

## 0. Objectivos

Pretende-se implementar uma loja electrónica de produtos de um tipo pré-determinado (por exemplo: Computadores). A loja deverá disponibilizar aos potenciais clientes as seguintes funcionalidades:

- Listagem completa de produtos disponíveis.
- Ordenação das listagens por preço, descrição, etc
- Pesquisa por preço, descrição, ...
- Apresentação detalhada de produto com imagem
- Selecção de produtos para carrinho de compras
- Listagem de carrinho de compras e *ckeck-out*
- Apresentação Multilingue (Português + Inglês + ....)
- Aspeto gráfico cuidado, definido através de ficheiro CSS

Outras funcionalidades propostas:

- Listagens de produtos com imagens (icons)
- Listagem de produtos hierarquizada por classe / sub-classe.
- Paginação das listagens
- Pré-registo de clientes
- Rastreio de encomenda
- *Back-office* para consulta e gestão de encomendas
- *Back-office* para edição de produtos
- *Back-office* para gestão de stocks

O serviço deverá ser implementado utilizando funcionalidades básicas PHP-MySQL até 04-Jan-2018. A implementação proposta neste guião é meramente exemplificativa.

## 1. Base de dados

Crie uma nova tabela para registo de produtos na sua Base de Dados localizada no servidor `ave.dee.isep.ipp.pt`:

```
SHOW DATABASES;  
USE cinemXXXXXX;  
CREATE TABLE Produtos(  
    Sku CHAR(15) NOT NULL PRIMARY KEY,  
    Marca CHAR(15),  
    Modelo VARCHAR(20),  
    Processador VARCHAR(20),  
    Monitor FLOAT,  
    Ram FLOAT,
```

```

        Disco FLOAT,
        Peso FLOAT,
        Autonomia FLOAT,
        Preco DECIMAL (6,2)
    );

```

Introduza alguns registos de teste na sua tabela.

```
cat ~/jml/WWW/cinem/lab11/produtos.sql | mysql -u nr_aluno cinemXXXXXX -p
```

## 2. Listagem de produtos

Exemplo (lista.php):

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >
<html>
  <head>
    <title>Nome da Loja</title>
  </head>
  <body>
<?php
    print "    <h1>Lista de Produtos</h1>\n";
    $conn= new mysqli("localhost", "00000", "yyyyy", "cinem00000");
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $query = "SELECT Sku, Marca, Modelo, Preco FROM Produtos";
    $result = $conn->query($query);
    if($result->num_rows > 0)
    {
        print "    <table> \n";
        while ( $row = $result->fetch_row() ) {
            print "        <tr> \n";
            foreach ( $row as $val )
                print "            <td>$val</td> \n";
            print "        </tr> \n";
        }//while
        print "    </table> \n";
    }
    else
        echo "0 results";
    mysqli_close($conn);
?>
  </body>
</html>

```

Acrescente uma linha de cabeçalho à tabela com o nome dos campos apresentados:

```

$row = $result->fetch_assoc();
print "    <tr> \n";
foreach ( $row as $name => $val )
    print "        <th>$name</th>\n";

```

```
print "          </tr> \n";
$result->data_seek(0);
```

Organize o seu código em funções. Arrume as funções que poderão ser reaproveitadas num ficheiro distinto (funcs.php) para que possa ser incluído usando a diretiva PHP **include**:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >
<html>
  <head>
    <title>Nome da Loja</title>
  </head>
  <body>
<?php
  Include("funcs.php");
  print "<h1>Lista de Produtos</h1>\n";
  $conn=LigaBD();
  ListaProdutos($conn);
?>
  </body>
</html>
```

Acrescente a cada produto da lista uma ligação para a página de detalhe do produto:

```
print "          <td> <a href='mostra.php?sku=$row[0]'+info </a> </td>\n";
```

## 3. Detalhe de produto

Exemplo (mostra.php):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >
<html>
<?php
  Include('funcs.php');
  $sku = $_GET['sku'];
  print "
  <head>
    <title>Nome da loja - Produto: $sku</title>
  </head>
  <body>\n";
  print "    <h1>Produto $sku</h1>\n";
  LigaBD();

  $query = "SELECT * FROM Produtos WHERE Sku='$sku'";

  $result = $conn->query($query);
  if($result->num_rows > 0)
  {
    $row = $result->fetch_assoc();
    print "<table>\n";
    foreach( $row as $fname => $val ) {
      print "  <tr><th>$fname</th><td>$val</td></tr>\n";
    }
    print "</table>\n";
```

```

}
    else
        echo "0 results";
mysqli_close($conn);

?>
</body>
</html>

```

Acrescente uma ligação que permita regressar à lista de produtos.

