

# Ejercicios de Cookies y sesiones

1. Hacer un script que la primera vez que se le llame muestre un mensaje “Hola desconocido. A partir de ahora te recordaré”. En ese momento se generará una cadena aleatoria (utilizando la función que se adjunta aquí, llamándola sin parámetros) y se colocará una cookie llamada “UID” con el valor devuelto por la función. Las siguientes veces que se conecte el usuario se mostrará un mensaje que diga. “Hola de nuevo. Te conozco como UID = <UID>”. Observar los “response-headers” y “request-headers” en cada consulta. Probar a borrar la cookie y rehacer la ejecución del script.

```
function generaCadenaAleatoria($longitud = 5) {  
    $charset = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";  
    $base = strlen($charset);  
    $result = '';  
    $now = explode(' ', microtime())[1];  
    while ($now >= $base){  
        $i = $now % $base;  
        $result = $charset[$i] . $result;  
        $now /= $base;  
    }  
    return substr($result, -5);  
}
```

2. Hacer un script que lleve la cuenta del número de visitas realizadas hasta el momento. La primera vez que se visita pondrá “BIENVENIDO”. El resto de veces pondrá. “Hola es la vez número <numVeces> que visitas esta página”.
3. Hacer una aplicación con el siguiente aspecto:

## Listado de cookies por nivel

Nivel0 --> [Listado de cookies](#)

Nivel0 --> Nivel1 --> [Listado de cookies](#)

Nivel0 --> Nivel1 --> Nivel2 --> [Listado de cookies](#)

**Creación de cookies en diferentes niveles (directorios) por un script ubicado en /**  
**Se permite fijar nombre y contenido de la cookie, así como el nivel**

Creación de cookie

Nombre

Contenido

Nivel

Crea cookie

...que me permita crear ciertas cookies al nivel en el que está el script (nivel 0), o a un par de niveles más profundos, con el nombre y el contenido que queramos. Al pinchar en los hipervínculos de cada nivel se mostrarán las cookies que son “visibles” a ese nivel.

Listado de cookies de nivel **dwes/pruebas/p1/nivel1/nivel2/**

Nombre cookie	Contenido cookie
dos	contenido2
uno	contenido1
cero	contrnido0

4. Hacer un programa que simule un proceso de login con el siguiente aspecto:

## Tratamiento de cookies

Primera ejecución. Usted no está conectado.

Usuario

Clave

Si se introduce un usuario/clave adecuados (almacenados en un array asociativo) se muestra una página con el siguiente aspecto.

Conectado como **alfa**. Número de visitas: 4.



Se informará del usuario con el que se ha conectado. El número de conexiones que ha hecho y mostrará un listado de banderas de tamaño 27x18 (pueden descargarse de [aquí](#) las imágenes) cuya selección se recordará en el siguiente login.

Al desconectar, se mostrará una pantalla similar a la primera, despidiéndose del usuario:

## Tratamiento de cookies

Adiós **alfa**. Usted no está conectado.

Usuario

Clave

Almacénese toda la información que se requiera de cada usuario en una sola cookie. Investigar las funciones “explode/implode” de PHP, o bien “serialize/unserialize”

- Hacer un script que implemente un contador de visitas utilizando sesiones y llamándose a sí mismo. Mantener la sesión sin utilizar cookie de sesión, para evitar que alguien pudiera bloquear las cookies en el navegador (hacer la prueba de bloquearlas y ver cómo sigue funcionando).
- Hacer un par de scripts crearSesion.php y recuperarSesion.php que, respectivamente, creen una sesión, borren los datos de la sesión anterior, asignen 6 valores de sesión, y a continuación los recupere y visualice. Los valores deben ser de los siguientes tipos:
  - entero: un número entero

- real: un número real
- texto: una cadena de texto
- fecha: una fecha (objeto DateTime. Utilizar date\_create('yyyy-mm-dd'))
- semaforo (array asociativo ['r'=>'rojo', 'a'=>'amarillo','v'=>'verde'])
- un Punto (objeto de la clase Punto(x,y) que representa a un punto geométrico)

7. Utilizando sesiones, hacer una simulación de un registro por pasos en el que se va progresando por varios formularios hasta una pantalla que muestra el resumen de la información introducida hasta el momento. En cualquier momento se podrá volver a un paso anterior y se deberá mantener la información introducida en el formulario previamente. Los iconos se pueden encontrar [aquí](#).

