

# MÓDULO

# PROGRAMAÇÃO PHP

## PRÁTICA 29

CRIAÇÃO DE CARRINHO DE COMPRAS,  
INCLUINDO PROCESSO DE LOGIN,  
RECORRENDO A PHP

## CRIAÇÃO DE CARRINHO DE COMPRAS, INCLUINDO PROCESSO DE LOGIN, RECORRENDO A PHP

---

### 1.1. DESCRIÇÃO DA PRÁTICA

Tempo estimado: 2 h.

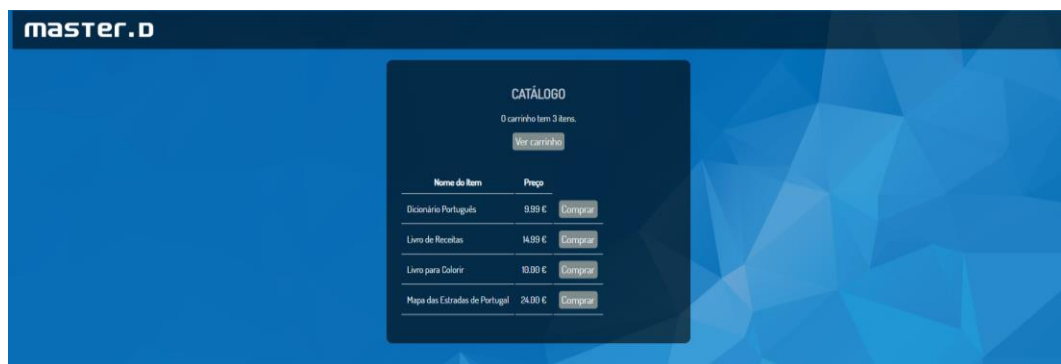
REQUISITOS
<ul style="list-style-type: none"><li>■ Ter concluído a unidade didática “Funções avançadas em PHP”.</li></ul>
OBJETIVOS
<ul style="list-style-type: none"><li>■ Aprofundar os conhecimentos de PHP e a sua utilidade na utilização de sessões.</li><li>■ Aprender a base de login com PHP.</li></ul>
MATERIAL E FERRAMENTAS NECESSÁRIAS PARA A PRÁTICA
<ul style="list-style-type: none"><li>■ Editor de código.</li><li>■ Servidor web.</li></ul>

## 1.2. DESENVOLVIMENTO DA PRÁTICA

Nesta prática, pretende-se explorar um pouco da dinâmica do PHP, realizando um exercício simples de um login que verifique se o e-mail e a senha inseridos pelo utilizador são iguais aos definidos no código, visto que ainda não abordámos como criar e armazenar logins diferentes para cada utilizador (guardar e ler dados na base de dados). Também se pretende que seja desenvolvido um catálogo e um carrinho de compras utilizando o PHP. Esta prática é bastante relevante se pretender criar uma loja online, por exemplo.

O funcionamento da página consiste em efetuar um login em que, caso os dados inseridos estejam corretos e iguais aos definidos no código, a página será direcionada para o catálogo dos produtos. Se, por outro lado, os dados não estiverem corretos, a página permanecerá na área de login. A página de catálogo exibirá alguns produtos que poderão ser adicionados ao carrinho de compras e, ao consultar o carrinho, surgirá uma lista dos produtos adicionados, assim como o valor total da “encomenda”.

As imagens seguintes ilustram o que se pretende:





De referir que os ficheiros referentes a esta prática, e tendo eles código PHP, só funcionarão se estiverem num servidor web. Deverá por isso incluí-los na pasta “htdocs” do XAMPP/MAMP, não se esquecendo de ativar o Apache e o MySQL.

Iremos iniciar esta prática com o desenvolvimento do processo de login. Sendo que esta prática está mais focada na lógica do PHP, serão facultados o primeiro ficheiro PHP (pratica29.php) e a respetiva folha de estilos, fornecidos na pasta de apoio à prática, disponível no seu Campus Virtual.

Reforçando alguns pontos do ficheiro base fornecido, o método do formulário é o post, logo, recuperará os valores, no segundo ficheiro (o ficheiro “processologin.php” criado posteriormente e ao longo desta prática), através do post e não do get. Estes valores também serão recuperados através do name e não do id, embora normalmente sejam iguais para evitar qualquer tipo de confusão.

```
<form method="post" action="processologin.php">  
  
<input type="text" id="email" placeholder="Inserir o email"  
name="email">
```

O primeiro passo será, então, criar um novo ficheiro chamado “processologin.php” (se lhe atribuir outro nome, não se esqueça de alterar o mesmo no action do formulário).

