

Arquitetura e Administração de Bases de Dados

2024/2025

Trabalho Prático

1. Introdução

Este documento apresenta o enunciado do trabalho prático da unidade curricular de Arquitetura e Administração de Bases de Dados, integrada na licenciatura de Engenharia Informática, ramo de Sistemas de Informação do Instituto Superior de Engenharia de Coimbra

Pretende-se com este trabalho que os alunos desenhem e implementem uma base de dados relacional, recorrendo ao SGBD Oracle, que lhes permita aplicar os conhecimentos adquiridos no decorrer das aulas da unidade curricular bem como estimular as capacidades de pesquisa e trabalho autónomo.

Embora se descreva um sistema completo, numa arquitetura cliente/servidor, o resultado pedido aos alunos prende-se apenas com a componente servidor, não se exigindo a implementação do sistema completo, nomeadamente, a implementação da interface com o utilizador.

No entanto, e para aquilo que é pedido, os alunos devem ter a preocupação de propor soluções cujas características de legibilidade, completude e eficiência seja o mais elevado possível, já que residirá aí grande parte da ponderação dos critérios de avaliação do trabalho.

Nas secções seguintes faz-se uma descrição do sistema a implementar e discrimina-se a forma como devem ser produzidos e entregues os vários elementos requeridos para este trabalho.

Notas Importantes:

- Considera-se que não realizam um checkpoint se não entregarem a documentação requerida no moodle, ou que falem à defesa do checkpoint
- Considera-se que não realizam o trabalho prático, se não entregarem a documentação do checkpoint 3 ou que falem à defesa desse checkpoint.
- Requisito de acesso a exames: avaliação mínima de 25% na nota global do trabalho prático

Prazos a cumprir:

16-02 – Definição dos grupos (**grupos de 3 alunos**)

10-03 às 09h – Checkpoint **1** (avaliado nas aulas práticas de **10 a 13 de março**).

07-04 às 09h – Checkpoint **2** (avaliado nas aulas práticas de **07 e 10 de abril**).

01-06 às 12h – Checkpoint **3** (avaliado a **02 a 05 de junho**).

2. Descrição do sistema

A empresa **VendingPLUS**, especialista na colocação e exploração de máquinas de *vending* de produtos na região Centro, necessita construir um sistema informático para a gestão das máquinas, dos produtos vendidos e da logística relacionada com a reposição dos produtos nas máquinas. A empresa apenas utiliza máquinas que permitem a venda de produtos não confeccionados, isto é, apenas produtos que possam ser dispensados sem requererem processamento adicional (ex. garrafas e latas de bebidas, pacotes de bolachas, chocolates,...). A figura 1 ilustra uma dessas máquinas.



Fig. 1 – pedido de entrega

O sistema será responsável por toda a logística desde a colocação e manutenção das máquinas, configuração da quantidade de cada produto em cada máquina, e a manutenção e reabastecimento dos produtos. Em cada máquina é possível configurar quais os compartimentos que serão atribuídos a cada produto, e consequentemente a capacidade máxima de cada produto que pode ser colocada numa máquina. Em cada máquina, a cada compartimento está definido um produto (ex. Água Luso 33cl) e o respetivo preço, a capacidade máxima desse produto e um código que o cliente terá de digitar quando pretende adquirir o produto desse compartimento.

As máquinas permitem a aquisição de produtos exclusivamente através de meios de pagamento digitais: Multibanco e MBWay. Para isso está equipado com um terminal de pagamentos e uma ligação à internet. Cada venda de um produto é registada pela máquina, que envia imediatamente essa informação para o sistema a implementar.

Além disso, cada máquina está equipada com um sistema que periodicamente (de 10 em 10 minutos) envia informação do estado da máquina (ex. operacional, com erro, bloqueada, em manutenção). Quando num intervalo de 30 minutos não é recebida nenhuma informação de uma máquina, considera-se que está offline e que deve ser visitada o mais rapidamente possível. Para manter o bom funcionamento das máquinas, todas as máquinas devem ser visitadas pelo menos uma vez por semana, para manutenção e/ou reposição de produtos.

