nome\_contacto: varchar(60))

FROTA\_VEICULOS(id: integer, nif\_cliente: integer)

ESTADOS\_EQUIPAMENTOS(estado: varchar(20))

EQUIPAMENTO\_GPS(id: integer, estado: varchar(20))

VEICULO(matricula: varchar(8), nome\_cond\_atual: varchar(60), telef\_cond\_atual: integer, id\_equip: integer, id\_frota: integer, n\_alarmes: integer)

ZONA\_VERDE(id: integer, latitude: decimal(7, 5), longitude: decimal(8, 5), raio: integer, matricula: varchar(8))

REGISTO\_NAO\_PROCESSADO(id\_reg: integer, id\_equip: integer, marca\_temporal: timestamp, latitude: decimal(7, 5), longitude: decimal(8, 5))

REGISTO\_PROCESSADO(id\_reg: integer, marca\_temporal\_proc: timestamp, id\_equip: integer, latitude: decimal(7, 5), longitude: decimal(8, 5))

REGISTO\_INVALIDO(equip\_id: integer, marca\_temporal\_inval: timestamp, latitude: decimal(7, 5), longitude: decimal(8, 5))

ALARME(id\_reg: integer, matricula: varchar(8), marca\_temporal: timestamp)

Em CLIENTE, *ref\_cliente* é uma chave estrangeira para o cliente que referenciou o cliente que pode ser null; *removido* é um bit que está a '1' se o cliente foi removido; *tipo* pode tomar os valores 'P' se o cliente é particular ou 'I' se o cliente é institucional.

Para ser possível adicionar mais estados aos equipamentos de GPS, foi criada a tabela ESTADOS\_EQUIPAMENTOS que guarda os possíveis estados dos equipamentos e em EQUIPAMENTO\_GPS, *estado* é uma chave estrangeira para ESTADOS\_EQUIPAMENTOS.

Em ZONA\_VERDE, *matricula* é uma chave estrangeira que referencia o veículo ao qual a zona verde pertence.

Em REGISTO\_PROCESSADO, *equip\_id* é uma chave estrangeira para o equipamento de GPS que gerou o registo.

Em ALARME, *id\_reg* é uma chave estrangeira que referencia o registo processado que originou o alarme e *matricula* é uma chave estrangeira para o veículo que saiu das suas zonas verdes.

## 3.2 Modelo ER

Em seguida é apresentado o modelo ER da solução implementada.