Neste novo ficheiro, a primeira ação deve ser abrir e fechar o PHP, informando o sistema de que estamos a escrever código nesta linguagem, através de:

```
<?php
```

```
?>
```

Todo o código PHP que iremos desenvolver ao longo desta prática deverá ser escrito entre estas tags.

Começamos por colocar um excerto de código PHP, onde se irá indicar que se pretende recolher os valores inseridos no formulário anterior (o e-mail, definido com name “email”, e a senha, definida com name “pwd”).

```
$email = htmlspecialchars($_POST["email"]);
```

Com esta linha de código, criamos uma variável chamada “email” que, através do método post, irá recolher o valor usando o campo do formulário com o nome “email”. O mesmo será feito para a variável da senha.

```
$pwd = htmlspecialchars($_POST["pwd"]);
```

Para definir se o login foi feito com sucesso ou não, será utilizada uma condição **if**. Visto que não existe ligação a uma base de dados, serão definidos nesta mesma condição um e-mail e uma senha – não necessita de os alterar, pois não irá confirmar se é um e-mail que já existe nem enviar nenhum e-mail; vai apenas verificar se o e-mail inserido na primeira página é igual ao da condição (e o mesmo com a senha). O e-mail e a senha escolhidos foram, respetivamente, “pratica@teste.com” e “senha”. Como se pretende que a página seja redirecionada para o catálogo dos produtos caso os dados inseridos estejam de acordo com o definido, podemos desde já criar este redirecionamento através da função **header**. Se reparar, o header está a direcionar o documento para uma nova página chamada “catalogo php”, que será definida posteriormente.

```
if ($email == "pratica@teste.com" && $pwd == "senha") {  
    //echo "<h2>Olá $email <br> Bemvindo.</h2>";  
  
    //entrar no carrinho de compras  
    Header("Location:catalogo.php");  
  
}
```

Este redirecionamento só irá funcionar se ambas as condições passadas pelo parâmetro forem verdadeiras. Caso o e-mail, a senha, ou ambos, não coincidirem com os dados definidos, não se pretende que seja feito o redirecionamento. Pretende-se que seja exibida uma página a indicar que o login é inválido.

**Login invalido**

Será, então, indicado no **else** da condição **if**.

```
else {  
    echo "<h2>Login invalido</h2>";  
}
```

O código compilado do ficheiro "processologin.php" será:

```
<?php  
  
$email = htmlspecialchars($_POST["email"]);  
  
$pwd = htmlspecialchars($_POST["pwd"]);  
  
if ($email == "pratica@teste.com" && $pwd == "senha") {
```

```
//echo "<h2>Olá $email <br> Bemvindo.</h2>";  
  
//entrar no carrinho de compras  
  
Header("Location:catalogo.php");  
  
}  
  
else {  
  
    //Header("Location:pratica8.php?invalid=&email=$email");  
  
    echo "<h2>Login invalido</h2>";  
  
}  
  
?>
```

Passamos agora para o desenvolvimento da página do catálogo dos produtos.

O ficheiro “catalogo.php” e o respetivo ficheiro “estilos.css” estão disponíveis no seu Campus Virtual, como base para utilizar nesta prática.

No ficheiro “catalogo.php”, pode observar que o catálogo está feito manualmente e de forma estática com a construção de “linha a linha” da tabela. Como não são muitos produtos, tal faz-se rapidamente, porém, se a quantidade de produtos fosse maior, seria uma tarefa muito complexa e teria de se repetir muito código, pelo que, para evitar que isso seja necessário, é utilizado um **array** e um **loop for** em PHP para fazer dinamicamente a tabela.

