

4. Xeración dinámica de páxinas web.....	2
4.3. Xestión de usuario coa linguaxe PHP	2
As cookies.....	2
As sesións.....	7
Creación o apertura da sesión	7
Utilización da sesión	8
Destrucción ou peche da sesión	8
Exemplo de manexo de sesións.....	9

4. Xeración dinámica de páxinas web

4.3. Xestión de usuario coa linguaxe PHP

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 mecanismos para que isto sexa posible: as cookies e as sesións. Nesta actividade estudaranse ambos, e ademais, aplicarase o mecanismo de sesións no sitio web de exemplo, estruturando o código en funcións.

Polo tanto, os conceptos e destrezas que se manexarán nesta actividade son os que seguen:

- Estudo das cookies e o seu manexo desde PHP.
- Xestión das sesións en PHP.

As cookies

As cookies son un mecanismo que utilizan os servidores web para gardar información no ordenador do usuario e recuperala cada vez que o navegador lle pide unha páxina. A información gárdase no ordenador do usuario en formato de texto. Esta información utilízase para fines diversos como autenticación, selección de preferencias, ítems seleccionados nun carrito de comprar, etc. A intención desta información será sempre a de identificar ao usuario e personalizar as súas páxinas.

As cookies créanse cando o servidor lle pide ao navegador. Cando o servidor envía unha páxina ao navegador, pode incluír nas cabeceiras de resposta HTTP a petición de creación dunha ou varias cookies. O navegador crea a cookie, gardando o nome da cookie, o seu valor, e tamén o nome do servidor que creou a cookie.

setcookie()

En PHP, as cookies se crean mediante a función `setcookie()`, como se mostra no exemplo:

Exemplo de creación de cookie

```
<?php
    setcookie("nome", "María");
?>
```

No caso que se desexe gardar unha cookie nun array, é necesario crear cada elemento do array nunha cookie distinta, como se amosa no seguinte exemplo.

Exemplo de creación de cookie que conteña un array

```
<?php
    setcookie("datos[nome]", "Xosé");
    setcookie("datos[apellidos]", "López Pérez");
?>
```

Cómpre ter en conta que a función `setcookie()` define unha cookie para ser enviada xunto co resto da cabeceira de HTTP. As cookies deben ser enviadas antes de que o script xere ningunha saída (é unha restrición del protocolo). Isto implica que as chamadas a esta función se coloquen antes de que se xere calquera saída, incluíndo las etiquetas `<html>` y `<head>` ao igual que calquera espazo en branco.

No seguinte **exemplo**, úsase un formulario onde se introducen os **datos dun automóbil**. Unha vez os formulario é enviado tras superar o proceso de verificación e validación, a información se garda en diferentes cookies. Deste xeito, esa información poderá ser recuperada desde outras páxinas do sitio web. Cómpre observar que no código, as chamadas á función `setcookie()` realízanse antes de calquera outro envío de cabeceiras HTTP.

```

index.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=="")
        $cadeaErros.= "O campo N° km non pode estar baleiro. ";
    else
        if (filter_var($nKm, FILTER_VALIDATE_INT)){
            if ((int)$nKm<=0)
                $cadeaErros.= "O N° km deber ser maior que 0. ";
        } else {
            $cadeaErros.= "O N° 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 ($cadeaErros == ""){
    setcookie("marcaModelo",$_POST["marMod"]);
    setcookie("numKm",$_POST["nKm"]);
    setcookie("urlFabricante",$_POST["urlFab"]);
    setcookie("combustible",$_POST["com"]);
    for ($i=0; $i<count($_POST["eqEx"]); $i++)
        setcookie("equipEx[$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>N° km: <input type="text" name="nKm" value="<?php echo $nKm;?>" /></p>
    <p>URL do fabricante: <input type="text" name="urlFab" value="<?php echo $urlFab;?>" /></p>
    <p>Combustible: <input type="radio" name="com" value="Diésel" <?php if ($com=="Diésel") echo "checked"?>> Diésel
    <input type="radio" name="com" value="Gasolina" <?php if ($com=="Gasolina") echo "checked"?>> Gasolina
    <input type="radio" name="com" value="Eléctrico/Híbrido" <?php if ($com=="Eléctrico/Híbrido") 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 dianteiro", $eqEx)){echo "selected=\'selected\'";}?>>Reposabrazos central diantei-
```

```

ro</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>

```

Exemplo de uso do formulario anterior:

Marca e modelo:
 Nº km:
 URL do fabricante:
 Combustible: ☐ Diesel ☒ Gasolina ☐ Eléctrico/Híbrido
 Equipamento extra:

Cando o **navegador solicita unha páxina PHP** a un servidor que gardou previamente cookies nese ordenador, o navegador **inclúe na cabeceira da petición HTTP todas as cookies creadas anteriormente** por ese servidor. O programa PHP recibe os nomes e valores das cookies e se gardan automaticamente na matriz **\$_COOKIE**, que é un array asociativo no que se accede a cada elemento indicando o nome da cookie.

