

## 5. PHP i formularis

En les aplicacions web podem utilitzar formularis perquè l'usuari interactuï amb l'aplicació introduint o seleccionant dades. Aquests formularis són **formularis HTML** i les dades introduïdes o seleccionades per l'usuari, en el nostre cas, **es recullen en els scripts PHP**.

La utilització dels formularis dona dinamisme a les aplicacions web, ja que segons les dades introduïdes o seleccionades per l'usuari, l'aplicació mostrarà un contingut o altre.

### 5.1 Introducció als formularis en PHP

Mitjançant un exemple veurem com funciona la interacció entre els formularis HTML i els scripts PHP. En aquest cas, l'exemple constarà d'un formulari HTML on l'usuari podrà introduir un nom, un missatge i enviar les dades introduïdes, i un script PHP on es recolliran les dades introduïdes per l'usuari.

Comencem veient la **pàgina HTML** on es troba el formulari:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8"/>
5          <title>Formulari HTML simple</title>
6      </head>
7      <body>
8          <!-- Construïm un formulari que passarà al fitxer _lmostrarDades.php
9              les dades introduïdes per l'usuari en els
10             diferents inputs mitjançant l'URL --&gt;
11             &lt;form method="get" action="_laccesDades.php"&gt;
12
13                 &lt;p&gt;
14                     &lt;!--Construïm un input de tipus text per introduir un nom--&gt;
15                     Nom: &lt;input type="text" name="usuari"&gt;
16                     &lt;/p&gt;
17
18                     &lt;!--Construïm un input de tipus textarea per introduir un missatge
19                         --&gt;
20                     &lt;p&gt;Missatge: &lt;/p&gt;
21                     &lt;textarea name="missatge" rows="5" cols="40"&gt;&lt;/textarea&gt;
22
23                     &lt;!--Construïm un element de tipus button per enviar les dades
24                         introduïdes per l'usuari --&gt;
25                     &lt;p&gt;&lt;button type="submit" name="indexEnviar" value="enviar"&gt;Enviar
26                         Dades&lt;/button&gt;&lt;/p&gt;
27
28             &lt;/form&gt;
29         &lt;/body&gt;
30     &lt;/html&gt;</pre>
```

En aquest formulari, es pot introduir un nom mitjançant l'input de tipus “text”

nom “**missatge**” i enviar les dades mitjançant un element HTML de tipus “**button**” de nom “**indexEnviar**” i valor “**enviar**”.

Aquestes dades s'enviaran fent servir el mètode “**get**” que ja vam veure en l'apartat 1 “Instal·lació i configuració del servidor Apache”, que és el valor assignat a la variable “**method**” de l'element HTML “**form**”.

Finalment, les dades introduïdes per l'usuari es recolliran en l'script **\_1accesDades.php**, que és el valor assignat a la variable “**action**” de l'element HTML “**form**”.

Quan accedim a la pàgina, el que es mostrarà per pantalla és el que es pot veure en la figura 8.1:

**FIGURA 5.1.** Formulari HTML simple

The screenshot shows a web browser displaying a simple HTML form. The address bar shows the URL: localhost/M09ASIX/\_11Formularis/\_1formularioSimple.html. The form itself has two text input fields: one labeled "Nom:" and another labeled "Missatge:". Below these fields is a large empty text area. At the bottom of the form is a single button labeled "Enviar Dades".

Quan enviem dades introduïdes o seleccionades mitjançant un formulari, el sistema guarda aquestes dades en una de les variables predefinides globals de PHP **\$\_POST**, **\$\_GET** o bé **\$\_REQUEST**, depenent del mètode fet servir per enviar-les, és a dir, si el mètode ha estat **get**, les dades es guardaran en les variables **\$\_GET** i **\$\_REQUEST**, i si ha estat **post**, les dades es guardaran en les variables **\$\_POST** i també **\$\_REQUEST**. Aquestes variables són arrays associatius que utilitzen els noms (name) dels inputs del formulari com a índexs i en cada índex guarden el valor associat al nom (name) de l'input corresponent. Com són arrays associatius, podem treballar amb ells com treballaríem amb qualsevol array associatiu.

El següent script correspon a l'script **\_1accesDades.php** què és on es recolliran les dades introduïdes per l'usuari mitjançant el formulari anterior.

