

UT2

Lenguaje PHP



Lenguaje PHP: recordando al usuario: Sesiones

- **Sesiones:** mecanismo que permite **conservar información del usuario** cuando éste cambia de página dentro del sitio web, y hasta que éste la cierre = **mientras lo esté visitando.**



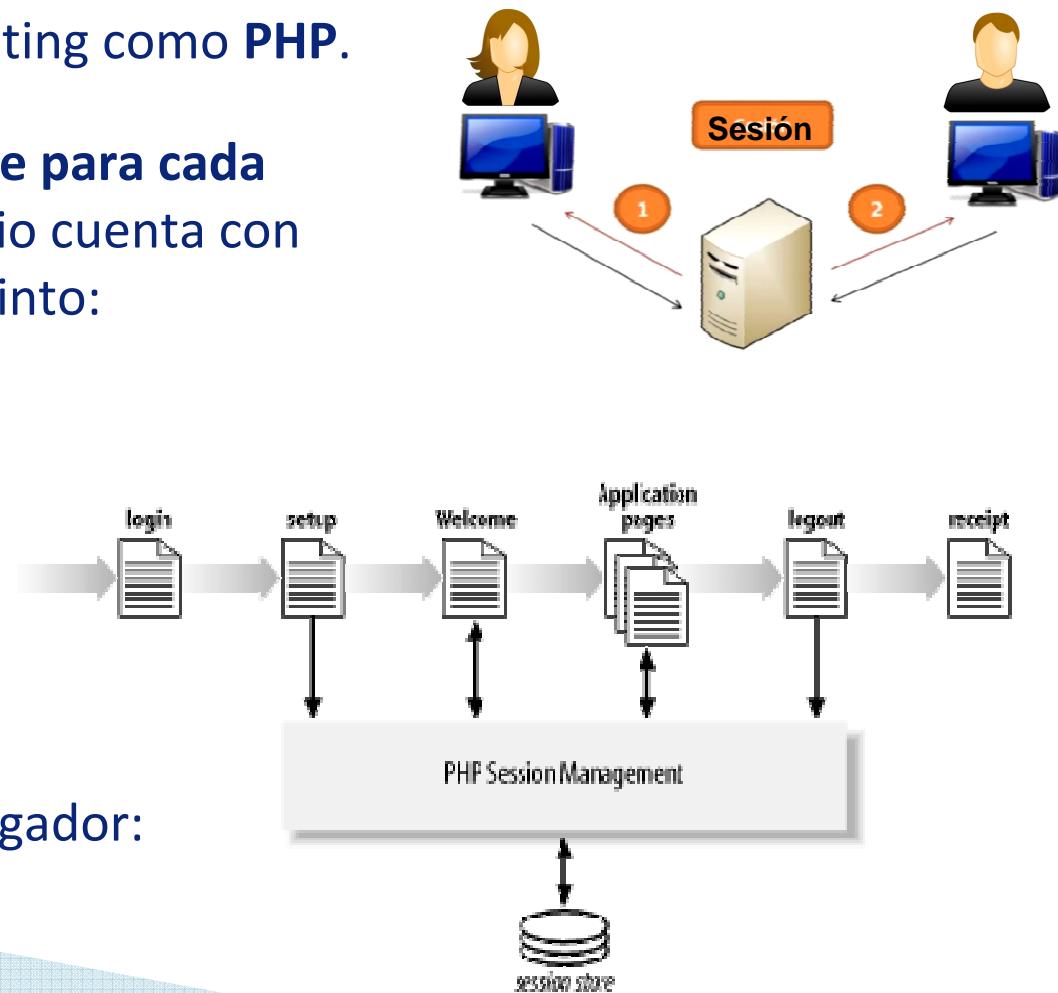
- Útiles en aplicaciones web en las que el usuario **navega a través de varias páginas del sitio** y se necesita saber los datos que introdujo anteriormente → así la aplicación no lo tratará como peticiones independientes; sino del mismo usuario, con lo que podrá recordar sus datos **propagándolos entre páginas** (por medio de *variables de sesión*):



Aplicación web de reservas que almacena mediante una sesión los datos que el cliente va introduciendo en páginas anteriores del mismo sitio.

Lenguaje PHP: Sesiones

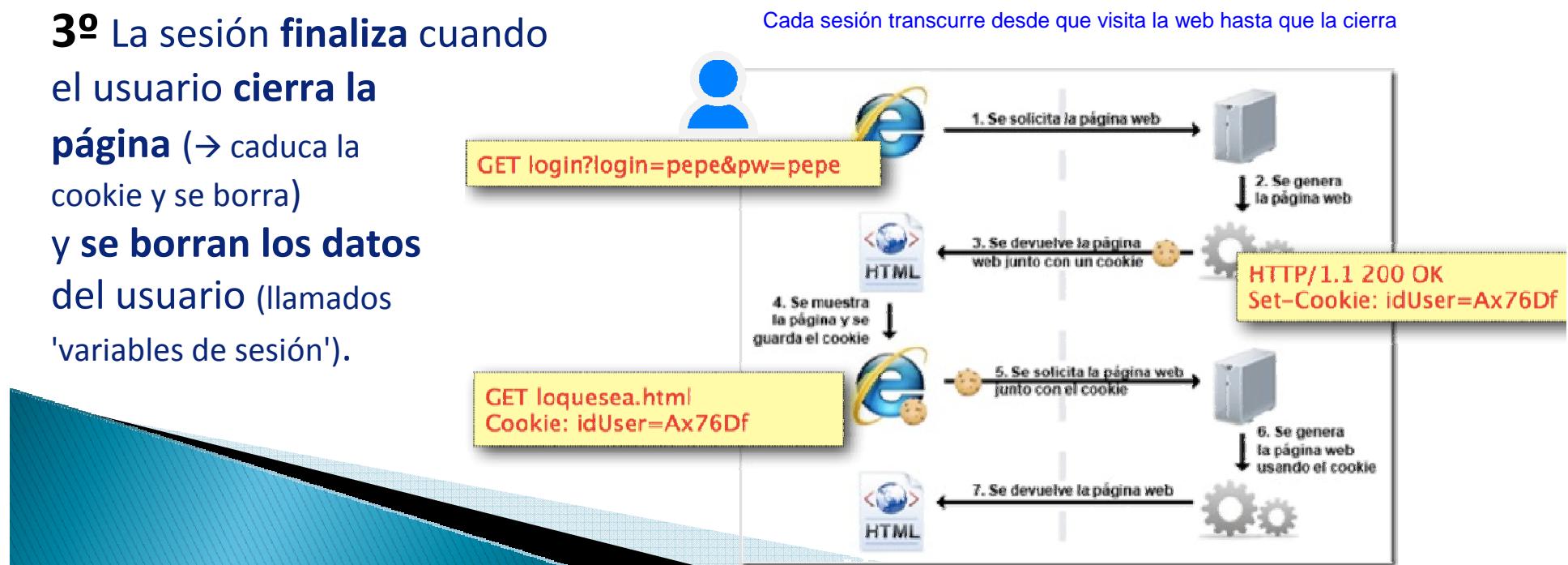
- Una sesión es el **espacio de tiempo** transcurrido desde que el usuario accede a una web hasta que finaliza su visita (se desloguea o cierra el navegador).
- Las sesiones se almacenan en el **lado servidor** usando algún lenguaje de scripting como **PHP**.
- Cada sesión es **independiente para cada usuario**; por lo que cada usuario cuenta con un **identificador de sesión** distinto:
 - La sesión **se crea cuando** el usuario **se loguea** para propagar datos suyos mientras navegue entre las páginas del sitio; y termina cuando cierre sesión (**logout**) o cierre el navegador:



Lenguaje PHP: Funcionamiento de las Sesiones

- 1º** La sesión se crea asignando un Identificador de sesión que es almacenado en una cookie en el cliente (setCookie) para poder recuperar sus datos posteriormente = "cookie de sesión".
- 2º** Cuando el usuario pasa de una página a otra de la aplicación web, el server lee la cookie para obtener su ID de sesión y determinar mediante el ID qué datos de sesión almacenados previamente se debe usar (variables de sesión):

- 3º** La sesión finaliza cuando el usuario cierra la página (→ caduca la cookie y se borra) y se borran los datos del usuario (llamados 'variables de sesión').



Lenguaje PHP: Sesiones

Las sesiones no perviven entre visitas (se olvidan al cerrar la página) → no pueden propagar datos entre visitas; sino entre páginas de la misma visita.

- Las cookies sí pueden perdurar tras una visita (hasta que caduquen).

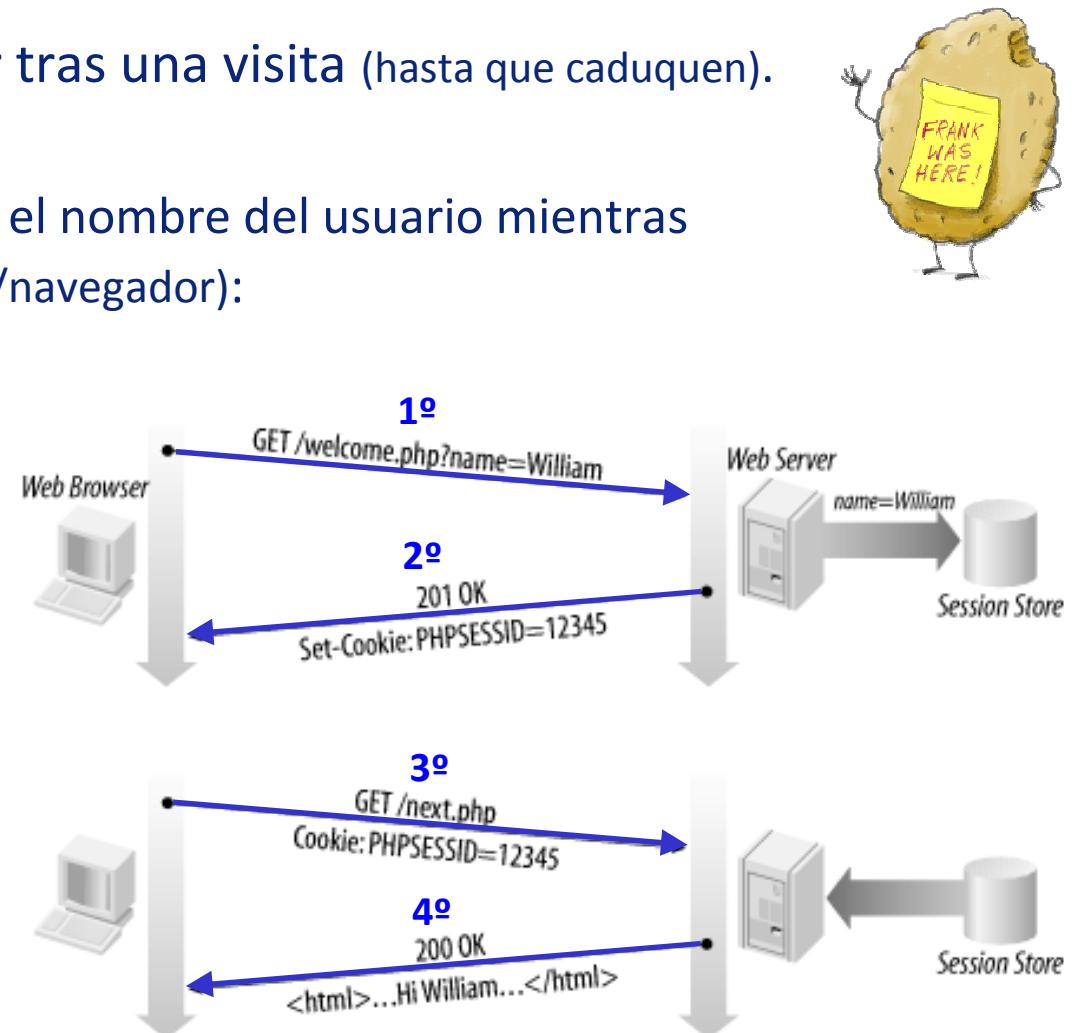
- **Ejemplo:** sesión que recuerda el nombre del usuario mientras esté online (hasta que cierre el cliente/navegador):

1º El usuario visita una web y envía su nombre por medio de un form.

2º El servidor crea una sesión para él dándole un ID y lo guarda en una cookie. Guarda su nombre en la sesión creada.

3º El usuario visita otra página del sitio → el servidor lee el ID de su cookie.

