

# Projeto de BD – Entrega 3

Professor: Gonalo Babo Freire

Ant3nio Oliveira – 100721 – 20 Horas (50%)

Filipe Resendes – 96859 – 20 Horas (50%)

Martim Antunes - 90753 – 0 Horas (0%)

Grupo 100 – Turno BDL13

## queries.sql

Ficheiro com as consultas sucintas de SQL das 4 situações pedidas.

## populate.sql

Ficheiro com as instruções para criar o esquema de Base de Dados correspondente ao esquema relacional apresentado e o carregamento para preencher essa mesma Base de Dados com dados teste.

## ICs.sql

Ficheiro com o código usado para implementar as restrições de integridade com as extensões procedimentais SQL (Stored Procedures e Triggers).

## view.sql

Ficheiro com as instruções para criar uma vista que resume as informações mais importantes sobre as vendas.

## analytics.sql

Ficheiro com as consultas OLAP que permitem analisar os número de artigos vendidos em duas situações.

# Indices

## 7.1

```
CREATE INDEX retail_frutos ON responsavel_por USING HASH (nome_cat);
```

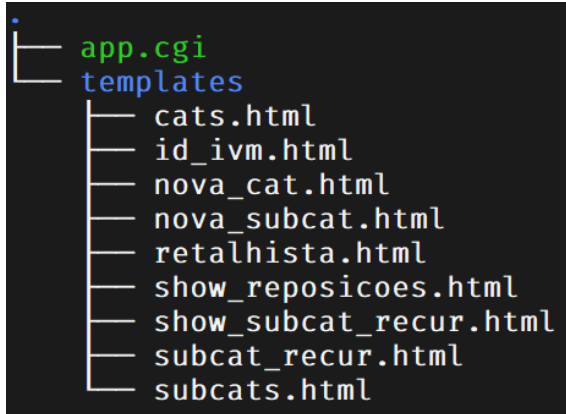
A razão pela qual ter escolhido apenas a segunda igualdade é devido ao facto de a primeira igualdade fazer uma comparação, no pior caso, entre um atributo tin da tabela retalhista e todos os outros atributos tin da tabela responsavel\_por. Isto leva a um tempo de operação quadrática no pior caso, enquanto que obter as linhas sobre a qual o atributo nome\_cat possui o valor 'Frutos' tem no pior caso uma complexidade linear. Para além disso o método HASH foi preferido sobre BTREE uma vez que trata-se de uma comparação de igualdade e não de um intervalo.

## 7.2

```
CREATE INDEX produto_desc ON produto USING BTREE (desc);
```

Pela mesma razão da query anterior, a segunda igualdade toma prioridade no index dado que o custo operacional de comparar cada valor de duas colunas é pior que encontrar o valor sobre o qual uma coluna se encontra compreendido. Assim, se a wildcard for interpretada como uma compreensão, onde os valores que o atributo desc da tabela produto toma, qualquer valor maior que 'A' será selecionado. Acontece que um index da forma BTREE é o mais indicado ao que toca a intervalos de valores. Sendo que neste caso, o intervalo será Qualquer palavra com A e qualquer outra sequência de letras.

# Arquitetura da aplicação



URL base onde todas as páginas se inserem (basta adicionar a palavra da ação a este link):

<https://web2.tecnico.ulisboa.pt/ist1100721/app.cgi>

5.a. Ver categorias: /cats

Ver sub-categorias: /sub\_cats

Eliminar categoria: Coluna Remove em /cats

Eliminar sub-categoria: Coluna Remove em /sub\_cats

Adicionar categoria: Botão “Adicionar Categoria”

Adicionar sub-categoria: Botão “Adicionar Sub Categoria”

5.b. Adicionar e remover um retalhista: /retalhista

5.c. Listar todos os eventos de reposição da IVM, agregados por categoria: /id\_ivm

5.d. Pesquisar todas as sub-categorias de uma super\_categoria: /sub\_cat\_recurativo