Xestión de usuarios coa linguaxe PHP.

Introdución

Cando un usuario accede a un sitio web có seu navegador, posiblemente este inicie sesión ou intercambie datos co servidor. Pode ocorrer que certa información relativa ao usuario, durante a visita do sitio web, sexa necesario que estea dispoñible en calquera das páxinas do sitio web. PHP implementa dous mecanismo para que isto sexa posible: as **cookies** e as **sesións**.

Cookies

Unha *cookie* é unha cantidade limitada de información que garda o navegador no ámbito do usuario. Cada *cookie* está asociada a un sitio web determinado, de tal forma que soamente as súas páxinas podan acceder ao seu contido.

Orixinariamente os navegadores gardaban o contido de cada *cookie* nun pequeno ficheiro de texto. Hoxe en día a maioría de navegadores modernos almacenan a información das *cookies* de xeito centralizado nunha base de datos, pero sempre mantendo independentes unhas de outras, de xeito que cada sitio web poderá acceder soamente as súas *cookies* e non as almacenadas por outros sitios web.

As especificacións suxiren que os navegadores deben soportar un número mínimo de *cookies* ou unha cantidade mínima de memoria para almacenalas. En concreto, espérase que un navegador sexa capaz de almacenar polo menos 300 *cookies* de 4 Kbytes cada unha e polo menos 20 *cookies* por servidor ou dominio.

O contido das *cookies* é xestionado polo sitio web que as crea. O seu uso máis típico é o almacenamento das preferencias do usuario (por exemplo, o idioma en que se deben mostrar as páxinas), para que non teña que volver indicalas a próxima vez que visite o sitio.

Cookies e seguridade

Aínda que as *cookies* por si mesmas non son perigosas, algúns sitios web as empregan para rastrexar a navegación dos usuarios. Unha páxina web pode conter imaxes e outros compoñentes que residan noutros dominios; nas conexións que se crean cos servidores destes dominios para obter os compoñentes da páxina tamén se poden crear *cookies*.

Por exemplo, os sitios web que engaden publicidade ás páxinas polas que pasamos poden engadir as súas propias *cookies* e recoller a información daqueles sitios web visitados sempre que tamén conteñan a súa publicidade. A estas *cookies* noméaselles "*cookies* de terceiros" (*third-party cookies*).

Principalmente por este motivo os navegadores inclúen na súa configuración opcións para indicar as *cookies* que queremos almacenar e as que non.

Os servidores poden especificar unha data de caducidade para as *cookies*. Se non se especifica data de caducidade, a *cookie* elimínase cando o usuario pecha o navegador (ao "final da sesión"). A aquelas *cookies* que indican unha data de caducidade tamén se lles chama "persistentes", pois non se eliminan ao pechar o navegador. Cando chega a data de vencemento dunha *cookie* persistente, o navegador a elimina de xeito automático.

Emprego de *cookies* en PHP

Almacenamento de información nunha cookie

En PHP, podes utilizar a función "`setcookie`" para almacenar unha *cookie* no navegador do usuario. O único parámetro obrigatorio que tes que usar é o nome da *cookie*, que debe ser unha cadea de texto ASCII que non pode conter os seguintes caracteres: coma "`,`", punto e coma "`;`", espazo "` `" e o signo igual "`=`". Pero admite varios parámetros máis opcionais: o segundo parámetro é o seu valor e o terceiro é a data de caducidade da *cookie* (é importante subliñar que se trata dunha data, non dun prazo de tempo).

Unha vez almacenada unha *cookie* persistente no navegador é posible eliminar o seu contido antes de que expire. Faise empregando a mesma función "`setcookie`" pero indicando unha data de caducidade anterior á actual).

Por exemplo, se queres almacenar nunha *cookie* a lingua que prefire o usuario para o sitio web, podes facer:

```
setcookie('Lingua', 'Galego');
```

A *cookie* anterior elimínase ao pechar a sesión do navegador. A mesma *cookie* cun prazo de caducidade de unha hora (persistente) faríase:

```
setcookie('Lingua', 'Galego', time()+3600);
```

Cando en PHP facemos unha chamada á función "`setcookie`", o servidor envía un encabezado "`Set-Cookie`" ao navegador. Por iso, as chamadas á función "`setcookie`" deben enviarse antes de que o navegador mostre información algunha en pantalla.