4º A partir del ID rescata la sesión del usuario y recupera del server su nombre.



Al cerrar el navegador se cerrará la sesión y se borrará la cookie con el ID.

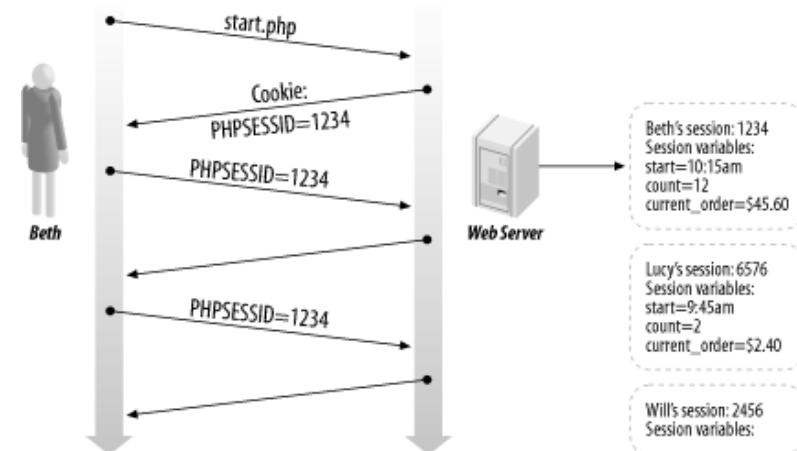
Lenguaje PHP: Sesiones

- Gracias a que el ID de sesión se guarda tanto en el servidor como en una cookie en el cliente, es posible tratar simultáneamente distintas sesiones de distintos usuarios:

Los datos del cliente (las variables de sesión) **sólo se almacenan en el servidor.**

Comparado con las cookies, las sesiones son un mecanismo más:

- **Seguro:** los datos del usuario están en el server y no en su equipo.
- **Volátil:** las sesiones caducan al cerrar el navegador → se olvidan/borran los datos del usuario.



Para usar sesiones, el usuario debería tener habilitadas las cookies en su navegador (cliente web).

¿Qué ocurre con la sesión si borramos la cookie en el cliente?
Se pierde la sesión y se inicia otra (otro ID) al recargar el sitio

Lenguaje PHP: manejo de Sesiones con PHP

- Verificar que las sesiones están habilitadas en el intérprete de PHP del servidor web:

archivo C:\xampp\php\php.ini
o mediante la función **phpinfo()**;

- **Proseguir** con la sesión actual si existe la cookie con el ID, o **crear** una sesión si no existe:

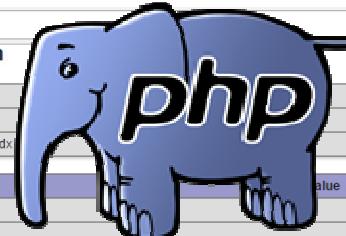
```
bool session_start();
```

- **Consultar** la sesión (una existente; o la recién creada si no existe una):

```
string session_name(); // Devuelve el nombre de la sesión actual: 'PHPSESSID' por
                      // defecto, definido en el parámetro session.name de php.ini)

session_name('nombre'); // Asigna un nombre a la sesión actual (distinto a PHPSESSID)
                       // Será el mismo nombre que se dará a la cookie del cliente
                       // que almacenará el ID de la sesión.
                       // Debe invocarse antes que session_start() para que cambie.
```

```
string session_id(); // Devuelve el ID de la sesión actual
```



A screenshot of a web browser window showing the output of the `phpinfo()` function. The page title is "localhost/sesion.php" and the section title is "session". The table lists various session-related configuration directives and their values. A large blue elephant icon with the word "PHP" is overlaid on the right side of the table. The \$_SESSION variable is also visible at the bottom of the table.

Directive	Local Value	Value
Session Support	enabled	
Registered save handlers	files user	
Registered serializer handlers	php_serialize php php_binary wddx	
Directive	Local Value	Value
session.auto_start	Off	
session.cache_expire	180	
session.cache_limiter	nocache	
session.cookie_domain	no value	
session.cookie_httponly	Off	
session.cookie_lifetime	0	
session.cookie_path	/	
session.cookie_secure	Off	
session.entropy_file	no value	
session.entropy_length	0	
session.gc_divisor	1000	
session.gc_maxlifetime	1440	
session.gc_probability	1	
session.hash_bits_per_character	5	
session.hash_function	0	
session.name	PHPSESSID	PHPSESSID
session.referer_check	no value	no value
session.save_handler	files	files
session.save_path	C:\xampp\tmp	C:\xampp\tmp

Lenguaje PHP: Sesiones

Las sesiones se usan para mantener/propagar datos del usuario entre secciones de una aplicación web; como: una búsqueda- configuración por pasos, una compra (carrito), etc.

- **Ejemplo:** web que crea una sesión con el nombre por defecto y muestra su ID en cada visita:

```
<!DOCTYPE html>
<html>
<head> <title> Sesión básica </title> </head>
<body>
<?php
    session_start(); // inicia la sesión con nombre PHPSESSID o la continua si existe
    echo "<h4> Datos de la sesión actual: </h4>";
    echo "ID: ".session_id();
    echo "<br> Nombre: ".session_name()."<br>";
    echo "Parámetros de la cookie de sesión: <br>";
    print_r(session_get_cookie_params());
    echo "<br> <br> Estos datos se perderán al cerrar la página";
    echo "<br> y se creará una nueva sesión al abrirla otra vez.";
?
<hr> <br> <a href=<?php echo $_SERVER['PHP_SELF']; ?>>RECARGAR LA PÁGINA</a>
</body>
</html>
```

El **nombre de la cookie** que se creará en el cliente para almacenar el ID de la sesión será **el mismo nombre que el de la sesión**.

Lenguaje PHP: Sesiones

- Al visitar la web por primera vez se crea una sesión; y mientras recargue la web, se prosigue con la misma sesión (el mismo ID):



Datos de la sesión actual:

ID: 1bctig4uti7liab7pas5ib6v3
Nombre: PHPSESSID

Parámetros de la cookie:

Array ([lifetime] => 0 [path] => / [domain] => [secure] => [httponly] =>)

Estos datos se perderán al cerrar la página
y se creará una nueva sesión al abrirla otra vez.

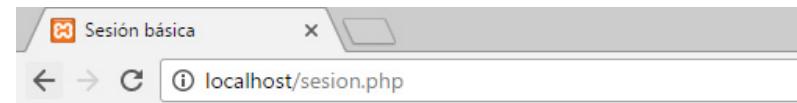
[RECARGAR LA PÁGINA](#)

Podemos comprobar en el cliente cómo existe la cookie con el mismo nombre que la sesión (PHPSESSID) y con el ID de la sesión.

La cookie caducará (será eliminada) cuando el usuario cierre el navegador dando por concluida la sesión:

- Si el usuario cierra el navegador termina la sesión.

Si luego vuelve a visitar la web, se crea una nueva sesión (un nuevo ID):



Datos de la sesión actual:

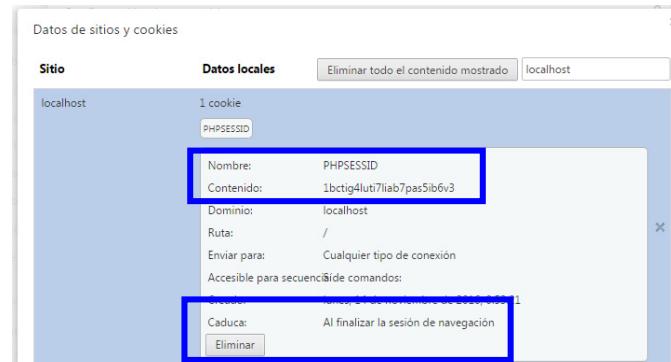
ID: 72amo9mcb9h7k7mbt9fb16cci3
Nombre: PHPSESSID

Parámetros de la cookie:

Array ([lifetime] => 0 [path] => / [domain] => [secure] => [httponly] =>)

Estos datos se perderán al cerrar la página
y se creará una nueva sesión al abrirla otra vez.

[RECARGAR LA PÁGINA](#)

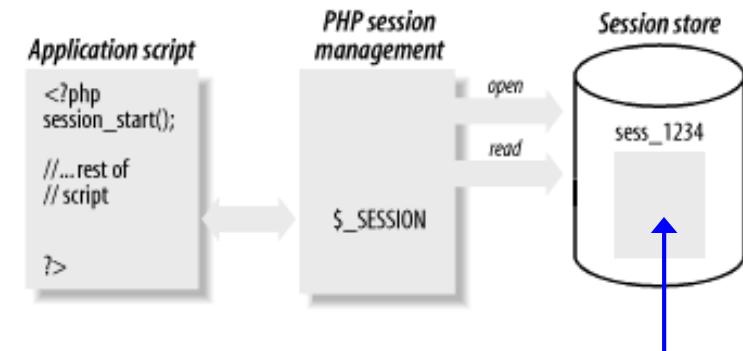


En Chrome podemos editar las cookies en:
(CTRL+SHIFT+SUPR)
chrome://settings →
Ajustes → Mostrar configuración avanzada → Privacidad →
Configuración de contenido → Cookies →
Todas las cookies y datos de sitios

Lenguaje PHP: manejo de Sesiones con PHP

- **Variables de sesión:** datos del usuario que se almacenan en el servidor (y no en la cookie del cliente). Están asociadas al ID de la sesión y se manipulan con el array asociativo superglobal `$_SESSION[]`. Se propagan automáticamente entre las páginas del sitio que sean visitadas durante la sesión.

Las variables de cada sesión se almacenan en un archivo de texto en el server (C:\xampp\tmp). Se guardan hasta que se finalice la sesión: se cierre el navegador y se borre la cookie con el ID. El archivo se llama como el ID de la sesión a la que representa.



- **Modificar/Consultar las variables de sesión (datos de usuario):**

```
$_SESSION['dato1'] = $valor1;  
$_SESSION['dato2'] = $valor2;
```

```
if (isset($_SESSION['dato']))  
    $valor = $_SESSION['dato'];
```

- **Borrar una sesión (su ID) junto con sus variables y la cookie asociada:**

```
session_start();  
$_SESSION = array(); // Borra todas las variables de sesión. Igual que unset($_SESSION);  
setcookie(session_name(), '', time() - 3600, '/'); // Borra la cookie de sesión  
session_destroy(); // Destruye la sesión.
```



Si no la borramos explícitamente → se borrará al cerrar el navegador

Lenguaje PHP: Sesiones

- **Ejemplo:** contador implementado con sesiones. Cuenta las veces que la web es visitada durante la sesión hasta que se cierre el navegador y se termine la sesión perdiéndose el contador.

```
<?php
    session_name('mi_contador'); // Nombre de la sesión a crear/continuar. Mismo nombre que la cookie a crear
    session_start(); // Crea la sesión con el nombre especificado o se continua si ya existe la sesión
    if (!isset($_SESSION['contador'])) { // No existe la variable-> primera vez que se visita->contador a 1
        $contador = 1;
        $mensaje = "Es la primera vez que me visitas en esta sesión.";
        $color = "#7799FF";
    } else { // Ya existe la variable-> ya ha visitado-> incrementar contador
        $contador = $_SESSION['contador']; // Lee el valor de la variable de sesión
        $contador++;
        $mensaje = "Me has visitado en esta sesión con este navegador " . $contador . " veces.";
        $color = "#FFFF00";
    }
    $_SESSION['contador'] = $contador; // Guarda en la variable de sesión
?>
<html> <head> <title>Contador durante la sesión</title> </head>
<body bgcolor="<?php echo $color; ?>">
    <h3> <?php echo $mensaje; ?> </h3>
    <br> Al cerrar el navegador me olvidaré del número de visitas.
    <br> Lo siento, soy una sesión, no una cookie. <br> <br>
    <a href="<?php echo $_SERVER['PHP_SELF']; ?>">RECARGAR LA PÁGINA</a>
</body> </html>
```

- **Si cerramos el navegador:** se cierra la sesión, se borra la cookie 'mi_contador' y se borra el valor del contador.
- **Al abrir el navegador de nuevo y visitar la web:** se crea una nueva sesión → reinicia el contador a 1:



Lenguaje PHP: Sesiones

- **Actividad:** crea una página llamada sesionarray.php que permita almacenar y aplicar (propagar) durante la sesión varias preferencias del usuario:

- Su nombre.
- El color de fondo que prefiere para la web (rojo, verde o azul).
- El lenguaje elegido para el texto de la web (español o inglés).