```

1 <?php
2
3 /*Les dades enviades es guarden en la variable predefinida global $_POST, $_GET
   o bé $_REQUEST.
4 *Aquesta variable és un array associatiu que utilitza els noms (name) dels
   inputs del formulari
5 *com a índexs i en cada índex guarda el valor associat al nom (name) de l'
   input corresponent.*/
6
7 //Mostrem el contingut de l'array associatiu $_GET
8 echo "<p>El contingut de l'array \$_GET és:<br/>";
9 print_r($_GET);
10 echo "</p>";

```

```

11 //Accedim a les dades de l'array $_GET
12 $usuari=$_GET["usuari"];
13 $missatge=$_GET["missatge"];
14
15 //Mostrem l'usuari i el missatge
16 echo"<p>L'usuari és: $usuari<br/>";
17 echo"<p>El missatge és: $missatge<br/>";
18
19 //Mostrem el contingut de l'array associatiu $_POST
20 echo"<p>El contingut de l'array \$_POST és:<br/>";
21 print_r($_POST);
22 echo"</p>";
23
24 //Mostrem el contingut de l'array associatiu $_REQUEST correponent al mètode
25 //get o post, dependent del
26 //mètode fet servir al formulari.
27 echo"<p>El contingut de l'array \$_REQUEST és:<br/>";
28 print_r($_REQUEST);
29 echo"</p>";
30
31 //Accedim a les dades mitjançant $_REQUEST
32 $usuari=$_REQUEST["usuari"];
33 $missatge=$_REQUEST["missatge"];
34
35 //Mostrem l'usuari i el missatge
36 echo"<p>L'usuari és: $usuari<br/>";
37 echo"<p>El missatge és: $missatge<br/>";
38
39 //Afegim un element nou als arrays $_POST i $_GET
40 $_GET["edat"] = 2;
41 $_POST["edat"] = 3;
42
43 //Mostrem contingut dels arrays $_POST, $_GET i $_REQUEST
44 echo"<p>El nou contingut de l'array \$_GET és:<br/>";
45 print_r($_GET); // Afegit element "edat" igual a 2
46 echo"</p>";
47 echo"<p>El nou contingut de l'array \$_POST és:<br/>";
48 print_r($_POST); // Afegit element "edat" igual a 3
49 echo"</p>";
50 echo"<p>El nou contingut de l'array \$_REQUEST és:<br/>";
51 print_r($_REQUEST); // No s'afegeixen cap dels elements afegits amb GET o POST
52 echo"</p>";

```

Si l'usuari mitjançant el formulari anterior ha introduït el nom de “Xesca” i el missatge “Això és un missatge”, el resultat que es mostrerà per pantalla d'aquest script serà el que es pot veure en la figura 5.2:

Com podem veure, en la variable \$\_GET i \$\_REQUEST, es guardaran els valors “Xesca” corresponent a líndex “usuari”, “Això és un missatge” corresponent a líndex “missatge” i “enviar” corresponent a líndex “indexEnviar”. En el cas de la variable \$\_POST, el seu contingut és buit perquè el mètode utilitzat per enviar les dades ha estat get, no post.

També podem observar, que en tractar-se d'arrays associatius, hem accedit als seus noms i missatges com qualsevol array associatiu, guardant els seus valors, en les variables \$usuari i \$missatge respectivament. De la mateixa manera, finalment, hem afegit contingut a les variables \$\_GET i \$\_POST, com fem amb qualsevol array associatiu.

En el cas de la variable \$\_GET, el contingut l'hem afegit al final de l'array conservant les dades provinents del formulari. En el cas de \$\_POST, l'únic contingut que conté és el nou valor afegit, és a dir, l'índex “edat” amb valor 2.

**FIGURA 5.2.** Resultat script \*\*\_1accesDades.php\*\*

```

← → C ⓘ localhost/M09ASIX/_1Formularis/_1accesDades.php?usuari=Xesca&missatge=Això+és+un+missatge&indexEnviar=enviar

El contingut de l'array $_GET és:
Array ( [usuari] => Xesca [missatge] => Això és un missatge [indexEnviar] => enviar )

L'usuari és: Xesca
El missatge és: Això és un missatge
El contingut de l'array $_POST és:
Array ( )

El contingut de l'array $_REQUEST és:
Array ( [usuari] => Xesca [missatge] => Això és un missatge [indexEnviar] => enviar )

L'usuari és: Xesca
El missatge és: Això és un missatge
El nou contingut de l'array $_GET és:
Array ( [usuari] => Xesca [missatge] => Això és un missatge [indexEnviar] => enviar [edat] => 2 )

El nou contingut de l'array $_POST és:
Array ( [edat] => 3 )

El nou contingut de l'array $_REQUEST és:
Array ( [usuari] => Xesca [missatge] => Això és un missatge [indexEnviar] => enviar )

```

## 5.2 Formularis amb inputs checkbox

Mitjançant un exemple veurem com funciona la interacció entre els formularis HTML amb inputs de tipus checkbox i els scripts PHP. En aquest cas, l'exemple constarà d'un formulari HTML on l'usuari podrà introduir un nom, seleccionar productes i enviar les dades introduïdes.

Comencem veient la **pàgina HTML** on es troba el formulari:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Seleccionar Productes</title>
5          <meta charset="UTF-8"/>
6      </head>
7      <body>
8          <form action="_2mostrarProductes.php" method="post">
9              <p>
10                 Nom: <input type="text" id="nom" name="usuari" />
11             </p>
12
13             <fieldset>
14
15                 <legend>Selecciona alguns dels productes:</legend><br/>
16
17                 <input type="checkbox" name="productes[]" value="poma"/> Poma<br/>
18
19                 <input type="checkbox" name="productes[]" value="pera"/> Pera<br/>
20
21                 <input type="checkbox" name="productes[]" value="plàtan"/> Plàtan
22
23             </fieldset>
24
25             <button type="submit" name="enviar" value="enviar">Enviar productes</button>

