



UNIVERSIDAD
DE GRANADA



Tecnologías Web

Grado en Ingeniería Informática

Tema 3 – Programación en el lado del servidor PHP y aplicaciones web

Este documento está protegido por la Ley
de Propiedad Intelectual ([Real Decreto Ley
1/1996 de 12 de abril](#)).
Queda expresamente prohibido su uso o
distribución sin autorización del autor.

© Javier Martínez Baena
jbaena@ugr.es

Departamento de Ciencias de la
Computación e Inteligencia Artificial
<http://decsai.ugr.es>



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor – PHP+web

1. El lenguaje PHP
2. PHP y aplicaciones web
 1. Uso de PHP en la web
 2. Procesamiento de formularios
 3. Saneamiento de cadenas
 1. URL encoding
 2. Saneamiento de cadenas
 3. Query strings
 4. Recordando el estado de las aplicaciones
 1. Cookies
 2. Sesiones
 5. Envío de encabezados
 3. PHP y conexión con BBDD



Uso de PHP en la web

Ejemplo



Situación típica

Todas las páginas de un sitio mantienen elementos comunes: encabezados, pie de página, menú de navegación, etc

Índice

[Inicio](#) [Alan Turing](#) [Ada Lovelace](#) [Contacto](#)

Personajes históricos

Pulsa para ver la biografía de alguno de ellos.

(C) Pepito Pérez

Índice

[Inicio](#) [Alan Turing](#) [Ada Lovelace](#) [Contacto](#)

Contacto

Envía un correo a PepitoPerez@servidor.de.correo.com

(C) Pepito Pérez

Índice

[Inicio](#) [Alan Turing](#) [Ada Lovelace](#) [Contacto](#)

Alan Turing

Alan Mathison Turing, OBE (Paddington, Londres, 23 de junio de 1912-Wilmslow, Cheshire, 7 de junio de 1954), fue un matemático, lógico, científico de la computación, criptógrafo, filósofo, maratoniano y corredor de ultra distancia británico. Es considerado uno de los padres de la ciencia de la computación y precursor de la informática moderna. Proporcionó una influyente formalización de los conceptos de algoritmo y computación: la máquina de Turing. Formuló su propia versión de la hoy ampliamente aceptada tesis de Church-Turing (1936).

(C) Pepito Pérez

Índice

[Inicio](#) [Alan Turing](#) [Ada Lovelace](#) [Contacto](#)

Ada Lovelace

Augusta Ada King, Condesa de Lovelace , (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 - Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la denominada máquina analítica. Entre sus notas sobre la máquina se encuentra lo que se reconoce hoy como el primer algoritmo destinado a ser procesado por una máquina, por lo que se la considera como la primera programadora de ordenadores.

(C) Pepito Pérez

Uso de PHP en la web

Ejemplo



pag_comun.php

```
<?php
function HTMLinicio($titulo) { ... }
function HTMLfin() { ... }
function HTMLnav($activo) { ... }
function HTMLfooter() { ... }

function HTMLpag_inicio() { ... }
function HTMLpag_alan() { ... }
function HTMLpag_ada() { ... }
function HTMLpag_contacto() { ... }
?>
```

```
function HTMLinicio($titulo) {
echo <<< HTML
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="pag_estilo.css">
<title>$titulo</title>
</head>
<body>
HTML;
}

function HTMLfin() {
echo <<< HTML
</body>
</html>
HTML;
}

function HTMLfooter() {
echo <<< HTML
<footer>
<small>(C) Pepito Pérez</small>
</footer>
HTML;
}
```

Uso de PHP en la web**Ejemplo****pag_inicio.php**

```
<?php
require "pag_comun.php";
HTMLinicio("Ejemplo de PHP");
HTMLnav(0);
HTMLpag_inicio();
HTMLfooter();
HTMLfin();
?>
```

pag_contacto.php

```
<?php
require "pag_comun.php";
HTMLinicio("Ejemplo de PHP");
HTMLnav(3);
HTMLpag_contacto();
HTMLfooter();
HTMLfin();
?>
```

pag_ada.php

```
<?php
require "pag_comun.php";
HTMLinicio("Ejemplo de PHP");
HTMLnav(2);
HTMLpag_ada();
HTMLfooter();
HTMLfin();
?>
```

pag_alan.php

```
<?php
require "pag_comun.php";
HTMLinicio("Ejemplo de PHP");
HTMLnav(1);
HTMLpag_alan();
HTMLfooter();
HTMLfin();
?>
```

Uso de PHP en la web**Ejemplo**

```
function HTMLpag_inicio() {
echo <<< HTML
<section> <h1>Personajes históricos</h1>
<p>Pulsa para ver la biografía de alguno.</p>
</section>
HTML;
}
```

```
function HTMLpag_alan() {
echo <<< HTML
<section> <h1>Alan Turing</h1>
<p>Alan Mathison Turing, OBE (Paddington, Londres, 23 de junio de 1912-Wilmslow, Cheshire, 7 de junio de 1954), fue un matemático, lógico, científico de la computación, criptógrafo, filósofo, maratoniano y corredor de ultra distancia británico.</p>
<p>Es considerado uno de los padres de la ciencia de la computación y precursor de la informática moderna. Proporcionó una influyente formalización de los conceptos de algoritmo y computación: la máquina de Turing. Formuló su propia versión de la hoy ampliamente aceptada tesis de Church-Turing (1936).</p>
</section>
HTML;
}
```

```
function HTMLpag_ada() {
echo <<< HTML
<section> <h1>Ada Lovelace</h1>
<p>Augusta Ada King, Condesa de Lovelace , (nacida Augusta Ada Byron en Londres, 10 de diciembre de 1815 – Londres, 27 de noviembre de 1852), conocida habitualmente como Ada Lovelace, fue una matemática y escritora británica conocida principalmente por su trabajo sobre la máquina calculadora mecánica de uso general de Charles Babbage, la denominada máquina analítica. Entre sus notas sobre la máquina se encuentra lo que se reconoce hoy como el primer algoritmo destinado a ser procesado por una máquina, por lo que se la considera como la primera programadora de ordenadores.</p>
</section>
HTML;
}
```

```
function HTMLpag_contacto() {
```

```
echo <<< HTML
<section> <h1>Contacto</h1>
<p>Envía un correo a
PepitoPerez@servidor.de.correo.com</p>
</section>
HTML;
}
```

Uso de PHP en la web**Ejemplo**

```
function HTMLnav($activo) {
echo <<< HTML
<nav> <h1>Índice</h1> <ul>
HTML;
$items = ["Inicio", "Alan Turing", "Ada Lovelace", "Contacto"];
$links = ["pag_inicio.php", "pag_alan.php", "pag_ada.php", "pag_contacto.php"];
foreach ($items as $k => $v)
    echo "<li".($k==$activo?" class='activo'":"").">.<a href='".$links[$k]."'>".$v."</a></li>";
echo <<< HTML
</ul> </nav>
HTML;
}
```

Si \$activo==2
(página de Ada)

```
<nav>
<h1>Índice</h1>
<ul>
<li><a href='pag_inicio.php'>Inicio</a></li>
<li><a href='pag_alan.php'>Alan Turing</a></li>
- 

```

```
.activo { font-weight: bold; }
nav .activo a:link { color: Brown; }
nav .activo a:visited { color: Brown; }
```

Uso de PHP en la web**Ejemplo**

Simplificando más ...

Sustituir todas las páginas (pag_inicio.php, pag_alan.php, pag_ada.php, pag_contacto.php) por una única página (pag2.php):

```
<?php
require "pag_comun.php";
HTMLinicio("Ejemplo de PHP");

if (!isset($_GET["p"]))
    $_GET['p']=0;
else if ($_GET["p"]<0 || $_GET["p"]>3)
    $_GET['p']=0;
HTMLnav_alternativo($_GET["p"]);
switch ($_GET['p']) {
    case 0: HTMLpag_inicio(); break;
    case 1: HTMLpag_alan(); break;
    case 2: HTMLpag_ada(); break;
    case 3: HTMLpag_contacto(); break;
}

HTMLfooter();
HTMLfin();
?>
```

URL para solicitar páginas:

- pag2.php?p=0
- pag2.php?p=1
- pag2.php?p=2
- pag2.php?p=3

Uso de PHP en la web

Ejemplo

Simplificando más ...