Cuando el usuario cargue la página; elegirá sus preferencias y pulsará ACCEDER para visitar el sitio → se le conduce a la página siguiente.php; que se mostrará aplicándole sus preferencias. El usuario podrá volver atrás para cambiar sus preferencias (le mostrará las que tenía). Guarda también en la sesión la fecha y hora en la que ésta comenzó, y muéstrasela al usuario al acceder. Implementa el cerrar sesión (logout) para que cierre la sesión y le devuelva al inicio (fin.php):

Hace un logout, pulsa Back, o cierra el navegador y vuelve a visitarla

Mantiene las preferencias al recargar

Al ACCEDER, se guardan sus preferencias en variables de sesión y se propagan para aplicarlas en la página siguiente:

Session started at: 09:48 - 28/10/2016

Hi Alumn@,
This web saves your preferences in a session with:
your name, background color and language selected.
These values will be lost if you finish the session closing the browser.

Lenguaje PHP: Sesiones

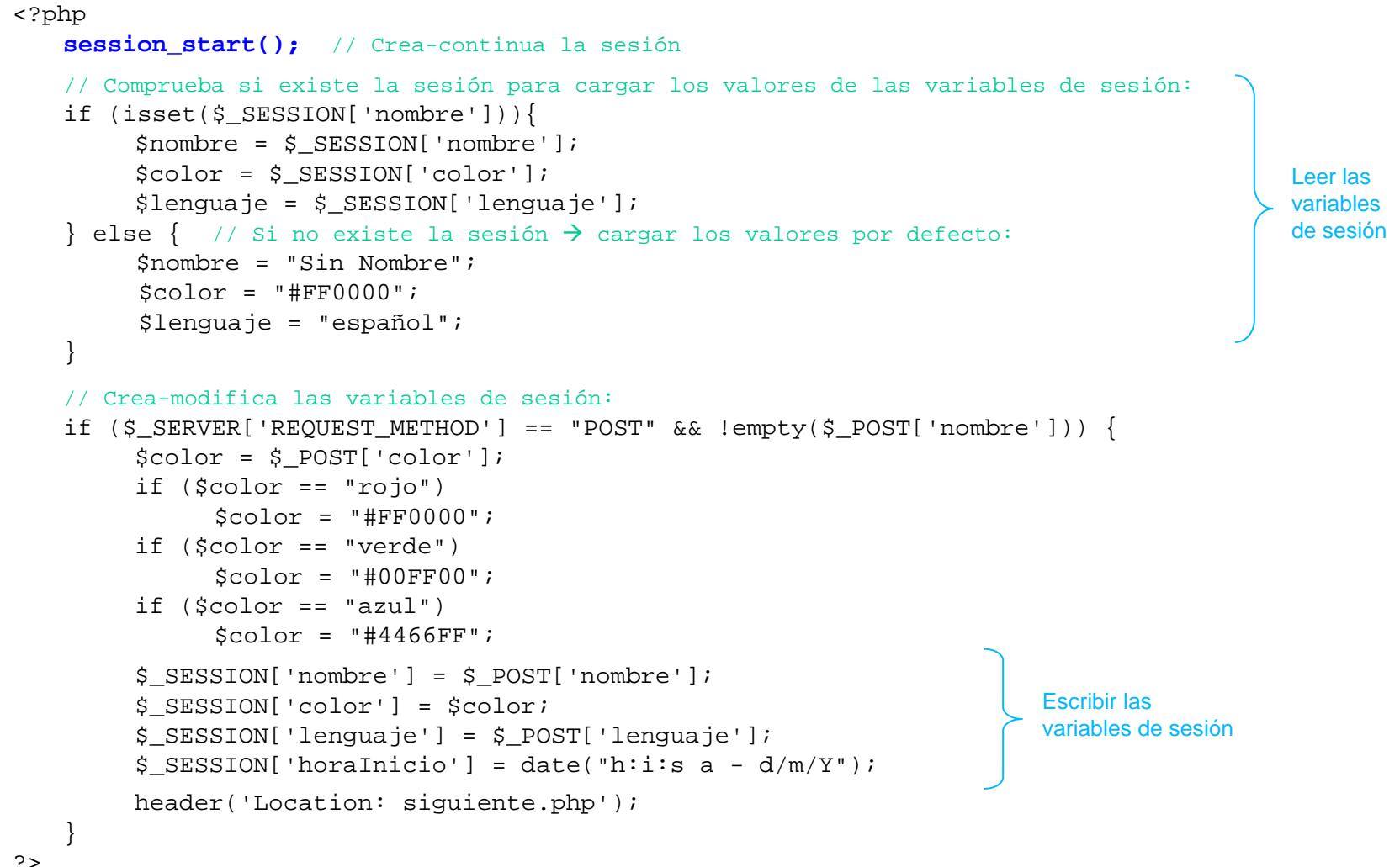
Archivo **sesionarray.php**

```
<?php
    session_start(); // Crea-continua la sesión

    // Comprueba si existe la sesión para cargar los valores de las variables de sesión:
    if (isset($_SESSION['nombre'])) {
        $nombre = $_SESSION['nombre'];
        $color = $_SESSION['color'];
        $lenguaje = $_SESSION['lenguaje'];
    } else { // Si no existe la sesión → cargar los valores por defecto:
        $nombre = "Sin Nombre";
        $color = "#FF0000";
        $lenguaje = "español";
    }

    // Crea-modifica las variables de sesión:
    if ($_SERVER['REQUEST_METHOD'] == "POST" && !empty($_POST['nombre'])) {
        $color = $_POST['color'];
        if ($color == "rojo")
            $color = "#FF0000";
        if ($color == "verde")
            $color = "#00FF00";
        if ($color == "azul")
            $color = "#4466FF";

        $_SESSION['nombre'] = $_POST['nombre'];
        $_SESSION['color'] = $color;
        $_SESSION['lenguaje'] = $_POST['lenguaje'];
        $_SESSION['horaInicio'] = date("h:i:s a - d/m/Y");
        header('Location: siguiente.php');
    }
?>
```



Leer las variables de sesión

Escribir las variables de sesión

Lenguaje PHP: Sesiones

```
<!-- continuación del archivo sesionarray.php -->

<html>
    <head>
        <title>Sesion array</title>
    </head>
    <body bgcolor="#FFEEBB">
        <p style ="color:rgb(180,180,0); font-family:verdana; font-size:26px;"> Mi Web </p>
        <p>Personalice el sitio web antes de acceder:</p>
        <form method="post" action="">
            Nombre: <input type="text" name="nombre" value=<?php echo $nombre; ?>" required>
            <select name="color" required>
                <option value="rojo" <?php if ($color == "#FF0000") echo "selected";?> >rojo</option>
                <option value="verde" <?php if ($color == "#00FF00") echo "selected";?> >verde</option>
                <option value="azul" <?php if ($color == "#4466FF") echo "selected";?> >azul</option>
            </select>
            <select name="lenguaje" required>
                <option value="español" <?php if($lenguaje=="español") echo "selected";?>>español</option>
                <option value="ingles" <?php if($lenguaje=="ingles") echo "selected";?>>ingles</option>
            </select>
            <input type="submit" name="submit" value="ACCEDER >>">
        </form>
        <hr>
    </body>
</html>
```

Lenguaje PHP: Sesiones

Archivo **siguiente.php**

```
<?php
    session_start(); // Continuar con la sesión
    if (isset($_SESSION['nombre'])){ // Comprobar si existe la sesión
        $nombre = $_SESSION['nombre'];
        $color = $_SESSION['color'];
        $lenguaje = $_SESSION['lenguaje'];
        $horaInicio = $_SESSION['horaInicio'];
    } else { // Si no existe la sesión es que ha caducado
        $mensaje = "Su sesión ha caducado. <br><br>";
        $mensaje .= "<a href=\"sesionarray.php\"><< VOLVER </a>";
        echo $mensaje;
        exit;
    }
?>
<html>
    <head> <title>Siguiente</title> </head>
    <body bgcolor=" <?php echo $color; ?> ">
        <p style ="color:rgb(180,180,0); font-family:verdana; font-size:26px;"> Mi Web </p>
        <hr> <br>
        <?php
            if ($lenguaje == "español") {
                echo "Sesión iniciada a las: ".$horaInicio;
                echo "<br> <br> <h4>";
                echo "Hola " . $nombre . ",<br>";
                echo "Esto es una página que guarda tus preferencias en una sesión con:<br>";
                echo "tu nombre, color de fondo y el lenguaje seleccionados." ;
                echo "</h4> <br>";
                echo "<a href=\"sesionarray.php\"><< VOLVER</a> &nbsp; &nbsp; &nbsp; ";
                echo "<a href=\"".$_SERVER['PHP_SELF']."\">RECARGAR</a>";
            }
        }
    </body>
</html>
```



Leer las variables de sesión

Lenguaje PHP: Sesiones

```
<!-- continuación del archivo siguiente.php -->

    if ($lenguaje == "ingles") {
        echo "Session started at: ".$horaInicio;
        echo "<br> <br> <h4>";
        echo "Hi " . $nombre . ",<br>";
        echo "This web saves your preferences in a session with:<br>";
        echo "your name, background color and language selected.";
        echo "</h4> <br>";
        echo "<a href=\"sesionarray.php\"><< BACK</a> &nbsp; &nbsp; &nbsp; ";
        echo "<a href=\"".$_SERVER['PHP_SELF']."\">RELOAD</a>";
    }
?>

<div style="position:absolute; left:500px; top:10px; width:95px; height:45px; background:#FFEEBB;
border:2px solid gray; padding:5px; text-align:center; font-family:verdana; color:green; font-size:14px;">
<?php echo $nombre; ?> <br>
<a href="fin.php">Logout</a>
</div>
</body>
</html>
```

Archivo fin.php

```
<?php
    session_start();      // engancha con la sesión actual
    $_SESSION = array(); // y destruye la sesión. Igual que unset($_SESSION);
    setcookie(session_name(),'',time()-3600,'/');
    session_destroy();
?>

<html> <head> <title>Sesion array</title> </head>
<body bgcolor="#FFEEBB">
    <p style ="color:rgb(180,180,0); font-family:verdana; font-size:26px;">Mi Web</p>
    <p>Ha cerrado su sesión. Puede volver a iniciarla:</p>
    <a href="sesionarray.php"><< VOLVER</a> <hr>
</body> </html>
```

Lenguaje PHP: Sesiones

- **Actividad:** desarrolla un asistente online para configurar seguros de coche.

El configurador deberá propagar entre sus 4 páginas (pasos) los datos que vaya introduciendo el usuario. La última página mostrará todos los valores elegidos y permitirá al usuario indicar su email para que la web le envíe el presupuesto. Al enviarlo, se redirigirá a al archivo fin.php; que mostrará un mensaje de despedida, cerrará (destruirá) la sesión y creará un

enlace para que el usuario acceda de nuevo al inicio del configurador (index.php):

¿Cuál es la marca del coche que quieras asegurar?

					
AUDI	BMW	CITROËN	DACIA	FIAT	FORD

[index.php](#) [SIGUIENTE>>](#)

¿Qué tipo de combustible usa tu coche?

Diesel
Gasolina

[paso2.php](#) [<<VOLVER](#) [SIGUIENTE>>](#)

¿Cuándo lo compraste?

MES	enero	febrero	marzo	abril
mayo	junio	julio	agosto	
septiembre	octubre	noviembre	diciembre	

[paso3.php](#) [<<VOLVER](#) [SIGUIENTE>>](#)

Te enviaremos los resultados al email que introduzcas

Email:

[paso4.php](#) [<<VOLVER](#) [ENVIAR>>](#)

Fin
El presupuesto ha sido enviado a tu email

[fin.php](#) [CONFIGURAR OTRO COCHE>>](#)

El usuario podrá volver hacia atrás desde cada paso; mostrándole los valores de sesión que había introducido previamente por si desea modificarlos.

Lenguaje PHP: Cookie de Sesión

- **Cookie de sesión:**

- Se crea al iniciar una sesión para almacenar el ID de la sesión.
- Tiene el mismo nombre que la sesión: PHPSESSID o el indicado en `session_name('nombre')`.
- No almacena datos del usuario (eso se hace en el server usando variables de sesión).
- Se destruye cuando se termina la sesión (cierra el navegador).