Recuperación da información almacenada nunha cookie

O proceso de recuperación da información que almacena unha *cookie* é moi simple. PHP recolle a información que o servidor recibe nos encabezados "`Cookie`" das peticións HTTP e a pon accesible aos guións por medio dunha variable superglobal, o array asociativo "`$_COOKIE`".

```
echo 'A lingua preferida é ' . $_COOKIE['Lingua'];
```

Sempre que unha aplicación web empregue *cookies*, se debe ter ben presente que en última instancia a súa dispoñibilidade está controlada polo cliente. Por exemplo, algúns usuarios deshabilitan por completo as *cookies* no navegador porque pensan que a información que almacenan pode supoñer un potencial problema de seguridade. Ou a información que almacenan pode chegar a perderse porque o usuario decide acceder empregando outro navegador, ou simplemente eliminalas do seu sistema.

Forzar o borrado dunha *cookie*

Como rematamos de ver, as *cookies* persistentes elimínanse cando vence o seu tempo de vida. Por outra banda, as *cookies* nas que non se especifica o tempo de vida son eliminadas ao pechar a sesión do navegador.

As veces pode acontecer que queiramos obrigar ao borrado dunha *cookie* do navegador. Isto faise reescribindo a mesma *cookie* cun tempo de vida que sinale ao pasado.

Por exemplo, se nalgún momento engadimos a seguinte *cookie*:

```
setcookie('Lingua', 'Galego');
```

E agora queremos eliminala, podemos executar o seguinte código PHP:

```
setcookie('Lingua', 'Galego', time() - 1000);
```

Almacenamento de arrays nas cookies

Aínda que o almacenamento que ofrecen as *cookies* é limitado, pode ser que nalgún intre necesitemos pór nelas os datos dun array. Isto faise engadindo tantas *cookies* como membros teña o array, de unha en unha, indicando os seus nomes coa notación típica dos arrays.

```
<?php

setcookie('usuario[nome]', 'Xaquín');

setcookie('usuario[apelido1]', 'Lorenzo');

setcookie('usuario[apelido2]', 'Fernández');

...

?>
```

Cando máis adiante recuperemos as *cookies*, veremos que se converten nun array cos seus valores como membros.

```
<?php

if(isset($_COOKIE['usuario'])) {

foreach ($_COOKIE['usuario'] as $nome => $valor) {

echo "$nome : $valor <br />";

}

}

?>
```

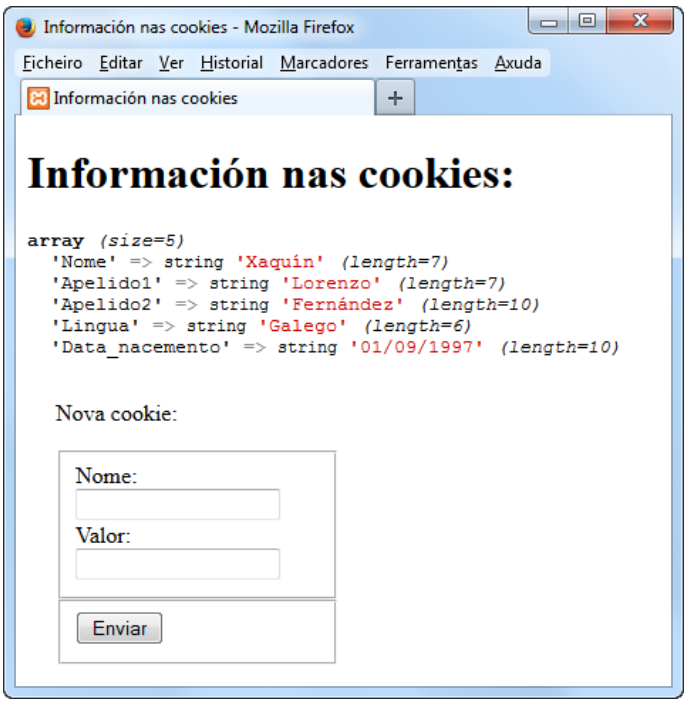
Tarefa: Almacenamento de información nas *cookies*

a. Tarefa 1_a

Imos comezar creando unha pequena aplicación web que:

- Amose as *cookies* recibidas do navegador.
- Incorpore un formulario que permita introducir unha nova *cookie* no navegador, indicando o seu nome e o seu valor.