Sustituir todas las páginas (pag_inicio.php, pag_alan.php, pag_ada.php, pag_contacto.php) por una única página (pag2.php):

```
function HTMLnav_alternativo($activo) {
echo <<< HTML
<nav>
<h1>Índice</h1>
<ul>
HTML;

$item = ["Inicio", "Alan Turing", "Ada Lovelace", "Contacto"];
foreach ($item as $k => $v)
    echo "<li".($k==$activo?" class='activo'":"").">".
        "<a href='pag2.php?p=".($k)."'">$v.</a></li>";

echo <<< HTML
</ul>
</nav>
HTML;
}
```

Uso de PHP en la web

Acoplamiento PHP/HTML

Inconvenientes

- Todo el código HTML es generado desde PHP: menos eficiente que mostrar páginas HTML
- Hay mucho código HTML mezclado con código PHP

```
<?php
require "templ_comun.php";
include "templ_head.html";
if (!isset($_GET["p"]))
    $_GET['p']=0;
else if ($_GET["p"]<0 || $_GET["p"]>3)
    $_GET['p']=0;
HTMLnav_alternativo($_GET["p"]);
switch ($_GET['p']) {
    case 0: include "templ_inicio.html"; break;
    case 1: include "templ_alan.html"; break;
    case 2: include "templ_ada.html"; break;
    case 3: include "templ_contacto.html"; break;
}
include "templ_foot.html";
?>
```

Cambiar llamadas a funciones
por include de código HTML

Uso de PHP en la web**Acoplamiento PHP/HTML****templ_head.html**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="pag_estilo.css">
<title>Título de la página</title>
</head>
<body>
```

templ_foot.html

```
<footer>
<small>(C) Pepito Pérez</small>
</footer>
</body>
</html>
```

Ahora no es un parámetro: se podría haber dividido en 2 HTML

templ_alan.html

```
<section>
<h1>Alan Turing</h1>
<p>Alan Mathison Turing, OBE (Paddington, Londres, 23 de junio de 1912-Wilmslow, Cheshire, 7 de junio de 1954), fue un matemático, lógico, científico de la computación, criptógrafo, filósofo, maratoniano y corredor de ultra distancia británico.</p> <p>Es considerado uno de los padres de la ciencia de la computación y precursor de la informática moderna. Proporcionó una influyente formalización de los conceptos de algoritmo y computación: la máquina de Turing. Formuló su propia versión de la hoy ampliamente aceptada tesis de Church-Turing (1936).</p>
</section>
```

Uso de PHP en la web**Usando plantillas****Con el sistema de plantillas**

- Se separa mejor el código PHP del HTML
- Permite diseñar mejor la página (aspecto)
- Se mantiene la mezcla PHP/HTML cuando lo requiere la lógica de la aplicación
- ... ¿cómo resolver la parametrización?

templ_head.plantilla

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="pag_estilo.css">
<title>##TITULO##</title>
</head>
<body>
```

Marcado especial

Uso de PHP en la web
Usando plantillas

Con el sistema de plantillas

- Eficiencia:
 - Búsqueda y sustitución de cadenas
 - Mantener plantillas “cacheadas”
- Separación efectiva de HTML y PHP

```
<?php
function expandir($fich, $tags) {
    if ($f=fopen($fich, 'r')) {
        $plantilla = fread($f, filesize($fich));
        fclose($f);
        foreach ($tags as $k => $v)
            $plantilla = str_replace("##{$k}##", $v, $plantilla);
    } else
        $plantilla = '';
    return $plantilla;
}

$tags = ['TITULO' => 'Titulo de la página'];
echo expandir('templ_head.plantilla', $tags);
?>
```

Hay muchos sistemas para trabajar con plantillas: Twig, Smarty, Mustache, ...

Tecnologías Web
Grado en Ingeniería Informática

Programación en el lado del servidor – PHP+web

UNIVERSIDAD DE GRANADA

DECSAI

»

1. El lenguaje PHP
2. PHP y aplicaciones web
 1. Uso de PHP en la web
 2. Procesamiento de formularios
 3. Saneamiento de cadenas
 1. URL encoding
 2. Saneamiento de cadenas
 3. Query strings
 4. Recordando el estado de las aplicaciones
 1. Cookies
 2. Sesiones
 5. Envío de encabezados
 3. PHP y conexión con BBDD

Procesamiento de formularios
Variables del formulario

Formularios

Los datos del formulario son accedidos a través de las variables globales:

- `$_GET`
- `$_POST`

Dependiendo del método de envío

```
conversor.html
<body>
<h1>Conversor de temperaturas</h1>
<form action="conversor.php" method="get">
    <label>Temperatura en Celsius:<br/>
        <input type="text" name="celsius"/>
    </label>
    <input type="submit" value="Convertir"/>
</form>
</body>
```

Conversor de temperaturas

Temperatura en Celsius: 37.2

Convertir

localhost/tw/php/conversor2.php?celsius=37.2

Grados Celsius: 37.2

Grados Fahrenheit: 98.96

Calcule otra conversión

conversor.php
15

Procesamiento de formularios
Variables del formulario

El mismo ejemplo usando POST en lugar de GET

```
conversor.html
<body>
<h1>Conversor de temperaturas</h1>
<form action="conversor.php" method="post">
    <label>Temperatura en Celsius:<br/>
        <input type="text" name="celsius"/>
    </label>
    <input type="submit" value="Convertir"/>
</form>
</body>
```

```
POST /tw/php/conversor2_post.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
...
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 12
celsius=37.2
```

Conversor de temperaturas

Temperatura en Celsius: 37.2

Convertir

localhost/tw/php/conversor2_post.php

Grados Celsius: 37.2

Grados Fahrenheit: 98.96

Calcule otra conversión

conversor.php
16

Procesamiento de formularios
Diseño del formulario

Formularios

La misma página suele mostrar el formulario y procesarlo

```

echo "<h1>Conversor de temperaturas</h1>";
if (isset($_GET["celsius"])) {
    /* Si se han recibido datos del formulario */
    $cel = $_GET["celsius"];
    $fah = $_GET["celsius"]*9/5+32;
    echo "<p>Grados Celsius: $cel</p>";
    echo "<p>Grados Fahrenheit: $fah</p>";
    echo "<p><a href='".$SERVER["PHP_SELF"]."'>Calcule otra conversión</a></p>";
} else {
    /* Si no se han recibido datos del formulario */
    echo "<form action='".$SERVER["PHP_SELF"]."' method='get'>
        <label>Temperatura en Celsius:
            <input type='text' name='celsius'/>
        </label>
        <input type='submit' value='Convertir' />
    </form>";
}

```

\$_SERVER["PHP_SELF"] : nombre del script

Procesamiento de formularios
Controles del formulario: select

Formularios: controles de tipo select

```

if (isset($_GET["celsius"])) { /* Si se han recibido datos del formulario */
    $cel = $_GET["celsius"];
    echo "<p>Grados Celsius: $cel</p>";
    switch ($_GET["destino"]) {
        case "Fahrenheit" : echo "<p>Grados Fahrenheit: ".($cel*9/5+32)."</p>"; break;
        case "Kelvin" : echo "<p>Grados Kelvin: ".($cel+273.15)."</p>"; break;
        case "Rankine" : echo "<p>Grados Rankine: ".($cel*9/5+491.67)."</p>"; break;
    }
    echo "<p><a href='".$SERVER["PHP_SELF"]."'>Calcule otra conversión</a></p>";
} else { /* Si no se han recibido datos del formulario */
    echo "<form action='".$SERVER["PHP_SELF"]."' method='get'>
        <label>Temperatura en Celsius:
            <input type='text' name='celsius'/>
        </label><br>
        <label>A qué unidad desea convertir:
            <select name='destino'>
                <option>Fahrenheit</option>
                <option>Kelvin</option>
                <option>Rankine</option>
            </select>
        </label>
        <input type='submit' value='Convertir' />
    </form>";
}

```

Conversor de temperaturas

Temperatura en Celsius:

A qué unidad desea convertir:

Fahrenheit
 Kelvin
 Rankine

Procesamiento de formularios
Controles del formulario: radio

Formularios: controles de tipo radio

```

if (isset($_GET["celsius"]) && isset($_GET["destino"])){
    /* Si se han recibido datos del formulario */
    /* IDEM */
} else { /* Si no se han recibido datos del formulario */
    echo "<form action='".$SERVER['PHP_SELF']."' method='get'>
        <label>Temperatura en Celsius:
            <input type='text' name='celsius'/>
        </label><br>
        <label>A qué unidad desea convertir: <br>
            <input type='radio' name='destino' value='Fahrenheit'> Fahrenheit<br>
            <input type='radio' name='destino' value='Kelvin'> Kelvin<br>
            <input type='radio' name='destino' value='Rankine'> Rankine<br>
        </label>
        <input type='submit' value='Convertir'/'>
    </form>";
}

