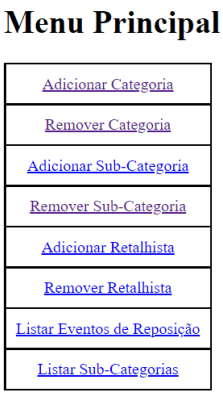
# ARQUITETURA:

A nossa aplicação tem um layout bastate simples e intuitivo. Em cada página visitada, caso se queira regressar à página anterior ter-se-á que usar o botão de retrocesso do browser em uso.

Menu principal:

O menu principal tem 8 opções à escolha:




1. Adicionar categoria: adiciona uma categoria nova (simples ou super) que não é sub-categoria de nenhuma outra.
2. Remover categoria: remove uma categoria que não esteja ligada a nenhuma outra
3. Adicionar Sub-Categoria: adiciona uma categoria nova (simples ou super) que tem que ser sub-categoria de outra.
4. Remover Sub-Categoria: Remove uma categoria que seja sub-categoria de outra.
5. Adicionar Retalhista: adiciona um retalhista
6. Remover Retalhista: remove um retalhista
7. Listat Eventos de Reposição: Lista as unidades repostas duma dada categoria duma IVM especificada
8. Listar Sub-Categorias: Lista as sub-categorias duma dada categoria

## 1. Adicionar Categoria


## 2. Remover Categoria


Clicar em Remover para remover uma categoria

## 3. Adicionar Sub-Categoria

**Adicionar sub categoria**

Nome da Categoria:

Categoria Simples  Super Categoria

Categoria Pai: Pastilhas

Confirmar

Mesmo processo que no ponto 1., tendo uma secção para selecionar a categoria Pai.

## 4. Remover Sub-Categoria: Mesmo processo que no ponto ponto 2.

## 5. Adicionar Retalhista

**Adicionar Retalhista**

Nome do Retalhista:

TIN do Retalhista:

Confirmar

Colocar todos os dados do retalhista e confirmar

## 6. Remover Retalhista

**Remover Retalhista**

| TIN | Nome | Remover |
|---|---|---|
| 1 | João | Remover |
| 2 | Carlos | Remover |
| 3 | Pedro | Remover |
| 4 | Teresa | Remover |
| 5 | Joana | Remover |
| 6 | Carolina | Remover |

Processo semelhante aos pontos 2. e 4..

## 7. Listar Eventos de Reposição

**IVM's**

| Número de Série | Fabricante | Listar |
|---|---|---|
| 1 | Maquineiros | Listar eventos |
| 2 | Maquineiros | Listar eventos |
| 3 | SmartVM | Listar eventos |
| 4 | Maquineiros | Listar eventos |
| 5 | SmartVM | Listar eventos |
| 6 | Compraqui | Listar eventos |

Clicar em listar eventos para mostrar todas as unidades repostas das categorias

**Eventos IVM**

| Categoria | Unidades |
|---|---|
| Pastilhas Sabores | 8 |

## 8. Listar Sub-Categorias

**Lista de Super Categorias**

| Hierarquias | |
|---|---|
| Pastilhas | Listar |
| Pastilhas Elásticas | Listar |
| Batatas | Listar |
| Refrigerante | Listar |
| Chocolate | Listar |
| Chocolate Negro | Listar |
| Chocolate de Leite | Listar |
| Bolachas | Listar |
| TEST!@#$%^ | Listar |

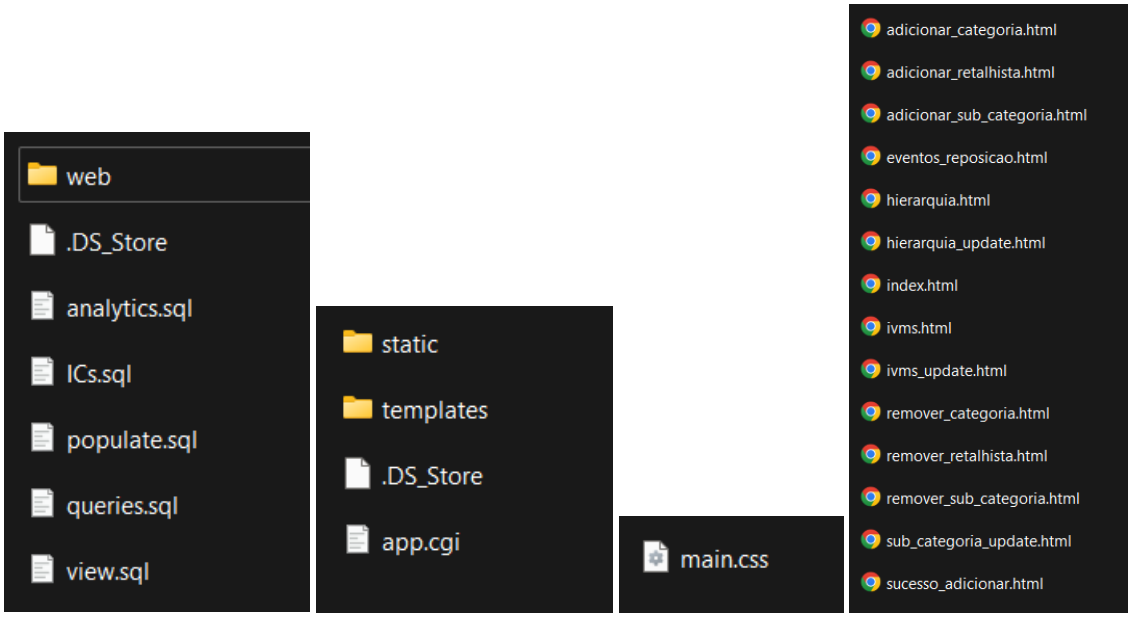
Clicar em Listar para mostrar todas as sub-categorias da categoria selcionada

**Lista de Sub Categorias**

| Super Categoria | Sub Categorias |
|---|---|
| Pastilhas Elásticas | Pastilhas Sabores |
| Pastilhas Elásticas | Pastilhas Diferentes |
| Pastilhas Elásticas | Pastilhas s/ Açúcares |

# RELAÇÕES ENTRE OS FICHEIROS:

Listagem de todos os ficheiros e diretorias por ordem descendente descendente do que está listado nas imagens seguintes:




# ÍNDICES:

```
CREATE INDEX idx_tin ON responsavel_por USING HASH(tin);
```

Para a primeira query, uma vez que nenhum dos atributostestados é uma chave primária e por cosneguinte não indexar automaticamente e que a relação responsavel_por é testada por igualdades tanto para o atributo tin, como para nome_cat. Sendo igualdades a solução lógica para indexar a query apresentada é fazê-lo por  hashing. Sendo que tanto o nome_cat, como o tin não são atributos primários seria redundante indexar a query por ambos os atributos. Foi escolhido o tin para ser indexado por ser um inteiro e ser de mais fácil comparação que uma string.

```
CREATE INDEX idx_nome ON produto USING HASH(cat);
CREATE INDEX idx_descr ON produto USING BTREE(descr);
```

Para a segunda query  usou-se uma composição de índices, pois ambos os índices não são chaves primárias. Para o primeiro índice sendo que se está a testar uma condição através duma comparação direta recorreu-se à indexação através de hashing. Para o segundo índice sendo que poderá haver várias combinações, pois está-se a usar o 'LIKE' podendo associar a várias strings que cumpram a restrição. Posto isto ao usar  a indexação de B-Tree faz-se uma procura muito mais rápidsa por todas as combinações que satisfaçam a condiçam da query.