```

```

26   </form>
27   </body>
28 </html>
```

En aquest formulari, es pot introduir un nom mitjançant l'input de tipus “**text**” de nom “**usuari**”, seleccionar els productes que l'usuari vulgui mitjançant els inputs de tipus “**checkbox**” de nom “**productes**” amb valors “**pera**”, “**poma**” i “**plàtan**”, i enviar les dades mitjançant un element HTML de tipus “**button**” de nom “**enviar**” i valor “**enviar**”.

Aquestes dades s'enviaran fent servir el mètode “**post**” que ja vam veure en l'apartat 1 “Instal·lació i configuració del servidor Apache”, que és el valor assignat a la variable “**method**” de l'element HTML “**form**”. Finalment les dades introduïdes per l'usuari, es recolliran en l'script **\_2mostrarProductes.php**.

Quan accedim a la pàgina, el que es mostrarà per pantalla és el que es pot veure en la figura 5.3:

**FIGURA 5.3.** Formulari HTML amb inputs de tipus checkboxes

El següent script correspon a l'script **\_2mostrarProductes.php** al qual s'envien les dades del formulari anterior:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Mostrar productes seleccionats</title>
5     <meta charset="UTF-8"/>
6     <?php include "_2accesProductes.php"; ?>
7   </head>
8   <body>
9     <p>Benvingut/da, <strong><?php echo $usuari; ?></strong>!</p>
10    <p>Els productes seleccionats són:<br>
11      <?php llistarProductes($productes); ?>
12    </p>
13  </body>
14 </html>
```

En aquest cas, tenim un fitxer que mostra el contingut de la variable **\$usuari** i el retorn de la funció **llistarProductes(\$productes)** passant-li com a paràmetre la variable **\$productes**, però enlloc veiem on es recullen les dades del formulari. Doncs bé, tant la recollida de dades del formulari, com el contingut de les variables **\$usuari** i **\$productes**, com la definició de la funció **llistarProductes(\$productes)**, els trobarem en el fitxer **\_2accesProductes.php**, inclòs dins d'aquest script.

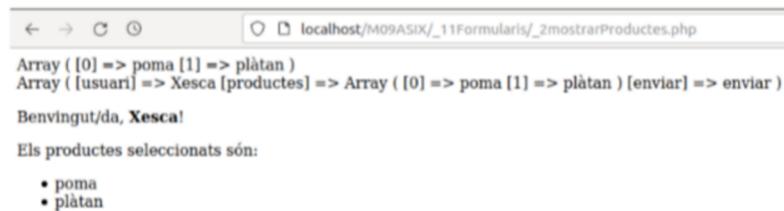
L'script corresponent a **\_2accesProductes.php** és el següent:

```

1 <?php
2
3 $enviar = $_POST["enviar"]; //Recollim name del button
4
5
6 if ($enviar) { //Si s'han enviat les dades...
7
8     //Accedim al nom d'usuari i productes
9     $usuari = $_POST["usuari"];
10    $productes = $_POST["productes"];
11
12    //Mostrem el contingut de l'array productes
13    print_r($productes);
14    echo "<br/>";
15
16    //Mostrem el contingut de la variable $_POST
17    print_r($_POST);
18
19    //Accedim als productes seleccionats i els mostrem en una llista HTML
20    function llistarProductes($productes) {
21
22        if (!empty($productes)) { //Si la variable productes no està buida...
23
24            echo "<ul>"; //Creem llista html
25            //Fem un recorregut per l'array productes..
26            foreach ($productes as $producte) {
27                echo "<li>$producte</li>"; //Mostrem producte actual com li
28            }
29            echo "</ul>";
30
31        } else { //Si està buida
32            echo "No has seleccionat cap producte";
33        }
34    }
35
36
37 } else { //Si les dades no s'han enviat correctament..
38     echo "Les dades no s'han enviat correctament";
39 }
```

Si l'usuari mitjançant el formulari ha introduït el nom de “Xesca” i ha seleccionat la poma i el plàtan, el resultat que es mostrarà per pantalla de l'script **\_2mostrarProductes.php** serà el que es pot veure en la figura 5.4:

**FIGURA 5.4.** Resultat script **\_2mostrarProductes.php**



Com podem veure, en la variable `$_POST` de l'script `_2accesProductes.php`, es guardaran els valors “Xesca” corresponent a l'índex “usuari”, un array numèric amb els valors dels checkboxs seleccionats (poma i plàtan) corresponent a l'índex “productes” i “enviar” corresponent a l'índex “enviar”.

A les variables `$usuari` i `$productes` se'ls assignen el nom “Xesca” introduït per l'usuari i l'array numèric amb la selecció de poma i plàtan respectivament.

Finalment, la funció **llistarProductes(\$productes)**, el que fa és crear un llistat HTML amb el contingut de l'array passat per paràmetre, en el nostre cas, l'array de productes \$productes.

### 5.3 Formularis amb inputs radio

Mitjançant un exemple veurem com funciona la interacció entre els formularis HTML amb inputs de tipus radio i els scripts PHP. En aquest cas, l'exemple constarà d'un formulari HTML on l'usuari haurà de seleccionar el menú que han escollit els comensals d'una taula d'un restaurant (tots han escollit el mateix menú) i introduir el nombre de comensals de la taula per finalment calcular l'import que han de pagar en total.

Comencem veient la **pàgina HTML** on es troba el formulari:

```
1 <!DOCTYPE html>
2 <html>
3
4     <head>
5         <title>Càcul import taules</title>
6         <meta charset="UTF-8"/>
7     </head>
8
9     <body>
10        <strong>Càcul import total de les taules</strong>
11        <form method=POST action="_3restaurant.php"><br/>
12            <input type="hidden" name="operacio" value="00005">
13
14            Quin d'aquests menús ha escollit la taula?<br/><br/>
15            <input name="preuMenu" type="radio" value=20 />Menú migdia de 20
16                euros<br/>
17            <input name="preuMenu" type="radio" value=30 />Menú nit de 30 euros
18                <br/>
19            <input name="preuMenu" type="radio" value=35 />Menú cap de setmana
20                de 35 euros<br/>
21                <br/>
22            Quants comensals són? <input type="text" name="comensals" size=5 />
23                <br/><br/>
24            <input type="submit" value="Calcular">
25        </form>
26
27    </body>
28
29 </html>
```

En aquest formulari, es pot seleccionar el menú mitjançant un input de tipus “radio” de nom “**preuMenu**” entre els tres inputs amb valors “20”, “30” i “35”, per després introduir el nombre de comensals mitjançant un input de tipus “text” de nom “**comensals**”, per, finalment, enviar les dades mitjançant un input de tipus “submit”. Aquestes dades s’enviaran fent servir el mètode “post” a l’script **\_3restaurant.php**.

Quan accedim a la pàgina, el que es mostrarà per pantalla és el que es pot veure en la figura 5.5:

**FIGURA 5.5.** Formulari HTML amb inputs de tipus radio

Càcul import total de les taules

Quin d'aquests menús ha escollit la taula?

Menú migdia de 20 euros  
 Menú nit de 30 euros  
 Menú cap de setmana de 35 euros

Quants comensals són?

**Calcular**

El següent script correspon a l'script **\_3restaurant.php** al qual s'envien les dades del formulari anterior:

```

1  <!DOCTYPE html>
2  <html>
3
4      <head>
5          <title>Resultat</title>
6          <meta charset="UTF-8"/>
7
8      </head>
9
10     <body>
11         <?php
12             //Contingut de l'array $_POST
13             print_r($_POST);
14
15             $codiOperacio = $_POST["operacio"]; //Codi operació del càlcul el total
16                 de l'import dels menús
17             $comensals = $_POST["comensals"];
18
19             //Calculem l'import total
20             $import = $_POST["preuMenu"] * $comensals;
21
22             //Mostrem resultat
23             echo "<h2>Operacio $codiOperacio</h2> El preu total dels menús per
24                 $comensals comensals és $import euros.";
25
26         ?>
27     </body>
28
29 </html>
```

Si l'usuari mitjançant el formulari ha seleccionat el menú de migdia i ha introduït 4 comensals, el resultat que es mostrarà per pantalla d'aquest script serà el que es pot veure en la figura 5.6:

**FIGURA 5.6.** Resultat script **\_3restaurant.php**

Array ( [operacio] => 00005 [preuMenu] => 20 [comensals] => 4 )

## Operació 00005

El preu total dels menús per 4 comensals és 80 euros.

Com podem veure, en la variable `$_POST`, es guardaran els valors “00005” corresponent a l’índex “operacio”, 20 corresponent a l’índex “preuMenu” i “4” corresponent a l’índex “enviar”.

A la variable **\$codiOperacio** se li assigna el valor “00005” de l’input de tipus **hidden**, el qual no és un input que permet que l’usuari interactuï amb l’aplicació. A la variable **\$comensals** se li assignen el nombre de comensals introduïts per l’usuari. A la variable **\$import** se li assigna el preu que ha de pagar la taula, calculat a partir del preu del menú, és a dir, **\$\_POST[“preuMenu”]**, seleccionant un dels inputs radio, i el nombre de comensals.

## 5.4 Formularis amb elements select

Mitjançant un exemple veurem com funciona la interacció entre els formularis HTML amb elements select i els scripts PHP. En aquest cas, l’exemple constarà d’un formulari HTML on l’usuari haurà de seleccionar un menú.

Comencem veient la **pàgina HTML** on es troba el formulari:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8"/>
5     <title>Selecció</title>
6   </head>
7   <body>
8
9     <!-- construim una selecció d'opcions-->
10    <form method="post" action="_4mostrarSeleccio.php">
11      <fieldset>
12
13        <legend>Selecciona un dels menús:</legend><br/>
14        <select name="menu">
15          <option value="1">Menú 1</option>
16          <option value="2">Menú 2</option>
17          <option value="3">Menú 3</option>
18        </select>
19
20      </fieldset>
21      <button type="submit" value="enviar" name="enviar">Enviar Menú</
22        button>
23
24    </form>
25  </body>
</html>
```

