

# Sistemas de Informação e Bases de Dados

Enunciado da Etapa 4 do Projeto de 2024/2025

## Ponto de partida

Para esta última etapa do projeto, considere o esquema relacional da etapa 3, sobre a gestão de uma loja de equipamentos usados, com as tabelas *cliente*, *ficha*, *equipamento*, e *fatura*.

## Objetivos

1. Implementar um *pacote* PL/SQL para a gestão de dados, cuja especificação de funcionalidades vem na secção seguinte.
2. Escrever um *script* com invocações dos procedimentos e funções do pacote PL/SQL que demonstrem um cenário de gestão de dados.

## Funcionalidades

Os procedimentos e funções seguintes devem constar num pacote PL/SQL chamado *pkg\_loja*.

*registra\_cliente*(nif\_in, nome\_in, telemovel\_in, genero\_in, nascimento\_in, localidade\_in)

Regista um cliente com NIF, nome, telemóvel, género, ano de nascimento, e localidade.

Aquando do registo, a idade do cliente tem de ser pelo menos 16 anos (RIA 9, adaptada).

*registra\_ficha*(ean\_in, marca\_in, modelo\_in, tipo\_in, ano\_in, preco\_in)

Regista uma ficha de equipamento com EAN-13, marca, modelo, tipo, ano de lançamento, e preço. Se já existir uma ficha com o mesmo EAN-13, os seus dados devem ser atualizados. Aquando do registo, o ano de lançamento tem de ser anterior ao ano atual (RIA 10).

*registra\_equipamento*(ean\_in, estado\_in, preco\_in, data\_in := SYSDATE) -> NUMBER

Regista um equipamento usado com EAN-13 (de uma ficha de equipamento já existente), estado de conservação, preço de venda, e data de colocação na loja (sendo, por omissão, assumida a data atual). Tendo em conta o EAN-13 da ficha, é automaticamente atribuído ao equipamento um número sequencial de exemplar<sup>1</sup>, devolvido pela função.

*registra\_compra*(cliente\_in, ean\_in, exemplar\_in, fatura\_in := NULL) -> NUMBER

Regista a compra de um equipamento usado (com EAN-13 e número de exemplar) por um cliente (identificado pelo seu NIF), no âmbito de uma fatura, ou seja, acrescenta uma linha ao talão de uma fatura. Se o número de fatura for omitido, é criada uma nova fatura para o

---

<sup>1</sup> O número sequencial de exemplar *não* pode ser gerado por uma sequência Oracle, pois começa do um *para cada* ficha de equipamento (por exemplo, exemplar um de Apple iPhone 15 e exemplar um de Canon 7d). Também *não* é suficiente, embora possa ser útil, fazer `SELECT COALESCE(MAX(exemplar)+1, 1) FROM equipamento WHERE (ficha = ean_in)`, pois, num cenário com acessos concorrentes à base de dados, dois ou mais equipamentos poderiam ficar com o mesmo número de exemplar. Para garantir que cada invocação de *registra\_equipamento* obtém um número de exemplar único, sugere-se o uso de `SELECT...FOR UPDATE` sobre os dados estritamente necessários da tabela *ficha*, sendo considerado excessivo o uso de `LOCK TABLE`.

cliente, com um número sequencial gerado automaticamente<sup>2</sup> e com a data atual, ficando a compra associada a essa fatura. A data da fatura tem de ser posterior à data de colocação do equipamento na loja (RIA 12). A função devolve o número da fatura da compra.

`remove_compra(fatura_in, ean_in := NULL, exemplar_in := NULL) -> NUMBER`

Remove a compra de um equipamento usado no âmbito de uma fatura, ou seja, retira uma linha do talão da fatura. Se a fatura ficar sem linhas no talão, também é removida. Se o EAN-13 e o número de exemplar do equipamento forem omitidos, deve ser chamada a própria função `remove_compra` para cada um dos equipamentos mencionados na fatura. A remoção da compra de um equipamento faz com que este volte a ficar disponível na loja. A função devolve o número de linhas que ainda constam no talão da fatura, ou zero se a fatura tiver sido removida.