- Podemos consultar los valores de la cookie creada para la sesión mediante:

`$array = session_get_cookie_params();` Índices del array asociativo devuelto:

- 'lifetime': tiempo de duración de la cookie (Segs). 0 para que caduque al cerrar la sesión
- 'path': ruta donde almacenar la cookie en el cliente (navegador)
- 'domain': nombre de dominio de la página que la ha creado
- 'secure': 1 → solo se puede crear en conexiones seguras
0 → en cualquier tipo de conexión



Lenguaje PHP: Sesiones

- Los **valores por defecto** para crear las **cookies de sesión** de las páginas almacenadas en Apache se definen en **php.ini** como: **session.cookie_xxx**
- Podemos **establecer otros valores a la cookie** creada para la sesión mediante:

session_set_cookie_params(duración, path, dominio, segura);

Ajustando el parámetro *duración* podemos hacer que la sesión termine antes de cerrar el navegador; o que ésta perdure aún habiéndolo cerrado.

- **Ejemplo:**

```
<html> <head> <title> Sesión que expira en breve </title> </head> <body>
<?php
    session_set_cookie_params(10, "/", "localhost", 0);
    session_name('sesión_que_expira');
    session_start(); // inicia la sesión o sigue con ella
    echo "<h4> Datos de la sesión actual: </h4>";
    echo "ID: ".session_id();
    echo "<br> Nombre: ".session_name()."<br>";
    echo "Parámetros de la cookie de sesión:";
    $parametrosCookie = session_get_cookie_params();
    foreach ($parametrosCookie as $clave=>$valor)
        echo "<br>&nbsp;&nbsp;".$clave." = ".$valor;
    echo "<br> <br> Estos datos se perderán en 10 segundos o al cerrar la página";
    echo "<br> Se creará una nueva sesión (nuevo ID) al abrir la página otra vez."
?
<hr> <br> <a href="<?php echo $_SERVER['PHP_SELF']; ?>">RECARGAR LA PÁGINA</a> </body> </html>
```

← Los parámetros de la cookie de sesión deben ajustarse antes de llamar a **session_start()**



Modifica los parámetros de la cookie de sesión para que ésta **perdure aún habiendo cerrado** el navegador (**keep alive**) → crearla con una duración/caducidad prolongada.

Lenguaje PHP: Sesiones

- **¿Cómo saber** (desde cualquier página de una aplicación web) **si una sesión ha expirado?**

En cada página del sitio se debe comenzar comprobando si existe o no en el equipo del cliente la cookie con el nombre de la sesión:

```
if (!isset($_COOKIE['PHPSESSID'])) { // si no existe la cookie → la sesión ha expirado
    header("Location: index.php"); // redirige al usuario hacia la pág principal (login)
}
```

- **¿Cómo usar sesiones si el cliente tiene las cookies deshabilitadas?**

Para poder usar sesiones sin crear cookies que almacenen el ID, debemos propagar el ID de la sesión a través de las URLs que usemos para enlazar a otras páginas del sitio (parecido a la cadena de parámetros de GET):

```
<a href="siguiente.php?<?php echo session_name()."=".session_id(); ?>" >
SIGUIENTE
</a>
```

Si deseamos que el navegador no guarde las respuestas del servidor en su caché local; antes de ejecutar `session_start()`, llamaríamos a:
`session_cache_limiter('nocache,private');`

Lenguaje PHP: Sesiones

- ¿Cómo saber (desde cualquier página de una aplicación web) si estamos continuando con una sesión previa o si se trata de una nueva sesión?

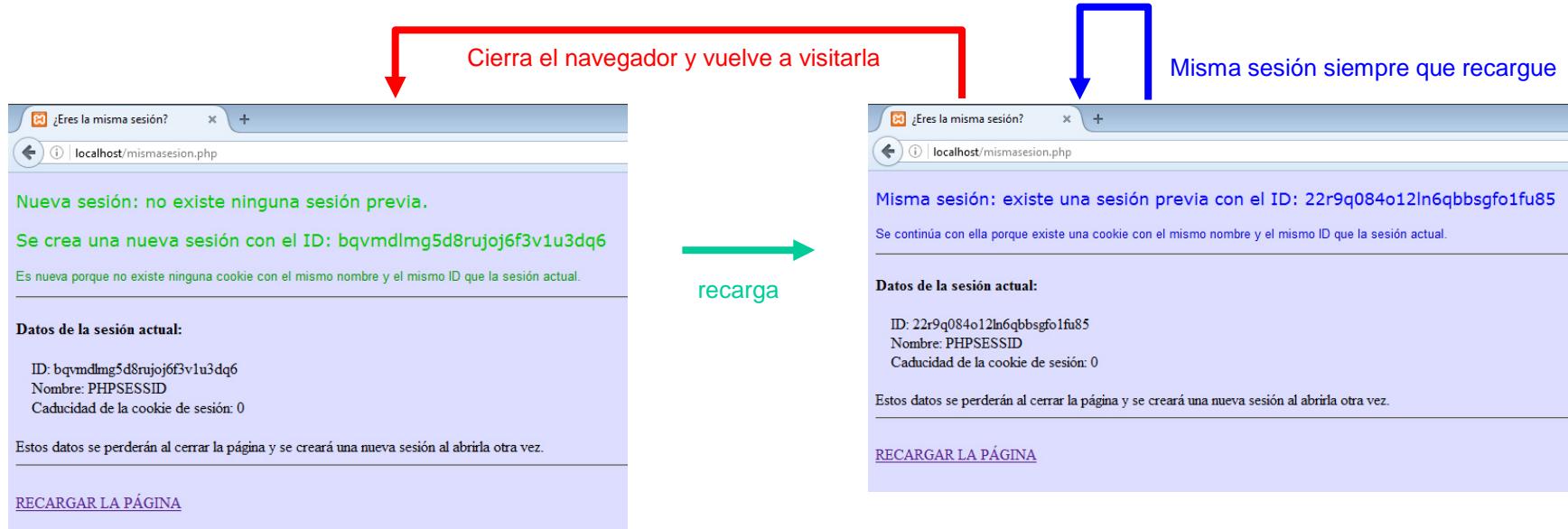
Consultando si existe en el cliente la cookie de sesión. Y si existe, comprobando si el ID de sesión que almacena coincide con el ID de la sesión actual que devuelve el servidor:

- **Ejemplo:**

```
<!DOCTYPE html> <html> <head> <title> ¿Es la misma sesión? </title> </head> <body bgcolor="#DDDDFF">
<?php
    if (isset($_COOKIE['PHPSESSID'])) // Si existe la cookie de la sesión -> leer el ID que contiene
        $IDanterior = $_COOKIE['PHPSESSID'];
    else
        $IDanterior = "";
    session_start(); // Inicia la sesión si no existe o prosigue con ella
    $IDactual = session_id();
    if ($IDanterior == $IDactual) { // Comprobar si el ID actual es el mismo ID que había en la cookie
        echo "<p style=\"color:rgb(0,0,255); font-family:verdana; font-size:18px;\">
Misma sesión: existe una sesión previa con el ID: ".$IDanterior."</p>";
        echo "<p style=\"color:rgb(0,0,200); font-family:arial; font-size:13px;\">
Se continúa con ella porque existe una cookie con el mismo nombre y el mismo ID
que la sesión actual.</p>";
    } else { // Sesión nueva
        echo "<p style=\"color:rgb(0,200,0); font-family:verdana; font-size:18px;\">
Nueva sesión: no existe ninguna sesión previa.</p>";
        echo "<p style=\"color:rgb(0,200,0); font-family:verdana; font-size:18px;\">
Se crea una nueva sesión con el ID: ".$IDactual."</p>";
        echo "<p style=\"color:rgb(0,150,0); font-family:arial; font-size:13px;\">
Nueva porque no existe la cookie con el mismo nombre e ID que la actual.</p>";
    }
}
```

Lenguaje PHP: Sesiones

```
echo "<hr>";
echo "<h4> Datos de la sesión actual: </h4>";
echo "&nbsp; &nbsp; ID: ".session_id();
echo "<br> &nbsp; &nbsp; Nombre: ".session_name()."<br>";
$cookie = session_get_cookie_params();
echo "&nbsp; &nbsp; Caducidad de la cookie de sesión: ";
echo $cookie['lifetime'];
echo "<br> <br> Estos datos se perderán al cerrar la página y se creará
una nueva sesión al abrirla otra vez.";
?>
<hr> <br>
<a href="=php echo $_SERVER['PHP_SELF']; ?&gt;"&gt; RECARGAR LA PÁGINA &lt;/a&gt;
&lt;/body&gt;
&lt;/html&gt;</pre
```



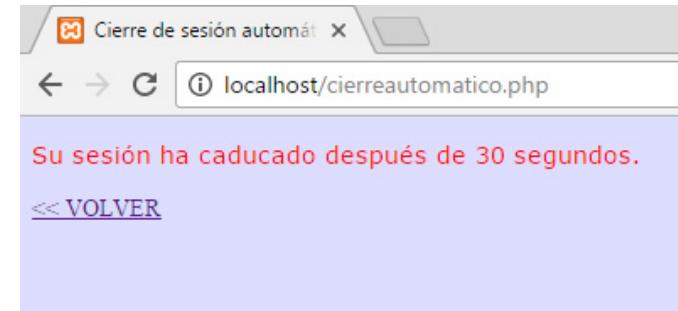
¿Para qué crees que podría ser útil saber si estamos continuando con una sesión previa o si se trata de otra nueva?

Lenguaje PHP: Cierre Automático de Sesiones

- **Actividad:** Desarrollar una aplicación web (cierreautomatico.php) que cree una sesión para el usuario cuando éste la visite y le muestre un vídeo de YouTube (autoplay). Al cabo de 30 segundos le deberá cerrar la sesión automáticamente, redirigiéndole a una página que cuenta con un enlace al inicio (cierreautomatico.php):



Al cabo de 30 seg
se cierra sola



Lenguaje PHP: Sesiones

- Algoritmo a implementar:

1. Al comenzar el script se comprueba si existe la cookie de sesión (si aún no se ha llegado a los 30 seg); es decir, si existe la sesión:

```
if (!isset($_COOKIE['PHPSESSID'])) {  
    // si no existe la cookie de sesión → la sesión ha expirado → redirige al usuario hacia  
    // la pág final, que destruye la sesión y tiene un enlace a la de inicio  
    header("Location: end.php");  
}
```

2. Crear la cookie de sesión configurada para que se borre (cierra la sesión) a los 30 seg:

```
session_set_cookie_params(120, "/", "localhost", 0);  
session_start();
```

- 3.1. Si es la primera vez que se visita la web (no existen las variables de sesión), se debe guardar en una variable de sesión la hora de visita.