En aquest formulari, es pot seleccionar un únic menú mitjançant un input de tipus “**select**” de nom “**menu**” entre les tres opcions amb valors “1”, “2” i “3”, per després enviar el menú seleccionat mitjançant un element “**button**”. Aquestes dades s’enviaran fent servir el mètode “**post**” a l’script **\_4mostrarSeleccio.php**.

Quan accedim a la pàgina, el que es mostrarà per pantalla és el que es pot veure en la figura 5.7:

**FIGURA 5.7.** Formulari HTML amb element select

localhost/M09ASIX/\_11Formularis/\_4mostrarSeleccio.php

Selecciona un dels menús:

Menú 1

Enviar Menú

El següent script correspon a l'script **\_4mostrarSeleccio.php** al qual s'envien les dades del formulari anterior:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Mostrar productes seleccionats</title>
5     <meta charset="UTF-8"/>
6     <?php include "_4seleccio.php"; ?>
7   </head>
8   <body>
9     <p> <?php llistarMenus(); ?></p>
10  </body>
11 </html>
```

En aquest cas, tenim un fitxer que mostra el retorn de la funció **llistarMenus()**, però enlloc veiem on es recullen les dades del formulari. Doncs bé, tant la recollida de dades del formulari, com la definició de la funció **llistarMenus()**, els trobarem en el fitxer **\_4seleccio.php**, inclòs dins d'aquest script.

L'script corresponent a **\_4seleccio.php** és el següent:

```

1 <?php
2 print_r($_POST);
3
4 function llistarMenus() {
5
6   $menu = $_POST["menu"];
7
8   if (!empty($menu)) { //Si la variable menú no està buida..
9     echo "El menú seleccionats és:<br>";
10    echo "<ul>"; //Cream llista html
11    echo "<li>$menu</li>"; //Mostrem menú actual com li
12    echo "</ul>";
13  } else { //Si està buida...
14    echo "No has seleccionat cap menú";
15  }
16}
```

Si l'usuari mitjançant el formulari ha seleccionat el menú 1, el resultat que es mostrarà per pantalla de l'script **\_4mostrarSeleccio.php** serà el que es pot veure en la figura 5.8:

**FIGURA 5.8.** Resultat script **\_4mostrarSeleccio.php**

localhost/M09ASIX/\_11Formularis/\_4mostrarSeleccio.php

Array ( [menu] => 1 [enviar] => enviar )

El menú seleccionats és:

\* 1

Com podem veure, en la variable **\$\_POST** de l'script **\_4mostrarSeleccio.php**, es

mitjançant l'element select, i “enviar” corresponent a l'índex “enviar” de l'element button.

La funció **llistarMenus()**, el que fa és crear un llistat HTML amb el valor 1 del menú seleccionat.

Quan el select és múltiple, és a dir, ens permet seleccionar més d'una opció, els valors de les opcions seleccionades es guardaràn en un array numèric què és el que s'enviarà al servidor, igual que fem amb els inputs de tipus checkbox.

## 5.5 Galetes ('cookies')

Una **galeta** o **cookie**, és un petit fitxer de text que envia el servidor web a l'usuari i que s'emmagatzema en el directori de configuració del navegador que en aquell moment estigui utilitzant l'usuari.

Les galetes emmagatzemen informació referent a la visita de l'usuari a una pàgina web determinada i s'utilitzen principalment per conèixer hàbits de navegació de l'usuari que, entre altres coses, ens poden ajudar a augmentar l'eficiència en l'accés.

Mitjançant scripts PHP podem crear, modificar i eliminar galetes. Veiem com fer-ho mitjançant el següent script d'exemple:

L'usuari pot consultar el contingut i eliminar les galetes des dels navegadors.