No seguinte ficheiro exemplo.php, móstrase o acceso ás cookies que se gardaron sobre a información do automóbil.

Facer exemplo:
Cookies Automobil

exemplo.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title></title>
  </head>
  <body>
    <?php
      $marMod=(isset($_COOKIE["marcaModelo"]))
      ?$_COOKIE["marcaModelo"]
      :"";

      $nKm=(isset($_COOKIE["numKm"]))
      ?$_COOKIE["numKm"]
      :"";

      $url=(isset($_COOKIE["urlFabricante"]))
      ?$_COOKIE["urlFabricante"]
      :"";

      $com=(isset($_COOKIE["combustible"]))
      ?$_COOKIE["urlFabricante"]
      :"";

      $eqEx=(isset($_COOKIE["equipEx"]))
      ? $_COOKIE["equipEx"]
      : array();

      echo "<p>Os valores contidos na cookie 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>
```

A seguinte táboa mostra o resultado de acceder a exemplo.php , antes de crear a cookie, e despois. No primeiro caso, pódese observar que a como as cookies aínda non foron creadas, non se mostra información, ao contrario que na segunda imaxe.

Exemplo de formulario que usa o método GET	
<p>Os valores contidos na cookie son:</p> <p>Marca e modelo:</p> <p>Nº km:</p> <p>URL do fabricante:</p> <p>Combustible:</p> <p>Equipamento extra:</p>	<p>Os valores contidos na cookie son:</p> <p>Marca e modelo: VOLVO C70 2.4 140cv</p> <p>Nº km: 78000</p> <p>URL do fabricante: http://www.volvocars.com</p> <p>Combustible: http://www.volvocars.com</p> <p>Equipamento extra: Pintura metalizada, Sistema de navegación.</p>

É moi importante que os nomes das cookies non coincidan cos nomes dos controis dos formularios xa que PHP inclúe os valores das cookies no array `$_REQUEST`. Polo tanto, si o nome dunha cookie coincide co nome dun control, en `$_REQUEST` só se gardará o valor da cookie, e non do control.

A creación de cookies ten límites, e cada navegador ten un límite diferente. Polo xeral o **límite é aproximadamente 20 cookies por dominio**, borrando as antigas no caso de crear máis. E o **límite do valor é normalmente 4096 bytes**. O número total de cookies polo xeral é de 300, aínda que estes valores poden variar bastante segundo as versións dos navegadores. No caso concreto do navegador **Firefox**, os ficheiros que conteñen as cookies son **cookies.sqlite** e **cookies.txt**. A información gárdase nun formato semellante ao dunha táboa de base de datos, de xeito que o navegador sabe como acceder a cada rexistro.

A función `setcookie()` admite outros argumentos ademais do nome e o valor: **expiración** (número de segundos de vida da cookie), **ruta** (directorios dominio aos que se enviará a cookie), **dominio** (subdominio do dominio ao que se enviará posteriormente a cookie), **seguridade** (establece se soamente se enviará a cookie baixo conexións seguras) e **solohttp** (establece se a cookie será só accesible desde o servidor, e no navegador desde linguaxes como Javascript).

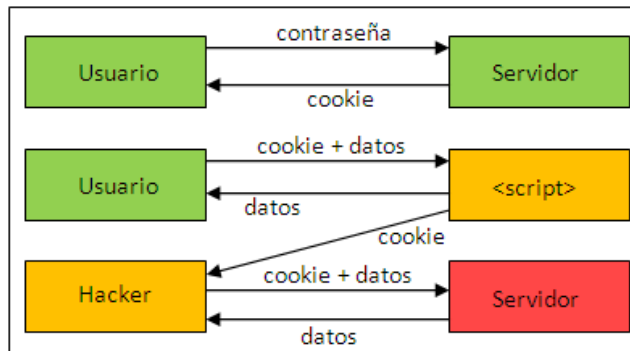
Para **modificar** unha cookie **se creará de novo mediante `setcookie()`**. Por outra banda, para **eliminar** unha cookie **pódese crear de novo con `setcookie()` co mesmo nome e asignarlle un parámetro expiración negativo**. Cómpre considerar que as cookies borranse ao pechar o navegador.

Aínda que as cookies non sexan programas, e polo tanto, non poden conter virus nin danar o ordenador, xeran certa desconfianza ata o punto que todos os navegadores inclúen opcións para impedir totalmente a creación de cookies, ou permitirllo unicamente a sitios autorizados. Loxicamente, impedir totalmente a creación de cookies complicará a visita dalgúns sitios web que baseen o seu funcionamento no uso de cookies. **Os motivos de desconfianza das cookies son debidos a que algúns sitios web gardan cookies de forma indiscriminada coa única finalidade de facer un seguimento dos hábitos de uso de internet do usuario.**

Existe un **tipo de ataque chamado XSS (Cross-Site Scripting)** que no que as cookies están involucradas. Un script do atacante, apodérase da sesión dun usuario nun servidor. Ao final o resultado é **secuestrar a sesión dun usuario**, roubarlle as cookies e cambiar a configuración da conta do usuario.

`setcookie()` argumentos

Modificar e eliminar cookie



As sesións

As sesións son un mecanismo que utilizan o servidor web para gardar información sobre o usuario e a súa actividade, para recuperala cada vez que o navegador lle pide unha páxina. Esta **información** estará **dispoñible** como moito **ata que o usuario peche o navegador**.

A diferenza con respecto ás cookies é que mediante as cookies a información se garda no ordenador do usuario e se envía en cada petición ao servidor, mentres que mediante as **sesións, a información gárdase no servidor**. De todos os xeitos, o uso de sesións implicará a creación dunha cookie. No uso das sesións distínguense as seguintes **tarefas: creación ou apertura da sesión, utilización da sesión e destrución ou peche da sesión**. A continuación explicarase cada unha destas accións, así como un exemplo de manexo de sesións.

Creación o apertura da sesión

`session_start()`

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();
?>
```

