ISEL

ADEETC

Área Departamental de Engenharia Electrónica e de Telecomunicações e de Computadores

Licenciatura em Engenharia Informática e de Computadores

Sistemas de Informação

Trabalho Prático

Parte 2

Autores:

Diogo Alexandre Ferreira de Jesus	48302
Miguel José Reys e Sousa Marmelete	48260
Henrique Valente Mareco Ferreira Águas	48929

Grupo 6 - Turma 41D

24/04/2022

Verão 2021 / 2022

Resumo

Neste relatório encontram-se informações acerca da realização da primeira fase do Trabalho Prático da disciplina de Sistemas de Informação.

Entre estas informações incluem-se os Modelos de Dados adoptados (Conceptual e Relacional) com explicações de conceitos que achámos relevante mencionar e a explicação dos mecanismos utilizados para a realização dos exercícios requiridos.

Abstract

In this report informations about the first phase of the practical assignment of Information Systems subject can be found.

Among this are included the Data Models (Conceptual and Relational) with explanations for decisions we though were worth explaining. We also explain the mecanisms we used in the development of this phase's exercises.

Table of Contents

Lista de Figuras

1 Objetivos do Trabalho e Enunciado

No final da primeira fase do trabalho, os alunos deverão ser capazes de:

- Desenvolver um modelo de dados adequado aos requisitos, normalizado até à 3NF;
- Conceber e implementar uma solução baseada em bases de dados dinâmicas, adequada aos requisitos;
- Utilizar corretamente controlo transacional;
- Utilizar corretamente níveis de isolamento;
- Utilizar corretamente vistas;
- Utilizar corretamente procedimentos armazenados;
- Utilizar corretamente gatilhos;
- Utilizar corretamente funções;
- Saber justificar o uso na solução apresentada de vistas, procedimentos armazenados, gatilhos e funções.
- Desenvolver código de teste, em PL/pgSQL, para cada uma das funcionalidades requeridas nos requisitos;
- Desenvolver código PL/pgSQL para criar todos os objetos necessários à solução, a partir de uma base de dados vazia;
- Escrever um relatório técnico sobre as decisões tomadas e o trabalho desenvolvido.

O tema do trabalho é acerca de uma empesa “OnTrack”, para a qual, ao longo das Fases 1 e 2, vamos desenvolver um sistema capaz de gerir e registar a localização de automóveis e camiões.
Esta empresa pretende guardar informações acerca dos seus clientes, que podem ser particulares ou institucionais.
Cada cliente tem uma frota de veículos, sendo que cada cliente particular pode ter, no máximo, 3 veículos na sua frota. Os clientes institucionais não têm qualquer limitação no número de veículos da sua frota.
Clientes particulares podem referenciar outros clientes (particulares ou institucionais).
A remoção de clientes da base de dados não significa remover os dados associados ao cliente.

Os veículos possuem um equipamento de GPS. A cada 10 segundos o equipamente emite um registo que é armazenado na base de dados. O equipamento GPS pode ter 3 estados, sendo possível adicionar outros estados através da futura aplicação.
Cada veículo tem Zonas Verdes associadas. Quando o veículo sai destas zonas verdes um alarme é criado.

Os registos emitidos pelos equipamentos GPS de cada veículo vão para uma tabela de registos não processados, migrando depois para os registos processados ou registos inválidos, caso algo não esteja correto nestes registos.

O enunciado completo do trabalho encontra-se no *Anexo 1*.

2 Modelos de Dados

2.1 Conceptual

Após a leitura do enunciado identificámos as seguintes entidades:
Nota: Algumas das seguintes foram criadas por conveniência e não por interpretação direta do enunciado;

- Cliente
    - Cliente Particular
    - Cliente Institucional
- Veículo
- Zona Verde
- GPS
- Estados GPS
- Registo
    - Registo processado
    - Registo não processado
    - Registo inválido
- Alarme

Na *Figura 1* encontra-se o modelo de dados conceptual desenvolvido de acordo com o enunciado, presente no *Anexo 1*, onde ficam evidentes as relações entre as entidades enumeradas acima.

