**Projeto de Bases de Dados**

**3ªEntrega**

**Turno Prático 10,** Prof. Flávio Martins

|  |  |  |  |
| --- | --- | --- | --- |
| Grupo 199 | | | |
| Nome | Nº | Esforço (Horas) | Contribuição (%) |
| Gonçalo Carvalho | 99227 | 6 | 33.(3) % |
| Jaime Costa | 95595 | 6 | 33.(3) % |
| Pedro Mateus | 99306 | 6 | 33.(3) % |

# Base de Dados

A base de dados foi baseada no modelo relacional presente no anexo A, optámos por todos os atributos como não nulos, de modo a evitar que as tabelas tenham entradas vazias em qualquer uma das suas colunas.

É usada uma constraint para verificar que uma categoria simples não está contida na tabela das super categorias, durante a inserção da mesma.

São utilizados um conjunto de **TRIGGERS** para automatizar o carregamento da base de dados:

(1) TRIGGER exists\_super\_categoria: Assim que inserimos um novo par na relação **Tem\_outra,** o argumento passado no argumento super\_categoria é passado da tabela **Categoria\_simples** para a tabela **Super\_categoria**, caso esta seja uma categoria simples (não sendo necessário inserir valores na tebla **Super\_categoria.**

(2) TRIGGER insert\_into\_categoria: trata de inserir um valor na tabela **categoria**, antes de este ser inserido na tabela **Categoria\_simples**.

(3) TRIGGER produto\_categorizado: trata de introduzir na tabela **Tem\_categoria** uma relação produto, categoria, depois de um produto ser introduzido, na tabela Produto, garantindo assim a (RI-RE6): O valor do atributo ean existente em qualquer registo da relação produto tem de existir também no atributo ean da relação **Tem\_categoria**.

São utilizados um conjunto de **TRIGGERS** para automatizar o processo de Remover Categoria e Remover Retalhista, proporcionando uma melhor facilidade de escrita de queries muito complexas no ficheiro app.cgi (aplicação FLASK):

(1) TRIGGER erase\_categoria: trata de apagar todas as entradas nas tabelas **Evento\_reposicao**, **Planograma**, **Tem\_categoria**, **Tem\_outra**, **Prateleira**, **Produto**, **Responsavel\_por** e **Categoria\_simples** ou **Super\_categoria** (dependendo da classificação da categoria) que possuam a categoria a seer removida antes de apagar a entrada da tabela **Categoria.** Este trigger facilita o passo de Remover Categoria na Aplicação Web.

(2) TRIGGER erase\_retalhista: trata de apagar todas as entradas nas tabelas **Responsavel\_por** e **Evento\_reposicao** que possuam o retalhista a ser removido.

# Arquitetura da Aplicação Web

A aplicação web(<http://web2.tecnico.ulisboa.pt/ist199227/app2.cgi/>) está estruturada em volta de uma página principal(no ficheiro *home.html*) a partir da qual é possível aceder a todas as funcionalidades pedidas pelo enunciado, em seguida vamos descrever a estrutura do site para cada ponto do enunciado mantendo em mente que todos os caminhos de funcionalidade se iniciam nesta página inicial.

## Inserir uma Categoria

Para inserir uma nova categoria escolhe-se a opção **Adicionar Categoria** na página principal sendo-se redirecionado para uma página(*add\_category.html*) que contém um campo de input em que é pedido o nome da nova categoria, após a inserção no campo de input deste atributo e após carregar no botão **Submit,** é-se redirecionado, novamente para uma página que mostra a query usada para inserir a nova categoria.

## Remover uma Categoria

Para remover uma categoria escolhe-se a opção **Remover Categoria** na página principal sendo-se redirecionado para uma página(*remove\_category.html*) que contém um tabela com duas colunas, na primeira coluna encontram-se as categorias presentes na base de dados , na segunda coluna encontram-se hiperlinks cada um relacionado com a categoria na mesma linha ao se carregar nesse hiperlink, a categoria correspondente é removida e somos redirecionados para uma página(*update\_remove\_category.html*) que mostra texto que informa que categoria foi removida da base de dados.

## Inserir uma Subcategoria

Para inserir uma subcategoria escolhe-se a opção **Adicionar Subcategoria**, sendo-se redirecionado para uma página(*add\_subcategory.html*) com uma tabela que, como no ponto anterior , tem as categorias presentes no sistema atualmente, e hiperlinks correspondentes a cada uma delas, ao escolher uma certa categoria, significa que vamos inserir uma nova subcategoria que terá a categoria escolhida como super-categoria, somos então redirecionados para uma nova página(*choose\_subcategory.html*) com uma tabela semelhante à anterior na qual vamos escolher que categoria será inserida como subcategoria da super-categoria escolhida anteriormente, depois desta ultima seleção seremos direcionados para uma página que informa que categoria foi adicionada como subcategoria de que super-categoria.

## Remover Subcategoria

Para remover uma subcategoria escolhe-se a opção **Remover Subcategoria,** sendo-se redirecionado para uma página(*remove\_subcategory.html*) onde se encontra uma tabela com duas colunas, a primeira coluna contém as super-categorias atualmente presentes na base de dados, e a segunda coluna, hiperlinks, um para cada super-categoria, carregando num destes hiperlinks, significando que queremos remover uma subcategoria desta super-categoria, somos redirecionados, para outra pagina(*choose\_subcategory\_to\_remove.html*) onde se encontra uma tabela com duas colunas, a primeira contém as subcategorias, da super-categoria escolhida anteriormente e a segunda hiperlinks, um para cada subcategoria , ao carregar num deles, a aplicação vai remover essa subcategoria das subcategorias da super-categoria a que pertencia anteriormente e redirecionar o utilizador para uma nova página que mostra um texto em que se encontra a subcategoria removida e a super-categoria de onde foi removida.