```
print "    <a href=lista.php> [Voltar &agrave; lista] </a>\n";
```

Inclua uma imagem ilustrativa do produto.

```
print "<img src=img/'.".$row['Sku'].".jpeg alt='Foto indisponivel'>\n";
```

## 4. Ordenação

Acrescente a possibilidade de ordenação por um campo chave especificado no pedido HTTP:

```

$ordem = $_GET['ordem'];
if ( $ordem != "" )          $query .= " ORDER BY $ordem";

```

Transforme os títulos das colunas em ligações para que o utilizador final possa seleccionar a chave de ordenação:

```

Function LinkOrdem($ordcamp)
{
    $txt = "<a href='?ordem=$ordcamp' class='titcoluna'> $ordcamp </a>";
    return $txt;
}
...
print "    <th>". LinkOrdem($name) . "</th>\n";

```

## 5. Pesquisa

5.1. Acrescente a possibilidade de pesquisa por uma chave especificada no pedido HTTP:

```

$filtro = $_GET['filtro'];
if ($filtro!="")
    $query.=" WHERE CONCAT(Sku,Marca,Modelo) LIKE '%$filtro%';

```

5.2 Introduza um formulário HTML com um campo que permita a especificação da chave de pesquisa:

```

print "<form method=GET action=lista.php>";
print "    Filtro:";
print "    <input name='filtro' value='".$_GET['filtro']."' size=8>";
print "    <input type=submit value='>'>";
print "</form>";

```

Verifique que não se encontra garantida a preservação da chave de ordenação quando se redefine o critério de pesquisa nem é mantido o critério de pesquisa quando se redefine a chave de ordenação.

Estes problemas de não preservação de estado serão resolvidos no capítulo 7, através da definição de uma sessão. Alternativamente, é possível manter o estado explicitamente, como se indica a seguir.

5.3. Para que o estado da ordenação seja mantido quando é alterada a chave de pesquisa, introduza-o num parâmetro *hidden* no formulário:

```
print " <input type=hidden name='ordem' value='$ordem'>";
```

5.4. Para que o estado da pesquisa seja mantido quando é alterada a chave de ordenação, inclua-o nas ligações de ordenação:

```
$ordem = $ordcamp . "&filtro=$filtro";  
$txt = "<a href='?ordem=$ordem' class='titcoluna'> $ordcamp </a>";
```

## 6. Estilo

Prepare um ficheiro com a definição CSS do estilo a aplicar nas suas páginas. Exemplo:

```
wget ave.dee.isep.ipp.pt/~jml/cinem/cinemovel/loja.css
```

No cabeçalho das suas páginas especifique o ficheiro CSS a utilizar:

```
<link href="loja.css" type="text/css" rel="stylesheet" >
```

## 7. Sessão

No início dos seus ficheiros PHP, inclua a directiva ***session\_start()*** para associar os diversos pedidos HTTP numa única sessão:

```
session_start();
```

Utilize o *array* `$_SESSION` para armazenar as mudanças de estado ordenadas através do *array* `$_GET`

```
foreach ( $_GET as $key => $val )  
    $_SESSION[$key] = $val;
```

Na construção da consulta SQL, passe a utilizar as variáveis de estado armazenadas no *array* `$_SESSION`.