```

Conversor de temperaturas

Temperatura en Celsius: 23
 A qué unidad desea convertir:
 Fahrenheit
 Kelvin
 Rankine
 Convertir

Conversor de temperaturas

Grados Celsius: 23
 Grados Kelvin: 296.15
[Calcule otra conversión](#)

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 19

Procesamiento de formularios
Controles del formulario: checkbox

Formularios: controles de tipo checkbox

```

if (isset($_GET["celsius"]) && isset($_GET["destino"])){
    /* Si se han recibido datos del formulario */
    /* IDEM */
} else { /* Si no se han recibido datos del formulario */
    echo "<form action='".$SERVER['PHP_SELF']."' method='get'>
        <label>Temperatura en Celsius:
            <input type='text' name='celsius'/>
        </label><br>
        <label>A qué unidad desea convertir: <br>
            <input type='checkbox' name='destino' value='Fahrenheit'> Fahrenheit<br>
            <input type='checkbox' name='destino' value='Kelvin'> Kelvin<br>
            <input type='checkbox' name='destino' value='Rankine'> Rankine<br>
        </label>
        <input type='submit' value='Convertir'/'>
    </form>";
}

```

Conversor de temperaturas

Temperatura en Celsius:
 A qué unidad desea convertir:
 Fahrenheit
 Kelvin
 Rankine
 Convertir

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 20

Procesamiento de formularios
Controles del formulario: checkbox

Formularios: controles de tipo checkbox

```

if (isset($_GET["celsius"]) && isset($_GET["destino"])){
    /* Si se han recibido datos del formulario */
    $cel = $_GET["celsius"];
    echo "<p>Grados Celsius: $cel</p>";
    if (in_array("Fahrenheit",$_GET["destino"]))
        echo "<p>Grados Fahrenheit: ." .($cel*9/5+32). "</p>";
    if (in_array("Kelvin",$_GET["destino"]))
        echo "<p>Grados Kelvin: ." .($cel+273.15). "</p>";
    if (in_array("Rankine",$_GET["destino"]))
        echo "<p>Grados Rankine: ." .($cel*9/5+491.67). "</p>";
    echo "<p><a href='".$._SERVER["PHP_SELF"]."">'Calcule otra conversión</a></p>";
} else { /* Si no se han recibido datos del formulario */
    echo "<form action='".$._SERVER["PHP_SELF"]."' method='get'>
        <label>Temperatura en Celsius:
            <input type='text' name='celsius'/>
        </label><br>
        <label>A qué unidad desea convertir: <br>
            <input type='checkbox' name='destino[]' value='Fahrenheit'> Fahrenheit<br>
            <input type='checkbox' name='destino[]' value='Kelvin'> Kelvin<br>
            <input type='checkbox' name='destino[]' value='Rankine'> Rankine<br>
        </label>
        <input type='submit' value='Convertir' />
    </form>";
}

```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 21

Procesamiento de formularios
Validación de datos

Formularios

Validación de datos con PHP

Conversor de temperaturas

Temperatura en Celsius:

A qué unidad desea convertir:

Fahrenheit
 Kelvin
 Rankine

No es un número

No hemos marcado ninguna escala

SI (formulario enviado)
 Validar datos
 FIN-SI

SI (formulario enviado y datos correctos)
 Procesar formulario
 SI-NO
 Mostrar formulario
 (incluyendo errores y valores previos)
 FIN-SI

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 22



Procesamiento de formularios

Validación de datos

```

if (isset($_GET['celsius'])) { /* El formulario ha sido enviado */
    /* Comprobar valor de Celsius */
    if (empty($_GET['celsius']))
        $hayerror['celsius'] = '<p style="color:red;">No ha indicado ningún valor</p>';
    else if (!is_numeric($_GET['celsius']))
        $hayerror['celsius'] = '<p style="color:red;">El valor debe ser un número</p>';
    else if ($_GET['celsius'] <-100)
        $hayerror['celsius'] = '<p style="color:red;">El número ha de ser mayor que -100</p>';
    else
        $celsius = $_GET['celsius'];
    /* Comprobar si hay alguna escala */
    if (!isset($_GET['destino']))
        $hayerror['destino'] = '<p style="color:red;">Ha de seleccionar al menos una escala</p>';
    else {
        if (in_array('Fahrenheit',$_GET['destino']))
            $destino['fah'] = 1;
        if (in_array('Kelvin',$_GET['destino']))
            $destino['kel'] = 1;
        if (in_array('Rankine',$_GET['destino']))
            $destino['ran'] = 1;
    }
}

```

SI (formulario enviado)
 Validar datos
 FIN-SI

Procesamiento de formularios

Validación de datos

```

if (isset($celsius) && isset($destino)) {
    /* Si no hay errores */
    echo "<p>Grados Celsius: $celsius</p>";
    if (array_key_exists('fah',$destino))
        echo '<p>Grados Fahrenheit: ' . ($celsius*9/5+32). '</p>';
    if (array_key_exists('kel',$destino))
        echo '<p>Grados Kelvin: ' . ($celsius+273.15). '</p>';
    if (array_key_exists('ran',$destino))
        echo '<p>Grados Rankine: ' . ($celsius*9/5+491.67). '</p>';
    echo "<p><a href='".$SERVER["PHP_SELF"]."'">Calcule otra conversión</a></p>";
} else {
}

```

SI (formulario enviado y datos correctos)
 Procesar formulario
 SI-NO

Procesamiento de formularios
Validación de datos

```

} else { /* Hay errores o no se ha enviado formulario */
    <form action=<?php echo $_SERVER['PHP_SELF']?> method='get'>
        <label>Temperatura en Celsius:
            <input type='text' name='celsius'
                <?php if (isset($celsius)) echo " value='".$celsius."';?>
                <?php if (isset($hayerror) && array_key_exists('celsius', $hayerror))
                    echo $hayerror['celsius']; ?></label><br>
        <label>A qué unidad desea convertir: <br>
            <input type='checkbox' name='destino[]' value='Fahrenheit'
                <?php if (isset($destino) && array_key_exists('fah', $destino))
                    echo ' checked';?> > Fahrenheit<br>
            <input type='checkbox' name='destino[]' value='Kelvin'
                <?php if (isset($destino) && array_key_exists('kel', $destino))
                    echo ' checked';?> > Kelvin<br>
            <input type='checkbox' name='destino[]' value='Rankine'
                <?php if (isset($destino) && array_key_exists('ran', $destino))
                    echo ' checked';?> > Rankine<br>
                <?php if (isset($hayerror) && array_key_exists('destino', $hayerror))
                    echo $hayerror['destino']; ?>
        </label>
        <input type='submit' value='Convertir'/>
    </form>
<?php }

```

Sticky form

SI-NO
Mostrar formulario
(incluyendo errores y
valores previos)
FIN-SI

Procesamiento de formularios
Subida de ficheros

Formularios: subir un fichero
Método: POST
enctype="multipart/form-data"

```

<form action=<?php echo $_SERVER['PHP_SELF']?>
    method='post'
    enctype='multipart/form-data'>
    <label for='fichero'>Fichero: </label>
    <input type='file' name='fichero'><br>
    <input type='submit' value='Subir'/'>
</form>

```

Subir fichero

Fichero: No se ha seleccionado ningún archivo.

Al enviar el formulario (submit):

- Se sube el fichero al servidor
- Se almacena en carpeta temporal (upload_tmp_dir de php.ini)
- Se almacena alguna información para manipularlo desde PHP

Al terminar la ejecución del script:

- Se borra el fichero temporal

Procesamiento de formularios
Subida de ficheros

Acceso al fichero desde PHP

```
<form action="<?php echo $_SERVER['PHP_SELF']?>" method='post' enctype='multipart/form-data'>
<label for='fichero'>Fichero: </label>
<input type='file' name='fichero'><br>
<input type='submit' value='Subir' name='subido' />
</form>
```

`$_FILES`: array con datos de los ficheros subidos
 Datos de cada fichero subido:

- name: nombre del fichero enviado
- type: Tipo de contenido
- size: Tamaño en bytes
- tmp_name: nombre del fichero en donde se ha almacenado temporalmente
- error: Código de error

```
echo "Has subido un fichero llamado ", $_FILE['fichero']['name'];
echo "Que se ha almacenado temporalmente en ", $_FILE['fichero']['tmp_name'];
echo "Y que ocupa estos bytes: ", $_FILE['fichero']['size'];
```

Procesamiento de formularios
Subida de ficheros

Comprobación de subida del fichero

SI (formulario enviado)
 Validar datos
 FIN-SI

```
if (sizeof($_FILES)>0) {
  if (array_key_exists("fichero",$_FILES)) {
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
      if (isset($_POST["subido"])) {
```