A empresa possui vários armazéns logísticos na região onde guardam os produtos a colocar nas máquinas. A quantidade em stock de cada produto em cada um dos armazéns é gerido de acordo com a quantidade de produtos vendidos e da quantidade colocada nas máquinas nas semanas anteriores. A quantidade a encomendar de cada produto para cada armazém é determinada em função da quantidade vendida e da estimativa das necessidades previsíveis de reabastecimento para as próximas 2 semanas.

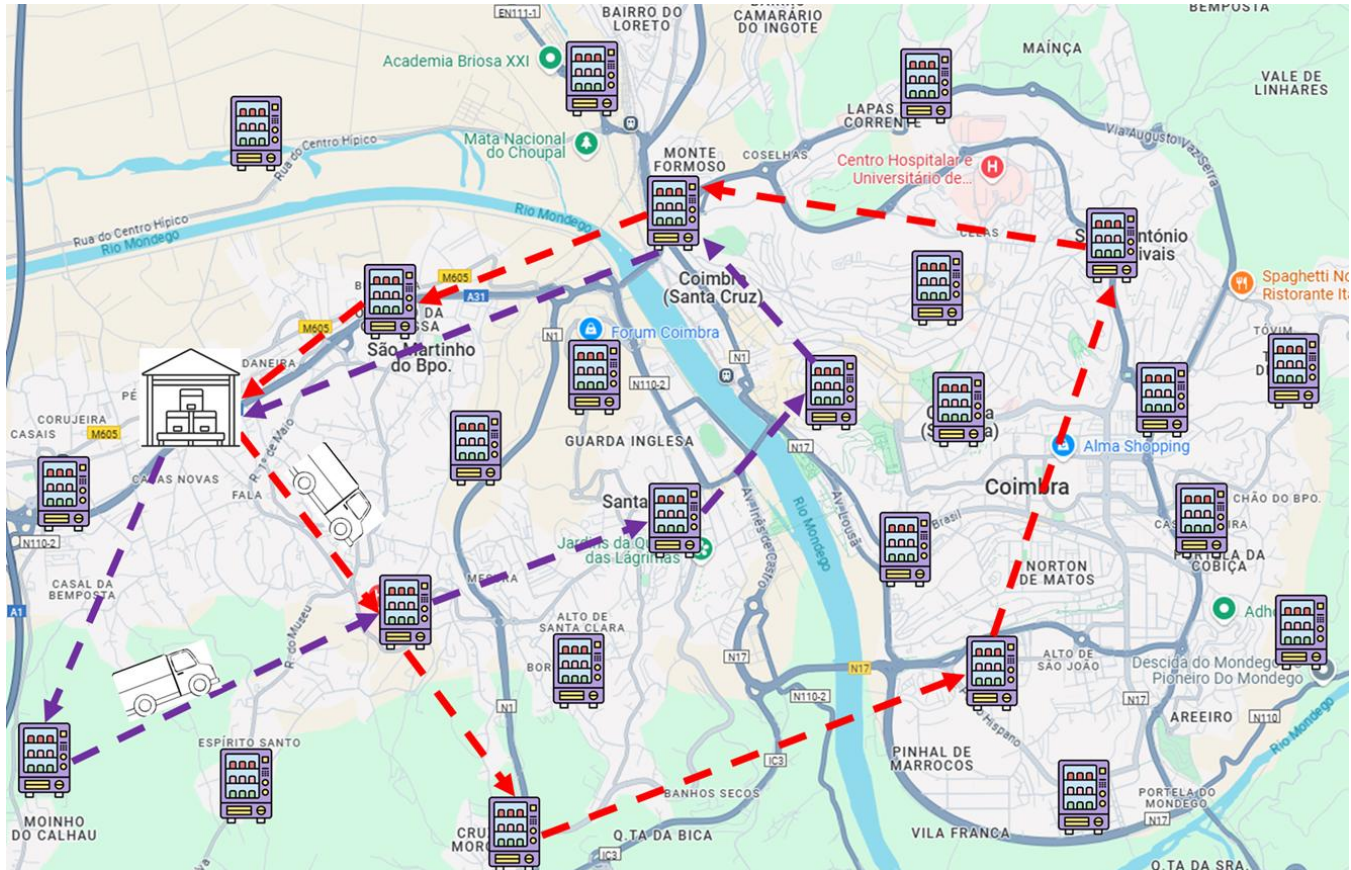


Fig. 2 – Localização das máquinas e exemplos de rotas de abastecimento

Cada armazém tem associado um conjunto de veículos elétricos que podem ser utilizados para o reabastecimento de produtos, mas que estão limitados a deslocações inferiores as 300km. Numa viagem de reabastecimento, um funcionário visita várias máquinas, e, quando necessário, repõe produtos que estejam em falta. Uma viagem pode seguir uma rota pré-definida, em que percorre todas as máquinas definidas na rota, mesmo aquelas em que não seja necessário repor stock (figura 2). No entanto, quando é detetada uma rutura de stock em algumas máquinas, uma viagem pode incluir a visita a outras máquinas não contempladas na rota. Por questões de manutenção e verificação do estado das máquinas, todas as máquinas são visitadas pelo menos uma vez em cada semana, mesmo que não seja necessário repor nenhum produto.

O sistema a projetar deve dar resposta às necessidades operacionais típicas de uma empresa de *Vending* de produtos (venda, reposição, manutenção, etc.), mas também servir de apoio à gestão através da disponibilização de informação detalhada e agregada acerca do estado e visita das máquinas, produtos vendidos, abastecimentos realizados, intervalo entre abastecimentos por cada tipo de produto, e análise da eficiência dos seus veículos, das rotas e do stock de produtos nos armazéns e na utilização e manutenção das máquinas.

3. Entregas

Pretende-se que o desenvolvimento do trabalho subjacente a este projeto seja efetuado de forma faseada ao longo do semestre. Deste modo, definiram-se três momentos para entregas parciais dos diversos elementos pedidos, bem como um momento adicional para defesa individual do trabalho, que se detalham nos pontos seguintes.