```
if ($_SESSION['filtro']!="")  
    $query.=" WHERE CONCAT(Sku,Marca,Modelo) LIKE '%" . $_SESSION['filtro'] . "%'";  
if ($_SESSION['ordem']!="") $query .= " ORDER BY " . $_SESSION['ordem'];
```

Passe também a utilizar o *array* `$_SESSION` para a inicialização do campo de especificação da chave de pesquisa:

```
print " <input name='filtro' value='". $_SESSION['filtro'] . "' size=8>";
```

## 8. Linguagens

Prepare um ficheiro com uma tabela de tradução dos termos e frases utilizados para Português e Inglês.  
Exemplo ('ling.php'):

```
<?php
// ling.php
$LANGs['Marca']      = array('en'=>'Brand');
$LANGs['Preco']      = array('en'=>'Price', 'pt'=>'Pre&ccedil;o');
$LANGs['Modelo']     = array('en'=>'Model');
$LANGs['Descricao']  = array('en'=>'Description', 'pt'=>'Descri&ccedil;&atilde;o');
$LANGs['CoresEcra']  = array('en'=>'Display colors', 'pt'=>'Cores ecr&atilde;');
$LANGs['Peso']       = array('en'=>'Weight');
$LANGs['Carrinho']   = array('en'=>'Cart');
$LANGs['+Carrinho']  = array('en'=>'Add to Cart', 'pt'=>'Adicionar do Carrinho');
$LANGs['Unitario']   = array('en'=>'Unit price', 'pt'=>'Unit&aacute;rio');
?>
```

Crie uma função que receba um termo um termo a traduzir e retorne a sua tradução na língua seleccionada:

```
Function Traduz($txt)
{
    include ('ling.php');
    if ( isset( $LANGs[$txt][$SESSION['ling']] ))
        return $LANGs[$txt][$SESSION['ling']];
    return $txt;
}
```

Aplique a função Traduz na tradução dos títulos das colunas e dos outros textos utilizados:

```
print Traduz("Filtro:");
...
Function LinkOrdem($ordcamp)
{
    $txt="<a href='?ordem=$ordcamp' class='titcoluna'>".Traduz($ordcamp)."</a>";
    return $txt;
}

... etc
```

Na página inicial, acrescente ligações que permitam alterar o idioma da página:

```
print "    <p>Idioma: ";
print "        <a href='?ling=pt'>[Portugu&ecirc;s]</a>";
print "        <a href='?ling=en'>[English]</a>";
print "    </p>";
```

## 9. Carrinho de compras

Na página de apresentação de detalhe de um produto, inclua uma ligação para adicionar o produto ao carrinho de compras

```
print "<a href='carrinho.php?comprar=$sku'>[" .
    Traduz("+carrinho") . "]</a>";
```

Para armazenar a lista de conteúdos do carrinho de compras pode ser utilizado o *array* \$\_SESSION. Prepare um *script* PHP (carrinho.php) que incremente o número de unidades do carrinho, identificadas pelo parâmetro \$\_GET['comprar']:

```
session_start();
if ( isset($_GET['comprar']) ) {
    $_SESSION['carrinho'][$_GET['comprar']] += 1;
    header("Location: carrinho.php");
}
```

Quando chamado sem parâmetros, este *script* deverá listar o conteúdo do carrinho de compras:

```
print "    <h1>". Traduz("Carrinho") . "</h1>\n";
LigaBD();
print "<table>\n";
foreach( $_SESSION['carrinho'] as $sku => $quant ) {
    $campos = "Sku, Marca, Modelo, Preco as Unitario";
    $query = "SELECT $campos FROM Produtos WHERE Sku='$sku'";
    // Consulta BD

    $result = mysqli_query($conn, $query);
    if ( ! $result ) echo "erro sql\n";
    $row = mysqli_fetch_row($result);

    print "    <tr>\n";
    foreach ( $row as $val )
        print "        <td>$val</td> \n";
    print "    </tr>";
}
print "</table>\n";
```

Acrescente duas colunas com o número de elementos encomendados de cada item e o valor total por item:

```
$campos .= ", '$quant' as Quantidade, Preco * $quant as Total";
```

Acrescente uma linha com o número total de elementos no carrinho e o valor total da encomenda:

```
$nprod = $total = 0;
foreach ... {
    ...
    $nprod += $row['Quantidade'];
    $total += $row['Total'];
}
print "    <tr> <th colspan=4>Total</th> " .
    "    <td>$nprod</td> <td>$total</td> </tr>\n";
```

Inclua uma linha com os títulos das colunas

```
print "    <tr>\n";
$campos = array('Sku', 'Marca', 'Modelo', 'Unitario', 'Quantidade', 'Total');
foreach( $campos as $c )
    print "        <th>" . Traduz($c) . "</th>\n";
```