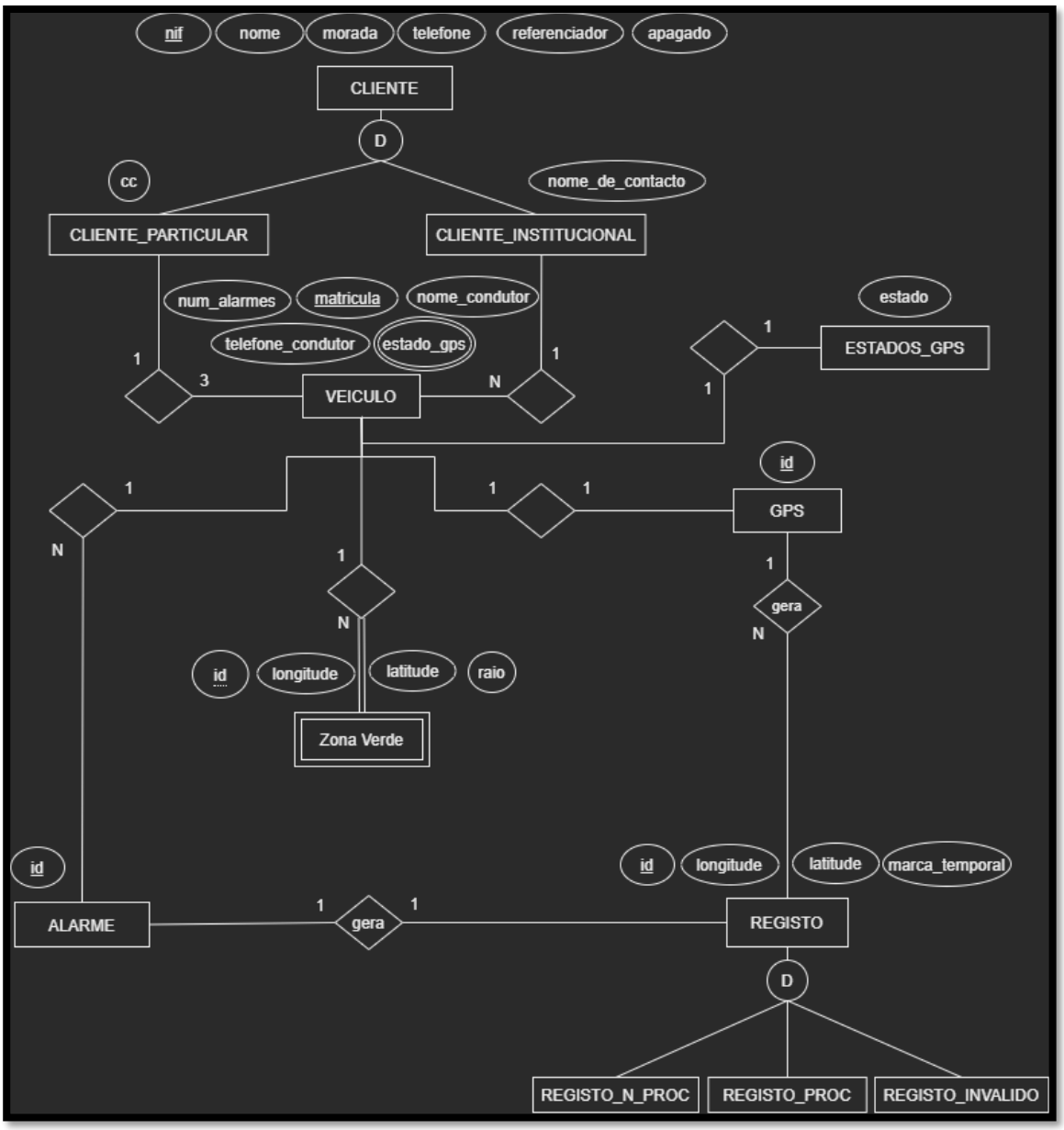


Figura 1. Modelo de Dados Conceptual

2.2 Relacional

O modelo relacional é uma representação do modelo de dados mais aproximada às tabelas em SQL.
É também neste modelo que se encontram as restrições de integridade que não conseguimos garantir até então e devem ser tidas em conta nas camadas superiores.
O modelo relacional encontra-se no *Anexo 3.*

3 Justificação do Modelo de Dados Adotado

Neste capítulo vamos abordar de forma breve algumas das decisões tomadas na criação do modelo de dados.

3.1 Remoção de um Cliente

O enunciado revela que no ato de apagar um utilizador os dados relativos ao mesmo não devem ser apagados.
Por forma a cumprir este requisito decidimos adicionar uma coluna à tabela *Cliente*, **apagado** que, caso esteja a True indica que o utilizador foi removido apesar de os seus dados continuarem disponíveis.