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  /**** Validación del formulario */

  /* Comprobar que se ha subido algún fichero */
  if ((sizeof($_FILES)==0) || !array_key_exists("fichero",$_FILES))
    $error = "No se ha podido subir el fichero";
  else if (!is_uploaded_file($_FILES['fichero']['tmp_name']))
    $error = "Fichero no subido. Código de error: ". $_FILES['fichero']['error'];
}
```

Si hay algún error se crea la variable \$error

Procesamiento de formularios
Subida de ficheros

Códigos de error de subida	
UPLOAD_ERR_OK	No hay error
UPLOAD_ERR_INI_SIZE	El tamaño supera el valor de upload_max_filesize (php.ini)
UPLOAD_ERR_FORM_SIZE	El tamaño supera el valor de max_file_size del formulario HTML <input type="hidden" name="MAX_FILE_SIZE" value="1024">
UPLOAD_ERR_PARTIAL	El fichero no se ha subido por completo
UPLOAD_ERR_NO_FILE	No se ha subido ningún fichero
UPLOAD_ERR_NO_TMP_DIR	Falta la carpeta temporal se subidas (upload_tmp_dir de php.ini)
UPLOAD_ERR_CANT_WRITE	No se puede escribir en el disco
UPLOAD_ERR_EXTENSION	Una extensión de PHP provocó el error

POST_MAX_SIZE (en php.ini): si es superado no se sube el fichero ni existe la entrada correspondiente en \$_FILES

Procesamiento de formularios
Subida de ficheros

Procesar el formulario

```

if (($_SERVER['REQUEST_METHOD'] == 'POST') && !$error) {
    /**
     * Procesar formulario
     */
    echo "<p>Nombre : {$_FILES['fichero']['name']}</p>";
    echo "<p>Tipo : {$_FILES['fichero']['type']}</p>";
    echo "<p>Nombre temporal: {$_FILES['fichero']['tmp_name']}</p>";
    echo "<p>Tamaño : {$_FILES['fichero']['size']}</p>";
    echo "<p>Cod. error : {$_FILES['fichero']['error']}</p>";

    move_uploaded_file($_FILES['fichero']['tmp_name'],
                      './subidos/' . $_FILES['fichero']['name']);

    // Si es una imagen: mostrar
    if (in_array($_FILES['fichero']['type'],
                ['image/jpeg', 'image/gif', 'image/png']))
        echo "<img src='./subidos/' . $_FILES['fichero']['name'].'" width='256' />";
}

```



Recomendaciones de seguridad

- Almacenar ficheros subidos en:
 - Directorio separado de aplicación web y no accesible desde web (si es posible)
 - Se podrían subir ficheros que afecten a la aplicación (.htaccess, *.html, *.php, ...)
- No confiar en el nombre del fichero enviado para almacenar en servidor
 - El HTTP header puede enviarse con un filename malicioso (/etc/passwd, ../../aplicacion/hack.php, ...).
 - El sistema de ficheros del servidor puede no ser el mismo que el sistema de ficheros del cliente (case-sensitive, caracteres especiales, etc).
 - Se podrían sobreescribir ficheros (`move_uploaded_file` sobreescribe)
- Configurar `php.ini` para limitar tamaños de ficheros
 - `POST_MAX_SIZE` y `MAX_FILE_SIZE`
- No confiar en el mimetype
 - Supongamos que el atacante ha podido subir un fichero `.htaccess` con el contenido: `AddType application/x-httpd-php .jpg`
 - Cuando Apache carga una imagen `.jpg` entiende que es un script PHP y lo ejecuta. Si el fichero `.jpg` tiene código malicioso ...



Mostrar el formulario (y errores si hay)

```

} else {
    /*** Hay errores o no se ha enviado formulario ***/
    if ($error)
        echo "<p>ERROR: ".$error."</p>";
    ?>
    <form action=<?php echo $_SERVER['PHP_SELF']?>" method='post'
          enctype='multipart/form-data'>
        <label for='fichero'>Fichero: </label><input type='file' name='fichero'><br>
        <input type='submit' name="subido" value='Subir'/>
    </form>
<?php } ?>

```



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor – PHP+web

- 1. El lenguaje PHP
- 2. PHP y aplicaciones web
 - 1. Uso de PHP en la web
 - 2. Procesamiento de formularios
 - 3. Saneamiento de cadenas
- 3. URL encoding
- 2. Saneamiento de cadenas
- 3. Query strings
- 4. Recordando el estado de las aplicaciones
 - 1. Cookies
 - 2. Sesiones
- 5. Envío de encabezados
- 3. PHP y conexión con BBDD

»»»

Saneamiento de cadenas

Query strings



```
$restaurante = "Food & fun";
$plato = "Tortilla";

$url = 'verpostyget.php';
$url .= '?restaurante='.$restaurante;
$url .= '&plato='.$plato;
echo '<a href="'.$url.'">Visitar</a>';

$url vale "verpostyget.php?restaurante=Food & fun&plato=Tortilla"
```

Codificación de la cadena de query de la URL: urlencode

Según el RFC 2396 (definición de URI / URL), algunos caracteres no pueden estar incluidos en una URL:

blanco < > # % “ { } | \ ^ [] `

Otros caracteres tienen significado especial:

? & = + ...

`$url = 'verpostyget.php';
$url .= '?restaurante=' . urlencode($restaurante);
$url .= '&plato=' . urlencode($plato);
echo 'Visitar';`

`$url vale "verpostyget.php?restaurante=Food%26fun&plato=Tortilla"`

restaurante = Food & fun
plato = Tortilla

Saneamiento de cadenas
Query strings

Codificación de la cadena de query de la URL: `rawurlencode`

`rawurlencode` es similar, sigue el RFC 3986, codifica los espacios como %20 en lugar de como '+'

Cadena original:
Mc "Donalds" + Burguer King & CIA = no comida

urlencode:
`Mc%22Donalds%22+%2B+Burguer+King+%26+CIA+%3D+no+comida`

rawurlencode:
`Mc%20%22Donalds%22%20%2B%20Burguer%20King%20%26%20CIA%20%3D%20no%20comida`

Decodificación de la cadena de query

`urldecode`
`rawurldecode`

Por ejemplo para almacenarla en una BBDD sin codificar

Saneamiento de cadenas
Cadenas y HTML entities

```
if (isset($_GET['nombre'])) {
    echo 'Hola ' . $_GET['nombre'];
    echo 'Bienvenido';
} else { ?>
    <form action=<?php echo $_SERVER['PHP_SELF']?>> method='get'>
        <input type='text' name='nombre'>
        <input type='submit' value='Enviar' name='enviado' />
    </form> <?php
}
```

Javier Hola Javier Bienvenido

<h2>Javier</h2> Hola Javier Bienvenido

Hola
Javier
Bienvenido

Hola
Javier
Bienvenido

Saneamiento de cadenas
Cadenas y HTML entities

Codificar caracteres en “HTML entities”

htmlentities Codifica los caracteres susceptibles de ello en HTML entities
htmlspecialchars Solo convierte &, “, ‘, <, >

```
echo 'Hola '.htmlentities($_GET['nombre']).'<br>';
echo 'Hola '.htmlspecialchars($_GET['nombre']).'<br>';
echo 'Bienvenido';
```

<h2>Martín Hola <h2>Martín
Hola <h2>Martín Bienvenido

<body>
Hola <h2>Martín

Hola <h2>Martin

Bienvenido
</body>

Decodificación

html_entity_decode
htmlspecialchars_decode

<https://dev.w3.org/html5/html-author/charref>

Saneamiento de cadenas
Eliminar tags HTML

Javier Hola Javier Bienvenido

Quitar tags HTML de la cadena

strip_tags Elimina cualquier tag de HTML o PHP

```
echo 'Hola '.strip_tags($_GET['nombre']).'<br>';

echo 'Hola '.strip_tags($_GET['nombre'], '<b>').'<br>';

$cad = '<p>Texto de prueba</p><!-- Comentario -->
<a href="http://www.ugr.es">Universidad de Granada</a>';
echo strip_tags($cad), PHP_EOL; // Texto de prueba Universidad de Granada
```



Saneamiento de cadenas

Escapado de CADENAS

Escapado de caracteres

addslashes	Escapa los caracteres: ‘, “, \
stripslashes	Quita el escapado

```
// Usa barras para escapar una cadena
$cad = "O'Reilly";
echo addslashes($cad), PHP_EOL; // O\'Reilly