```

1 <?php
2 /*setcookie defineix una cookie per ser enviada juntament amb la resta de capç
   aleres HTTP, per tant ha de ser
3 *enviades abans que el script generi cap sortida com els elements HTML*/
4
5 //Creem una cookie de nom producte, amb valor pastanaga, que caduqui en una
   hora, que estigui disponible per //qualssevol URL del servidor, amb domini
   buit i que es pugui enviar mitjançant una connexió no segura (http) amb el
   //servidor.
6 setcookie("producte","pastanaga",time()+3600, "/","",0);
7
8 print_r($_COOKIE);
9
10 if (isset($_COOKIE["producte"])) { //Si la cookie existeix en el navegador...
11   //Mostrem valor de la cookie.
12   echo "<p>Hola de nou! Has escollit: ".$_COOKIE["producte"]."</p>";
13
14   //Modifiquem el valor de la cookie. El canvi no el podrem mostrar per
     pantalla fins que no recarreguem la pàgina
15   ////(tornem a enviar la petició HTTP).
16   setcookie("producte","tomaquet",time()+3600, "/","",0);
17
18 } else { //Si no existeix (primer cop que executem script)...
19   //Mostrem missatge.
20   echo "<p>Hola. Aquesta és la teva primera visita.</p>";
21 }
22
23 //Eliminem la cookie creada
24 setcookie("producte","tomaquet",time()-3600, "/","",0);

```

Com podem veure, mitjançant la funció **setcookie** hem creat la galeta de nom

actual en segons que ens retorna la funció **time()** més **3600** segons que són els segons equivalents a una hora, que estarà disponible per qualsevol URL del servidor ("**/**"), que no estarà disponible per cap domini en concret ("**"**) i que per accedir a ella ho farem mitjançant el protocol no segur HTTP (**0**).

La funció predefinida de PHP **time()** ens retorna la data actual del sistema en format UNIX, és a dir, els segons transcorreguts des de l'1 de gener de 1970 a les 00:00:00 fins a la data actual.

En la variable predefinida global de PHP **\$\_COOKIE**, el sistema guarda com a índex el nom de la galeta creada, és a dir, **producte** i el contingut **pastanaga** de la galeta com a valor associat a l'índex. Si haguéssim creat més galetes, les guardaria totes dins d'aquesta variable, on per cada galeta crearia un índex associant-li el seu valor corresponent, tal com hem vist en el cas del nostre script.

En l'script modifiquem el contingut de la galeta creada mitjançant la funció **setcookie**, simplement canviant el valor del contingut **pastanaga** per **tomaquet**.

Finalment, eliminem la galeta creada també mitjançant la funció **setcookie** modificant la data de caducitat per una data que ja ha passat, en el nostre cas -3600 segons, és a dir, passada una hora.

## 5.6 Sessions

A vegades ens interessa emmagatzemar informació de manera temporal perquè quan naveguem per les diferents pàgines d'un lloc web, aquesta informació estigui disponible en qualsevol d'aquestes pàgines. Doncs bé, a PHP això ho aconseguim mitjançant les **sessions**.

Les **sessions**, per defecte, ens permeten guardar informació en el servidor accessible des de qualsevol pàgina d'una aplicació web fins que tanquem el navegador.

### 5.6.1 Introducció a les sessions

Mitjançant un exemple veurem com funcionen les sessions. En aquest cas, l'exemple és una aplicació web que permet a un usuari autenticar-se mitjançant un nom d'usuari i una contrasenya.

El següent codi pertany a la pàgina **\_4Inici.html** que és la pàgina per on comença l'aplicació:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8"/>
5   <title>Inici</title>
6 </head>
7 <body>
8

```

```

9   <form method="POST" action="_4Autenticacio.php">
10
11     <label for="usuari">Usuari: </label><input type="text" name="usuari" id="usuari" /><br/><br/>
12     <label for="contrasenya">Contrasenya: </label><input type="password" name="contrasenya" id="contrasenya" /><br/><br/>
13
14     <input type="submit" name="enviar" value="Accedir" />
15
16   </form>
17
18 </body>
19 </html>
```

Com podem veure, en la pàgina **\_4Inici.html** trobem el formulari perquè l'usuari pugui introduir el seu nom d'usuari i contrasenya. Les dades introduïdes per l'usuari arribaran al fitxer **\_4Autenticacio.php** mitjançant el mètode **POST**.

El següent codi pertany al fitxer **\_4Autenticacio.php**:

```

1 <?php
2 define("USUARI", "Xesca"); //Definim nom d'usuari vàlid
3 define("PASSWORD", "123"); //Definim contrasenya vàlida
4
5 if ($_POST["usuari"] == USUARI && $_POST["contrasenya"] == PASSWORD) {//Si l'autenticació és correcte...
6
7     session_start(); //Iniciem sessió.
8
9     //Guardem les dades de l'usuari autenticat en la sessió
10    $_SESSION["usuari"] = $_POST["usuari"];
11    $_SESSION["contrasenya"] = $_POST["contrasenya"];
12
13    $_SESSION["ultimAcces"] = time(); //Inicialitzem la data d'inici de sessió
14
15    //Mostrem la pàgina de l'aplicació.
16    //La funció header envia encapçalaments HTTP sense el format HTTP
17    header("Location:_4Aplicacio.php");
18
19 } else { //Si l'autenticació no és correcte...
20     header("location:_4Inici.html"); //Retornem a la pàgina inicial.
21 }
```