`session_name()`

A **sesión por defecto** ten o nome **PHPSESSID** asinando, aínda que ca función **session_name()** pódesele asignar outro nome diferente. Isto é necesario en casos nos que nos mesmo dominio se estean a executar varias aplicacións web que fagan uso de sesións. Se todas elas tivesen o mesmo nome de sesión, produciríase un conflito.

Exemplo de creación dunha sesión nominal

```
<?php
    session_name("AMiñaSesion");
```

```
session_start();
?>
```

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.

En PHP as sesións son ficheiros de texto gardados no disco do servidor. **O directorio onde se crean estes ficheiros ven especificado no arquivo de configuración php.INI, na directiva session.save_path.** Esta directiva se atopa á súa vez dentro da sección [Session] da configuración PHP. Este arquivos conteñen os valores dos diferentes elementos que se gardan na sesión.

Utilización da sesión

\$_SESSION

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.**

Destrución ou peche da sesión

session_destroy()

As sesións duran ata que o cliente pecha o navegador, aínda que as sesións tamén poden ser destruídas desde o script PHP mediante o uso da función session_destroy().

Código para a destrución dunha sesión

```
<?php
```



```
// Inicializar la sesión.
// Si está usando session_name("algo"), ¡no lo olvide ahora!
session_start();
// Destruir todas las variables de sesión.
$_SESSION = array();

// Finalmente, destruir la sesión.
session_destroy();
?>
```

Para poder destruir a sesión, primeiramente hai que abríla. Cando fose creada con nome, haberá que indicalo mediante a función `session_name()`. Neste caso, primeiramente destrúanse todas as variables de sesión. Unha vez feito o anterior, borrarase finalmente a sesión chamando á función `session_destroy()`.

Existen dúas directivas no ficheiro de configuración **php.INI** que controlan o tempo de vida das sesións. Unha é **session.cookie_lifetime**, que especifica o número de segundos de validez dunha cookie de sesión, e **por defecto o seu valor é cero indicando que a sesión estará dispoñible ata que se peche o navegador**. A outra variable é **session.gc_maxlifetime** que especifica o número de segundos máximos antes de que **o recolector de lixo elimine os ficheiros temporais**, entre os que se encontran os ficheiros de sesión, e por defecto o seu valor é 1440 segundos (**24 minutos**). No caso en se configure no servidor que os ficheiros temporais das sesións se estructuren en directorios dentro da carpeta por defecto, eses tempos poderían non ser moi fiables.

Exemplo de manexo de sesións

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 asignaráselle 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 ($com=="Gasolina") echo "checked"?> Gasolina
    <input type="radio" name="com" value="Eléctrico/Híbrido" <?php if ($com=="Eléctrico/Híbrido") 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 dianteiro", $eqEx)){echo "selected='selected'";}?>>Reposabrazos central diantei-
ro</option>
    </select></p>
    <p><input type="reset" value="Borrar" /> <input type="submit" value="Enviar" /></p>
</form>
<?php
}
?>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title></title>
</head>
<body>
<?php
if (count($_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 `exemplesessions.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 sesión que se quere abrir.

Código do ficheiro `exemplesessions.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>
```