- 3.2. Si no es la primera vez (sí existen las variables de sesión), se calcula (en cada recarga) el tiempo de sesión que ha transcurrido (comparando la hora actual con la de inicio de sesión) y:

- 4.1. Si éste es menor que los 30 seg, se recargará la página

- 4.2. Si excede los 30 seg se debe borrar la sesión y recargar la página

Lenguaje PHP: Sesiones

Archivo **cierreautomatico.php**

```
<?php

$X = 30; // Segundos tras los cuales se cerrará automáticamente la sesión
$mensaje = ""; $tiempoTranscurrido = 0; $finSesion = false;
if (!isset($_COOKIE['sesion_que_expira'])) { // Si no existe la cookie de sesión -> sesión caducada
    $finSesion = true;
    $mensaje = "<p style=\"color:red; font-family:verdana;\">Su sesión ha caducado después de $X seg</p>";
    $mensaje .= "<a href=\"cierreautomatico.php\"> << VOLVER </a>";
} else {
    $finSesion = false;
    $mensaje = "<p style=\"color:blue;font-family:verdana;\">Sesión activa. Programada para finalizar a los $X seg</p>";
}

session_set_cookie_params($X, "/", "localhost", 0); // Establece cuándo se cerrará la sesión automáticamente
session_name('sesion_que_expira');
session_start(); // Inicia la sesión si no existe o prosigue con ella
//echo "ID: ".session_id();

if (isset($_SESSION['horaInicio'])) { // Si existe la variable de sesión -> calcular el tiempo de sesión transcurrido
    $horaInicio = $_SESSION['horaInicio'];
    $ahora = date("Y-n-j H:i:s");
    $tiempoTranscurrido = strtotime($ahora) - strtotime($horaInicio);

    if ($tiempoTranscurrido >= $X) { // HA EXPIRADO EL TIEMPO --> CERRAR LA SESIÓN
        $_SESSION = array();
        session_destroy(); // destruyo la sesión
        header("Location: cierreautomatico.php");
    }
}
```

Lenguaje PHP: Sesiones

```
    } else { // NO SE HA CUMPLIDO EL TIEMPO --> RECARGAR LA WEB
        // Descomentar esta instrucción si se quiere RENOVAR X seg MÁS DE ACTIVIDAD cada vez que recargue la
        // web --> la sesión será menos restrictiva: en lugar de cerrarse cuando pasen X segundos después de
        // ser iniciada; SE CERRARÁ SI HAY X seg DE INACTIVIDAD después de la última recarga:
        //      $_SESSION['horaInicio'] = $ahora;

        $tiempoRestante = $X - $tiempoTranscurrido;
        $mensaje .= "<p style=\"color:blue;\">Quedan ".$tiempoRestante." seg para que se cierre la sesión.</p>";
        $mensaje .= "&nbsp;<a href=\"cierreautomatico.php\">RECARGAR CONTINUANDO EL VÍDEO</a> <br> <br>";
        $self = $_SERVER['PHP_SELF'];
        $tiempo = (int)$tiempoRestante + 2;
        header("refresh:$tiempo; url=$self"); // recarga la web para comprobar si ha alcanzado el tiempo
    }
} else { // No existe la sesión --> primera vez que se visita --> guardar la hora
    $_SESSION['horaInicio']= date("Y-n-j H:i:s");
}
?>

<!DOCTYPE html>
<html> <head> <title> Cierre de sesión automático a los X segundos </title> </head>
<body bgcolor="#DDDDFF">
<?php
echo $mensaje;
if (!$finSesion) {
    $paramVideo = "?rel=0&showinfo=0&autoplay=1&start=".$tiempoTranscurrido."\\" frameborder="0\" allowfullscreen";
    echo "<iframe width=\"580\" height=\"390\" src=\"https://www.youtube.com/embed/iueZPRb00ig\"$paramVideo.\"></iframe>";
}
?>
</body>
</html>
```

También se puede optar por refrescar la web a los 30 seg desde el cliente usando JavaScript.
Bastaría con añadir en la sección <head> de cierreautomatico.php el siguiente script:

```
<script>
window.onload =
    setTimeout(
        function(){
            "window.open('cierreautomatico.php', '_self');");
    }
    , 30000);
?>
```

Lenguaje PHP: Cesta de la compra con Sesiones

- **Ejemplo:** web que implementa un **cesta/carrito de la compra** mediante una sesión que almacena un array de productos con las cantidades seleccionadas (cesta.php):



```
<?php
    session_start();

    // Productos en venta:
    $productos[0] = "Portátil X";    $productos[1] = "Móvil Y";      $productos[2] = "Tablet Z";

    if (isset($_SESSION['cesta']))
        $cesta = $_SESSION['cesta'];    // si existe la sesión, carga la cesta
    else
        $cesta = array(0, 0, 0);        // si no existe la sesión pone a 0 todos los productos

    if ($_SERVER['REQUEST_METHOD'] == "POST") {    // incrementa/decrementa la cantidad del producto
        $operacion = $_POST['boton'];
        $producto = $_POST['producto'];
        if ($operacion == "+")
            $cesta[$producto]++;
        else {
            if ($cesta[$producto] != 0)
                $cesta[$producto]--;
        }
        $_SESSION['cesta'] = $cesta;    // y guarda el nuevo valor
                                        // en la variable de sesión
    }
?>
```

\$cesta[] es el array que se guarda como variable de sesión.
\$cesta[0] contiene la cantidad de \$productos[0] seleccionados.

A screenshot of a web browser window titled "Cesta de la compra". The address bar shows "localhost/cesta.php". The page content is titled "CESTA DE LA COMPRA" and contains the instruction "Seleccione sus productos:". Below this is a table with three rows, each representing a product with a quantity input field and +/- buttons. The table has two columns: "Producto" and "Cantidad".

Producto	Cantidad
Portátil X	2
Móvil Y	1
Tablet Z	3

Lenguaje PHP: Sesiones

```
<html>
    <head> <title> Cesta de la compra </title>
        <style>
            .principal { font-family: arial; font-size: 20px; color: #0022DD; font-weight: bold; }
            .titulo { font-family: verdana; font-size: 15px; color: #0000DD; }
        </style>
    </head>
    <body bgcolor="#DDDDFF">
        <p class="principal">CESTA DE LA COMPRA</p>
        <p class="titulo">Seleccione sus productos:</p>

        <table class="titulo" style="border: 1px solid black;" border="0" cellpadding="10">
            <tr><th>Producto</th><th>Cantidad</th><th></th></tr>
        <?php
            foreach ($productos as $posicion=>$nombre) {
                echo "<tr><td>".$nombre."</td>";
                echo "<td>".$cesta[$posicion]."</td><td>"; // escribe el valor de la variable de sesión
                echo "<form method=\"post\" action=\"\">";
                    echo "<input type=\"submit\" name=\"boton\" value=\"+\">";
                    echo "<input type=\"submit\" name=\"boton\" value=\"-\">";
                    echo "<input type=\"hidden\" name=\"producto\" value=\"".$posicion."\">";
                echo "</form>";
                echo "</td></tr>";
            }
        <?
        </table>
    </body>
</html>
```



Lenguaje PHP: Sesiones

- **Actividad:** modifica la aplicación web anterior para cargue los productos a vender a partir del archivo `productos.xml`. Además deberá calcular el importe total de los productos seleccionados en la cesta de la compra. El archivo incluye para cada producto su nombre, imagen y precio:



```
<?xml version = "1.0" encoding = "UTF-8" ?>
<!DOCTYPE Productos [
    <!ELEMENT Productos (Producto*)>
    <!ELEMENT Producto (Nombre, Imagen, Precio)>
    <!ELEMENT Nombre (#PCDATA)>
    <!ELEMENT Imagen (#PCDATA)>
    <!ELEMENT Precio (#PCDATA)>
]>
<Productos>
    <Producto>
        <Nombre>Tablet Samsung View</Nombre>
        <Imagen>http://www.samsung.com/global/galaxy/galaxy-view/images/galaxy-view_gallery_front_black_s3.png</Imagen>
        <Precio>640</Precio>
    </Producto>
    <Producto>
        <Nombre>Móvil iPhone7</Nombre>
        <Imagen>https://www.vodafone.es/img/dynamique/LARGE/a89e671b092c47c8b4c046bbcf74f1bd_275x385_4G_1.png</Imagen>
        <Precio>750</Precio>
    </Producto>
    <Producto>
        <Nombre>Portátil HP Pavilion</Nombre>
        <Imagen>http://d243u7pon29hni.cloudfront.net/images/products/Portatil_HP_Pavilion_15-P139NS_1Tb_ad_1.png</Imagen>
        <Precio>550</Precio>
    </Producto>
</Productos>
```

Archivo `productos.xml`

Cesta de la compra

localhost/cestatotal.php

CESTA DE LA COMPRA

Seleccione sus productos:

Producto	Precio	Cantidad	
Tablet Samsung View	640	2	[+][-]
Móvil iPhone7	750	1	[+][-]
Portátil HP Pavilion	550	1	[+][-]

Importe total de los productos seleccionados: **2580**

Lenguaje PHP: Sesiones

```
<?php
    session_start();
    // Carga todos los productos contenidos en el archivo XML en la matriz $productos:
    $Productos = simplexml_load_file("productos.xml") or die("No se encuentra el archivo productos.xml");
    $i = 0;
    foreach ($Productos->Producto as $Producto) {
        $productos[$i][0] = $Producto->Nombre;
        $productos[$i][1] = $Producto->Imagen;
        $productos[$i][2] = $Producto->Precio;
        $i++;
    }
    if (isset($_SESSION['cesta'])) { // Recupera la cesta si existe
        $cesta = $_SESSION['cesta'];
    } else { // Cesta vacía: inicializa todos los productos de la cesta con la cantidad 0
        $cesta = array();
        for ($i = 0; $i < count($productos); $i++)
            array_push($cesta, 0);
    }
    if ($_SERVER['REQUEST_METHOD'] == "POST") {
        $operacion = $_POST['boton'];
        $producto = $_POST['producto']; // recupera el valor del campo hidden con la posición
        if ($operacion == "+")
            $cesta[$producto]++;
        else {
            if ($cesta[$producto] != 0) // evita cantidades negativas para los productos
                $cesta[$producto]--;
        }
        $_SESSION['cesta'] = $cesta; // actualiza el valor del array $cesta
    }
?>
```

Lenguaje PHP: Sesiones

```
<html> <head> <title> Cesta de la compra </title>
<style>
    .principal { font-family: arial; font-size: 20px; color: #0022DD; font-weight: bold; }
    .titulo { font-family: verdana; font-size: 15px; color: #0000DD; }
</style>
</head>
<body bgcolor="#AADDFF">
    <p class="principal">CESTA DE LA COMPRA</p>
    <p class="titulo">Seleccione sus productos:</p>
    <table class="titulo" style="border: 1px solid black;" border="0" cellpadding="10">
        <tr><th> </th><th>Producto</th><th>Precio</th><th>Cantidad</th><th> </th></tr>
    <?php
        for ($i = 0; $i < count($productos); $i++) {
            echo "<tr><td><img src=\"".$productos[$i][1]."\\" width=\"64\" height=\"64\"></td>";
            echo "<td>".$productos[$i][0]."</td>";
            echo "<td>".$productos[$i][2]."</td>";
            echo "<td>".$cesta[$i]."</td><td>";
            echo "<form method=\"post\" action=\"\">";
            echo "<input type=\"submit\" name=\"boton\" value=\"+\">";
            echo "<input type=\"submit\" name=\"boton\" value=\"-\">";
            echo "<input type=\"hidden\" name=\"producto\" value=\"".$i."\">";
            echo "</form>";
            echo "</td></tr>";
        }
    ?>
    </table>
    <br>
    <span class="titulo">Importe total de los productos seleccionados:</span>
```

El campo **hidden** permite recuperar la posición del elemento de la cesta cuya cantidad se ha cambiado con el botón + ó -

Lenguaje PHP: Sesiones

```
<?php  
    // calcula el importe total de la cesta multiplicando el número de unidades  
    // de cada producto por el valor de su precio almacenado que está almacenado  
    // en la misma posición del array $productos  
  
    if (isset($_SESSION['cesta'])) {  
        $cesta = $_SESSION['cesta'];  
  
        $importe = 0;  
  
        for ($i = 0; $i < count($cesta); $i++) {  
            $importe += $cesta[$i] * $productos[$i][2];  
        }  
    } else {  
        $importe = 0;  
    }  
  
    echo "<span class=\"principal\">".$importe."</span>";  
?  
  
</body>  
</html>
```

\$cesta es un array que se guarda como variable de sesión.

\$cesta[0] contiene la cantidad de \$productos[0] seleccionados.

El precio de cada producto se encuentra en \$productos[][][2]

Lenguaje PHP: Sesiones y Login

- Cuando un usuario se *loguea* en una aplicación web, es deseable asociar/ligar la sesión que se va a crear con su nombre de usuario.
- Para crear la sesión debe ocurrir la **autentificación** exitosa del usuario: las credenciales aportadas por el usuario (su nombre y su clave) coinciden con las que hay registradas en la BD de la aplicación.

