Sesiones en PHP

Área de Ingeniería Telemática

Contenido

- Estado de la conexión con el navegador
- Cookies
- Sesiones

Estado de la conexión con el navegador

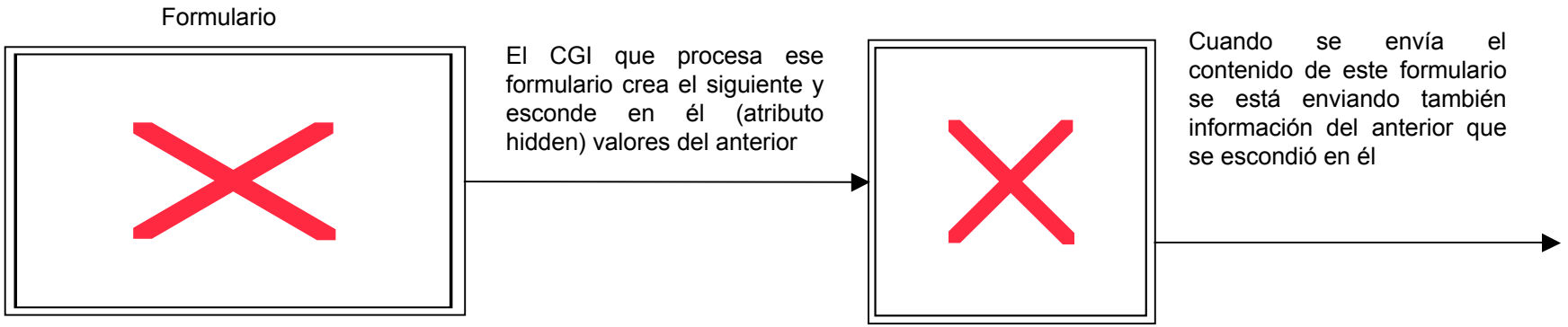
- Cuando el usuario solicita una página que es en verdad un PHP éste empieza a ejecutarse, manteniendo la conexión establecida para poder mandar el resultado del script
- Puede que el usuario aborte la conexión (botón STOP en el navegador)
- Para reconocer esta circunstancia un script PHP puede encontrarse en diferentes estados:
  - NORMAL: Mientras el script se ejecuta con normalidad como se ha descrito
  - ABORTED: Si el usuario corta la conexión el script pasa a este estado
  - TIMEOUT: Se puede configurar un máximo tiempo que puede ejecutarse el script, si se alcanza este tiempo pasa a este estado

Estado de la conexión con el navegador

- Normalmente cuando el usuario corta la conexión y el script pasa al estrado ABORTED termina abruptamente la ejecución del script
- Se puede cambiar este modo de funcionamiento para que los scripts se ejecuten siempre hasta finalizar (por ejemplo llamando a la función `ignore_user_abort()`)
- El tiempo máximo típico que está configurado que pueda ejecutarse un script sin ser abortado por TIMEOUT es de 30 segundos pero puede cambiarse por ejemplo con la función `set_time_limit()`

Persistent Client State HTTP Cookies

- Una de las limitaciones de la Web a la hora del desarrollo de aplicaciones/interfaces es que su funcionamiento es sin estado
- Cada petición de un URI es independiente de los anteriores y hay poca o ninguna información de lo que ha hecho el usuario anteriormente
- Eso quiere decir que formularios que ocupen varias páginas HTML son difíciles de implementar
- Un técnica clásica ha sido, al generar un CGI una página como resultado de un formulario esconder en esa página, en el siguiente formulario, la información que se quiere conservar...



Persistent Client State HTTP Cookies

- Las Cookies son el mecanismo más cómodo para almacenar información de estado en el cliente
- Al enviar una página Web el servidor puede indicar al cliente que almacene cierta información
- Ese cliente, cuando solicite otras páginas de ese servidor enviará en la solicitud esa información que se le pidió almacenar (la cookie)
- Cuando el cliente solicita un URL envía las cookies que pertenezcan a ese dominio y dentro del camino (path) especificado
- Las cookies se envían al servidor como parte de la cabecera HTTP

```
Cookie: NAME1=OPAQUE_STRING1 ;NAME2=OPAQUE_STRING2 ...
```

