



**Departamento de Engenharia de Eletrónica e Telecomunicações e  
de Computadores**

**Licenciatura em Engenharia Informática e de Computadores**

## **Trabalho prático** (Fases 1 e 2)

**Sistemas de Informação**

Semestre de verão de 2021/2022

Docentes: Afonso Remédios, Nuno Datia e Walter Vieira

## Planeamento

As datas importantes a recordar são:

- Lançamento do enunciado: 25 de março de 2022
- Entrega intermédia (Fase 1): 02 de maio de 2022 (+5 semanas)
- Entrega intermédia (Fase 2): 06 de junho de 2022 (5 semanas)

Cada entrega intermédia deve apresentar o relatório e código (se houver) referentes exclusivamente a essa fase. O relatório deve seguir um dos modelos fornecidos, obrigatoriamente, sob pena de penalização na nota. Este deve ser conciso e apresentar a justificação de todas as decisões tomadas. A capa do relatório deve indicar a composição do grupo, a unidade curricular e a fase do trabalho a que respeita. Caso tenha adendas e/ou correções a fazer a modelos já entregues, deve indicá-las de forma explícita no relatório seguinte.

O ficheiro pdf (ou, o zip) gerado deve seguir o padrão: `TPSISINF-2122SV-GrupoNNTFaseN.ext`. N representa um dígito, T a turma (D1, D2, N) e `ext` a extensão do ficheiro), exemplo: TPSI2-2122SV-Grupo14D2Fase1.zip ou TPSISINF-2122SV-Grupo02NFase1.zip.

## Primeira fase

### Objetivos de aprendizagem

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.

### Enunciado do trabalho (documento de requisitos do sistema)

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 actual, 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.

Nota:

-Deve utilizar os tipos de dados que achar adequados.

## Resultados pretendidos

Tendo em conta os objetivos de aprendizagem, deverão ser produzidos os seguintes resultados:

1. O modelo de dados (conceptual e relacional), incluindo todas as restrições de integridade;
2. O código PL/pgSQL que permite:
  - (a) Criar o modelo físico (1 script autónomo);
  - (b) Remover o modelo físico (1 script autónomo);
  - (c) Preenchimento inicial da base de dados (1 script autónomo);
  - (d) Criar os mecanismos que permitam inserir, remover e atualizar informação de um cliente particular (CC, NIF, nome, morada, cliente que o referenciou);
  - (e) Criar uma função que devolve o total de alarmes para um determinado ano e veículo passados como parâmetros; caso a matrícula do veículo seja vazia deve devolver as contagens de alarmes para todos os veículos nesse ano;
  - (f) Criar um procedimento que será chamado de forma periódica e que processa todos os registos não processados até esse momento. Deve tratar todos os registos existentes na tabela de registos não processados, de forma a copiar os dados para a tabela de registos processados ou de registos inválidos e remover as entradas tratadas. Deve garantir que processa todas as entradas não processadas até esse momento.
  - (g) Criar um gatilho que permita analisar o registo processado, aquando da sua criação. e que gere o respetivo alarme caso esteja fora de qualquer uma das suas zonas verdes. Se não existirem zonas verdes ou se o equipamento estiver em pausa de alarmes não gera alarme. Para a realização do gatilho, deverá usar a função `zonaVerdeValida` que recebe as coordenadas e o raio de uma zona verde e as coordenadas do registo em processamento e retorne true se as coordenadas do registo a ser processado se encontrarem dentro da zona verde e false caso contrário. Para teste implemente uma versão da função `zonaVerdeValida` que retorna true quando a o arredondamento da coordenada da latitude do registo for par e false quando for impar;
  - (h) Desenvolver um procedimento que crie um veículo com a respectiva informação do equipamento associado, e que o associe a um cliente. Caso sejam passados dados para a criação de uma zona verde, deve criar e associar o veículo a essa zona. Reutilize procedimentos já existentes ou crie novos se necessário; Deve garantir as restrições de negócio respectivas, nomeadamente a limitação do número de veículos.