## Inserir. Retalhista

Para inserir uma nova categoria escolhe-se a opção **Adicionar Retalhista** na página principal sendo-se redirecionado para uma página(*add\_retalhista.html*) que contém dois campos de input em que é pedido o tin e o nome do novo retalhista, após a inserção de valores nos campos de input após carregar no botão **Submit,** é-se redirecionado, novamente para uma página que mostra a query usada para inserir o novo retalhista.

## Remover. Retalhista

Para remover um retalhista escolhe-se a opção **Remover Retalhista** na página principal sendo-se redirecionado para uma página(*remove\_retalhista.html*) que contém um tabela com três colunas, na primeira coluna encontram-se os tin’s e na segunda coluna os nomes de todos os retalhistas atualmente presentes na base de dados, na terceira coluna encontram-se hiperlinks cada um relacionado com o retalhista na mesma linha ao se carregar nesse hiperlink, o retalhista correspondente é removido e somos redirecionados para uma página(*update\_remove\_retalhista.html*) que mostra texto que informa que retalhista foi removida da base de dados.

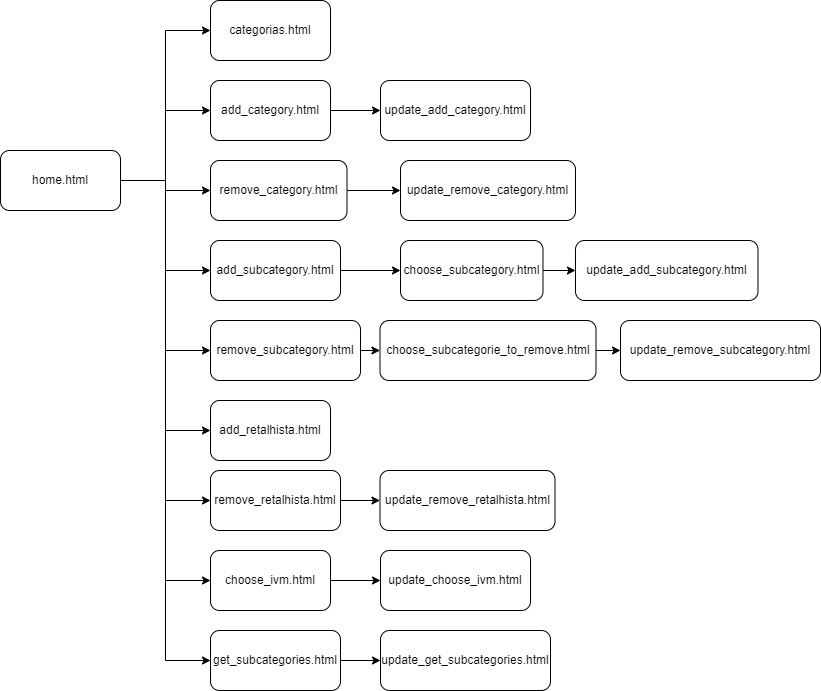
## Eventos de reposição por IVM e número de unidades repostas por categoria

Para aceder a esta funcionalidade escolhe-se a opção **Eventos por ivm e categoria** sendo-se redirecionado para uma nova página (*choose\_ivm.html*), em que se encontra uma tabela com 3 colunas em que se encontram nas primeiras duas colunas o número de série e o fabricante, respetivamente de todas as ivm’s atualmente presentes na base de dados, na terceira coluna econtram-se hiperlinks como nas tabelas acima, ao carregar em um deles estamos a selecionar a ivm a que ele está associado para ver o total de unidades repostas por categoria nessa ivm, para este propósito somos redirecionados para uma nova página que contém a tabela com essa informação(*update\_choose\_ivm\_.html*)

## Subcategorias de uma Super-Categoria

Para aceder a esta funcionalidade escolhe-se a opção **Subcategorias de uma Super-categoria,** somosentão redirecionados para uma nova página(*get\_subcategories.html*) onde se encontra uma tabela com duas colunas, a primeira é composta por todas as super-categorias atualmente presentes na base de dados, a segunda coluna como já referido acima contém os hiperlinks que ao serem clicados nos redirecionam para uma nova página(*update\_get\_subcategories.html*) em que se encontra a tabela com as subcategorias da super-categoria correspondente ao hiperlink que foi carregado.

De seguida apresentamos um esquema que representa a ligação entre as diferentes páginas html acima referidas, para facilitar a leitura.



# Indicies

## 7.1

Esta query pode ser otimizada ao criando-se um índice hash sobre o atributo tin da tabela retalhista. Isto permite que o join entre tabelas possa usar um inner index scan.É possível também melhorar a query com um outro indíce hash sobre o atributo nome\_cat da tabela responsavel\_por. Isto permite usar um index only scan para verificar a igualdade. O facto de as condições serem igualdades torna o uso de índices hash preferível.



## 7.2

Esta query pode ser otimizada criando-se um índice hash sobre o atributo cat da tabela produto,facilitando o join entre as duas tabelas. Pode-se também criar um segundo índice B+ Tree no atributo descr do produto, pois a operação "like 'A%'"é tornada mais eficiente quando as descrições do produto já se encontram agrupadas alfabeticamente.