Assim, iremos começar por definir as variáveis itens e preços, que vão ser geradas a partir de um array que contém as quatro opções do catálogo de produtos. Estas variáveis devem ser criadas logo no parágrafo h2.

```
<?php

    $itens = array ('Dicionário Português', 'Livro de Receitas'
, 'Livro para Colorir', 'Mapa das Estradas de Portugal');

    $precos = array (9.99, 14.99, 10, 24,99);

?>
```

Em seguida, iremos substituir as rows da tabela feita em HTML (todo o conteúdo entre as tags <tbody> e </tbody>) por código PHP, onde deverá ser criado um **loop for**.

```
for ($i = 0; $i < count($itens); $i++) {

}
```

Nesta linha, está indicado que \$i irá começar no zero (corresponde ao primeiro índice do array), irá parar quando \$i for maior do que o número de itens (obtido através da função **count**) e será incrementado num valor.

Dentro do **loop for**, terá vários echos, imprimindo uma tabela tal como em HTML, só que substituindo os nomes e os preços dos produtos pelas respetivas variáveis.

```
echo '<tr>';

echo '<td>' . $itens[$i] . '</td>';

echo '<td align = "right">' . number_format($precos[$i], 2) . ' €</td>';

echo '</tr>';
```

Para o preço, é utilizada a função predefinida **number\_format()**, sobre a qual poderá ler mais na página oficial do PHP:

- [https://www.php.net/manual/pt\\_BR/function.number-format.php](https://www.php.net/manual/pt_BR/function.number-format.php)



O próximo passo será guardar a lista de itens que forem colocados no carrinho de compras pelo utilizador. Isso será feito através de um novo array, pois é necessário que a variável seja persistente e permanente durante toda a visita do utilizador, utilizando as sessões do PHP. O excerto de código que se segue deverá ser colocado antes de todo o código HTML:

```
<?php
session_start();

if (!isset($_SESSION['carrinho'])) {
    $_SESSION['carrinho'] = array();
}

?>
```

A função **session\_start()** inicia uma nova sessão (e cria uma nova cookie de sessão) ou recupera as variáveis existentes de uma anterior sessão (se esta existir). A condição **if** verifica se a variável `$_SESSION['carrinho']` existe. Caso não exista, é criado um novo array vazio que representa um novo carrinho de compras.

Depois de adicionar este excerto de código, já é possível exibir o número de itens no carrinho de compras na página web e um botão “Comprar” que permita adicionar produtos ao carrinho. Para exibir o número de itens no carrinho de compras, basta colocar o seguinte código depois do título h2 da página:

```
<p>O carrinho tem <?php echo count($_SESSION['carrinho']); ?> i  
tems.</p>
```

Para adicionar o botão “Comprar” em cada um dos produtos exibidos na tabela, terá de ser adicionada uma nova linha no **loop for** criado anteriormente, ficando, então:

```
<?php

for ($i = 0; $i < count($itens); $i++) {

    echo '<tr>';

    echo '<td>' . $itens[$i] . '</td>';
```

```
        echo '<td align = "right">' . number_format($precos[$i], 2) . ' €
    </td>';

    echo '<td><a href="' . $_SERVER['PHP_SELF'] . '?comprar=' . $i .
    '" class="botao">Comprar</a></td>';

    echo '</tr>';

    }

?>
```

Com esta nova linha, cada produto do catálogo tem um link através do buy=n, onde **n** é o índice do item que é adicionado ao carrinho de compras (no array \$itens).

Na primeira tag PHP (antes do código HTML e depois do código colocado anteriormente), falta este excerto de código, onde pode observar a variável \$\_GET['comprar'], que fará com que, ao adicionar um item ao array \$\_SESSION['comprar'], o navegador não se atualize repetitivamente conforme forem adicionados novos itens.

```
if (isset($_GET['comprar'])) {

    $_SESSION['carrinho'][] = $_GET['comprar'];

    header('location: ' . $_SERVER['PHP_SELF'] . '?' . SID);

    exit();

}
```

Este excerto de código vai funcionar perfeitamente se o utilizador tiver os cookies ativos, mas, caso não tenha, o ajuste automático do PHP não irá afetar o header e, dessa forma, perderá o ID da sessão. Porém, este caso é facilmente resolvido com a constante SID (já incluída acima), que vai passar o ID de sessão e, assim, tornar o código compatível com os cookies desativados.

Para terminar a página do catálogo, deve acrescentar-se a seguinte linha depois da contagem dos itens adicionados ao carrinho:

```
<p><a href="carrinho.php" class="botao">Ver carrinho</a></p><br>
```

Esta linha será a responsável por redirecionar a página para o carrinho de compras, que iremos desenvolver no próximo ponto desta prática.