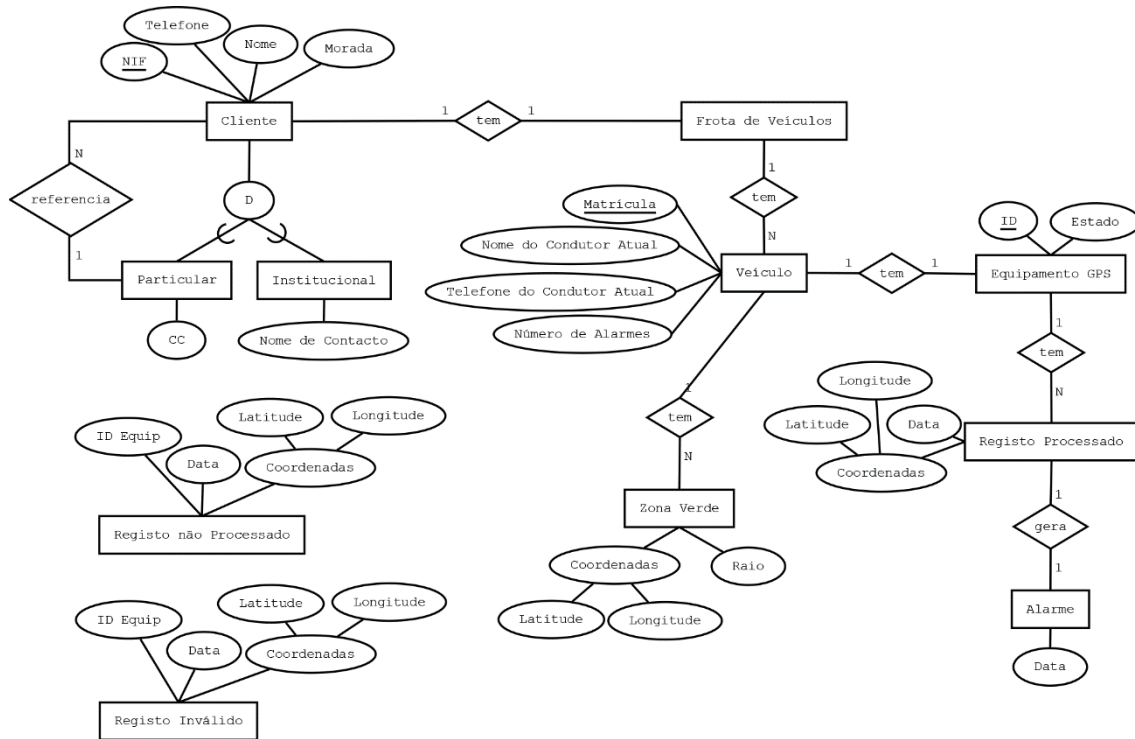


Figura 1 - Modelo ER da solução apresentada

### 3.3 Restrições de Integridade

É necessário garantir as seguintes restrições de integridade para o correto funcionamento do sistema:

1. Cada cliente tem associada 1 (e apenas uma) frota de veículos.
2. As datas (marcas temporais) têm precisão ao segundo.
3. O estado dos equipamentos pode ser {'Activo', 'PausaDeAlarmes', 'Inactivo'}.
4. Deve ser possível adicionar novos estados para os equipamentos.
5. O processamento dos registos tem de ser efetuado em lote a cada 5 minutos.
6. Os registos que tenham identificadores de GPS inexistentes, que não tenham data ou coordenadas devem ser movidos para os registos inválidos.
7. Os registos inválidos são removidos após 15 dias.
8. Os clientes particulares têm um máximo de 3 veículos.
9. Tem de ser suportada a remoção de clientes sem os remover da base de dados.
10. O número de alarmes de um veículo é atualizado sempre que é gerado um alarme.
11. O raio das zonas verdes é expresso em metros.





## 4. Avaliação Experimental

Para verificar o correto funcionamento da solução apresentada, o grupo realizou testes aos pontos mais importantes da solução.

### 4.1 Criação do Modelo Físico

Para criar o modelo físico devem ser executados os *scripts* pela seguinte ordem:

1. createTables.sql
2. createViews.sql
3. createFunctions.sql
4. createProcedures.sql
5. createTriggers.sql
6. insertRows.sql

### 4.2 Execução dos Testes

Para executar os testes criados, é necessário criar o modelo físico utilizando os passos descritos na secção anterior e executar o *script* “tests.sql”. Se um teste falhar é lançada uma exceção.

### 4.3 Integridade dos Dados

Para manter a integridade dos dados, o grupo utilizou transações com o nível de isolamento adequado nos testes efetuados.

De um modo geral,

- Onde se fazem vários *inserts/updates* seguidos, foi utilizado o nível de isolamento *repeatable read* (ex: inserção/atualização de clientes) dado que as instruções dependem das anteriores e é necessário que as alterações feitas num ponto não sejam alteradas antes de inserir/atualizar todos os registos necessários.
- Onde se fazem *deletes*, foi utilizado o nível de isolamento *read committed* para que não ocorram *dirty reads*.
- Foi utilizado o nível de isolamento *serializable* onde são efetuadas operações que não podem ser intercaladas com outras, por exemplo para verificar se entre uma instrução e outra foram criados/removidos X registos, não pode haver outra instrução que crie/remova um registo do mesmo tipo.

Após a execução de cada teste é efetuado *rollback*, para que as alterações não fiquem guardadas na base de dados.