As novas *cookies* introducidas no formulario deberán amosarse a continuación na lista da mesma *páxina*, do seguinte xeito.



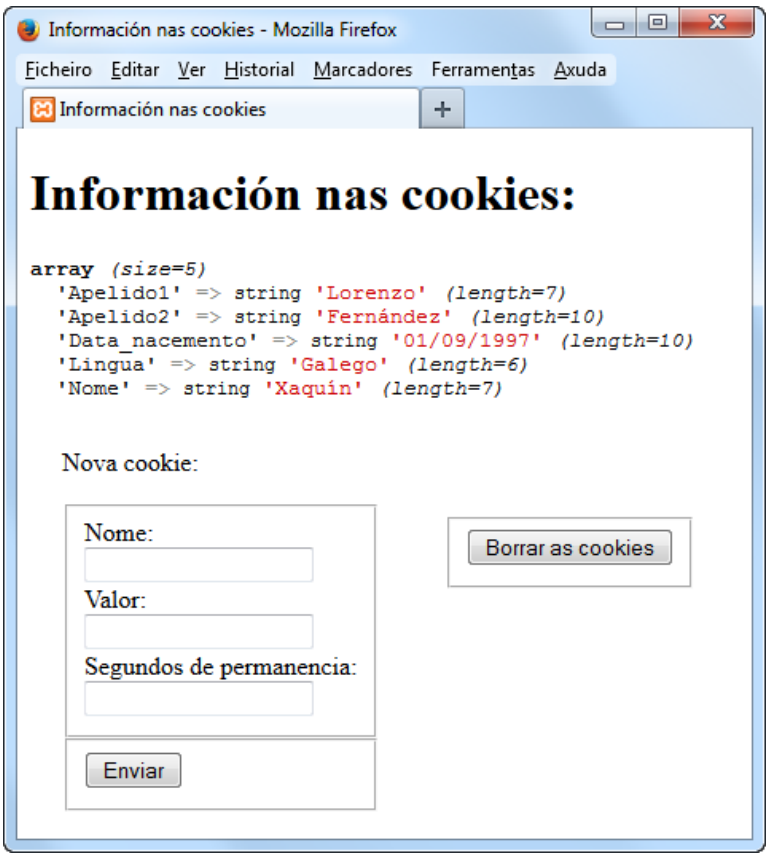
b. Tarefa 1_b

Agora imos engadirlle un par de detalles á páxina anterior:

- Deberán poderse introducir *cookies* sen valor, soamente co seu nome.
- No formulario haberá un dato máis, o tempo de vida da *cookie* que se indicará en segundos. Cando se cubra ese campo, a *cookie* que se envíe deberá ser persistente.

c. Tarefa 1_c

Imos continuar coa mesma páxina. Nesta ocasión temos que crear un novo botón que permita eliminar todas as *cookies* do navegador do usuario. Por exemplo:



Ao premer o botón de borrado, recargarase a páxina e a lista de *cookies* deberá quedar baleira.

As sesións

Unha sesión é un mecanismo para persistir información en diferentes páxinas. Para isto, PHP, a través das sesións, permite que páxinas distintas poden acceder a unha variable común. Esta variable común é a variable superglobal $_SESSION. Esta variable $_SESSION é unha matriz asociativa na que se poden definir valores como en calquera outra matriz. A diferenza é que $_SESSION é accesible desde páxinas diferentes (sempre que esas páxinas teñan asociada a mesma sesión), mantendo os valores dunha páxina a outra.

Non se deben confundir as sesións coas cookies. As cookies é un método que permite gardar información no ordenador do cliente para recuperalo no futuro, mentres que nas sesións a información mantense no servidor ata que se pecha a sesión (por intervención do usuario ou por tempo).