// Se suele usar para crea consultas SQL
$sql = "SELECT * FROM libros WHERE editorial = '" . $cad . "'";
echo $sql, PHP_EOL; // SELECT * FROM libros WHERE editorial = 'O'Reilly'

$sql = "SELECT * FROM libros WHERE editorial = '" . addslashes($cad) . "'";
echo $sql, PHP_EOL; // SELECT * FROM libros WHERE editorial = 'O\'Reilly'
```



Saneamiento de cadenas

Parseo de cadenas con formato GET

```
// Parsear variables (formato GET) desde un string
$cadena = "nombre=Javier&ape1=Martinez&ape2=Baena";
parse_str($cadena,$vars);
echo $vars['nombre'], PHP_EOL; // Javier
echo $vars['ape1'], PHP_EOL; // Martínez

// Si no se usa el segundo argumento ... (desaconsejado)
$nombre = "Pepe";
parse_str($cadena);
echo $nombre, PHP_EOL; // Javier
echo $ape1, PHP_EOL; // Martínez

// Cuidado con los nombres de las variables
$cadena = "var1-2=letra";
parse_str($cadena,$vars);
print_r($vars); // $vars=[ "var1-2"=>"letra" ]
echo $vars['var1-2'], PHP_EOL; // letra
// En este caso no se puede usar la técnica de omitir segundo parámetro

// Cuidado con los nombres de las variables
// espacios y puntos se cambian por -
$cadena = "var1_2=letra&var3.4=otra";
parse_str($cadena,$vars); // $vars=[ "var1_2"=>"letra", "var3_4"=>"otra" ]
print_r($vars);
```

Saneamiento de cadenas
Parseo de cadenas con formato GET

```

$variables = ['nombre' => 'Javier', 'apellidos' => 'Martínez Baena'];

// Construir un Query String desde un array de valores
$qqs = http_build_query($variables);
echo $qs;           // nombre=Javier&apellidos=Mart%C3%ADnez+Baena
echo urldecode($qs); // nombre=Javier&apellidos=Martínez Baena

$variables[] = 'Grado'; // Añadir valor enumerado (no asociativo)
$qqs = http_build_query($variables, 'VARI');
echo $qs;           // nombre=Javier&apellidos=Mart%C3%ADnez+Baena&VARI=Grado
echo urldecode($qs); // nombre=Javier&apellidos=Martínez Baena&VARI=Grado

$variables = ['nombre' => 'Javier', 'apellidos' => 'Martínez Baena'];
$qqs = http_build_query($variables, '', '#');
echo $qs;           // nombre=Javier#apellidos=Mart%C3%ADnez+Baena
echo urldecode($qs); // nombre=Javier#apellidos=Martínez Baena

$qqs = http_build_query($variables, '', '?', PHP_QUERY_RFC3986);
echo $qs;           // nombre=Javier?apellidos=Mart%C3%ADnez%20Baena
echo rawurldecode($qs); // nombre=Javier?apellidos=Martínez Baena
// Por defecto usa codificación PHP_QUERY_RFC1738

```

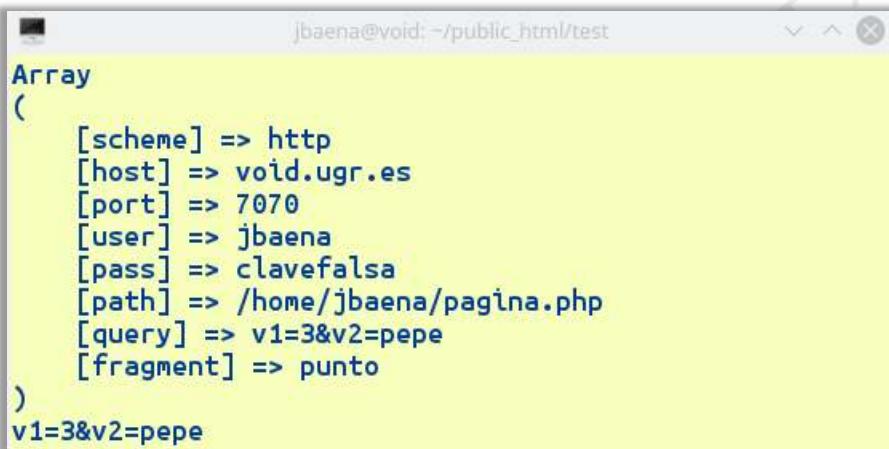
Saneamiento de cadenas
Parseo de cadenas con formato GET

```

// Parsear una URL
$cadena = "http://jbaena:clavefalsa@void.ugr.es:7070/home/jbaena/pagina.php?v1=3&v2=pepe#punto";
$result = parse_url($cadena); print_r($result);

// Obtener solo una parte
$result = parse_url($cadena, PHP_URL_QUERY);
echo $result, PHP_EOL;      // v1=3&v2=pepe

```



PHP_URL_SCHEME
PHP_URL_HOST
PHP_URL_PORT
PHP_URL_USER
PHP_URL_PASS
PHP_URL_PATH
PHP_URL_QUERY
PHP_URL_FRAGMENT

Saneamiento de cadenas
Nombres de ficheros

Nombres de ficheros provenientes de fuentes externas (formularios)

- Riesgo potencial

```

if (isset($_GET['nombre'])) {
    echo 'Contenido del fichero '.$_GET['nombre'].' :<br>';
    readfile($_GET['nombre']);
} else { ?>
    <form action=<?php echo $_SERVER['PHP_SELF']?> method='get'>
        <input type='text' name='nombre'>
        <input type='submit' value='Enviar' name='enviado' />
    </form> <?php
}

```

Contenido del fichero datos.txt:
 Línea 1 Línea 2 Línea 3

Contenido del fichero /etc/passwd:
 root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/u
 sync:x:4:65534:sync:/bin:/sync games:x:5:60:games:/usr/gam
 /sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9
 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:ww
 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x
 /lib/cnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/none

Saneamiento de cadenas
Nombres de ficheros

Nombres de ficheros provenientes de fuentes externas (formularios)

- Riesgo potencial

```

if (isset($_GET['nombre'])) {
    echo 'Contenido del fichero '.$_GET['nombre'].' :<br>';
    $f = '/home/data/www/'.$_GET['nombre'];
    readfile($f);
} ...

```

Error

Contenido del fichero /etc/passwd:
 root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/u
 sync:x:4:65534:sync:/bin:/sync games:x:5:60:games:/usr/gam
 /sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9
 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:ww
 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x
 /lib/cnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/none

```

preg_match('^{([/]*)$}', $_GET['nombre'], $base);
$f = '/home/data/www/'.$base[1]; // $base[1] es el nombre sin la ruta
readfile($f);

```



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor – PHP+web

- 1. El lenguaje PHP**
- 2. PHP y aplicaciones web**
 - 1. Uso de PHP en la web**
 - 2. Procesamiento de formularios**
 - 3. Saneamiento de cadenas**
 - 1. URL encoding**
 - 2. Saneamiento de cadenas**
 - 3. Query strings**
 - 4. Recordando el estado de las aplicaciones**
 - 1. Cookies**
 - 2. Sesiones**
 - 5. Envío de encabezados**
 - 3. PHP y conexión con BBDD**

»

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena

Recordar el estado de la aplicación

Introducción

Programación en la web

HTTP es un protocolo “stateless”:

- Cada petición es independiente del resto sin relación con peticiones previas.
- El servidor no recuerda peticiones previas

```
<?php
$x = $x + 3;
echo "X vale ", $x, PHP_EOL;
?>
```

Cada vez que se pide la página:

1. PHP da un mensaje de error al usar \$x sin estar definida
2. Imprime el mensaje X vale 3

¿Cómo recordar el estado entre solicitudes (ejecuciones)?

- Pasando variables por \$_GET o \$_POST
- Cookies
- Variables de sesión
- Almacenando datos en BBDD en el servidor

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 46

Recordar el estado de la aplicación
Con variables GET/POST

Recordar estado usando variables POST/GET
El paso de una página a otra se hace con formularios y controles “submit”

```

echo "Bienvenido<br>";

if (isset($_POST["autenticado"])) {
    // Ya pasó por formulario de login
    if ($_POST["autenticado"]=='si') {
        // Y se autenticó bien
        echo "Usted está autenticado<br>";
    } else if (($_POST['user']=='yo') &&
               ($_POST['passwd']=='1234')) {
        // No se había autenticado aún
        echo "Usted está autenticado<br>";
        $_POST["autenticado"] = "si";
    }
} else {
    // No pasó aún por formulario de login
    $_POST["autenticado"] = "no";
}

```

Recordar el estado de la aplicación
Con variables GET/POST

Recordar estado usando variables POST/GET
El paso de una página a otra se hace con formularios y controles “submit”

```

if ($_POST["autenticado"]=="no") {
    // Mostrar formulario de login
    echo "<form action={$ _SERVER['PHP_SELF']} method='POST'>
        Usuario: <input type='text' name='user'><br>
        Clave: <input type='password' name='passwd'><br>
        <input type='hidden' name='autenticado' value='no'>
        <input type='submit' value='Login'>
    </form>";
} else {
    // Mostrar resto de página e incluir dato de validación
    echo "<form action={$ _SERVER['PHP_SELF']} method='POST'>
        <input type='hidden' name='autenticado' value='{$ _POST['autenticado']}'>
        <input type='submit' value='Seguir ...'>
    </form> ";
    // Mostrar formulario para logout (no incluye dato de validación)
    echo "<form action={$ _SERVER['PHP_SELF']} method='POST'>
        <input type='submit' value='Logout'>
    </form> ";
}

```

Cookies
¿Qué son?

Cookies

Una cookie son unos pocos bits de información que el servidor envía al cliente y este los almacena en forma de cadena de caracteres.