3.1. CHECKPOINT 1 (submeter no moodle até 10 março às 09h) – Análise de dados (Entrega do modelo ER, do Diagrama de tabelas e do Script de criação de tabelas, ...)

Nesta fase deve ser feita a análise de dados do sistema proposto. Deverá ser entregue os seguintes documentos:

- Diagrama do modelo E/R (utilize designações sugestivas para as entidades, relacionamentos e atributos);
- Diagrama do modelo físico de tabelas para o SGBD Oracle 11g;
- *Script SQL* de criação da base de dados.

Todas as tabelas e restrições (*primary e foreign keys*) devem estar criadas na conta Oracle do grupo AABDGxx (em que xx é o número do grupo de 01 a 35).

Além disso, **cada elemento do grupo deve propor 2 VIEWS** que considere relevante, justificando a relevância para o negócio em questão. Estas vistas deverão posteriormente ser implementadas e entregues no checkpoint2:

- uma vista **VIEW_J_202xxxx** (em que 202xxxx é o nº de aluno do inforestudante), que inclua pelo menos um SELECT com GROUP BY e que considere relevante, justificando a sua relevância. A relevância e o grau de complexidade das mesmas influenciarão fortemente a sua avaliação. Ex.
- uma vista **VIEW_K_202xxxx** (em que 202xxxx é o nº de aluno do inforestudante), que inclua pelo menos um SELECT encadeado e que considere relevante, justificando a sua relevância. A relevância e o grau de complexidade das mesmas influenciarão fortemente a sua avaliação. Ex.

Estes elementos. devem serão alvo de análise nas aulas de **04 e 06 de março**.

3.2. CHECKPOINT 2 (07 abril às 09h) – Vistas com pesquisas à base de dados

Nesta fase, a base de dados deve estar preenchida com dados para que seja possível testar as suas funcionalidades. Para este checkpoint devem ser entregues os seguintes elementos:

- **Script SQL de inserção de dados** nas diversas tabelas.
- **Implementação de consultas** à base de dados, incluindo uma amostra dos respetivos resultados para verificação da correção das mesmas.
- Estas consultas devem ser obtidas através de comandos SELECT, e guardadas como vistas.
- Podem ser criadas vistas auxiliares para simplificar esses comandos.
- Serão desvalorizados os comandos que não sejam corretamente comentados.
- O formato do nome das vistas criadas deve ser do tipo **VIEW_alinea** (ex. **VIEW_A**, **VIEW_B**, ...).