En el codi podem veure que en primer lloc es defineixen dues constants, **USUARI** i **PASSWORD**, a les quals se'ls han assignat el nom d'usuari **Xesca** i la contrasenya **123** respectivament, vàlids perquè l'autenticació sigui un èxit. És a dir, perquè l'autenticació sigui correcta, el nom d'usuari i contrasenya introduïts per l'usuari han de coincidir amb els valors de les dues constants, que és el que es comprova amb la següent estructura condicional del codi.

Si l'autenticació és correcta (es compleix la condició de l'**if**), iniciem una sessió mitjançant la funció predefinida de PHP **session\_start()**. Un cop iniciada la sessió podem treballar amb la variable predefinida global de PHP **\$\_SESSION** que és on guardarem la informació que volem que estigui disponible per totes les pàgines de la nostra aplicació web. Aquesta variable és un vector associatiu on el contingut de cada element identificat per un índex, són les diferents informacions que guardarem perquè estiguin disponibles.

En el nostre cas guardarem el nom d'usuari en l'índex **usuari**, la contrasenya en l'índex **contrasenya** i la data del sistema actual, és a dir, la data del sistema en

el moment de crear la sessió, en l'índex **ultimAcces**. Finalment l'aplicació ens redirigirà al fitxer **\_4Aplicacio.php** mitjançant la funció **header**.

#### Funció predefinida PHP header

La funció predefinida de PHP **header** envia encapsulats sense format HTTP, així doncs s'ha de cridar abans que es mostri qualsevol altre contingut per pantalla com codi HTML. Com a paràmetre passem l'encapsulat que volem enviar. Si volem que en redirigeixi a una pàgina, com a encapsulat haurem d'escriure la cadena "location:urlFitxer" on urlFitxer és l'URL del fitxer a la qual volem que ens redirigeixi.

```

1 <?php
2 session_start(); //Iniciem sessió
3 include "_4GestioSessio.php";
4 controlSessio();
5 ?>
6
7 <!DOCTYPE html>
8 <html>
9   <head>
10     <meta charset="UTF-8"/>
11     <title>Aplicació</title>
12   </head>
13   <body>
14     <p>Benvinguda <strong><?php echo $usuari; ?></strong></p>
15     <p><a href="_4Logout.php">Tanca la sessió</a></p>
16     <p><a href="_4Inici.html">Torna a l'inici</a></p>
17   </body>
18 </html>
```

Com podem veure en aquest fitxer ens interessa mostrar el nom d'usuari de l'autenticació definit com a constant en el fitxer **\_4Autenticacio.php** què és el mateix que ha introduït l'usuari mitjançant el formulari de la pàgina **\_4Inici.html** i el que hem guardat en la variable **\$\_SESSION**. Això ho fem mostrant el contingut de la variable **\$usuari**. Així doncs les definicions d'aquesta variable i de la funció **controlSessio()** les trobarem en el fitxer **\_4GestioSessio.php** que hem inclòs.

Tant pel contingut de la variable com per la funció **controlSessio()**, necessitarem dades que hem guardat a la variable **\$\_SESSION**, així doncs per poder accedir a aquestes dades necessitem reiniciar la sessió que hem creat al fitxer **\_4Autenticacio.php**. Per reiniciar una sessió utilitzem la mateixa funció que per crear-la, és a dir, **session\_start()**, que és el que fem al codi abans d'incloure el fitxer **\_4GestioSessio.php**.

Per tant, cada cop que vulguem accedir al contingut de la variable **\$\_SESSION** des de qualsevol pàgina de l'aplicació web, abans hem de reiniciar la sessió amb **session\_start()**.

Finalment ens trobem amb dos enllaços HTML, un ens retornarà a la pàgina d'inici **\_4Inici.html** i l'altre a l'script **\_4Logout.php** que el que farà és tancar la sessió que hem creat.

El següent codi pertany al fitxer **\_4Logout.php**:

```

1 <?php
2 session_start(); //Iniciem sessió
3 include "_4GestioSessio.php";
4 logout();
5 ?>
6
7 <!DOCTYPE html>
8 <html>
```

```

9 <head>
10   <meta charset="UTF-8"/>
11   <title>Tancar Sessió</title>
12 </head>
13   <body>
14     <h3>Sessió finalitzada</h3>
15     <a href="_4Inici.html">Tornar a l'inici</a>
16   </body>
17 </html>
```

Com en el fitxer anterior `_4Aplicacio.php`, la definició de la funció `logout()` també la trobem en el fitxer `_4GestioSessio.php` que hem inclòs. Aquesta funció, entre altres coses, treballa amb la variable `$_SESSION`, per tant, de nou necessitem reiniciar la sessió creada al fitxer `_4Autenticacio.php` mitjançant la funció `session_start()`. L'objectiu d'aquesta funció és **tancar la sessió**, és a dir, destruir-la i buidar el contingut de la variable `$_SESSION`.

