

# Desarrollo de sitios web con PHP y MySQL



## Tema 6: Sesiones

Ana María Feroso García  
afermosoga@upsa.es



# Tema 6: Sesiones

1. Introducción
2. Manejo de sesiones
3. Autenticación de usuarios

# Introducción

- A veces es necesario mantener el estado de una conexión entre distintas páginas o entre distintas visitas a un mismo sitio
  - Ejemplos: aplicaciones personalizadas, carrito de la compra, control de acceso
- HTTP es un **protocolo sin estado**: cada conexión entre el cliente y el servidor es independiente de las demás
- Para mantener el estado entre diferentes conexiones hay que establecer lo que se conoce como una **sesión**
- Las sesiones permiten disponer de unas variables con valores persistentes durante toda la conexión del usuario. Estas variables pueden almacenarse en el cliente mediante *cookies* o en el servidor
- PHP dispone de una biblioteca de funciones para la gestión de sesiones



# Manejo de sesiones

- Funciones de PHP para el manejo de sesiones (register\_globals On)
  - **session\_start ()**
    - inicializa una sesión y le asigna un identificador de sesión único. Si la sesión ya está iniciada, carga todas las variables de sesión
  - **session\_register (variable)**
    - registra una variable de sesión
  - **session\_unregister (variable)**
    - elimina una variable de sesión
  - **session\_is\_registered (variable)**
    - comprueba si una variable está registrada. Devuelve *true* en caso afirmativo y *false* en caso contrario
  - **session\_destroy ()**
    - cierra una sesión

# Manejo de sesiones

- **Funciones de PHP para el manejo de sesiones** (*register\_globals* Off - valor *más habitual*)
  - **session\_start ()**
    - inicializa una sesión y le asigna un identificador de sesión único. Si la sesión ya está iniciada, carga todas las variables de sesión
  - **\$\_SESSION['nombre'] = valor;**
    - registra una variable de sesión
  - **unset (\$\_SESSION['nombre']);**
    - elimina una variable de sesión
  - **session\_unset ()**
    - Elimina todas las variables de sesión
  - **if (isset(\$\_SESSION['nombre']))**
    - comprueba si una variable está registrada. Devuelve *true* en caso afirmativo y *false* en caso contrario
  - **session\_destroy ()**
    - cierra una sesión



# Manejo de sesiones

- El manejo de las sesiones se realiza de la siguiente forma:
  - Todas las páginas deben realizar una llamada a `session_start()` para cargar las variables de la sesión  
Esta llamada debe estar colocada antes de cualquier código HTML
  - Asignamos una variable de sesión por cada valor que deseemos almacenar como sesión `$SESSION[nom_varianle_sesion]=valor`
  - Conviene llamar a `session_destroy()` para cerrar la sesión
- CEDER CONTROL en PHP
  - Muy habitualmente al trabajar con sesiones hay que ceder el control a un fichero (.php) que es quien procesa las variables de sesión. Para ceder el control de un fichero .php a otro se utiliza la sintaxis:  
**`header("location: nom_fich.php");`**

# Autenticación de usuarios

- Una cuestión frecuente en un sitio web es controlar el acceso de los usuarios a una zona determinada del mismo y para ello es muy habitual el uso de sesiones
- La autenticación de usuarios puede realizarse en el propio servidor web. Así, en Apache los ficheros `.htaccess` permiten limitar el acceso a un determinado recurso del servidor
- Una alternativa más flexible es utilizar PHP junto con una base de datos para controlar el acceso de los usuarios. Para ello se utilizan las sesiones



# Autenticación de usuarios

- Esquema de una página que utiliza sesiones para autenticar usuarios:

```
<?PHP
    session_start ();
?>
<HTML LANG="es">
<HEAD> ... </HEAD>
<BODY>
<?PHP
    if (isset($_SESSION["usuario_valido"]))
        // Código para usuarios autorizados
    else
        // Mensaje de acceso no autorizado
?>
</BODY>
</HTML>
```



# Autenticación de usuarios

- **Ejercicio:**

En el ejemplo dado sobre formulario de solicitud de datos (NIF, nombre, email, procedencia , sexo y fichero con foto), una vez introducidos los datos y validados hasta que resulten correctos, mostrar por pantalla los datos introducidos, excepto la foto, a través de otro fichero “procesar\_datos.php”