Segue-se o código completo da página do catálogo, para comparar, caso tenha algum erro.

```
<?php
session_start();

if (!isset($_SESSION['carrinho'])) {
    $_SESSION['carrinho'] = array();
}

if (isset($_GET['comprar'])) {
    $_SESSION['carrinho'][] = $_GET['comprar'];
    header('location: ' . $_SERVER['PHP_SELF'] . '?' . SID);
    exit();
}

?>

<html>

<head>

<meta charset="UTF-8">

<title>Catálogo - Prática 29</title>

<link href="estilos.css" rel="stylesheet">

</head>

<body>
```

```
<div class="caixa0">

    <span id="logo"></span>

</div>

<div class="caixa1">

    <h2>CATÁLOGO</h2>

    <p>0 carrinho tem <?php echo count($_SESSION['carrinho']); ?> itens.</p>

    <p><a href="carrinho.php" class="botao">Ver carrinho</a></p><br>

    <?php

        $itens = array ('Dicionário Português', 'Livro de Receitas'
, 'Livro para Colorir', 'Mapa das Estradas de Portugal');

        $precos = array (9.99, 14.99, 10, 24,99);

    ?>

    <table>

        <thead>

            <tr>

                <th>Nome do Item</th>

                <th>Preço</th>

            </tr>

        </thead>

        <tbody>

            <?php

                for ($i = 0; $i < count($itens); $i++) {

                    echo '<tr>';

                    echo '<td>' . $itens[$i] . '</td>';

                    echo '<td align="right">' . number_format($precos[$i], 2) . ' €</td>';
```

```

        echo '<td><a href="' . $_SERVER['PHP_SELF'] . '
?comprar=' . $i . '" class="botao">Comprar</a></td>';

        echo '</tr>';

    }

    ?>

</tbody>

</table>

</div>

</body>

</html>

```

O último ponto desta prática será a criação da página do carrinho de compras, que é bastante semelhante à página do catálogo. A diferença é que esta página recupera a variável `$_SESSION['carrinho']` e utiliza os valores armazenados para imprimir os itens guardados no array `$itens` numa tabela onde mostrará o total (também utilizando o **loop for** como no catálogo). Também fornecerá um link que carrega uma string vazia, de modo a esvaziar o carrinho de compras. Deste modo, sugere-se que tente fazer a página do carrinho e depois confirme se funciona, mas, se tiver dúvidas, segue-se o código da mesma:

```

<?php

    session_start();

    if (!isset($_SESSION['carrinho'])) {

        $_SESSION['carrinho'] = array ();

    }

    if (isset($_GET['vazio'])) {

        unset ($_SESSION['carrinho']);

        header('location: ' . $_SERVER['PHP_SELF'] . '?' . SID);

        exit();

    }

```

```
?>

<html>

<head>

<meta charset="UTF-8">

<title>Catálogo - Prática 29</title>

<link href="estilos.css" rel="stylesheet">

</head>

<body>

<div class="caixa0">

    <span id="logo"></span>

</div>

<div class="caixa1">

    <h2>CARRINHO</h2>

    <br>

    <?php

        $itens = array ('Dicionário Português', 'Livro de Receitas'
, 'Livro para Colorir', 'Mapa das Estradas de Portugal');

        $precos = array (9.99, 14.99, 10, 24,99);

    ?>

    <table>

        <thead>

            <tr>

                <th>Nome do Item</th>

                <th>Preço</th>

            </tr>

        </thead>

        <tbody>
```

```

        <?php
            $total = 0;

            for ($i = 0; $i < count($_SESSION['carrinho']); $i+
+) {

                echo '<tr>';

                echo '<td>' . $itens[$_SESSION['carrinho'][$i]]
. '</td>';

                echo '<td align="right">';

                echo number_format($precos[$_SESSION['carrinho'
][$i]], 2) . ' €</td>';

                echo '</tr>';

                $total = $total + $precos[$_SESSION['carrinho'
][$i]];

            }

        ?>

    </tbody>

    <tfoot>

        <tr>

            <th align="right"> Total:</th><br>

            <th align="right"><?php echo $total; ?> €</th>

        </tr>

    </tfoot>

</table>

    <p><a href="catalogo.php">Continuar a comprar</a> ou <a href="<
?php echo $_SERVER['PHP_SELF'] ; ?>?vazio=1">Esvaziar o carrinho</p>
>

</div>

</body>

</html>

```