3.2 Estados de GPS

Inicialmente pensamos que cada veículo poderia ter um atributo multi-valor que representasse o estado do equipamento GPS a ele associado mas, para permitir introdução de novos estados de equipamento para os GPS pensámos que fosse conveniente a criação de uma nova tabela com uma coluna apenas.
Nesta tabela cada linha representa um estado possível para os equipamentos GPS. Desta forma os veículos têm uma coluna denominada **estado_gps** que faz referência a uma linha desta tabela.

4 Técnicas utilizadas na concretização dos exercícios em PLPGSQL

As 14 alíneas do exercício 2 requiriam programação SQL. A linguagem utilizada foi PLPGSQL.

Estes exercícios foram realizados com recurso a mecanismos SQL como por exemplo:

- Funções
- Procedimentos Armazenados
- Gatilhos
- Vistas
- Tratamento de Erros
- Gestão Transacional

4.1 Funções

As **Funções** foram mais utilizadas quando havia necessidade de retornar algo, como por exemplo tabelas, contagens ou para a utilização de triggers.

4.2 Procedimentos Armazenados

Os **Procedimentos Armazenados** serviram para realizar procedimentos que poderiam ser programados para correr de forma rotineira ou operações para as quais o retorno não era relevante (*Ex.*: Criação de um cliente particular ou de um veículo).

4.3 Gatilhos

Os **Gatilhos** foram os mecanismos aos quais recorremos quando houve necessidade de realizar ações de forma automática ao acontecer algo (**INSERT**, **UPDATE** ou **DELETE** sobre tabelas/vistas). Com este mecanismo conseguimos, por exemplo, que a operação **DELETE** sobre a tabela *Cliente* não removesse a linha/dados correspondente ao cliente mas sim colocar um campo *apagado=true.*
Outro exemplo de utilização deste mecanismo foi para o incremento do número de alarmes de um veículo. Sempre que um alarme é criado um gatilho é acionado e incrementa o número de alarmes do veículo associado ao alarme.

4.4 Vistas

As **Vistas** foram utilizadas para mostrar informação mais relevante de um conjunto de tabelas.
Utilizámos este mecanismo para listar informações acerca dos alarmes mostrando assim o veículo ao qual o alarme está associado, o nome do condutor, coordenadas e data/hora do alarme. Desta forma temos uma vista que podemos interrogar a qualquer momento para obter informação mais detalhada acerca dos alarmes sem precisar de manualmente juntar informação de outras tabelas.

4.5 Tratamento de Erros

O **Tratamento de Erros** foi realizado principalmente quando o próprio **SGBD** (Sistema de Gestão de Base de Dados) não era capaz de detetar certas situações de erro uma vez que os *SQL Statements* eram válidos mas a base de dados ficaria num estado inconsistente do ponto de vista da nossa aplicação.

4.6 Gestão Transacional

A **Gestão Transacional** foi utilizada para prevenir situações de não atomicidade de um certo conjunto de instruções que necessitasse dessa atomicidade.
Este tipo de fenómenos poderia ser causado por concorrência de transações cujas operações envolvessem o mesmo conjunto de dados.
Desta forma ser-nos-ia possivel prevenir anomalias como por exemplo: *Dirty Reads, Dirty Writes, Lost Updates, Non-Repeatable Reads e Phantom Tuples*.

5 Conclusões

Com a realização deste trabalho colocámos em prática alguns dos conceitos abordados no âmbito da disciplina de Sistemas de Informação.

Nestes conceitos incluem-se:

- Gestão de transações SQL por forma a evitar anomalias derivadas de concorrência através de técnicas de Locking e Protocolos Multiversão tais como o Snapshot;

- Utilização de mecanismos dos SGBDs (procedimentos armazenados, funções, gatilhos, vistas, tratamento de erros).