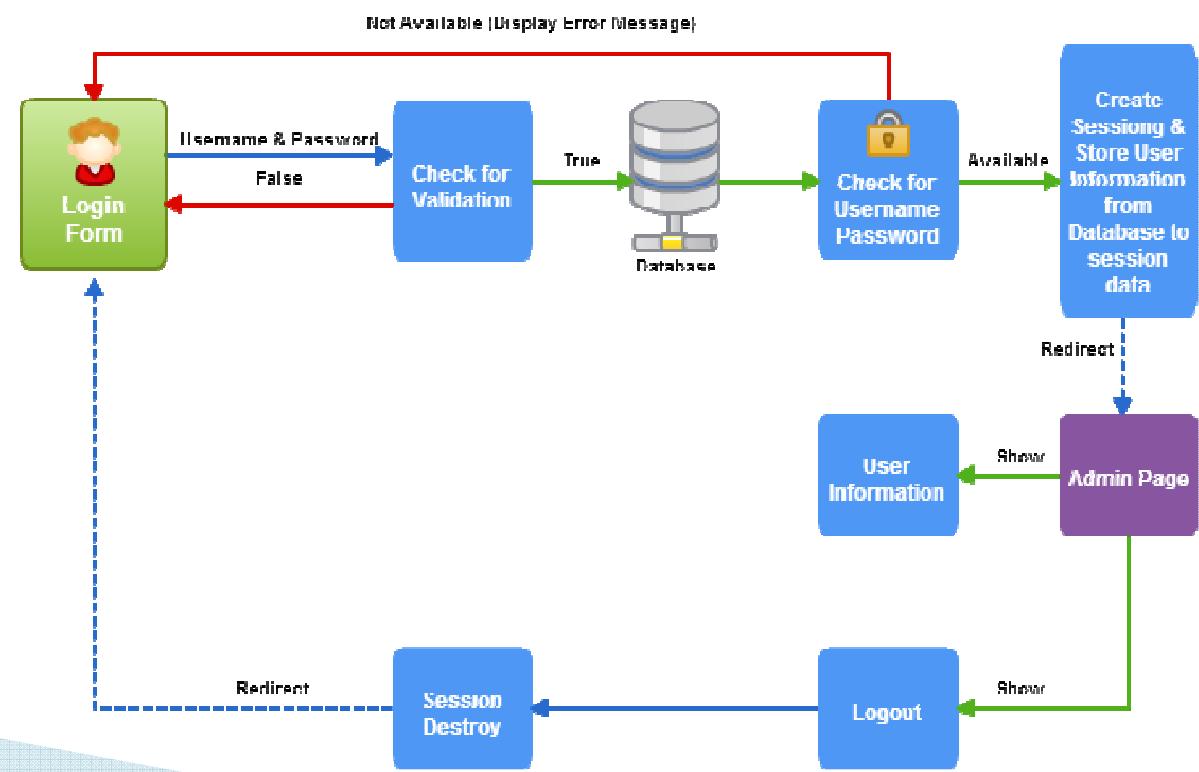
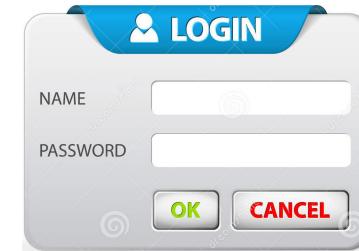


Diagrama de Navegabilidad de una aplicación web con inicio de sesiones:

Lenguaje PHP: Sesiones y Login

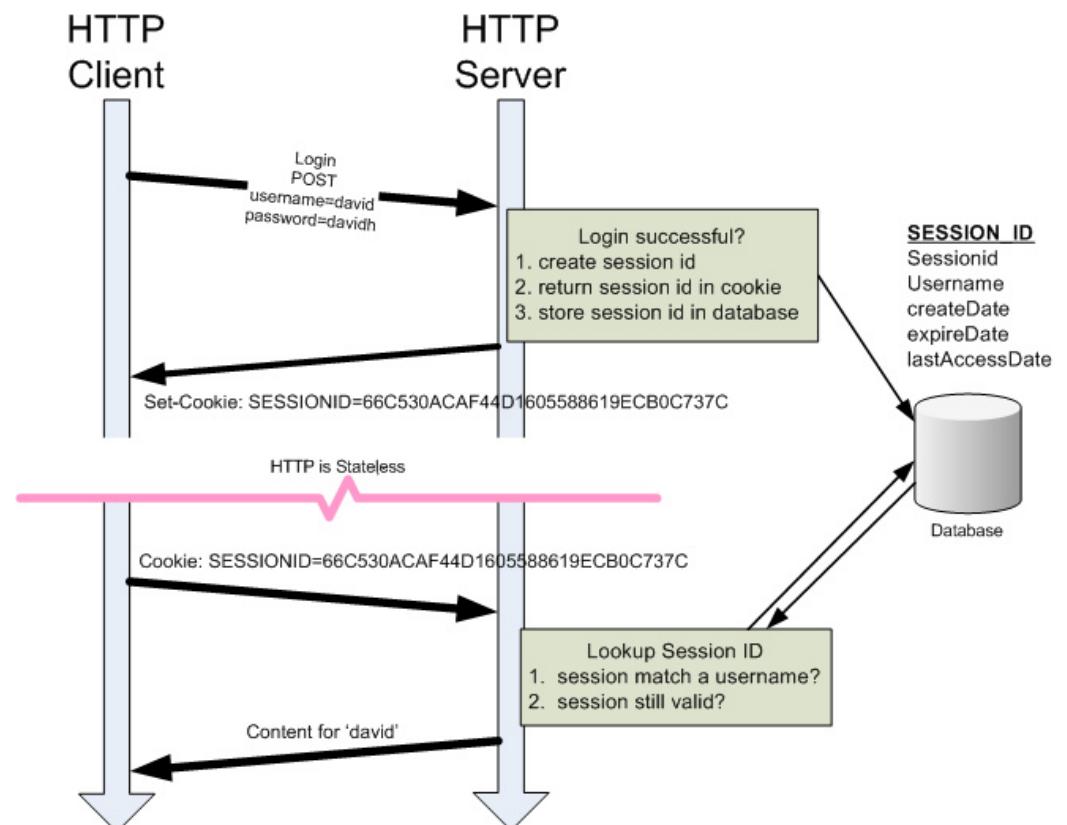
- Cada sesión es **independiente para cada usuario** y **cada usuario es único** en la aplicación web:

- Si asignamos a la sesión el mismo nombre que el del login de usuario: en el mismo cliente (navegador) será posible iniciar sesión en la web con varios usuarios distintos a la vez.

- Si no le asignamos un nombre a la sesión (**PHPSESSID**), solo se podrá iniciar sesión en el cliente con una única cuenta al mismo tiempo (todas las sesiones serán la misma, si no se usan distintos navegadores).



PHP Login Form with Session



Lo habitual es crear una **variable de sesión** con el **username** del usuario que acaba de iniciar sesión

Lenguaje PHP: Sesiones

- **Ejemplo:** sitio web con página de login, página principal y logout:

Archivo **login.php**

```
<?php
    session_start();
?>
<html> <head> <title>Login</title> </head> <body bgcolor="#BBDDAA">
<?php
    $error = "";
    if ($_SERVER['REQUEST_METHOD'] == "POST") {
        if (empty($_POST['usuario']) || empty($_POST['password'])) { // error en la validación
            $error = "Debes llenar todos los datos";
        } else {
            if ($_POST['usuario']=="test" && $_POST['password']=="test") { // Login OK
                $_SESSION['usuario'] = $_POST['usuario'];
                header("Location: principal.php");
            } else { // Login erróneo
                $error = "Usuario o clave incorrectos";
            }
        }
    }
?>
<br> <form name="login" method="post" action="">
<table width="300" border="0" cellpadding="3" cellspacing="4">
    <tr><td>Usuario:</td><td><input name="usuario" type="text" ></td></tr>
    <tr><td>Password:</td><td><input name="password" type="password" ></td></tr>
    <tr align="center"><td colspan="2"><input name="enviar" type="submit" value="ACCEDER" >
    </td></tr> </table> </form>
<?php echo $error; ?>
</body> </html>
```

La aplicación web asocia la sesión al nombre del **usuario** mediante una **variable de sesión** y la propaga entre las páginas del sitio

Lenguaje PHP: Sesiones

Archivo principal.php

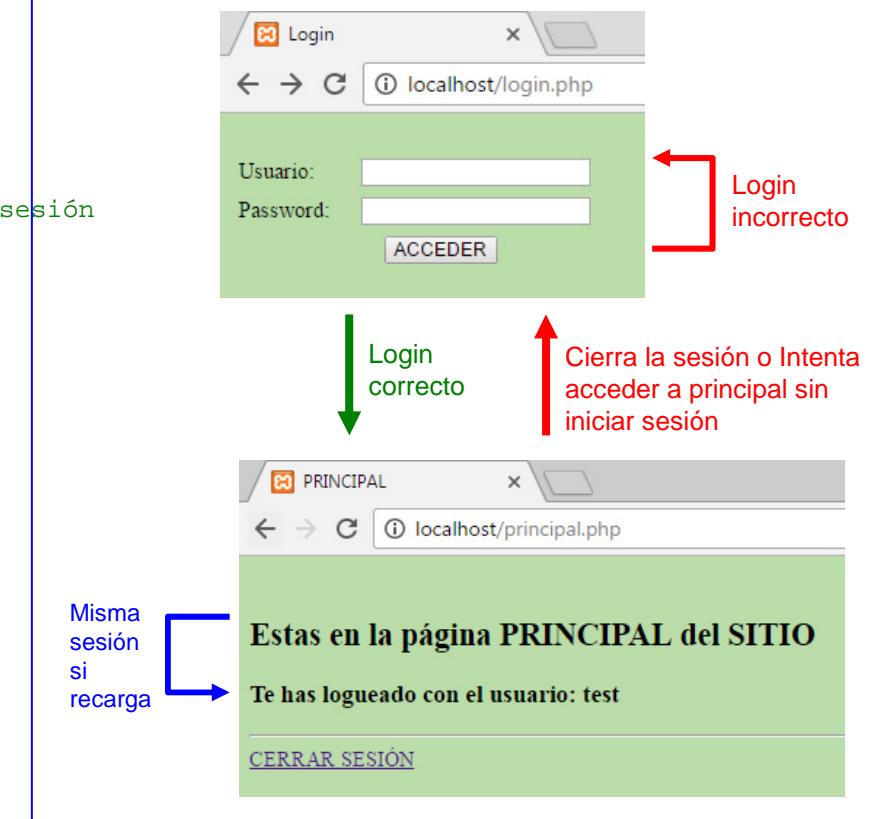
```
<?php  
    session_start();  
    $usuario = "";  
    if (isset($_SESSION['usuario'])) { //existe sesión  
        $usuario = $_SESSION['usuario'];  
    } else { // no existe la sesión  
        header("Location: login.php");  
    }  
?  
  
<head>  
<title>PRINCIPAL</title>  
</head>  
<body bgcolor="#BBDDAA">  
    <br>  
    <h2>Estás en la página PRINCIPAL del SITIO</h2>  
    <h3>Has iniciado sesión con el usuario:  
        <?php echo $usuario; ?>  
    </h3>  
    <hr>  
    <a href="logout.php">CERRAR SESIÓN</a>  
</body>  
</html>
```

<?php

?>

Archivo logout.php

```
session_start();  
if (isset($_SESSION['usuario'])) {  
    unset($_SESSION['usuario']);  
  
    setcookie(session_name(), '', time() - 10, '/');  
    session_destroy();  
}  
header("Location: login.php");
```



Lenguaje PHP: Sesiones

- **Actividad:** modifica el programa del ejemplo anterior para que autentifique a los usuarios comprobando sus credenciales que están almacenadas en un archivo XML (usuarios.xml) que debes crear.

Una vez logueado el usuario, se le debe dar acceso a la página principal.php mostrándole:

- Su nombre y password ingresadas para iniciar la sesión.
- Su hora de inicio de sesión y la dirección IP desde la que se conecta.
- Una lista con tres colores que el usuario puede elegir para aplicar como fondo a la página principal (form con 3 radiobuttons y un submit). El color elegido se almacenará en una cookie de forma que se cargue al iniciar la sesión el usuario. Si no existe la cookie se tomará el color #BBDDAA.
- Un enlace para el cierre de sesión.



El color de fondo se guarda en una cookie para ese usuario → el nombre de la cookie debe tener el nombre del usuario pq si la llamamos "color" se usaría la misma para distintos usuarios que se logueen usando el mismo equipo y navegador.

