

Projeto BD - Parte 3

Grupo: 202

Professor: Miguel Garção Silva

Laboratório: L08

**Percentagem relativa de contribuição
e
Esforço total de cada membro do grupo**

Nome e número do aluno	%Relativa de Contribuição	Esforço Total (horas)
Francisco Guilherme 99069	25	20
Henrique Anjos 99081	25	20
Tiago Caldas 99125	25	20
Vasco Vaz 99133	25	20

5. Desenvolvimento da Aplicação (Explicação da arquitetura web)

- a) Na página inicial clicar em CATEGORIAS, o que leva a uma nova página (menu categorias), aí inserir o nome da categoria que se quer inserir/remover e clicar no botão respetivo à ação que se quer realizar (i.e Inserir Super Categoria, Inserir Categoria Simples, Remover Categoria).
- b) Na página inicial clicar em RETALHISTAS, o que leva a uma nova página (menu retalhistas), aí para inserir Retalhistas preencher todos os campos pedidos (TIN, Nome, Categoria, Num Serie, Fabricante) e em seguida clicar no botão Inserir Retalhista. Para remover Retalhistas apenas é necessário preencher o campo do TIN e em seguida clicar no botão Remover Retalhista. Nota: se preencher todos os campos para remover um retalhista, esses vão ser ignorados apenas dando relevância ao TIN.
- c) Na página inicial clicar em EVENTOS DE REPOSIÇÃO, o que leva a uma nova página (menu eventos de reposição), aí inserir o número de série da IVM e o seu fabricante e clicar em Listar.
- d) Na página inicial clicar em CATEGORIAS, o que leva a uma nova página (menu categorias), aí inserir o nome da categoria da qual quer listar as sub-categorias a todos os níveis de profundidade e clicar no botão Listar sub-categorias

Para além disto em todas as páginas (exceto a inicial) existe um botão VOLTAR que retorna à página anterior.

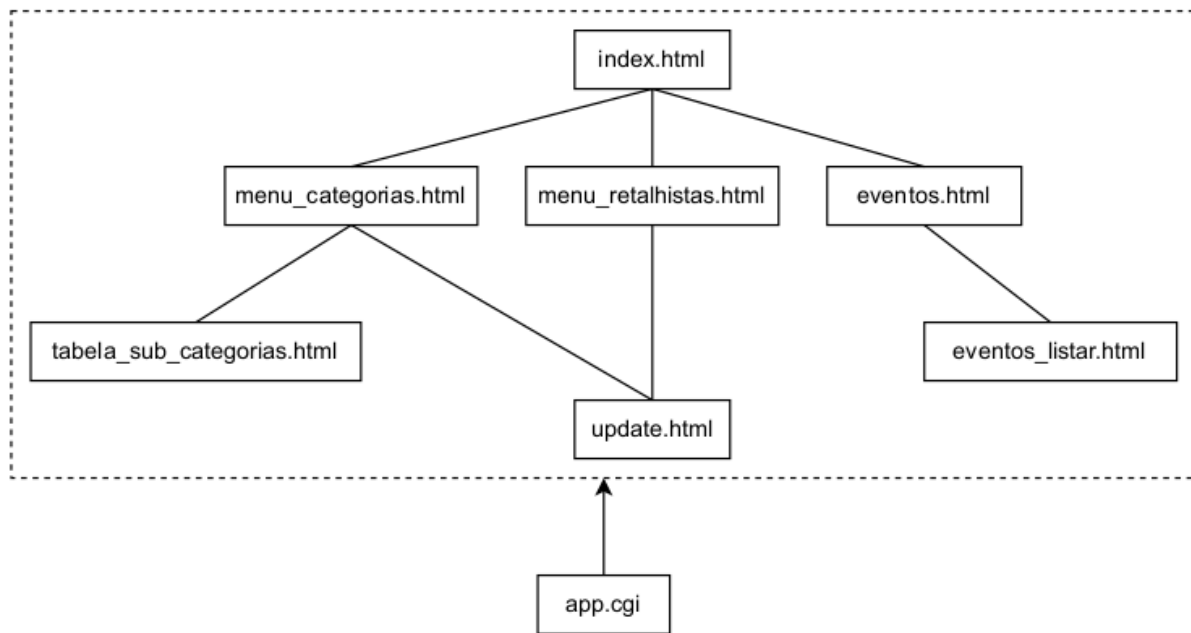
Link app: <https://web2.tecnico.ulisboa.pt/ist199133/app.cgi/>

Relações entre arquivos

Foram criados vários ficheiros: eventos.html, eventos_listar.html, index.html, menu_categorias.html, menu_retalhistas.html, tabela_sub_categorias.html, update.html e app.cgi.

A app.cgi relaciona os todos ficheiros html entre eles, com a base de dados e com queries.

Em baixo está um gráfico com as relações entre todos os nossos ficheiros.



7. Índices

- 1) O índice escolhido foi o hash no tin do retalhista. Logo o join das tabelas pode usar um inner index scan. Adicionando depois à query um segundo índice, desta vez à tabela responsavel_por na coluna nome_cat, para responder à condição sobre esta coluna com um index only scan. Ainda assim, a utilização de índices hash é mais adequada, pois as condições são apenas iguais e observamos que foi o index que demorou menos tempo após aplicarmos o comando EXPLAIN ANALYSE da query.

SQL:

```
CREATE INDEX tin_index ON retalhista using hash(tin);
```

```
CREATE INDEX responsavel_por_index ON responsavel_por using hash(nome_cat);
```

- 2) O primeiro índice desta query é do tipo B+ Tree, agrupado sobre a coluna cat da tabela produto, para facilitar a condição sobre esta coluna e o group by. Com a criação de um segundo índice sobre o nome da tabela tem_categoria, otimiza o join das duas tabelas. Enquanto que o primeiro índice deverá ser agrupado, não existe qualquer vantagem em agrupar o segundo índice, portanto um hash foi preferido.

SQL:

```
CREATE INDEX nome_index ON produto (cat);
```

```
CREATE INDEX nome_index_2 ON tem_categoria using hash (nome);
```