```

    graph LR
        subgraph Client [Client]
            direction TB
            CHD[Client's Hard Drive] --- EH[example.com test= 42]
            subgraph RSP1 [request set.php]
                direction TB
                PR1["page requested  
results $cookie  
write cookie"]
                RSP1 -- "request:" --> SR1["<?php  
setcookie('test', '42');  
?>"]
            end
            subgraph RSP2 [request read.php]
                direction TB
                PR2["send cookie  
request page and  
cookie results"]
                RSP2 -- "Session:  
request:" --> SR2["<?php  
echo $_COOKIE['test'];  
?>"]
            end
        end
        subgraph Server [Server]
            SR1
            SR2
        end
        PR1 -- "Cookie results" --> PR2
        PR2 -- "Results" --> SR2
    
```

1. El cliente solicita una página
 2. El servidor envía una cookie
 3. La cookie se almacena en el cliente
 4. En cada nueva petición, el cliente le envía esa información en el header.

Michele Davis, John Phillips "Learning PHP and MySQL (2ed)". O'Reilly, 2007

Cookies
Cómo se usan

Cookies

- En cada nueva petición del cliente al servidor le envía esa información en el header automáticamente.
- En el servidor, las cookies recibidas están almacenadas en una variable global (superglobal) llamada `$_COOKIE` (array asociativo).
- `$HTTP_COOKIE_VARS` (obsoleto). No es superglobal.

```

if (isset($_COOKIE["galleta"]))
    echo "Tenemos una cookie almacenada: ", $_COOKIE["galleta"], PHP_EOL;
else {
    setcookie("galleta", "Chocolate", time() + 10);
    echo "... acabo de almacenar una cookie";
}
    
```

The screenshot shows two browser tabs. The left tab has the URL `https://void....h_cookie2.php` and displays the text "... acabo de almacenar una cookie". The right tab has the same URL and displays the text "Tenemos una cookie almacenada: Chocolate". This illustrates how a cookie is stored and then retrieved in a subsequent request.

Cookies
Cómo se usan

Las cookies siguientes están guardadas en su equipo:

Sitio	Nombre de la cookie
localhost	galleta

Nombre: galleta
Contenido: Chocolate
Servidor: localhost
Ruta: /tw/php/
Enviar para: Cualquier tipo de conexión
Expira: Al finalizar la sesión

Eliminar seleccionada Eliminar todas Cerrar

moz_cookies (cookies)

Structure Data Constraints Indexes Triggers DDL

Grid view Form view

id	baseDomain	Attri	name	value	host	path	expiry	lastAccessed	creationTime	isSecure	isHttpOnly
1	localhost		galleta	Chocolate	localhost	/tw/php/	1490254742	1490254732658368	1490254732658368	0	0

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 51

Cookies
Cómo se establecen

Cookies

```
setcookie(name,value,expire,path,domain,secure,httponly);
```

- name: identificador único de la cookie
- value: Valor almacenado (máximo 3-4KB)
- expire: (Opc) Fecha en la que caduca (se borra automáticamente).
Por defecto: 0 (caduca al cerrar el navegador)
- path: (Opc) La cookie está disponible en esa subcarpeta y en las subcarpetas
Por defecto: carpeta actual
- domain (Opc): La cookie está disponible para ese dominio y subdominios
Por defecto: dominio actual
- secure: La cookie solo está disponible en conexiones seguras (HTTPS)
Por defecto: false
- httponly: La cookie solo está disponible a través del protocolo HTTP (no es accesible por JavaScript).
Por defecto: false

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 52

Cookies
Cómo se borran

Eliminando cookies

La forma de eliminar una cookie es hacer que caduque.

Ejecución de setcookie() con idénticos parámetros que en la creación salvo por el tiempo de caducidad, que debe establecerse en el pasado

```
if (isset($_COOKIE["galleta"])) {
    echo "Tenemos una cookie almacenada: ", $_COOKIE["galleta"], PHP_EOL;
    setcookie("galleta", "Chocolate", time()-2592000);
    echo "... y la hemos borrado", PHP_EOL;
} else {
    setcookie("galleta", "Chocolate", time()+60);
    echo "... acabo de almacenar una cookie";
}
```

2592000 = 1 mes ... por si el reloj del cliente no está bien

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 53

Cookies
Inconvenientes y seguridad

Inconvenientes

- No todos los clientes aceptan cookies
- El usuario puede deshabilitar las cookies
- Tamaño limitado a 4KB (nombre+valor+fecha caducidad+...)
- Máximo de cookies por dominio (20), configurable
- Máximo de cookies por cliente (300), configurable
- Si se superan esos máximos el cliente podría hacer caducar a otras cookies más antiguas que aún no debían caducar

Seguridad

- Las cookies se almacenan en el cliente: pueden modificarse
- Usa siempre (1) conexiones HTTPS verificadas para que las cookies viajen cifradas, (2) path y domain para restringir el acceso a la cookie, (3) httponly para evitar ataques XSS

Ataque “Session-hijacking”

1. U: acceso legal (transf. cookie)
2. A: intercepta cookie (sniffer)
3. A: inyecta cookie en su cliente
4. A: suplanta a U

The diagram shows a sequence of interactions between three entities: an Innocent User, a Website/Server, and a Black hat Hacker.
 1. The Innocent User initiates an "Authentic Request" to the Website/Server.
 2. The Black hat Hacker, positioned between the user and the server, intercepts the session ID being transmitted ("Hijacking Session ID").
 3. The Black hat Hacker uses this hijacked session ID to send an "Impersonate Request" back to the Website/Server, effectively impersonating the Innocent User.

Image created by Sarvesh Rughani

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 54

Sesiones
Qué son

Sesiones PHP

Permiten recordar el estado de la aplicación de forma muy sencilla: El servidor genera una cookie que identifica la sesión y almacena las variables asociadas a ella.

El diagrama ilustra el proceso de gestión de sesiones entre un cliente y un servidor:

- Client:** Representado por un cuadro que incluye un cilindro central y un cuadro superior.
- Server:** Representado por un cuadro que incluye un cilindro central y un cuadro superior.

Flujo de datos:

- El cliente hace una solicitud (request) a `example.com/set.php`. El servidor responde con el código PHP `<?php session_start(); $SESSION['test']=42; ?>`.
- El cliente responde con una cookie (`Cookie results`) que contiene la sesión ID: `example.com session_id: 19283232`.
- El cliente hace una solicitud (request) a `example.com/read.php`. El servidor responde con el código PHP `<?php session_start(); echo $_SESSION['test']; ?>`.
- El cliente responde con una cookie (`Cookie ID`) que contiene la sesión ID: `Session ID`.

Almacenamiento:

- El servidor almacena la sesión ID y las variables en un espacio de memoria centralizado.
- El cliente almacena la sesión ID en su cookie.

1. El cliente solicita una página
2. El servidor inicia la sesión y envía una cookie de ID de sesión
El servidor almacena las variables asociadas a la sesión
3. La cookie de ID se almacena en el cliente
4. En cada nueva petición, el cliente le la cookie de ID en el header.

Michele Davis, John Phillips "Learning PHP and MySQL (2ed)". O'Reilly, 2007

Sesiones
Cómo se usan

Sesiones PHP

Permiten recordar el estado de la aplicación de forma muy sencilla: El servidor genera una cookie que identifica la sesión y almacena las variables asociadas a ella.

Las variables de sesión se almacenan en el array asociativo `$_SESSION`

session1.php