Persistent Client State HTTP Cookies

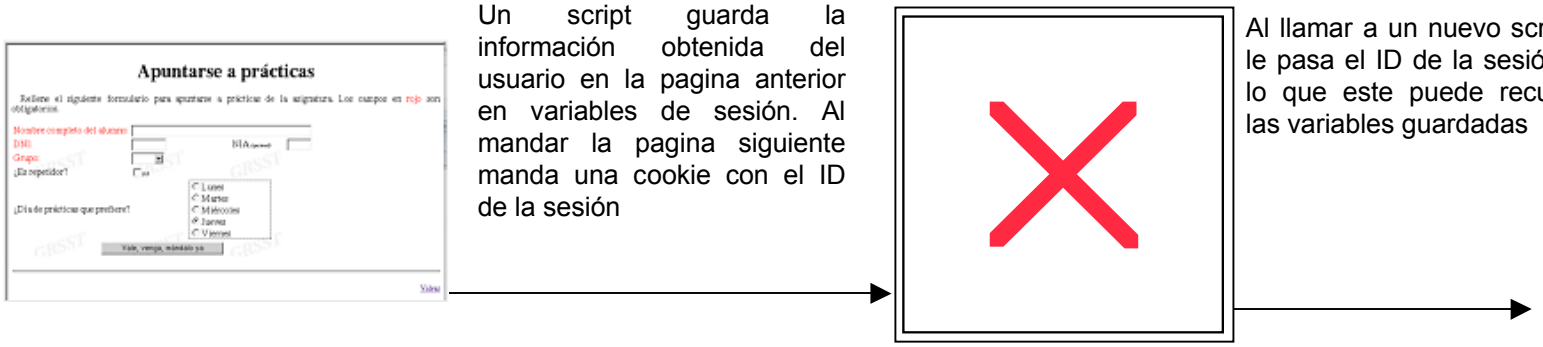
- Se introduce una cookie en el cliente mediante `Set-Cookie` en la cabecera HTTP:

  `Set-Cookie:` *`NAME`* `=`*`VALUE`* `; expires=` *`DATE`* `;path=` *`PATH`* `; domain=` *`DOMAIN_NAME`* `; secure`

  - El contenido de la cookie sigue el formato `NAME=VALUE` (ni punto-y-coma ni coma ni espacios).
  - Se le puede indicar una fecha máxima de validez a la cookie
  - Si se ha indicado el atributo `domain` el cliente, cuando haga una petición, enviará la cookie solo si en el URL al que se le solicita el nombre de dominio de la máquina está dentro de ese dominio
  - Con el atributo `path` se puede restringir el subconjunto de URLs del dominio a los que se les enviará la cookie al solicitar una página
  - Si se indica el atributo `secure` esta cookie solo se enviará si la conexión es segura (sobre SSL)

Sesiones en PHP

- Sucede lo mismo con los scripts PHP que con CGIs: no se guarda estado
- PHP nos permite guardar el contenido de unas variables asociándolas a una sesión
- En realidad lo que hará será guardar esas variables localmente y mandar al usuario un identificador de sesión asociado a ese conjunto de variables en forma de una cookie
- Cuando el cliente solicita otro script PHP envía su cookie con el identificador de sesión
- Se accede al fichero correspondiente recuperando esas variables de forma que parece que conservar el contenido asignado por el anterior script...



Variables y algunas funciones

- `$_SESSION`
  - Variable superglobal
  - Array con las variables que se guardan en la sesión (la clave es el nombre de la variable) => la clave tiene que cumplir las normas de una variable
- `bool session_start(void)`
  - Crea una nueva sesión o recupera las variables de una
  - En sesiones implementadas con cookies hay que llamar a esta función antes de que se envíe nada al navegador (para que se pueda poner la cookie en la cabecera HTTP)
- `bool session_destroy(void)`
  - Destruye los datos asociados a esta sesión (fichero donde guarda el servidor las variables de la sesión)
  - No invalida (unset) las variables ni la sesión
- `string session_encode(void)`
  - Devuelve una cadena con en contenido de la sesión codificado
- `session_decode(string data)`
  - Decodifica la cadena que se le pasa creando las variables que indica

PHP+Session: uso

- El identificador de la sesión va en la cabecera =>antes de enviar ningún dato hay que generar la sesión

```
<?php
session_start();
?>
HTML
<?php
$_SESSION['var'] = valor;
?>
Más HTML
```

```
<?php
session_start();
?>
HTML
<?php
$_SESSION['var'] = valor;
?>
Más HTML
```