Lenguaje PHP: refresco y redirección a otras páginas

- Refrescar una web por medio de código PHP:

```
<?php
    if ($_SERVER['REQUEST_METHOD'] == "POST") {
        $self = $_SERVER['PHP_SELF'];
        header("Location: $self"); // refrescar es redirigir hacia la misma web
    }
?>

<html> <head> <title> Refresco </title> </head>
<body bgcolor="#AA2299">
    <br><h3>Soy un ejemplo de web que se refresca al pulsar el botón.</h3>
    <form name="formulario" method="post" action="">
        <input type="submit" name="refrescar" value="REFRESCAR WEB">
    </form>
    <br>
    <?php echo "Hora actual: ".date("H:i:s"); ?>
</body></html>
```

- Refrescar una web cada cierto tiempo de manera automática con PHP:

```
<?php
    $time = 2; // segundos
    $self = $_SERVER['PHP_SELF'];
    header("refresh:$time; url=$self");
?>

<html> <head> <title> Refrescos </title> </head>
<body bgcolor="#AA2299">
    <br><h3>Soy un ejemplo de web que se refresca cada <?php echo $time; ?> segundos.</h3>
    <br>
    <?php echo "Hora actual: ".date("H:i:s"); ?>
</body></html>
```

Lenguaje PHP: refresco y redirección a otras páginas

- Redirigir hacia otra web por medio de código **PHP**:

```
<?php
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $url = "http://www.cifpvilladeaguimes.es";
    header("Location: $url");
}
?>

<html> <head> <title> Redirección </title> </head>
<body bgcolor="#442299">
<br><h3>Soy un ejemplo de web que redirige a otra al pulsar el botón.</h3>
<form name="formulario" method="post" action="">
    <input type="submit" name="redirigir" value="REDIRIGIR A LA WEB DEL INSTI">
</form>
</body></html>
```

- Refrescar una web cada cierto tiempo automáticamente mediante **HTML**:

```
<html>
<head>
    <title> Refresco </title>
    <META HTTP-EQUIV="REFRESH" CONTENT="1; URL=refresco.php"> </head>
</head>
<body bgcolor="#AA2299">
<br> <h3>Soy un ejemplo de web que se refresca sola cada segundo.</h3>
<script language="javascript">
    fecha = new Date();
    document.write("Hora actual: " + fecha.getHours() + ":" + fecha.getMinutes() + ":" +
                  fecha.getSeconds());
</script>
</body> </html>
```

Lenguaje PHP: refresco y redirección a otras páginas

- Refrescar una web por medio de código **JavaScript**:

```
<html> <head> <title> Refresco </title> </head>
<script language="javascript">
    function recargar () {
        location.reload();
    }
</script>
<body bgcolor="#AA2299">
<br> <h3>Soy un ejemplo de web que se refresca al pulsar el
    <a href="javascript:recargar()">enlace</a>.</h3>
<script language="javascript">
    fecha = new Date();
    document.write("Hora actual: " + fecha.getHours() + ":" + fecha.getMinutes() + ":" +
                    fecha.getSeconds());
</script>
</body> </html>
```

- Refrescar una web cada cierto tiempo de manera automática con **JavaScript**:

```
<html> <head> <title> Refresco </title> </head>
<script language="javascript">
    function recargar () {
        location.reload();
        alert("Me acabo de recargar");
    }
    setInterval('recargar()', 5000);
</script>
<body bgcolor="#AA2299">
<br> <h3>Soy un ejemplo de web que se refresca sola cada 5 segundos.</h3>
</body> </html>
```

Lenguaje PHP: refresco y redirección a otras páginas

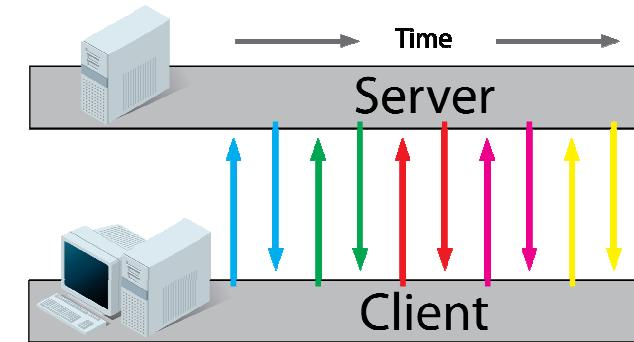
- Redirigir a otra página por medio de código **JavaScript**:

```
<html>
<head>
    <title> Redirección </title>
</head>
<script language="javascript">
    function redirigir() {
        location.href = "http://www.cifpvilladeaguimes.es";
        return;
    }
</script>
<body bgcolor="#AA2299">
    <br>
    <h3>Soy un ejemplo de web que redirige a otra al pulsar el
        <a href="javascript:redirigir()">enlace</a>.
    </h3>
</body>
</html>
```

Lenguaje PHP: comunicación Cliente-Servidor

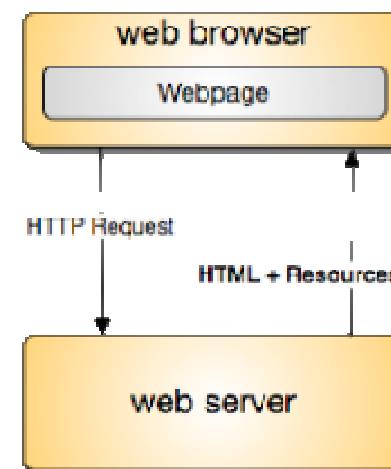
- El protocolo de aplicación HTTP/S presenta incomodidades debido a su comportamiento **stateless**:

Cada visita del cliente al servidor se toma como una conexión aislada e independiente (sin sesiones).

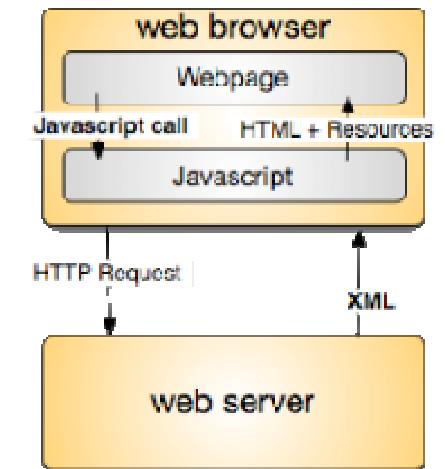


- Además la **comunicación entre cliente y servidor es síncrona**: el intercambio de información entre cliente y servidor sólo tiene lugar **si se recarga toda la web** (mediante GET o POST).

Traditional web model



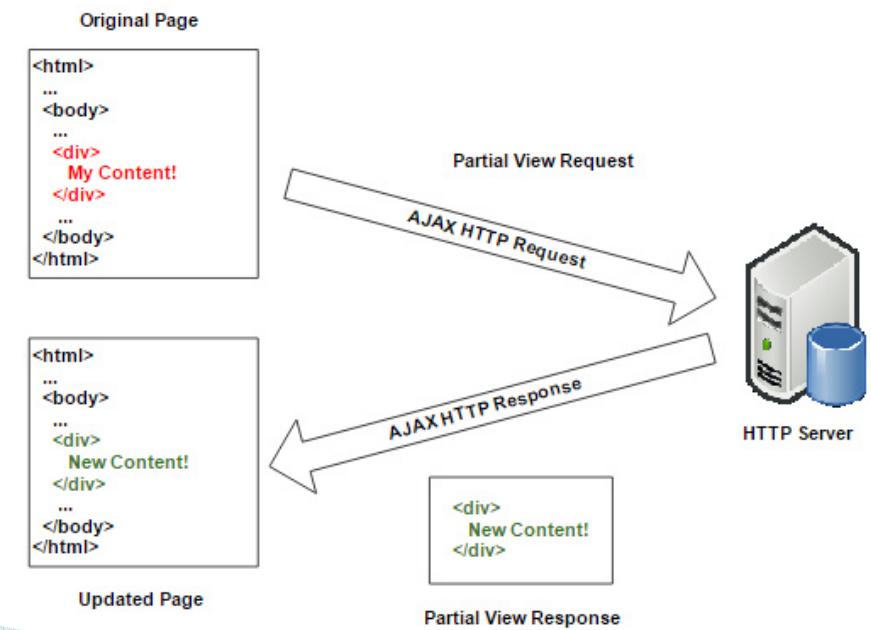
AJAX web model



Lenguaje PHP: actualización parcial de páginas - AJAX

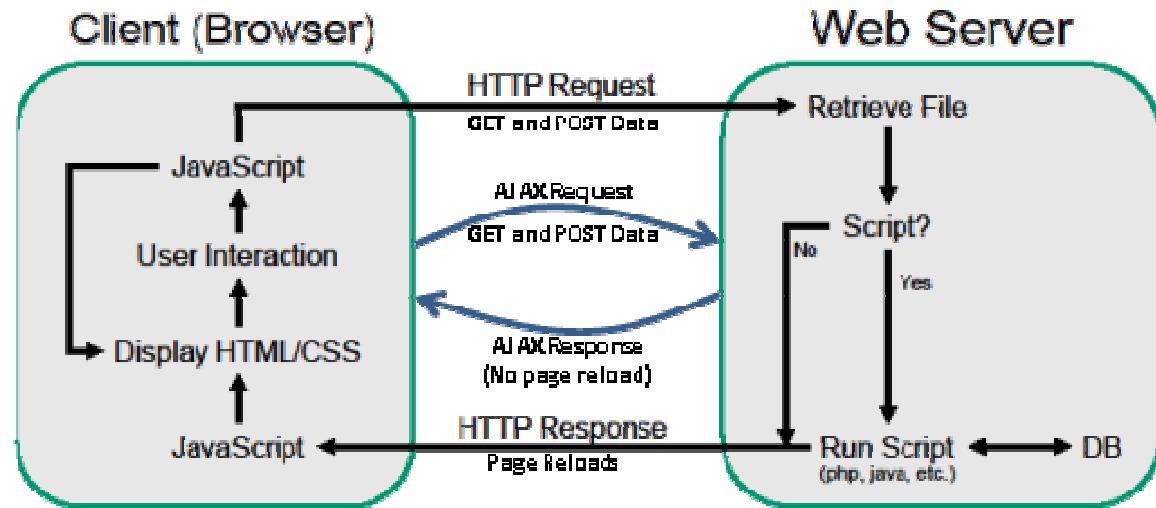
- **AJAX** (Asynchronous JavaScript + XML): consiste en actualizar el contenido de una página accediendo al servidor web pero **sin recargar toda la página**, sólo aquella sección que deseemos.

- **AJAX evita el comportamiento síncrono** de las aplicaciones web:
permite la comunicación asíncrona entre cliente y servidor (no al unísono): el navegador (cliente) se puede comunicar con el servidor sin tener que recargar toda la web
→ permite recargar el contenido de parte/s de una web.

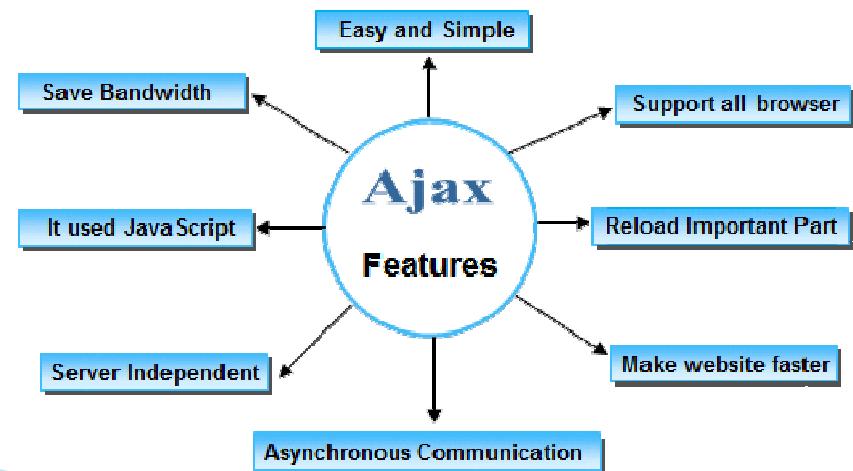


Lenguaje PHP: AJAX

- Recargar toda una web para actualizar únicamente el contenido de una sección es ineficiente.
- AJAX permite realizar un **refresco selectivo** de las secciones de la web cuyo contenido haya cambiado.