(i) Criar uma vista, que liste todos os alarmes. A vista deve apresentar a matrícula do veículo, o nome do condutor, as coordenadas e a data/hora do alarme;

(j) Adicione suporte de modo que a execução da instrução INSERT sobre a vista da alínea anterior permita adicionar um alarme e o respectivo registo tratado;

(k) Implemente o procedimento que será chamado diariamente e que apaga os registos inválidos existentes com duração superior a 15 dias;

(l) Crie os mecanismos necessários para que a execução da instrução DELETE sobre a tabela de cliente permita desativar o(s) Cliente(s) sem apagar os seus dados. Assuma que podem ser apagados vários clientes em simultâneo;

(m) Crie os mecanismos necessários para que, de forma automática, quando é criado um alarme, o número de alarmes do veículo seja actualizado;

(n) Crie um *script* autónomo com os testes das funcionalidades de 2d a 2m para cenários normais e de erro. Este *script*, ao ser executado, deve listar, para cada teste, o seu nome e indicação se ele correu ou não com sucesso;

Ex: “teste 1: Inserir Cliente com dados bem passados: Resultado OK”

Ou caso falhe: “teste 1: Inserir Cliente com dados bem passados: Resultado NOK”

Nota:

Todos os procedimentos armazenados devem ter tratamento de erros e gestão transaccional utilizando o nível de isolamento adequado.

**Data limite para entrega:** 2 de maio de 2022 até às 23:50.

A entrega tem de incluir um relatório (em formato PDF) e o código PL/pgsql, entregues via moodle. O relatório deverá obedecer ao padrão fornecido pelos docentes.

## Segunda fase

### Objetivos de aprendizagem

No final da segunda fase do trabalho, os alunos devem ser capazes de:

- Desenvolver uma camada de acesso a dados, que use uma implementação de JPA e um subconjunto dos padrões de desenho DataMapper, Repository e UnitOfWork;
- Desenvolver uma aplicação em Java, que use adequadamente a camada de acesso a dados;
- Utilizar corretamente processamento transacional, através de mecanismos disponíveis no JPA;
- Garantir a correta libertação de ligações e recursos, quando estes não estejam a ser utilizados;
- Garantir a correta implementação das restrições de integridade e/ou lógica de negócio;

### Resultados pretendidos

Tendo em conta os objetivos de aprendizagem, deverão ser produzidos os seguintes resultados:

1. Criação de uma aplicação Java que permita:
  - (a) Aceder às funcionalidades 2d a 2l, descritas na fase 1 deste trabalho;
  - (b) Realizar a funcionalidade 2h, descrita na fase 1 deste trabalho, sem usar qualquer procedimento armazenado;
  - (c) Realizar a funcionalidade 2h, descrita na fase 1 deste trabalho, reutilizando os procedimentos armazenados que a funcionalidade original usa;
2.
  - (a) Reimplementação da funcionalidade 2f usando optimistic locking.
  - (b) Teste de optimistic locking, apresentando uma mensagem de erro adequada em caso de alteração concorrente conflituante que inviabilize a operação. No relatório deve estar descrita a forma como as situações de erro foram criadas para teste desta alínea;
3. Pretende-se guardar em *document store* os relatórios semanais dos alarmes ocorridos. Estes relatórios terão existência independente dos dados que os originaram, isto é, devem continuar a existir, mesmo que os dados que lhes deram origem sejam apagados. Deve realizar as alterações necessárias à aplicação desenvolvida para permitir:
  - a) Realizar o a criação do relatório semanal, dados o ano e a semana no ano cujo relatório se pretende.
  - b) Obter informação dos alarmes relativos a um dado cliente num período de N semanas consecutivas, dado o ano e semana no ano relativo à primeira semana e o número de semanas pretendido.

Notas:

Todas as alíneas devem ter tratamento de erros e, nos casos relevantes, gestão transacional utilizando o nível de isolamento adequado de forma explícita.

O código a desenvolver deverá estar organizado em vários projetos que reflitam a estrutura modular da aplicação, a qual deverá reduzir o acoplamento entre módulos e facilitar a identificação das dependências entre eles.

**Data limite para entrega:** 06 de junho de 2022 até às 23:50.

A entrega tem de incluir um relatório (em formato PDF), os projetos *MAVEN*, o código Java e o código PL/pgSQL, entregues via moodle. O relatório deverá obedecer ao padrão fornecido pelos docentes.