```
<?php
session_start();
$_SESSION["nombre"] = "Javier";
echo '<a href="session1_ver.php">Pulsa para ir a otra página</a><br>';
?>
```

session1_ver.php

```
<?php
session_start();
echo "Nombre = ", $_SESSION["nombre"];
?>
```

session_start() se llama al comienzo, antes de generar código HTML

Sesiones Ejemplo

Formulario de login

Crear una página de login usando sesiones



```

function htmlLogin() {
echo <<< HTML
<p>Introduzca sus credenciales:</p>
<form action="session1.php" method="post">
<label>Usuario</label>
<input type="text" name="usuario"> <br>
<label>Clave</label>
<input type="password" name="pwd"> <br>
<input type="submit" name="login" value="Login">
</form>
HTML;
}

```



```

function htmlBienvenido($nombre) {
echo <<< HTML
<p>Bienvenido $nombre, sesión establecida</p>
<form action="session1.php" method="post">
<input type="submit" name="logout" value="Logout">
</form>
HTML;
}

```

Sesiones Ejemplo

Formulario de login

Crear una página de login usando sesiones

```

session_start(); // Antes de comenzar HTML

// Comprobar estado previo
if (isset($_POST["usuario"])) {
    // Acceso desde formulario de login,
    // Comprobar credenciales [...]
    $_SESSION["usuario"] = $_POST["usuario"];
} else if (isset($_POST["logout"])) {
    // Acceso desde formulario de logout
    acabarSesion();
}

htmlInicio();
if (isset($_SESSION["usuario"])) {
    // Si la sesión está establecida
    htmlBienvenido($_SESSION["usuario"]);
} else {
    // Si la sesión NO está establecida
    htmlLogin();
}
htmlFin();

```

```

function htmlInicio() {
echo <<< HTML
<!DOCTYPE html>
<html>
<head>
<meta content="text/html; charset=utf-8"
      http-equiv="content-type">
<title>Ejemplo de sesión</title>
</head>
<body>
HTML;
}

function htmlFin() {
echo <<< HTML
</body>
</html>
HTML;
}

```

Sesiones
Ejemplo

Almacenamiento

Las sesiones se almacenan en un fichero del servidor



```
<?php  
session_start();  
$_SESSION["nombre"] = "Javier";  
...  
?>
```

Servidor Apache en Ubuntu (un fichero por sesión):
`/var/lib/php/session(sess_1nrfdairbgi0q117afj0ogj40)`
`nombre|s:6:"Javier";`

El nombre del fichero incluye el ID de la Cookie de sesión

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 59

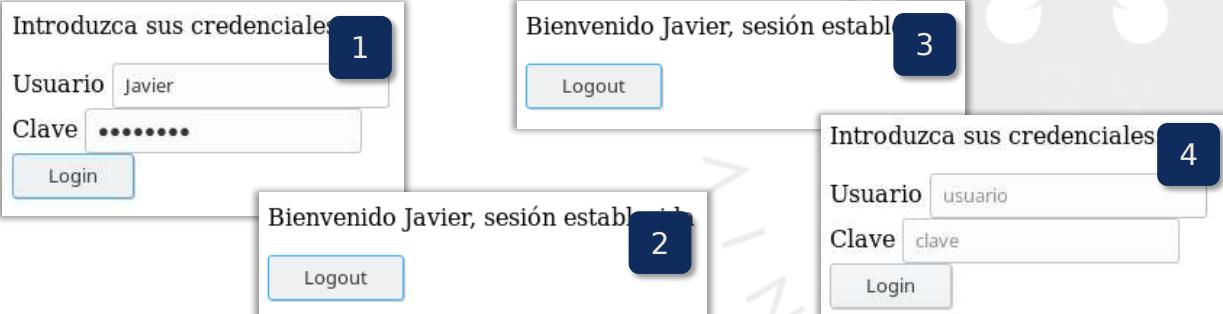
Sesiones
Cómo se finalizan

Finalizar una sesión

La función `session_destroy()` destruye la sesión pero:

- No destruye las variables de la sesión (se pueden seguir usando)
- No elimina la cookie de la sesión (se sigue enviando en cada página)

```
function acabarSesion() {  
    // La sesión debe estar iniciada  
    if (session_status() == PHP_SESSION_NONE)  
        session_start();  
  
    // Destruir sesión  
    session_destroy();  
}
```



1. Introduzca sus credenciales
2. Bienvenido Javier, sesión establecida
3. Logout
4. Introduzca sus credenciales

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 60

Sesiones
Cómo se finalizan

Finalizar una sesión

La función `session_destroy()` destruye la sesión pero:

- No destruye las variables de la sesión (se pueden seguir usando)
- No elimina la cookie de la sesión (se sigue enviando en cada página)

```

function acabarSesion() {
    // La sesión debe estar iniciada
    if (session_status()==PHP_SESSION_NONE)
        session_start();

    // Borrar variables de sesión
    //$_SESSION = array();
    session_unset();

    // Obtener parámetros de cookie de sesión
    $param = session_get_cookie_params();

    // Borrar cookie de sesión
    setcookie(session_name(), $_COOKIE[session_name()], time()-2592000,
              $param['path'], $param['domain'], $param['secure'], $param['httponly']);

    // Destruir sesión
    session_destroy();
}

```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 61

Sesiones
Sobre la configuración del servidor

Sesiones PHP

Permiten recordar el estado de la aplicación de forma muy sencilla: El servidor genera una cookie que identifica la sesión y almacena las variables asociadas a ella.

¿Y si el cliente tiene deshabilitadas las cookies?

Se puede pasar el ID de la sesión por GET/POST

- En la URL
- En el caso de formularios se puede pasar como un campo hidden

PHP puede añadir automáticamente el ID de sesión a la URL o a formularios

```

function iniciarSesion() {
    ini_set("session.use_cookies", 0);
    ini_set("session.use_only_cookies", 0);
    ini_set("session.use_trans_sid", 1);
    session_start();
}

function iniciarSesion() {
    session_start(["use_cookies" => "0",
                  "use_only_cookies" => "0",
                  "use_trans_sid" => "1"]);
}

```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 62

Sesiones

Sobre la configuración del servidor

```

function htmlBienvenido($nombre) {
echo <<< HTML
<p>Bienvenido $nombre, sesión establecida</p>
<form action="session2.php" method="get">
<input type="submit" name="logout" value="Logout">
</form><br>
<a href="session2.php">Seguir en la página</a>
HTML;
}

```

Código HTML generado por PHP

```

<!DOCTYPE html>
<html>
<head>
<meta content="text/html; charset=utf-8" http-equiv="content-type">
<title>Ejemplo de sesión</title>
</head>
<body>
<p>Bienvenido Javier, sesión establecida</p>
<form action="session2.php" method="post">
<input type="hidden" name="PHPSESSID" value="i141sv0d1pc23alum3ejt742j1" />
<input type="submit" name="logout" value="Logout">
</form><br>
<a href="session2.php?PHPSESSID=i141sv0d1pc23alum3ejt742j1">Seguir en la página</a>
</body>
</html>

```

Añadidos por PHP automáticamente

Sesiones

Seguridad

Sesiones PHP

No se recomienda pasar el ID de sesión en URL

- Queda registrado en logs que pueden ser externos
- Facilita ataques man-in-the-middle

La cookie de ID de sesión o la variable POST del formulario también se pueden interceptar.

Recomendación:

- Usar cookies
- Usar HTTPS para cifrar las cookies

Medidas de seguridad si no se puede usar HTTPS:

- Almacenar alguna información adicional del cliente que establece la sesión y comprobarlo en cada acceso (dirección IP, user agent, ...)
- Cambiar el ID de la sesión (session_regenerate_id())

<http://blog.teamtreehouse.com/how-to-create-totally-secure-cookies>



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor – PHP+web

- 1. El lenguaje PHP
- 2. PHP y aplicaciones web
 - 1. Uso de PHP en la web
 - 2. Procesamiento de formularios
 - 3. Saneamiento de cadenas
 - 1. URL encoding
 - 2. Saneamiento de cadenas
 - 3. Query strings
 - 4. Recordando el estado de las aplicaciones
 - 1. Cookies
 - 2. Sesiones
 - 5. Envío de encabezados
 - 3. PHP y conexión con BBDD



Envío de encabezados

Redirección y envío de datos

Header: envío de encabezados HTTP

```
header($msg);  
Debe ponerse antes de enviar HTML
```

```
<?php  
header("Location: http://www.google.es");  
?>
```

```
<?php  
header("Refresh: 3; url=http://www.google.es");  
echo "Redirigiendo en 3 segundos ...";  
?>
```