Traballar con sesións ten tres partes:

- Creación ou apertura da sesión:

Cando unha páxina crea unha sesión usando a función correspondente, o servidor asocia un identificador de usuario único ao explorador do usuario. O identificador gárdase no usuario en forma de cookie ou, se o navegador do usuario non permite a creación de cookies, engádese o identificador no enderezo da páxina.

- Uso da sesión:

Se a sesión xa foi creada, as páxinas solicitadas polo mesmo navegador poden gardar e recuperar información no servidor, información que está asociada co identificador de usuario, polo que non é accesible a outros usuarios. A información consérvase ata que o usuario ou servidor destrúe a sesión.

- Destrución ou peche da sesión:

Tanto o usuario como o servidor poden pechar sesión. O usuario pode destruír a sesión pechando o navegador. O servidor pode destruír a sesión cando calquera páxina usa a función correspondente ou despois dun determinado tempo.

Creación o apertura da sesión

Cando algunha páxina crea unha sesión, facendo uso da función correspondente, o servidor asocia ao navegador do usuario un identificador de usuario único. Este identificador de usuario único se garda en forma de cookie. Se o navegador non permite a creación de cookies, existe a posibilidade de que o identificador do usuario se inclúa na dirección da páxina, o cal pode supoñer un risco de seguridade.

A sesións se crean mediante a función session_start(). Se a sesión no existe, esta función crea a sesión e lle asocia un identificador de sesión único. Se a sesión xa existía, esta función permite que a páxina teña acceso á información vinculada á sesión.

Exemplo de creación dunha sesión

```
<?php
session_start();
?>
```

Como no caso das cookies, débese ter precaución ao utilizar a función session_start() antes de comezar a escribir o contido da páxina. O identificador da sesión utilízase nas cabeceiras de resposta HTTP, e estas cabeceiras deben enviarse antes que o texto da páxina.

As sesións teñen como vantaxes que permiten almacenar grandes cantidades de datos facilmente, aforran ancho de banda ao pasar soamente o identificador do usuario, e os datos ao almacenarse no servidor fan que as sesións sexan máis seguras ao non poder ser vistas ou editadas polo cliente.

Utilización da sesión

Cando unha páxina creou unha sesión, ou accedeu a unha sesión xa existente, mediante session_start(), a páxina ten acceso ao vector $_SESSION que contén as variables desa sesión.

O vector $_SESSION é asociativo, de xeito que cada elemento será accesible mediante un nome, e terá o seu valor almacenado no array. SESSION é accesible desde páxinas diferentes do sitio web para as que se lles asociou a sesión mediante session_start().

No seguinte exemplo faise uso dunha sesión e se almacena un elemento no array $_SESSION. Supóñase que o script PHP está nun ficheiro exemplo01.php.

exemplo01.php

```
<?php
session_start();
$_SESSION["nome"] = "Uxío Varela";
print "<p>Gardouse o nome.</p>";
?>
```

No exemplo que segue, accédese ao elemento que se almacenou no anterior script. Supóñase que o script está almacenado nun ficheiro exemplo02.php.

Exemplo02.php

```
<?php
session_start();
print "<p>O nome é $_SESSION[nome].</p>";
?>
```

Con respecto ás cookies, cando unha páxina pide ao navegador que cree unha cookie, o valor da cookie non está dispoñible en $_COOKIE nesa páxina. O estará nas páxinas posteriores cando o navegador pida outra páxina e envíe o valor na petición. No caso das sesións, cando unha páxina crea un valor en $_SESSION, ese valor está dispoñible na mesma páxina.

Mentres a sesión permanece aberta, podes utilizar a variable superglobal "`$_SESSION`" para engadir información á sesión do usuario, ou para acceder á información almacenada na sesión. Por exemplo, o seguinte guión conta o número de veces que o usuario visita a páxina, gardando o contador na sesión do usuario:

```
<?php

// Iniciamos a sesión ou recuperamos a anterior sesión existente

session_start();

// Comprobamos se a variable xa existe

if (isset($_SESSION['visitas'])) $_SESSION['visitas']++;

else $_SESSION['visitas'] = 1;

?>
```