`remove_equipamento(ean_in, exemplar_in)`

Remove o equipamento usado com EAN-13 e número de exemplar. Se o equipamento tiver sido comprado por um cliente, essa compra também deve ser removida, por invocação de `remove_compra`.

`remove_ficha(ean_in)`

Remove a ficha de equipamento com EAN-13, bem como todos os exemplares de equipamentos usados que estão associados à ficha. A remoção dos equipamentos deve ser feita invocando `remove_equipamento`.

`remove_cliente(nif_in)`

Remove o cliente com NIF, bem como todas as suas compras de equipamentos usados. A remoção das compras deve ser feita invocando `remove_compra`.

`lista_compras(cliente_in) -> CURSOR`

Devolve um cursor com as compras de equipamentos realizadas por um cliente (identificado pelo seu NIF), por ordem descendente da data da fatura e do preço do equipamento. Cada linha do cursor deve ter o número e a data da fatura, o EAN-13, número de exemplar, marca, modelo, e preço (que o cliente pagou) do equipamento.

## A ter em conta

- É fundamental que o pacote PL/SQL possa ser *compilado sem erros*.
- Os procedimentos e funções devem sinalizar o não cumprimento de restrições de integridade ou outras situações de erro através de *exceções com mensagens inteligíveis*.
- As exceções com origem em restrições de integridade declarativas (PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE, e CHECK) devem ser convertidas em *exceções com mensagens inteligíveis*. Pode ser útil usar `SQLERRM` e `REGEXP_SUBSTR` para extrair dados relevantes de uma mensagem de exceção. Também pode ser criado e reutilizado um procedimento privado do pacote, responsável pela conversão de exceções Oracle em exceções mais inteligíveis.

---

<sup>2</sup> O número de fatura deve ser gerado automaticamente por uma sequência Oracle, cujo comando de criação deve estar no início do *script* de demonstração do cenário de gestão de dados.

- A codificação do pacote *não deve repetir as condições que já estão em restrições de integridade declarativas* de tipo CHECK, pois isso causa problemas de manutenção.
- A especificação do pacote deve ter, logo no início, um bloco de comentários com a *lista de todas as exceções* que podem ser lançadas pelos seus procedimentos e funções.
- O pacote PL/SQL deve estar comentado, as variáveis devem ter *nomes inteligíveis*, e devem ser respeitadas as regras de alinhamento dos blocos de código.
- Nos procedimentos de remoção de dados, deve ser lançada uma *exceção se os dados a remover não existirem*. Pode ser útil usar SQL%ROWCOUNT.
- As *operações do pacote não devem terminar transações* com COMMIT ou ROLLBACK, pois essa é uma responsabilidade do contexto em que as transações foram iniciadas. Por exemplo, o *script* de demonstração inicia uma ou mais transações (uma de cada vez), devendo ser aí que se colocam os comandos COMMIT e ROLLBACK.
- O cenário de gestão de dados deve ter invocações de *todos os procedimentos e funções do pacote* PL/SQL, podendo haver uma parte do cenário dedicada a invocações com argumentos que causem propositadamente o lançamento de exceções.

## Estrutura do relatório

- O relatório deve ser constituído por três ficheiros: a especificação do pacote (PKG\_L0JA.PKS), a codificação do pacote (PKG\_L0JA.PKB), e o *script* de demonstração (PKG\_L0JA.SQL).
- No início dos ficheiros deve estar, em comentário, a sigla da unidade curricular e o ano letivo, a etapa do projeto, o número do grupo, e o nome, número, e turma dos alunos.
- No início do *script* de demonstração deve constar, para cada aluno, um sumário das contribuições para o trabalho e uma proporção aproximada do seu esforço face ao total.

## Entrega do relatório

- O relatório deve ser entregue num ficheiro com nome SIBD-2425-GXX-E4.ZIP, onde XX representa o número do grupo, com dois dígitos (ex. 01).
- A entrega do relatório é feita através de uma atividade na página de entrada do Moodle da unidade curricular.
- O prazo de entrega do relatório é **13 de dezembro de 2024** (sexta-feira).

Bom trabalho nesta etapa do projeto!