```
<?php  
header("Content-Type: text/plain");  
echo "Esto es texto plano que se enviará desde el servidor";  
?>
```

Envío de encabezados

Envío de ficheros

Header: envío de encabezados HTTP

```
header($msg);  
Debe ponerse antes de enviar HTML
```

```
<?php  
header("Content-Type: image/png");  
readfile("./smiley.png");      readfile: lee un fichero y lo envía a la salida  
?>
```

```
<?php  
header('Content-Type: application/octet-stream');  
header('Content-Disposition: attachment; filename="fichero.txt"');  
echo "Esto es texto plano que se enviará desde el servidor";  
?>
```

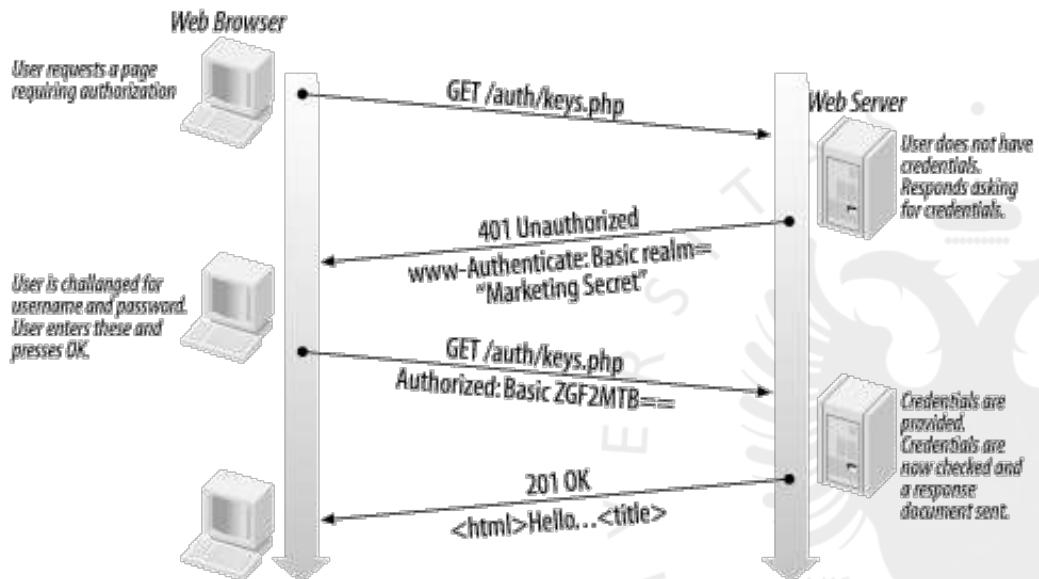
```
<?php  
header('Content-Type: application/octet-stream');  
header('Content-Disposition: attachment; filename="smiley.png"');  
header('Content-Length: ' . filesize("smiley.png"));  
readfile("smiley.png");  
?>
```

Envío de encabezados

Autenticación en el servidor

Autenticación HTTP

Usa la autenticación mediante una ventana similar a la que se usa cuando se configura htaccess en el servidor web



<http://www.techflirt.com/http-basic-authentication-php>

Williams, Lane "Web database applications with PHP & MySQL". O'Reilly. 2002

Envío de encabezados
Autenticación en el servidor



Autenticación HTTP

- Pide autenticación con cada cambio de “realm”
- Almacena credenciales en `$_SERVER['PHP_AUTH_USER']` y `$_SERVER['PHP_AUTH_PW']`
- No cifra la clave en la transmisión
- No permite hacer logout

```

if (isset($_SERVER['PHP_AUTH_USER']) && isset($_SERVER['PHP_AUTH_PW']) &&
    $_SERVER['PHP_AUTH_USER']=="ElUsuario" &&
    $_SERVER['PHP_AUTH_PW']=="LaClave") {
    echo "Usuario autenticado ", $_SERVER['PHP_AUTH_USER']
} else {
    header('WWW-Authenticate: Basic realm="Acceso restringido"');
    header('HTTP/1.0 401 Unauthorized');
    // Si pulsamos cancelar llegamos aquí
    die("Las credenciales no son válidas");
}

```

Ejemplo
Sistema de login con redirección a origen



The diagram illustrates a login process with redirection:

- The user starts at **Página 1**, which displays a navigation bar with links to Página 1, Página 2, Página 3 (id), Página 4 (id), Login, and Logout.
- The user clicks the **Login** link, which triggers a redirection.
- The user is redirected to a **Login page** (a separate window). This page has its own navigation bar and displays fields for **Usuario:** Javi and **Clave:** Below these fields is a **Acceder** button.
- The user enters the credentials and clicks **Acceder**.
- The user is redirected back to the **Página 3** (the original page from which they started).
- The **Página 3** now displays a message: **Bienvenido, ya está identificado** (Welcome, you are identified).

Arrows indicate the flow of the redirection process from the original page through the login page back to the original page.

Recuerda la página desde la que se ha llegado al login



Ejemplo

Sistema de login con redirección a origen

pagina1.php

```
<?php
require('nochecklogin.php');
require('include.php');
HTMLinicio('Página 1');
HTMLencabezado();
echo '<h1>Página 1</h1>';
HTMLpiepagina();
HTMLfin();
?>
```

nochecklogin.php

```
<?php
if (session_status()==PHP_SESSION_NONE)
    session_start();
unset($_SESSION['desdedonde']);
?>
```

pagina3.php

```
<?php
require('checklogin.php');
require('include.php');
HTMLinicio('Página 3');
HTMLencabezado();
echo '<h1>Página 3</h1>';
HTMLpiepagina();
HTMLfin();
?>
```

checklogin.php

```
<?php
if (session_status()==PHP_SESSION_NONE)
    session_start();
if (!isset($_SESSION['usuario'])) {
    $_SESSION['desdedonde']=$_SERVER['REQUEST_URI'];
    header("Location: login.php");
}
?>
```



Ejemplo

Sistema de login con redirección a origen

login.php

```
<?php
if (session_status()==PHP_SESSION_NONE)
    session_start();
require('include.php');
HTMLinicio('Login');
HTMLencabezado();
if (isset($_SESSION['usuario'])) {
    if (isset($_SESSION['desdedonde'])) {
        echo "<h1>Bienvenido, {$_SESSION['nombre']} . Identificación correcta</h1>";
        echo "<h2>... en unos segundos podrá continuar su navegación ...</h2>";
        header("Refresh:3; url={$_SESSION['desdedonde']}");
    } else
        echo "<h1>Bienvenido, {$_SESSION['nombre']}. Identificación correcta</h1>";
} else if (isset($_POST['submit']) && isset($_POST['usuario']) &&
           isset($_POST['password'])) {
    // *** En este punto hay que autenticar al usuario ***
    $_SESSION['usuario'] = $_POST['usuario'];
    $_SESSION['nombre'] = "Pepito Pérez ({$_POST['usuario']})";
    header("Location: {$_SERVER['SCRIPT_NAME']}"); // Para actualizar encabezado
} else
    FORM_login('');
HTMLpiepagina();
HTMLfin();
?>
```

Ejemplo

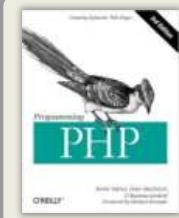
Sistema de login con redirección a origen

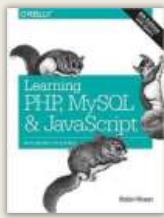
logout.php

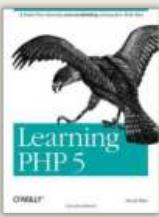
```
<?php
if (session_status() == PHP_SESSION_NONE)
    session_start();
require('include.php');
HTMLinicio('Logout');
HTMLencabezado();
if (isset($_SESSION['usuario'])) {
    acabarSesion();
    // Para actualizar encabezado
    header("Location: {" . $_SERVER['SCRIPT_NAME'] . "}");
} else
    echo '<h1>La sesión ha terminado</h1>';
HTMLpiepagina();
HTMLfin();
?>
```

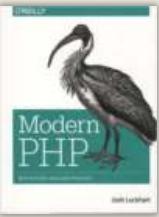
Programación en el lado del servidor. PHP+Web

Bibliografía

 Kevin Tatroe, Peter MacIntyre, Rasmus Lerdorf
Programming PHP
O'Reilly. 2013

 Robin Nixon
Learning PHP, MySQL, & JavaScript (4th ed)
O'Reilly. 2014
<http://lpmj.net/>

 David Sklar
Learning PHP
A gentle introduction to the web's most popular language
O'Reilly. 2016

 Josh Lockhart
Modern PHP
New features and good practices
O'Reilly. 2015