



# Database Project, Part 3

Group: 12

Shift: 4<sup>a</sup>f 16:30-18:00

Professor: Paulo Carreira

Student	Relative Contribution	Total Effort
Gonalo Veiga, 96738	33 %	12 h
Pedro Carmine, 95493	34 %	12 h
Diogo Baptista, 96733	33 %	12 h

# Architecture of the web application

Link of the application:

<http://web2.tecnico.ulisboa.pt/ist195493/homepage.cgi>

Ao entrar no link, temos uma página inicial (homepage.cgi) que contém todas as funcionalidades do projeto, separadas por alíneas.

Na alínea a), para inserir categoria, é pedido o nome da categoria e a página redireciona para o category\_update.cgi e para remover vai para remove\_category\_update.cgi, que remove qualquer categoria. Para inserir uma sub categoria é redirecionado para sub\_category.cgi.

Na alínea b), para inserir um produto é pedido os inputs necessários e a página é redirecionada para o insert\_product.cgi. Para remover vai para o remove\_product.cgi.

Na alínea c), é pedido um ean do produto e ao clicar no botão, a página é redirecionada para o replenish\_event.cgi.

Na alínea d), é pedido um ean do produto e ao clicar no botão, a página é redirecionada para o `description.cgi`. Dentro desta página é pedida uma nova descrição e leva então a `description_update.cgi`.

E para a alínea e), é pedida uma super categoria e ao clicar no botão, a página é redirecionada para o `list_categories.cgi`.

## Indexes

5.1)

```
CREATE INDEX product_category_idx ON product  
USING HASH(category);
```

Visto que se trata de uma query de igualdade, faz mais sentido usar Hash para este tipo de query, porque divide todas as categorias em blocos e aponta para o bloco procurado, de forma eficiente.

5.2)

Neste caso (BTree), para melhorar a execução da query dada seria mais rápido contar quantos suppliers secundários cada produto tem e organizar o índice pelo número obtido, para que, ao executarmos a query, só tivéssemos que buscar os produtos com um COUNT superior a 1.