Se en lugar do número de visitas, se quixera almacenar o instante en que se produce cada unha, a variable de sesión "`visitas`" deberá ser un array e polo tanto habería que cambiar o código anterior por:

```
<?php

// Iniciamos a sesión ou recuperamos a anterior sesión existente

session_start();

// En cada visita engadimos un valor ao array "visitas"

$_SESSION['visitas'][] = time();

?>
```

Peche manual da sesión

Aínda que é posible configurar PHP para que elimine de forma automática os datos dunha sesión pasado certo tempo, en ocasións pode ser necesario pechala de forma manual nun momento determinado. Por exemplo, cando se empregan sesións para recordar a información de autenticación, convén darlle ao usuario do sitio web a posibilidade de pechar a sesión cando o crea conveniente.

En PHP poden empregarse dúas funcións para eliminar a información almacenada na sesión:

- `session_unset()`. Elimina as variables almacenadas na sesión actual, pero non elimina a información da sesión do dispositivo de almacenamento usado.
- `session_destroy()`. Elimina completamente a información da sesión do dispositivo de almacenamento.

Ejemplos de sesiones:

- El servidor crea un número único para cada nueva sesión. Si quieres obtener el identificador de sesión, puedes usar la función session_id, como muestra el siguiente fragmento de código:

```
<?php

session_start();

echo session_id();

?>
```

Esto debería darte el identificador de sesión actual. La función session_id es interesante en porque también puede recibir un argumento— un identificador de sesión. Si quieres reemplazar el identificador de sesión generado por el sistema por el tuyo propio, puedes suministrarlo como el primer argumento de la función session_id.

```
<?php

session_id(YOUR_SESSION_ID);
```

```
session_start();

?>
```

Es importante hacer notar que la función session_id debe estar situada antes que la llamada a `session_start();`

Crearase un ficheiro sesions.php que conterá unha páxina web sinxela onde se amosa un formulario no que se encherán datos sobre un automóbil. No caso no que se verifique e valide o formulario, crearase unha sesión e asignaránselle como elementos ao array $_SESSION os datos introducidos no formulario. Cómpre destacar que cando se crea a función asígnaselle primeiro un nome mediante a función session_name() e a continuación se chama á función session_start(). Estas dúas funcións chámanse antes de enviar ningunha outra cabeceira HTTP.