Algumas das VIEWS a implementar são:

- a) **VIEW_A** que, para o tipo de produtos ‘Snacks’, retorne a listagem das máquinas localizadas na cidade de Coimbra que foram reabastecidas ontem, ordenadas pela quantidade abastecida e que atualmente não possuem stock desse tipo de produto. Exclua as máquinas que estão inativas. Ordene pela quantidade abastecida.

IDMAQUINA	DATA_HORA_ABAST	LOCAL	QUANT_ABASTECIDA	NUM_PRODUTOS_DIFERENTES
2023041105	06/02/25 08H11	CoimbraShopping	24	4
...				

CREATE VIEW **VIEW_A** AS SELECT ...

- b) **VIEW_B** que, para a viagem de reabastecimento com o código 2025031105, mostre a listagem cronológica das máquinas visitadas. Para cada máquina visitada, mostre a quantidade existente de cada produto e a quantidade abastecida. Ordene pela data/hora da visita a cada máquina, e decendentemente pela quantidade abastecida.

IDMAQUINA	LOCAL	REF_PRODUTO	PRODUTO	QUANT_EXISTENTE	QUANT_ABASTECIDA	CAPACIDADE
63	CS CELAS	9920	MARS	12	4	20
...						

CREATE VIEW **VIEW_B** AS SELECT ...

- c) **VIEW_C** que, para cada máquina, mostre qual o produto mais vendido no mês anterior. Mostre a quantidade vendida desse produto nesse mês, e a quantidade vendida desde o último reabastecimento. Exclua os produtos que representam mais do que 50% da capacidade máxima configurada para essa máquina. Na listagem dos produtos, os produtos menos consumidos devem aparecer no final.

IDMAQUINA	LOCAL	REF_PRODUTO	PRODUTO	QUANT_VENDIDA_MES	QUANT_VEND_DESDE_ULTIMO
63	CS CELAS	9924	LION	67	23
...					

CREATE VIEW **VIEW_C** AS SELECT ...

- d) **VIEW_D** que, obtenha as máquinas localizadas num raio de 30 kms em relação ao armazém localizado em Taveiro, que possuem o produto ‘KitKat’. Considere a distância linear (em kms) entre a localização geográfica do armazém e a localização geográfica das máquinas. Para cada máquina, mostre a data do último abastecimento e a quantidade total de produtos que possui. Apresente as máquinas de acordo com a proximidade das máquinas.

MAQUINAID	LOCAL	DISTANCIA_LINEAR	DATA_ULT_ABAST	QUANT_TOTAL_PRODUTOS
787	CHUC COVOES	6.3	21/02/2025 09H07	56
...				

```
CREATE VIEW VIEW_D AS SELECT ...
```

- e) **VIEW_E** que, considerando as vendas dos anos de 2023 e 2024, obtenha, para cada tipo de produto, a média da quantidade vendida em cada mês. Considere apenas as máquinas que estejam atualmente operacionais, e que tiveram um número de reabastecimentos acima da média. Ordene decendentemente pela média, e pelo produto

IDMAQUINA	PRODUTO	MEDIAMENSAL
773	AGUA LUSO 33c1	152

```
CREATE VIEW VIEW_E AS SELECT ...
```

- f) **VIEW_F** que, para a máquina que vendeu mais produtos do tipo ‘AGUA’ nas últimas 72h, obtenha para cada produto desse tipo, o número de produtos vendidos no mês de fevereiro, e que percentagem representa na quantidade total de produtos vendidos pela máquina nesse mês. Mostre também a quantidade reabastecida nesse mês.

IDMAQUINA	REFPRODUTO	QUANT_VENDIDA	PERCENTAGEM	QUANT_REABASTECIDA
884	8878	456	32	490

```
CREATE VIEW VIEW_F AS SELECT ...
```

- g) **VIEW_G** que, considerando os reabastecimentos registados no ano passado, mostre quais os 2 tipos de produtos mais frequentemente abastecidos nas máquinas de Coimbra. Considere apenas as viagens de abastecimento que visitaram mais do que 3 máquinas. Ordene decendentemente pelo volume total dos pedidos de transportes entregues por cada cidade de destino.