Finalment ens trobem amb un enllaç HTML que ens retornarà a la pàgina d'inici `_4Inici.html`.

Només ens queda conèixer el contingut del fitxer `_4GestioSessio.php`. El següent codi pertany al contingut d'aquest fitxer:

```

1 <?php
2
3 function controlSessio() {
4   global $usuari; //Variable que podrem utilitzar fora de la funció
5
6   $usuari=$_SESSION["usuari"]; //Usuari guardat en la sessió
7
8   ...
9
10  print_r($_SESSION); //Mostrem contingut de la sessió
11
12  ...
13 }
14
15 function logout(){
16   session_destroy(); //Destruïm sessió
17   print_r($_SESSION);
18   $_SESSION=array(); //Buidem array $_SESSION. És el mateix què unset(
19   //$_SESSION) o session_unset()
20   echo "</br></br>Hem buidat la sessió: ";
21   print_r($_SESSION);
}
```

Primer trobem la definició de la funció `controlSessio()` que es crida en l'script `_4Aplicacio.php`. Com podem veure el que fa és definir la variable `$usuari` com a global perquè la puguem utilitzar fora de la funció, en el nostre cas, com hem vist, en l'script `_4Aplicacio.php`, per després assignar-li un valor, en concret, el nom d'usuari guardat dins la variable `$_SESSION` i, finalment, imprimir per pantalla el seu contingut mitjançant `print_r($_SESSION)`.

En segon lloc trobem la definició de la funció `logout()` que es crida en l'script `_4Logout.php`. Com hem dit abans, el que fa aquesta funció és destruir la sessió, això ho fa cridant a la funció `session_destroy()`, i buidar la variable `$_SESSION`.

En el nostre cas buidem aquesta variable assignant-li un array buit creat mitjançant la funció `array()` sense passar-li com a paràmetres cap contingut. Tot i això,

aquesta no és l'única forma de buidar un array, també podem fer servir la funció `unset($_SESSION)`, on com podem veure, com a paràmetre li passem l'array que volem buidar, en el nostre cas `$_SESSION`. Finalment, una altra manera de buidar la variable `$_SESSION`, és cridant a la funció `session_unset()`.

### 5.6.2 Caducitat d'una sessió

Quan parlem de **caducitat d'una sessió**, ens referim al fet que la sessió es tancarà un cop transcorregut un temps determinat sense que hi hagi cap interacció amb el lloc web. Per defecte aquest temps és de 1440 segons.

#### Temps per defecte

El temps per defecte de 1440 segons en què caduca una sessió, el trobem definit en el fitxer `php.ini`, com a valor assignat a la directiva `session.gc_maxlifetime`.

Nosaltres podem modificar aquest temps perquè la caducitat es produueixi quan a nosaltres ens interessi.

Mitjançant la funció `controlSessio()` del següent script, podem veure com treballar amb la caducitat d'una sessió programada. Recordem que aquesta funció la cridem cada cop que carreguem el fitxer **\_4Aplicacio.php**.

```

1 <?php
2
3 function controlSessio() {
4
5     ...
6
7     define( "TEMPSINACTIU", 5 ); //Segons màxims que pot estar l'aplicació
8         inactiva
9
10    ...
11
12    //Temps transcorregut des de l'últim accés a la pàgina i la data actual.
13    $tempsTranscorregut = time() - $_SESSION["ultimAcces"];
14
15    if ( $tempsTranscorregut > TEMPSINACTIU ) { //Si la sessió ha caducat, han
16        passat 5 segons o més des de l'últim accés...
17        session_destroy(); //Destruïm sessió
18        header( "Location: _4Caducitat.html" ); //Mostrem la pàgina de
19            caducitat
20    } else { //Si no ha caducat...
21        $_SESSION["ultimAcces"] = time(); //Actualitzem la data per tornar a
22            començar
23    }
24 }
```

Quan cridem a la funció `controlSessio()`, el que primer es fa és crear una constant “`TEMPSINACTIU`” a la que li assignem 5, aquests seran els segons màxims que pot estar l'aplicació inactiva, és a dir, sense que interactuem amb ella.

A continuació, definim la variable `$tempsTranscorregut` a la que inicialitzarem amb el resultat de restar a la data actual del sistema obtinguda mitjançant la funció `time()` el valor assignat a `$_SESSION["ultimAcces"]`, d'aquesta manera sabrem quant de temps ha passat sense interactuar amb l'aplicació des que hem carregat la pàgina **\_4Aplicacio.php** què és on cridem a la funció.