

# 1 Curso POO PHP Sesións

## 1.1 Sesións

Como acabas de ver, unha forma para gardar información particular de cada usuario é utilizar **cookies**. Non obstante, existen diversos problemas asociados ás cookies, como o número delas que admite o navegador, ou o seu tamaño máximo. Para solucionar estes inconvenientes, existen as **sesións**. O termo sesión fai referencia ao conxunto de información relativa a un usuario concreto. Esta información pode ser tan simple como o nome do propio usuario, ou máis complexa, como os artigos que depositou na cesta de compra dunha tenda online.

Cada usuario distinto dun sitio web ten a súa propia información de sesión. Para distinguir unha sesión doutra úsanse os identificadores de sesión (SID). Un SID é un atributo que se asigna a cada un dos visitantes dun sitio web e identifica. Desta forma, se o servidor web coñece o SID dun usuario, pode relacionalo con toda a información que posúe sobre el, que se mantén na sesión do usuario. Esa información almacénase no servidor web, xeralmente en ficheiros aínda que tamén se poden utilizar outros mecanismos de almacenamento como bases de datos.

Como xa terás suposto, a cuestión agora é: e onde se almacena ese SID, o identificador da sesión, que é único para cada usuario? Pois existen dúas **maneiras de manter o SID** entre as páxinas dun sitio web que visita o usuario:

- Utilizando cookies, tal e como xa viste.
- Propagando o SID nun parámetro da URL. O SID engádese como unha parte máis da URL, da forma:

```
http://www.misitioweb.com/tienda/listado.php&PHPSESSID=34534fg4ffg34ty
```

No exemplo anterior, o SID é o valor do parámetro PHPSESSID.

Ningunha das dúas maneiras é perfecta. Xa sabes os problemas que ten a utilización de cookies. Malia iso, é o mellor método e o máis utilizado. Propagar o SID como parte da URL leva consigo maiores desvantaxes, como o feito de que compartir a URL con outra persoa implica compartir tamén o identificador de sesión.

A boa noticia, é que o proceso de manexo de sesións en PHP está automatizado en boa medida. Cando un usuario visita un sitio web, non é necesario programar un procedemento para ver se existe un SID previo e cargar os datos asociados con este. Tampouco tes que utilizar a función `setcookie` se queres almacenar os SID en cookies, ou ir pasando o SID entre as páxinas web do teu sitio se te decides por propagalo. Todo isto PHP faíno automaticamente.

Á información que se almacena na sesión dun usuario tamén coñéceselle como cookies do lado do servidor (*server side cookies*). Debes ter en conta que aínda que esta información non viaxa entre o cliente e o servidor, si o fai o SID, ben como parte da URL ou nun encabezado HTTP se se garda nunha cookie. En ambos os dous casos, isto formula un posible problema de seguridade. O SID pode ser conseguido por outra persoa, e a partir deste obter a información da sesión do usuario. O xeito máis seguro de utilizar sesións é almacenando os SID en cookies e utilizar HTTPS para encriptar a información que se transmite entre o servidor web e o cliente.

### 1.1.1 Configuración das sesións en PHP

Por defecto, PHP inclúe soporte de sesións incorporado. Non obstante, antes de utilizar sesións no teu sitio web, debes configurar correctamente PHP utilizando as seguintes directivas no ficheiro `php.ini` segundo corresponda:

Directiva	Significado
<code>session.use_cookies</code>	Indica se se deben usar cookies (1) ou propagación na URL (0) para almacenar o SID
<code>session.use_only_cookies</code>	Débese activar (1) cando utilizas cookies para almacenar os SID, e ademais non queres que se recoñezan os SID que se poidan pasar como parte da URL (este método pódese usar para usurpar o identificador doutro usuario)
<code>session.save_handler</code>	Utilízase para indicar a PHP como debe almacenar os datos da sesión do usuario. Existen catro opcións: en ficheiros (files), en memoria (mm), nunha base de datos SQLite (sqlite) ou utilizando para iso funcións que debe definir o programador (user). O valor por defecto (files) funcionará sen problemas na maioría dos casos
<code>session.name</code>	Determina o nome da cookie que se utilizará para gardar o SID. O seu valor por defecto é PHPSESSID
<code>session.auto_start</code>	O seu valor por defecto é 0, e neste caso deberás usar a función <code>session_start</code> para xestionar o inicio das sesións. Se usas sesións no sitio web, pode ser boa idea cambiar o seu valor a 1 para que PHP active de forma automática o manexo de sesións
<code>session.cookie_lifetime</code>	Se utilizas a URL para propagar o SID, este perderase cando peches o teu navegador. Non obstante, se utilizas cookies, o SID manterase mentres non se destrúa a cookie. No seu valor por defecto (0), as cookies destrúense cando se pecha o navegador. Se queres que se manteña o SID durante máis tempo, debes indicar nesta directiva ese tempo en segundos

Directiva	Significado
<code>session.gc_maxlifetime</code>	Indica o tempo en segundos que se debe manter activa a sesión, aínda que non haxa ningunha actividade por parte do usuario. O seu valor por defecto é 1440. É dicir, pasados 24 minutos dende a última actividade por parte do usuario, péchase a súa sesión automaticamente

### 1.1.2 Xestión das sesións

O inicio dunha sesión pode ter lugar de dúas formas. Se activaches a directiva `session.auto_start` na configuración de PHP, a sesión comezará automaticamente en canto un usuario se conecte ao teu sitio web. En caso de que ese usuario xa abrise unha sesión con anterioridade, e esta non se eliminase, en lugar de abrir unha nova sesión continuarase a anterior. Para iso utilízase o SID anterior, que estará almacenado nunha cookie (se usas propagación do SID, non poderás restaurar sesións anteriores; o SID figura na URL e pérdese cando pechas o navegador).

Se pola contra, decides non utilizar o inicio automático de sesións, deberás executar a función `session_start` para indicar a PHP que inicie unha nova sesión ou continúe a anterior. Aínda que anteriormente esta función devolvía sempre true, a partir da versión 5.3.0 de PHP o seu comportamento é máis coherente: devolve false en caso de non poder iniciar ou restaurar a sesión.

Como o inicio de sesión require utilizar cookies, e estas transmítense nos encabezados HTTP, debes ter en conta que para poder iniciar unha sesión utilizando `session_start`, terás que facer as chamadas a esta función antes de que a páxina web mostre información no navegador. Ademais, todas as páxinas que necesiten utilizar a información almacenada na sesión, deberán executar a función `session_start`.

Mentres a sesión permanece aberta, podes utilizar a variable superglobal `$_SESSION` para engadir información á sesión do usuario, ou para acceder á información almacenada na sesión. Por exemplo, para contar o número de veces que o usuario visita a páxina, podes facer:

```
<?php
// Iniciamos a sesión ou recuperamos a anterior sesión existente
session_start();
// Comprobamos se a variable xa existe
if (isset($_SESSION['visitas'])) $_SESSION['visitas']++;
else $_SESSION['visitas'] = 0;
?>
```

Se en lugar do número de visitas, quixeses almacenar o instante en que se produce cada unha, a variable de sesión "visitas" deberá ser un array e polo tanto terás que cambiar o código anterior por:

```
<?php
// Iniciamos a sesión ou recuperamos a anterior sesión existente
session_start();
// En cada visita engadimos un valor ao array "visitas"
$_SESSION['visitas'][] = mktime();
?>
```

Aínda que como xa viste, podes configurar PHP para que elimine de forma automática os datos dunha sesión pasado certo tempo, en ocasións pode ser necesario pechala de forma manual nun momento determinado. Por exemplo, se utilizas sesións para recordar a información de autenticación, deberás darlle ao usuario do sitio web a posibilidade de pechar a sesión cando o crea conveniente.

En PHP tes dúas funcións para eliminar a información almacenada na sesión:

- **`session_unset`**. Elimina as variables almacenadas na sesión actual, pero non elimina a información da sesión do dispositivo de almacenamento usado.
- **`session_destroy`**. Elimina completamente a información da sesión do dispositivo de almacenamento.

--Víctor Lourido 11:40 15 jul 2013 (CEST)