Referências

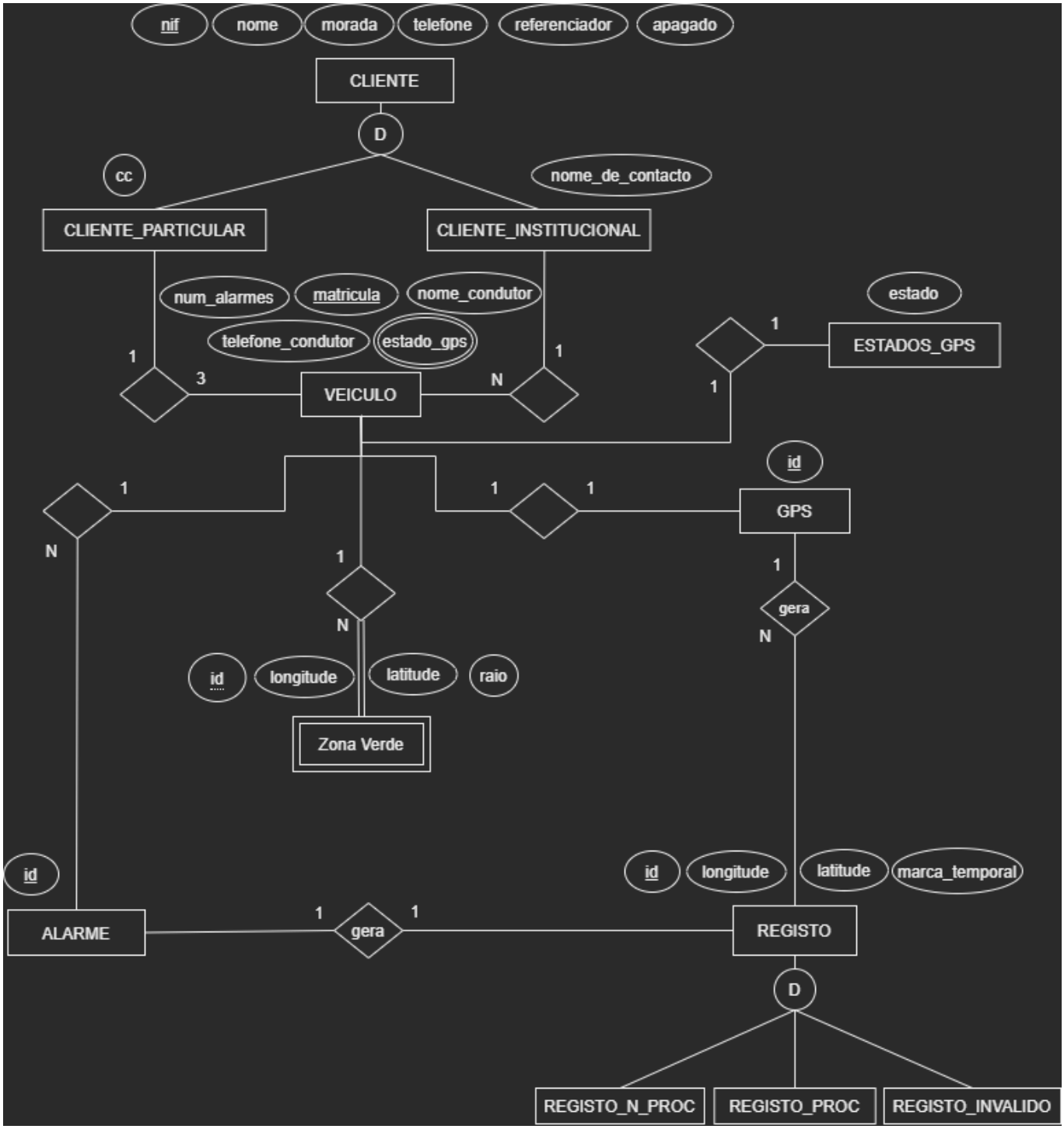
1: Walter Vieira, SISINF_M1_Transações(v6)
2: Walter Vieira, SisInf_M2_SP_Trig_Func(v2)

