

Dados Projeto de

Grupo 64

Turno L08

Prof. Miguel Garção Silva

Aluno	Esforço (horas)	Esforço Relativo
Bernardo Couto Melo (99057)	12 horas	33%
Guilherme Marques Pascoal (99079)	12 horas	33%
Pedro Cerqueira Lobo (99115)	12 horas	33%

1 Desenvolvimento da Aplicação

A aplicação foi desenvolvida utilizando *HTML*, *CSS*, *Python* e o framework *Flask*.

Todos os ficheiros necessários para o funcionamento da aplicação encontram-se na pasta *web*. Os ficheiros *HTML* estão na subdiretoria **templates**. Existe também um ficheiro *CSS* na subdiretoria **static** que visa tornar mais apelativa a interface da aplicação. Os detalhes da base de dados encontram-se no cabeçalho do ficheiro **app.cgi**.

A página inicial corresponde ao ficheiro **menu.html** apresenta ao utilizador os submenus onde é possível realizar todas as operações exigidas.

O requisito a) corresponde às opções:

- Adicionar Categoria
- Remover Categoria
- Adicionar Sub-Categoria
- Remover Sub-Categoria

O requisito b) corresponde às opções:

- Adicionar Retalhistas
- Remover Retalhistas

O requisito c) corresponde à opção:

- Listar Eventos de Reposição

O requisito d) corresponde à opção:

- Listar Sub-Categorias

Ao realizar uma operação com êxito não é apresentada nenhuma mensagem de sucesso ao utilizador. Em vez disso, o utilizador é reencaminhado para o menu principal. Se uma operação não for executada com êxito, é mostrada ao utilizador a mensagem de erro emitida pelo SGBD.

1.1 Menu Principal

O utilizador é recebido com um menu principal, a partir do qual pode aceder a todos os submenus, onde pode realizar todas as operações suportadas.

1.2 Adicionar Categoria

O utilizador é apresentado com uma caixa de texto, onde insere o nome categoria a ser criada e um menu de opção, onde especifica se esta se trata de uma categoria simples ou super categoria. Ao carregar no botão **Confirmar**, a categoria é adicionada ao sistema.

1.3 Remover Categoria

É apresentada uma tabela com todas as categorias, com um botão **Remover**, junto a cada uma delas. Ao carregar nesse botão, o utilizador remove a categoria do sistema.

1.4 Adicionar Sub-Categoria

São apresentadas duas tabelas. Na primeira seleciona-se apenas uma das sub-categorias e na segunda apenas uma das super-categorias. Ao carregar no botão **Confirmar**, a sub-categoria selecionada é guardada como sub-categoria da super-categoria selecionada.

1.5 Remover Sub-Categoria

É apresentada uma tabela com as várias relações **Super-Categoria/Sub-Categoria**. Ao clicar no botão **Remover**, junto a cada uma destas relações, a cada uma das entradas da lista, a relação entre as duas categorias apresentadas é removida do sistema.

1.6 Adicionar Retalhista

São apresentadas uma caixa de input numérico, que recebe o TIN do retalhista, uma caixa de texto para inserir o nome do mesmo e, por fim, uma tabela. A tabela é originada através de um produto cartesiano de todas as IVM's, ainda sem retalhista responsável, com todas as categorias registadas no sistema. Isto permite uma mais fácil utilização da aplicação por parte do utilizador, pois diminui os possíveis erros que este pode cometer ao introduzir os vários campos requisitados. Contudo, incorretamente, o utilizador pode selecionar a mesma IVM, para duas categorias diferentes. Para evitar isto, é apresentado ao utilizador um aviso, imediatamente acima da tabela. Selecionar uma ou mais entradas da tabela, equivale a tornar o retalhista responsável pela categoria selecionada, na IVM em questão. Ao carregar no botão **Confirmar**, é adicionado um retalhista, com todas essas responsabilidades.

1.7 Remover Retalhista

É apresentada uma lista com os vários retalhistas. Ao clicar no botão **Remover**, junto a cada um deles, o retalhista é removido do sistema.

1.8 Listar Eventos de Reposição

É apresentada uma lista com as várias IVM's. Ao clicar no botão **Listar**, junto a cada uma das IVM's, o utilizador é reencaminhado para uma página, onde são listados todos os eventos de reposição que dizem respeito a essa IVM.

1.9 Listar Sub-Categorias

É apresentada uma lista com as várias super-categorias existentes no sistema. Ao clicar no botão **Listar**, junto a cada uma delas, o utilizador é reencaminhado para uma página onde são listadas todas as sub-categorias da super-categoria selecionada, a todos os níveis de profundidade.

1.10 Link

<https://web2.ist.utl.pt/ist199115/app.cgi/>

2 Índices

Consulta 7.1

```
CREATE INDEX idx_tin_nome_cat ON responsavel_por USING HASH(tin, nome_cat);  
CREATE INDEX idx_nome ON retalhista(nome);
```

- As condições de filtragem sugerem a utilização de um índice em **tin**, para ambas as tabelas ou a utilização de um índice em **nome_cat** para a tabela **responsavel_por**. Obtámos por criar um índice de chave composta em (**tin**, **nome_cat**) em **responsavel_por**. Aproveitando o índice criado implicitamente na chave primária de **tin** em **retalhista**, a condição de filtragem será otimizada. O índice será não ordenado e de dispersão (*HASH*), devido à igualdade da condição de filtragem.
- Poderia ser apenas utilizado um índice de dispersão apenas sobre o atributo **nome_cat** ou apenas sobre o atributo **tin** de **responsavel_por**, dependendo da seletividade de cada coluna.
- O uso de **DISTINCT** provoca uma ordenação, pelo que o uso de um índice de **nome** em **retalhista** pode acelerar a consulta. O índice será não ordenado e Btree.

Consulta 7.2

```
CREATE INDEX idx_desc ON produto(desc);  
CREATE INDEX idx_nome_ean ON tem_categoria(nome, ean);
```

- Poderá ser criado um índice de **desc** em **produto**, de modo a tirar partido de que a **desc** pretendida começa por 'A'. O índice será não ordenado e Btree.
- A consulta beneficiaria também de um índice de (**nome**, **ean**) em **tem_categoria**, de modo a otimizar a operação de agregação. O índice será não ordenado e Btree.
- É de notar que o índice criado implicitamente sobre (**ean**, **nome**) em **tem_categoria** não traria o mesmo benefício que o índice anteriormente referido pois, neste **nome** surge depois de **ean**.