VIAGEM	TIPO_PRODUTO	QUANT_ABASTECIDA	NUM_MAQ_ABASTECIDAS
773	AGUA	8993	323
...			

```
CREATE VIEW VIEW_G AS SELECT ...
```


- h) **VIEW_H** que mostre a listagem dos 5 veículos mais utilizados em viagens em que foram reabastecidas 3 ou mais máquinas com produtos do tipo AGUA. Considere apenas as viagens superiores a 50 km, realizadas no mês passado. Ordene decendentemente pelo número de viagens que cada veículo efetuou.

MATRICULA	MARCA	MODELO	N_MAQUINAS_REABASTE
DD-66-77	SCANIA	SERIE L	4
...			

```
CREATE VIEW VIEW_H AS SELECT ...
```

- i) **VIEW_I** que, considerando as viagens realizadas desde o início do ano, mostre a listagem das 3 máquinas mais frequentemente visitadas, indicando o número de visitas, a quantidade total abastecida, a quantidade média abastecida por visita e o número médio de produtos distintos abastecido por visita.. Ordene decendentemente pelo número de visitas. Considere apenas as viagens iniciadas no armazém com mais viagens.

ARMAZEM	MAQUINA	N_VISITAS	QUANT_TOTAL	QUANT_MEDIA_VISITA	N_PROD_DIF	
TAVEIRO	ARM01	MAQ01 USCELAS	22	7229	32	5
...						

```
CREATE VIEW VIEW_I AS SELECT ...
```

- j) **Cada elemento do grupo** deve criar a vista com o formato **VIEW_J_<naLuno>**, que se propôs a fazer no checkpoint1, que inclua um SELECT com GROUP BY e que considere relevante, justificando a sua relevância. A relevância e o nível de complexidade das mesmas influenciarão fortemente a sua avaliação.

```
CREATE VIEW VIEW_J_20xxxx AS SELECT ...
```

- k) **Cada elemento do grupo** deve criar a vista com o formato **VIEW_K_<naLuno>**, que se propôs a fazer no checkpoint1, que inclua um SELECT encadeado e que considere relevante, justificando a sua relevância. A relevância e o nível de complexidade das mesmas influenciarão fortemente a sua avaliação.

```
CREATE VIEW VIEW_N_20xxxx AS SELECT ...
```

Além disso, **cada elemento do grupo** deve propor, para ser implementada até ao checkpoint3:

- l) Propor uma **FUNÇÃO** com o nome **P_FUNC_202xxxx** (em que 202xxxx é o nº de aluno), que considere relevante, justificando a sua relevância, e que pretende implementar até ao checkpoint 3. A relevância e o grau de complexidade das mesmas influenciarão fortemente a sua avaliação.
- m) Propor um **PROCEDIMENTO** com o nome **Q_PROC_202xxxx** (em que 202xxxx é o nº de aluno), que considere relevante, justificando a sua relevância, e que pretende implementar até ao checkpoint 3. A relevância e o grau de complexidade das mesmas influenciarão fortemente a sua avaliação.

- n) Propor um TRIGGER com o nome **R_TRIG_202xxxx** (em que 202xxxx é o nº de aluno), que considere relevante, justificando a sua relevância, e que pretende implementar até ao checkpoint 3. A relevância e o grau de complexidade das mesmas influenciarão fortemente a sua avaliação.

Os elementos propostos nas alíneas l) m) e n) neste *checkpoint* serão alvo de análise nas aulas de **07 e 10 de abril**.

3.3. CHECKPOINT 3 (01 junho às 12h) - Procedimentos, funções, *triggers* e parâmetros físicos

O trabalho a desenvolver nesta fase inclui a construção de procedimentos/funções em PL/SQL que facilitem a manipulação dos dados da base de dados (BD), a utilização de *triggers* para garantia da integridade dos dados (ou outras tarefas onde façam sentido) e a aplicação do conceito de transação. Pretende-se também que seja feito o cálculo dos parâmetros físicos e o tratamento das exceções.