Anexos

A.1 – Enunciado do Trabalho

A empresa “OnTrack” pretende desenvolver um sistema para gerir e registar a localização de automóveis e camiões. O sistema deve registar os dados dos clientes da empresa, os quais podem ser particulares ou institucionais. Dos clientes particulares, interessa registar o CC, NIF, nome, morada, telefone e dos clientes institucionais, o NIF, nome, morada, telefone e nome de contacto.
Cada cliente tem associada uma frota de veículos, onde cada um terá um equipamento de GPS que envia o id do equipamento, uma marca temporal (com precisão ao segundo) e as coordenadas da sua localização a cada 10 segundos. Para os veículos é necessário guardar a matrícula, nome e telefone do condutor atual, podendo este não estar preenchido, o identificador do equipamento associado ao veículo e o estado do equipamento {‘Activo’,’PausaDeAlarmes’,’Inactivo’}. Deve ser possível adicionar novos estados através da aplicação. Para cada veículo é possível criar grupos de zonas verdes, onde cada zona verde tem as coordenadas GPS em graus decimais de latitude e longitude de um ponto central e um raio associado. Os grupos de zonas verdes associados aos veículos servem para gerar alarmes quando os veículos saem fora do conjunto das suas zonas verdes. Pressupõe-se a existência de um processo altamente otimizado que recebe as coordenadas dos equipamentos e que tem de inserir os registos com estes valores (registos não processados) numa tabela o mais rápido possível. O processamento destes registos é efetuado em lote, a cada 5 minutos, e pressupõe que os registos são movidos para outra tabela (registos processados) com as devidas restrições de integridade. Os registos que tenham identificadores não existentes, ou que não tenham data ou coordenadas devem ser colocados numa tabela própria de registos inválidos onde devem ficar durante 15 dias. Os registos processados que violem as zonas verdes dos veículos/equipamentos respetivos, geram alarmes que identificam os registos que os originaram. Os clientes particulares estão limitados a um máximo de 3 veículos e os clientes institucionais não têm limite. Para ajudar na aquisição de novos clientes para a empresa, os clientes particulares podem referenciar outros clientes (particulares ou institucionais) de forma a obter futuros descontos. Apenas interessa registar para um cliente a referência de quem o referenciou, caso exista. Deve ser suportada a remoção de clientes mas sem os remover da base de dados. Os veículos têm também o número de alarmes associado, recalculado sempre que é gerado um novo alarme.

A.2 - Modelo de Dados Conceptual



A.3 - Modelo de Dados Relacional

cliente (nif, nome, morada, telefone, referenciador, apagado)
PK = nif
FK = referenciador ref. cliente_particular.id_cliente
R.I.:

- referenciador -> Opcional
- Na app remover um utilizador não leva à sua remoção da base de dados.
- Apenas colocar 'apagado' = true
- Quando, na app, o cliente é apagado fazer apagado=true ao invés de remover o tuplo

cliente_particular (id_cliente, cc)
PK = id_cliente
FK = id_cliente ref. cliente.nif
R.I.:

- Cliente particular tem no máximo 3 veículos (1-3)

cliente_institucional (id_cliente, nome_de_contacto)
PK = id_cliente
FK = id_cliente ref. cliente.nif

veiculo (matricula, nome_condutor, telefone_condutor, estado_gps, num_alarmes, id_cliente, id_gps)
PK = matricula
FK = id_cliente ref. cliente.nif -- Cliente ao qual o veículo pertence
FK = estado_gps ref. estados_gps.estado
FK = id_gps ref. gps.id
R.I.:

- telefone_condutor -> Opcional
- O estado do equipamento é multivalor e pode ser: Activo, PausaDeAlarmes, Inactivo.
- Os estados podem ser geridos (adicionar/remover/editar) pela aplicação
- 'num_alarmes' é recalculado quando é gerado um novo alarme

gps (id)
PK = id

zona_verde (id, longitude, latitude, raio, id_veiculo)
PK = id
FK = id_veiculo ref. veiculo.matricula

registo (id, longitude, latitude, marca_temporal, id_gps)
PK = id
FK = id_gps ref. gps.id
R.I.:

- A cada 10s é emitido um registo a partir de um equipamento GPS
- Quando as coordenadas de um registo para um 'id_gps'(atribuido a um único veículo) saem das coordenadas de alguma das 'zonas verdes' associadas a esse veiculo gerar um 'alarme'
- Quando é criado um registo ele fica nesta tabela e também em 'registo_n_proc'

registo_n_proc (id_registo)
PK = id_registo
FK = id_registo ref, registo.id
- A cada 5 mins passar todos para a tabela 'registo_proc' caso esteja tudo bem.
Caso alguma das seguintes condições se verifique o registo vai para 'registo_invalido':

- id não existente na tabela 'registo'
- data('marca_temporal') a null
- latitude a null
- longitude a null

registo_proc (id, id_registo)
PK = id_registo
FK = id_registo ref, registo.id

registo_invalido (id_registo)
PK = id_registo
FK = id_registo ref, registo.id
R.I.:

- Registos aqui inseridos são removidos ao fim de 15 dias

alarme (id, id_registo, id_veiculo)
PK = id
FK = id_registo ref, registo.id
FK = id_veiculo ref. veiculo.matricula