```
print " </tr>\n";
```

## 10. Encomenda

Na página de listagem do conteúdo do carrinho de compras, inclua um formulário para submeter o processo de encomenda:

```
<h2>Encomenda</h2>
<form action='encomenda.php' method=GET>
  Email: <input name=email> <br>
  Nome: <input name=nome> <br>
  Morada: <input name=morada> <br>
  Telefone: <input name=telefone> <br>
  <input type=submit value='Confirmar Encomenda'>
</form>
```

ou

```
print "<h2>". Traduz("Encomenda"). "</h2>\n";
print "<form action='encomenda.php' method=GET>\n";
print Traduz("Email"). " : <input name=email> <br>\n";
print Traduz("Nome"). " : <input name=nome> <br>\n";
print Traduz("Morada"). " : <input name=morada> <br> \n";
print Traduz("Telefone").": <input name=telefone> <br>\n";
print "<input type=submit value='". Traduz("Encomendar"). "'>\n";
print "</form>\n";
```

Crie uma nova tabela na base de dados para registar os dados dos clientes.

```
CREATE TABLE Clientes (
  ID char(120) NOT NULL,
  Nome varchar(120) NOT NULL,
  Morada varchar(200) default NULL,
  Telefone varchar(20) default '',
  PRIMARY KEY (ID)
);
```

Crie uma nova tabela na base de dados para registar a data de cada encomenda e associá-la ao cliente respectivo.

```
CREATE TABLE Encomendas (
  EncID int(11) NOT NULL auto_increment,
  `Data` datetime NOT NULL,
  ClienteID char(80) NOT NULL,
  PRIMARY KEY (EncID)
);
```

Crie uma nova tabela na base de dados para registar os produtos encomendados e as respectivas quantidades.

```
CREATE TABLE EncProdutos (
  EncID int(11) NOT NULL,
  Sku char(40) NOT NULL,
  Quantidade int(11) NOT NULL,
```



```

        UNIQUE KEY EncID (EncID, Sku)
    );

```

Prepare um *script* (encomenda.php) para receber os dados da encomenda e gravá-los na base de dados.

```

<?php
    // encomenda.php
    session_start();
    include('funcs.php');

    LigaBD();

    // Gravar Cliente
    $dados = "'". $_GET['email'].      "' , " .
              "'". $_GET['nome'].      "' , " .
              "'". $_GET['morada'].    "' , " .
              "'". $_GET['telefone']. "' " ;
    mysql_query("REPLACE INTO Clientes VALUE ($dados)");

    // Registrar Encomenda
    $dados = "Now(), '". $_GET['email'] . "'";
    mysql_query("INSERT INTO Encomendas(Data,ClienteID) VALUE ($dados)");

    // Obter nr de encomenda
    $res = mysqli_query($conn, "SELECT EncID FROM Encomendas ORDER BY Data DESC");
    $row = mysqli_fetch_row($res);
    $encID = $row[0];

    // Registrar produtos encomendados
    foreach( $_SESSION['carrinho'] as $sku => $qtd )

    mysqli_query($conn,"INSERT INTO EncProdutos VALUE ('$encID','$sku',$qtd)");

    // Esvaziar carrinho
    unset($_SESSION['carrinho']);
?>

```

Acrescente uma página informativa sobre o sucesso da encomenda:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >
<html>
    <head><title>Encomenda realizada</title></head>
    <body>
    <?php
        print "      <h2>A encomenda $encID foi realizada</h2>\n";
    ?>
    </body>
</html>

```

## 11. Autoavaliação e Entrega

A entrega do trabalho é feita através do preenchimento e submissão do formulário Web disponível para o efeito em:

<http://ave.dee.isep.ipp.pt/~jml/cinem/lab11/autoaval.php>

No mesmo formulário deve ser inserida a auto avaliação do trabalho realizado. A versão funcional do trabalho deve ser disponibilizada utilizando a partir de qualquer servidor Web acessível. O código fonte respetivo deve ser disponibilizado no servidor ave.dee.isep.ipp.pt.

A avaliação final decorre no laboratório com a apresentação do trabalho realizado, incluindo as funcionalidades adicionais implementadas.

## Histórico

CINEM-Loja Online - v2.0 {crc, jml} 12/2018

CINEM-Loja Online - v2.1 {crc, jml} 12/2019