AJAX mejora la interactividad, velocidad y amigabilidad de las aplicaciones web



Lenguaje PHP: AJAX

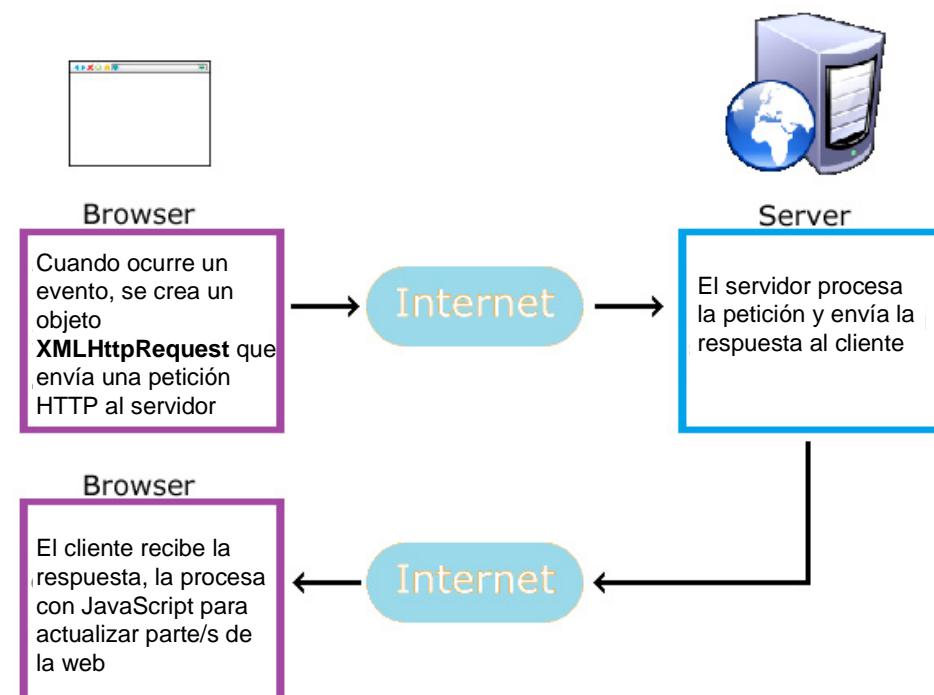
- **Funcionamiento de AJAX:**

1º.- Cuando sucede un **evento** en el cliente: cierto código JavaScript de la web llama a un fichero PHP del lado del servidor para pasarle variables.

2º.- El fichero PHP del lado del servidor genera un código JavaScript de respuesta.

3º.- Este código JavaScript se ejecuta en la página para modificar el código HTML de la página, o bien realizar alguna otra acción.

El objeto XMLHttpRequest permite a los navegadores comunicarse con el servidor sin requerir una recarga completa de la página.



Lenguaje PHP: AJAX

- **Ejemplo:** Aplicación que genera un número aleatorio refrescando sólo la parte de la web donde éste se visualiza (div) → el vídeo no se refresca.

```
<html>
<head> <title>Ejemplo recargar con AJAX</title>
      <script language="JavaScript" type="text/javascript">
          function generar () {
              var nuevoscript = document.createElement("script");
              nuevoscript.src = 'recargar.php';
              nuevoscript.setAttribute('language', 'javascript');
              nuevoscript.setAttribute('type', 'text/javascript');
              document.body.appendChild(nuevoscript);
          }
      </script>
  </head>
<body bgcolor="#AAAAFF" onLoad="generar();">
    <br> <iframe width="560" height="315"
    src="https://www.youtube.com/embed/515Xn_LOEY4?rel=0&autoplay=1" frameborder="0"
    allowfullscreen></iframe> <br>
    <p> Nuevo número aleatorio generado: <p>
        <div id="numero">
            <!-- el contenido es generado por AJAX -->
        </div>
        <br>   <input type="submit" onClick="generar();" value="GENERAR OTRO NÚMERO" >
    </body>
</html>
```

Cuando sucede el **evento** `onClick` en el cliente, el código JavaScript de la web llama a un fichero PHP que a su vez genera código JavaScript para actualizar el contenido de la sección (div) donde se muestra el número → genera un nuevo número.

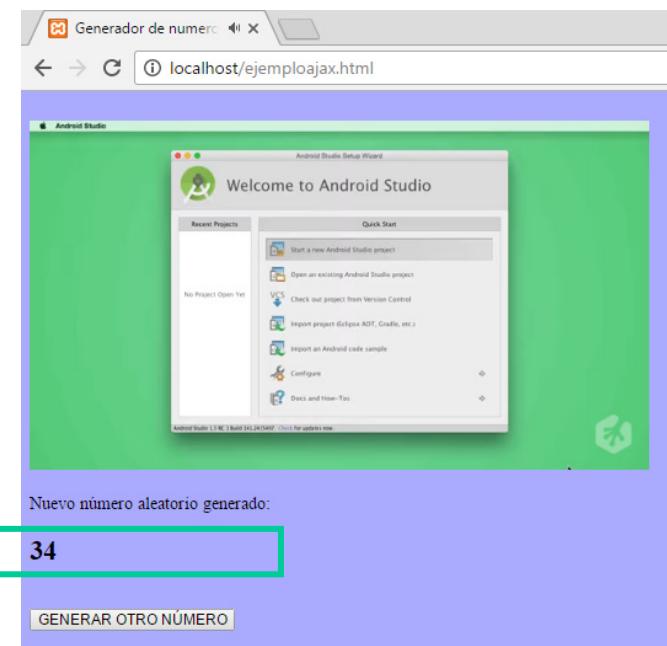
Lenguaje PHP: AJAX

- Archivo **recargar.php**

```
<?php  
    header( "Content-type: text/javascript" );  
?  
  
    <!-- código JavaScript que es enviado al cliente -->  
    document.getElementById("numero").innerHTML=<h2><?php echo rand(1,99);?></h2>;
```



El código JavaScript generado se encarga de actualizar el contenido de la sección (div) donde se muestra el número. El resto de la web permanece igual ;-)



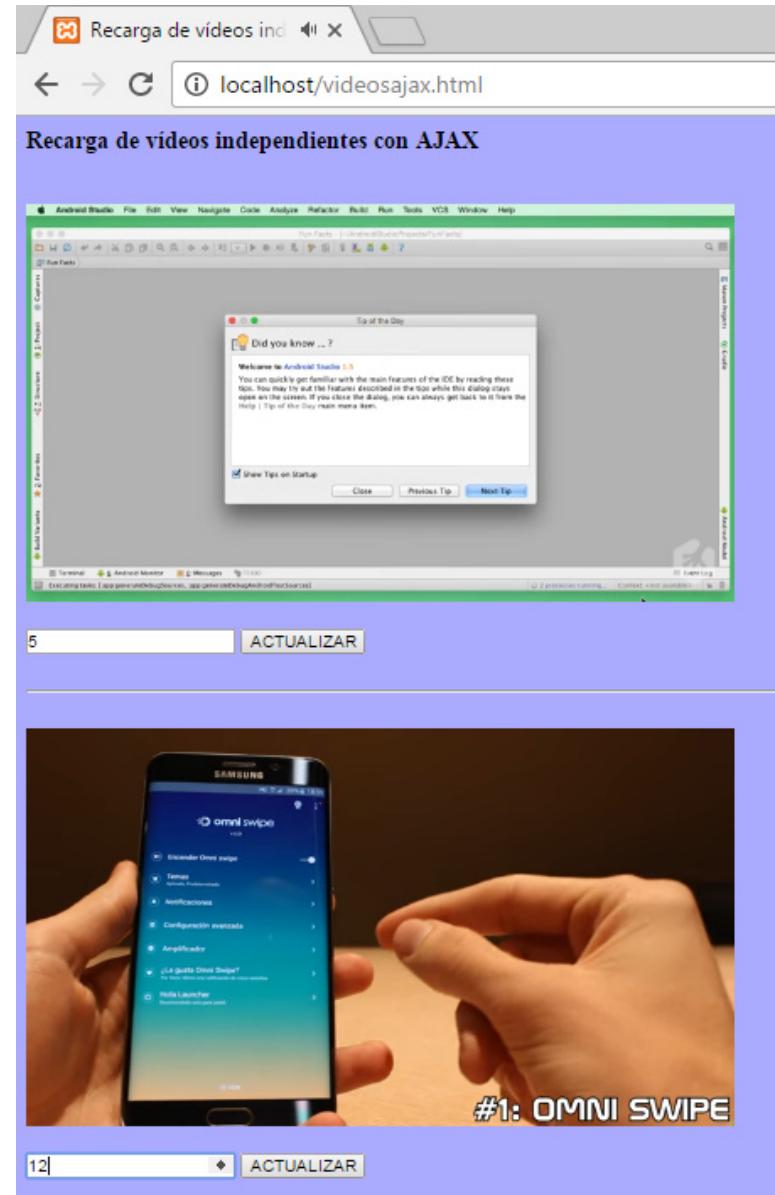
Lenguaje PHP: AJAX

- **Actividad:** crear una aplicación web que muestre dos vídeos diferentes de YouTube embebidos en la propia página.

Cada vídeo contará con un control para introducir el tiempo (en segundos) a partir del cual queremos que se reproduzca.

Si se actualiza el tiempo de comienzo de un vídeo, el otro vídeo debe permanecer reproduciéndose por donde iba.

El código de recarga de ambos vídeos debe implementarse en *un único archivo PHP* que reciba mediante el método GET (en la URL) los parámetros necesarios.



Lenguaje PHP: AJAX

• Archivo **videosAJAX.html**

```
<html> <head> <title>Recarga de v&iacute;deos independientes con AJAX</title>
    <script language="JavaScript" type="text/javascript">
        function recargar (id, tiempo) {
            var nuevoscript = document.createElement("script");
            nuevoscript.src = 'recargarVideos.php?id=' + id + '&tiempo=' + tiempo;
            nuevoscript.setAttribute('language', 'javascript');
            nuevoscript.setAttribute('type', 'text/javascript');
            document.body.appendChild(nuevoscript);
        }
    </script>
</head>
<body bgcolor="#AAAAFF">
    <h3>Recarga de v&iacute;deos independientes con AJAX</h3> <br>
    <div id="video1">
        <iframe width="560" height="315" src="https://www.youtube.com/embed/515Xn_LOEY4?rel=0&autoplay=1"
            frameborder="0" allowfullscreen></iframe>
    </div>
    <br> <input type="number" id="tiempoVideo1" min="0">
    <input type="submit" onClick="recargar('video1',document.getElementById('tiempoVideo1').value);"
        value="ACTUALIZAR">
    <br> <br> <hr> <br>
    <div id="video2">
        <iframe width="560" height="315" src="https://www.youtube.com/embed/voywIbE8P7M?rel=0&autoplay=1"
            frameborder="0" allowfullscreen></iframe>
    </div>
    <br> <input type="number" id="tiempoVideo2" min="0">
    <input type="submit" onClick="recargar('video2',document.getElementById('tiempoVideo2').value);"
        value="ACTUALIZAR">
</body> </html>
```

Para que JavaScript invoque al archivo PHP que recarga pasándole parámetros, se emplea una URL con el método GET: URL?campo=valor&campo=valor...

Lenguaje PHP: AJAX

- Archivo **recargarVideos.php**

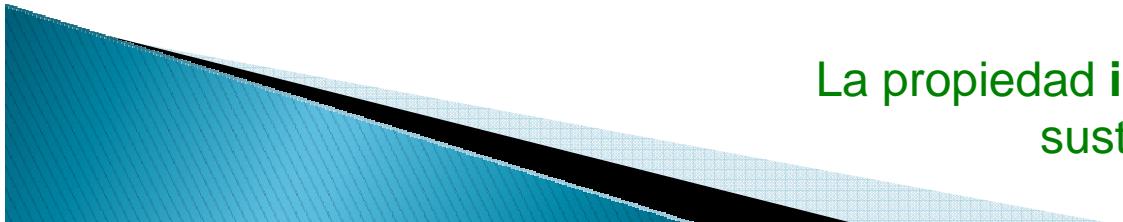
```
<?php
    header("Content-type: text/javascript");

    if (isset($_GET['id'])) {
        // Parámetros enviados en la URL para recargar un vídeo en particular:
        $id = $_GET['id'];
        $tiempo = $_GET['tiempo'];

        $paramVideo = "?rel=0&showinfo=0&autoplay=1&start=".$tiempo."' frameborder='0' allowfullscreen";

        if ($id == 'video1') {
            $video = "<iframe width='560' height='315' src='https://www.youtube.com/embed/515Xn_LOEY4"
                    . $paramVideo . "></iframe>";
        } else {
            $video = "<iframe width='560' height='315' src='https://www.youtube.com/embed/voywIbE8P7M"
                    . $paramVideo . "></iframe>";
        }

        // código JavaScript generado para recargar el vídeo en cuestión y en el segundo indicado:
        echo "document.getElementById(\"" . $id . "\").innerHTML=\"$video.\"";
    }
?>
```



La propiedad **innerHTML** de un elemento del DOM sustituye el contenido HTML que tiene el elemento por el indicado.