Assim, pretende-se que sejam entregues os elementos necessários à implementação dos seguintes pontos:

- a) Crie a função **quantidade_vendida**, que recebe o código de uma máquina, a referência de um produto, e um intervalo temporal, a data de início e a data de fim (por defeito a data atual), e retorna a quantidade vendida desse produto, nessa máquina nesse período. A função deve retornar as seguintes exceções: -20801, -20802, -20809

```
FUNCTION quantidade_vendida(idmaquina NUMBER, idproduto NUMBER, dataInicio DATE, dataFim DATE)
RETURN NUMBER
```

- b) Crie a função **quantidade_em_falta**, que recebe o código de uma máquina e a referência de um produto, e retorna a quantidade em falta desse produto nessa máquina. A quantidade em falta é a diferença entre a capacidade máxima definida para essa máquina e a quantidade existente. A função deve retornar as seguintes exceções: -20801, -20802

```
FUNCTION quantidade_vendida(idmaquina NUMBER, idproduto NUMBER) RETURN NUMBER
```

- c) Crie a função **quantidade_media_diaria**, que recebe o código de uma máquina e a referência de um produto, e retorna a quantidade média diária vendida desde o penúltimo reabastecimento desse produto nessa máquina. A função deve retornar as seguintes exceções: -20801, -20802

```
FUNCTION quantidade_media_diaria(idmaquina NUMBER, idproduto NUMBER) RETURN NUMBER
```

- d) Crie a função **data_ultimo_abastec**, que recebe o código de uma máquina e a referência de um produto, e retorna a data do último reabastecimento dessa máquina com esse produto. A função deve retornar as seguintes exceções: -20801, -20802

```
FUNCTION data_ultimo_abastec(idmaquina NUMBER, idproduto NUMBER) RETURN DATE
```

- e) Crie a função **distancia_entre_maquinas**, que recebe o código de duas máquinas e retorna a distância linear entre essas duas máquinas. A função deve retornar as seguintes exceções: -20801, -20810

```
FUNCTION distancia_entre_maquinas(idmaquina1 NUMBER, idmaquina2 NUMBER) RETURN NUMBER
```

- f) Crie a função **distancia_viagem**, que recebe o código de uma viagem de abastecimento, e considerando a sequência das máquinas visitadas, retorna a distância total da viagem. Assuma que a viagem termina no armazém onde a viagem iniciou. A função deve retornar as seguintes exceções: -20807

```
FUNCTION distancia_viagem(idviagem NUMBER) RETURN NUMBER
```

- g) Crie a função **maquina_mais_proxima**, que recebe a referência de um produto e as coordenadas de um local, e retorna o código da máquina mais próxima desse local que tem esse produto disponível para venda. A função deve retornar as seguintes exceções: -20802

```
FUNCTION maquina_mais_proxima (idproduto NUMBER, lat NUMBER, long NUMBER) RETURN NUMBER
```

- h) Crie a função **prox_maquina_sem_produto**, que recebe a referência de um produto e o código de uma máquina, e retorna o código da máquina mais próxima que não tem esse produto disponível para venda. A função deve retornar as seguintes exceções: -20802

```
FUNCTION prox_maquina_sem_produto (idproduto NUMBER, idmaquina NUMBER) RETURN NUMBER
```

- i) Crie o procedimento **cria_viagem_abast** que, recebe o código de um armazém e que considerando os produtos em falta num determinado raio (em kms) do armazém, planeia uma viagem de abastecimento, começando pela máquina com maior rutura de stock. Considere apenas as 10 máquinas com menos stock e cria a sequência de acordo com a rutura de stock. O procedimento deve retornar as seguintes exceções: -20806, -20811

```
PROCEDURE cria_viagem_abast (cod_armazem NUMBER, raio NUMBER)
```

- j) Crie o procedimento **encomenda_produtos** que recebe o código de um armazém e uma data, e tomando em consideração a quantidade vendida desde essa data, a quantidade existente nas máquinas e a quantidade em stock no armazém, encomende a quantidade de produtos necessários para a próxima semana. O procedimento deve retornar as seguintes exceções: -20806, -20812

```
PROCEDURE cria_viagem_abast (cod_armazem NUMBER , datainicio DATE)
```