sesions.php

```
<?php
if (count($_POST)!=0){
$cadeaErros="";
$marMod= htmlspecialchars(trim(strip_tags($_POST["marMod"])), ENT_QUOTES, "ISO-8859-1");
if ($marMod=="")
$cadeaErros.= "O campo Marca e modelo non pode estar baleiro. ";
$nKm= htmlspecialchars(trim(strip_tags($_POST["nKm"])), ENT_QUOTES, "ISO-8859-1");
if ($nKm=="")
$cadeaErros.= "O campo Nº km non pode estar baleiro. ";
else
if (filter_var($nKm, FILTER_VALIDATE_INT)){
if ((int)$nKm<=0)
$cadeaErros.= "O Nº km deber ser maior que 0. ";
} else {
$cadeaErros.= "O Nº km non é un número enteiro. ";
}
$urlFab= htmlspecialchars(trim(strip_tags($_POST["urlFab"])), ENT_QUOTES, "ISO-8859-1");
if ($urlFab == "")
$cadeaErros.= "O campo URL do fabricante non pode estar baleiro. ";
else{
if (!filter_var($urlFab, FILTER_VALIDATE_URL))
$cadeaErros.= "A URL do fabricante especificada non é válida.";
}
$mod=(isset($_POST["com"]))
? $_POST["com"]
: "";
$eqEx=(isset($_POST["eqEx"]))
? $_POST["eqEx"]
: array();
if ($cadeaErros == ""){
session_name("AMiñaSesion");
session_start();
$_SESSION["marcaModelo"]=$_POST["marMod"];
$_SESSION["numKm"]=$_POST["nKm"];
$_SESSION["urlFabricante"]=$_POST["urlFab"];
$_SESSION["combustible"]=$_POST["com"];
for ($i=0; $i<count($_POST["eqEx"]); $i++)
$_SESSION["equipEx"][$i]=$_POST["eqEx"][$i];
}
}
?>
```

```
<?php
function incForm($marMod, $nKm, $urlFab, $com, $eqEx){?>
<form action="<?php echo $_SERVER["PHP_SELF"];?>" method="post">
<p>Marca e modelo: <input type="text" name="marMod" value="<?php echo $marMod;?>" /></p>
<p>Nº km: <input type="text" name="nKm" value="<?php echo $nKm;?>"/></p>
<p>URL do fabricante:<input type="text" name="urlFab" value="<?php echo $urlFab;?>"/></p>
<p>Combustible: <input type="radio" name="com" value="Diesel" <?php if ($com=="Diesel") echo "checked"?>/> Diesel
<input type="radio" name="com" value="Gasolina" <?php if ($com=="Gasolina") echo "checked"?>/> Gasolina
<input type="radio" name="com" value="Eléctrico/Híbirdo" <?php if ($com=="Eléctrico/Híbirdo") echo "checked"?>/> Eléctrico/Híbrido
</p>
<p>Equipamento extra:<select name="eqEx[]" size="3" multiple="multiple">
<option <?php if (in_array("Pintura metalizada", $eqEx)){echo "selected=\"selected\"";}?>>Pintura metalizada</option>
<option <?php if (in_array("Sistema de navegación", $eqEx)){echo "selected=\"selected\"";}?>>Sistema de navegación</option>
<option <?php if (in_array("Reposabrazos central delanteiro", $eqEx)){echo "selected=\"selected\"";}?>>Reposabrazos central
delanteiro</option>
</select></p>
<p><input type="reset" value="Borrar"/> <input type="submit" value="Enviar" /></p>
</form>
<?php
}
?>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title></title>
</head>
<body>
<?php
if (count($_POST)==0)
incForm("", "", "", "", array());
else{
if ($cadeaErros != ""){
incForm($marMod,$nKm, $urlFab, $mod, $eqEx);
echo "<strong><em>$cadeaErros</em></strong>";
}else{
echo "<p>O formulario encheuse correctamente. Os datos recibidos son:</p>";
echo "<p>Marca e modelo: ".$_POST["marMod"]."</p>";
echo "<p>Nº km: ".$_POST["nKm"]."</p>";
echo "<p>URL do fabricante: ".$_POST["urlFab"]."</p>";
echo "<p>Combustible: ".$_POST["com"]."</p>";
echo "<p>Equipamento extra: ";
for ($i=0; $i<count($_POST["eqEx"]); $i++){
echo $_POST["eqEx"][$i];
if ($i==(count($_POST["eqEx"])-1))
echo ".";
else
echo ", ";
}
echo "</p>";
}
}



?>
</body>
</html>
```

No ficheiro exemplosesions.php accédese aos valores almacenados no array $_SESSION. Pasa isto é necesario abrir antes a sesión existente mediante a función session_start(), pero como neste caso é unha sesión nominada, cómpre chamar antes á función session_name() asignándolle o nome da sessión que se quere abrir.

Código do ficheiro exemplosesions.php

```
<?php
session_name("AMiñaSesion");
session_start();
?>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title></title>
</head>
<body>
<?php
$marMod=(isset($_SESSION["marcaModelo"]))
?$_SESSION["marcaModelo"]
:"";
$nKm=(isset($_SESSION["numKm"]))
?$_SESSION["numKm"]
:"";
$url=(isset($_SESSION["urlFabricante"]))
?$_SESSION["urlFabricante"]
:"";
$com=(isset($_SESSION["combustible"]))
?$_SESSION["urlFabricante"]
:"";
$eqEx=(isset($_SESSION["equipEx"]))
? $_SESSION["equipEx"]
: array();
echo "<p>Os valores contidos na sesión son:</p>";
echo "<p>Marca e modelo: ".$marMod."</p>";
echo "<p>Nº km: ".$nKm."</p>";
echo "<p>URL do fabricante: ".$url."</p>";
echo "<p>Combustible: ".$com."</p>";
echo "<p>Equipamento extra: ";
for ($i=0; $i<count($eqEx); $i++){
echo $eqEx[$i];
if ($i==(count($eqEx)-1))
echo ".";
else
echo ", ";
}
echo "</p>";
?>
</body>
</html>
```