- k) Crie o procedimento **abastece_produto** que recebe o código de um armazém, o código de um produto e uma quantidade, e planeia uma viagem de abastecimento em que distribui essa quantidade pelas máquinas que requerem esse produto. Começa pela máquina com maior rutura de stock. O procedimento deve retornar as seguintes exceções: -20802, -20806, -20813

```
PROCEDURE abastece_produto(cod_armazem NUMBER , cod_produto NUMBER, quantidade NUMBER)
```

- l) Crie um trigger **update_stock** que quando é efetuado a venda um produto, atualize a quantidade desse produto existente na máquina. Se, após a venda, a máquina ficar completamente vazia, sem stock de nenhum produto, deve mudar o estado da máquina para 'SEM STOCK'

```
TRIGGER update_stock
```

- m) Crie um trigger **abastece** que quando é registado o abastecimento de um produto, não deve permitir registar uma quantidade inválida, por exemplo, negativo ou superior à capacidade definida nessa máquina para o produto abastecido.

```
TRIGGER abastece
```

- n) Crie um trigger **update_viagem** que quando após o registo da quantidade abastecida de um produto, deve atualizar a quantidade transportada no veículo, durante a viagem, em conformidade com a quantidade abastecida.

```
TRIGGER update_viagem
```

- o) Crie a função **distancia_linear**, que recebe as coordenadas (latitude, longitude) de dois pontos e retorna a distância linear (em kms) entre eles.

```
FUNCTION distancia_linear(lat1 NUMBER, long1 NUMBER, lat2 NUMBER, long2 NUMBER) RETURN NUMBER
```

Além disso, **cada elemento do grupo** deve, considerando o proposto no checkpoint2:

- p) Criar **a FUNÇÃO** com o nome **P_FUNC_202xxxx** (em que 202xxxx é o nº de aluno), proposta no checkpoint2, justificando a sua relevância. A relevância e o grau de complexidade influenciarão fortemente a sua avaliação.

```
FUNCTION P_FUNC_2021xxxx
```

- q) Criar **o PROCEDIMENTO** com o nome **Q_PROC_202xxxx** (em que 202xxxx é o nº de aluno), proposto no checkpoint2, justificando a sua relevância. A relevância e o grau de complexidade influenciarão fortemente a sua avaliação.

```
PROCEDURE Q_PROC_202xxxx
```

- r) Criar **o TRIGGER** com o nome **R_TRIG_202xxxx** (em que 202xxxx é o nº de aluno), proposto no checkpoint2, justificando a sua relevância. A relevância e o grau de complexidade influenciarão fortemente a sua avaliação.

```
TRIGGER R_TRIG_2021xxxx
```

- s) Identifique os mecanismos necessários para assegurar a integridade dos dados que não seja assegurada por restrições da BD, incluindo as restrições de valor (ex. duração e custo negativo), datas inválidas (termino anterior ao início), ...
- t) Entrega do cálculo dos parâmetros físicos das 5 tabelas do sistema, que considerem que irão ocupar mais espaço.

Tratamento de exceções:

Sempre que for necessário, devem ser lançadas, entre outras, as exceções seguintes:

Código	Mensagem
-20801	Código de máquina inexistente
-20802	Código de produto inexistente
-20803	Código tipo de produto inexistente
-20804	Máquina sem stock do produto
-20805	Viagem sem stock do produto
-20806	Código de armazém inexistente
-20807	Viagem de abastecimento inexistente
-20808	Viagem de abastecimento ainda não foi iniciada
-20809	Inválido intervalo temporal
-20810	Máquinas inválidas. Devem ser diferentes.
-20811	Distância Inválida
-20812	Data inválida. Deve ser anterior à data atual.
-20813	Quantidade inválida
-208xx	...

Serão desvalorizados os procedimentos, funções e *triggers* que não forem corretamente comentados.

A entrega final, no *moodle* deverá ocorrer até ao dia **01 de junho**.

3.4. Defesa individual do trabalho

As defesas dos trabalhos serão realizadas a **03 e 04 de junho**. Espera-se que cada um dos elementos do grupo consiga, individualmente, explicar e/ou alterar qualquer parte do trabalho entregue.

Os *scripts* entregues que não funcionem não serão considerados para avaliação.

Considera-se que os alunos que não façam a defesa na data marcada não realizaram o trabalho.