8. Realizar un script que simule el juego del ahorcado con el siguiente aspecto

Al comenzar, el juego escogerá al azar una palabra oculta desde una Base de Datos de palabras (codificada por ejemplo en un array de palabras).

Se podrá introducir una letra, y el script se llamará a sí mismo considerando la nueva situación. Si la letra ya se había introducido antes, lo avisará y no computará para el número de fallos. Si la letra pertenece a la palabra oculta se desvelará en el panel, y si no, se informará y se

incrementará el número de fallos. La partida termina cuando se alcanza el número máximo de fallos o cuando se acierte la palabra.

El juego informará de todas las letras probadas hasta el momento.

El juego debe ser insensible a mayúsculas/minúsculas/tildes/diéresis. Para aclarar cómo hacerlo echar un vistazo [aquí](#)

Toda la información relevante entre una situación y otra debe conservarse utilizando variables de sesión y no cookies (exceptuando la propia cookie que contiene el PHPSESSID)

9. (PRÁCTICA) Realizar una pequeña aplicación de mensajería, en la que existen las siguientes pantallas.

- (**index.php**) Redirige incondicionalmente a la pantalla de **login.php**.
- LOGIN (**login.php**): Formulario con usuario/password y checkbox "recordar". Sólo si se marca este último la casilla usuario aparecerá rellena con el nombre del último usuario que entró (el password habrá que introducirlo siempre). Adicionalmente habrá un link que llevará a la pantalla de registro. (Para ver si funciona, NO UTILIZAR el sistema de recuerdo de login/password del navegador)



- REGISTRO (**registrar.php**): Formulario con usuario / password que permite crear nuevos usuarios. En un caso real, los usuarios se almacenarían en una BD. En este caso, lo haremos en una variable de sesión (lo cual no tiene mucho sentido ya que la BD de usuarios no estará disponible desde otro navegador, ni cuando se cierre la sesión).



- LISTA de USUARIOS (**listaUsuarios.php**): si el login no tiene éxito, se informará y volverá automáticamente a la pantalla de login (pasados 3 s.). Si tiene éxito, se indicará en primer lugar el nombre del usuario activo y a continuación una lista de usuarios (exceptuando a uno mismo) indicándose entre paréntesis los mensajes que le ha enviado cada uno de ellos. Al lado de cada usuario habrá un par de links: "leer" y "escribir". "leer" sólo estará visible si el número de mensajes es mayor que cero.

Usuario actual maria

**Lista de usuarios / mensajes**

pepe (2) [Leer](#) [Escribir](#)

manolo (0) [Escribir](#)

[Volver a login](#)

- LEER MENSAJES (**leer.php**): se accede a esta página desde el link “Leer” de “LISTA de USUARIOS” y mostrará la fecha y el texto de todos los mensajes que le ha enviado el usuario correspondiente. ¡OJO: Se debe impedir que un usuario pueda leer mensajes que no van dirigidos a él, aunque accedan manualmente operando sobre los parámetros GET.

Usuario actual maria

**Lista de mensajes de pepe**

Fecha	Texto
28/10/2015 (11:16)	uno
28/10/2015 (11:17)	dos

[Volver a lista de usuarios](#)

- ESCRIBIR MENSAJE (**escribir.php**): Formulario que permite escribir un mensaje al usuario escogido en la pantalla “LISTA de USUARIOS”. Los campos “De” y “Para” son informativos: vendrán rellenos y no se podrán modificar. Los mensajes serán acumulativos (no se pueden borrar), y el usuario tan sólo escribirá el texto del mensaje, la fecha y hora la calculará el servidor y la almacenará junto al mensaje.

De: maria

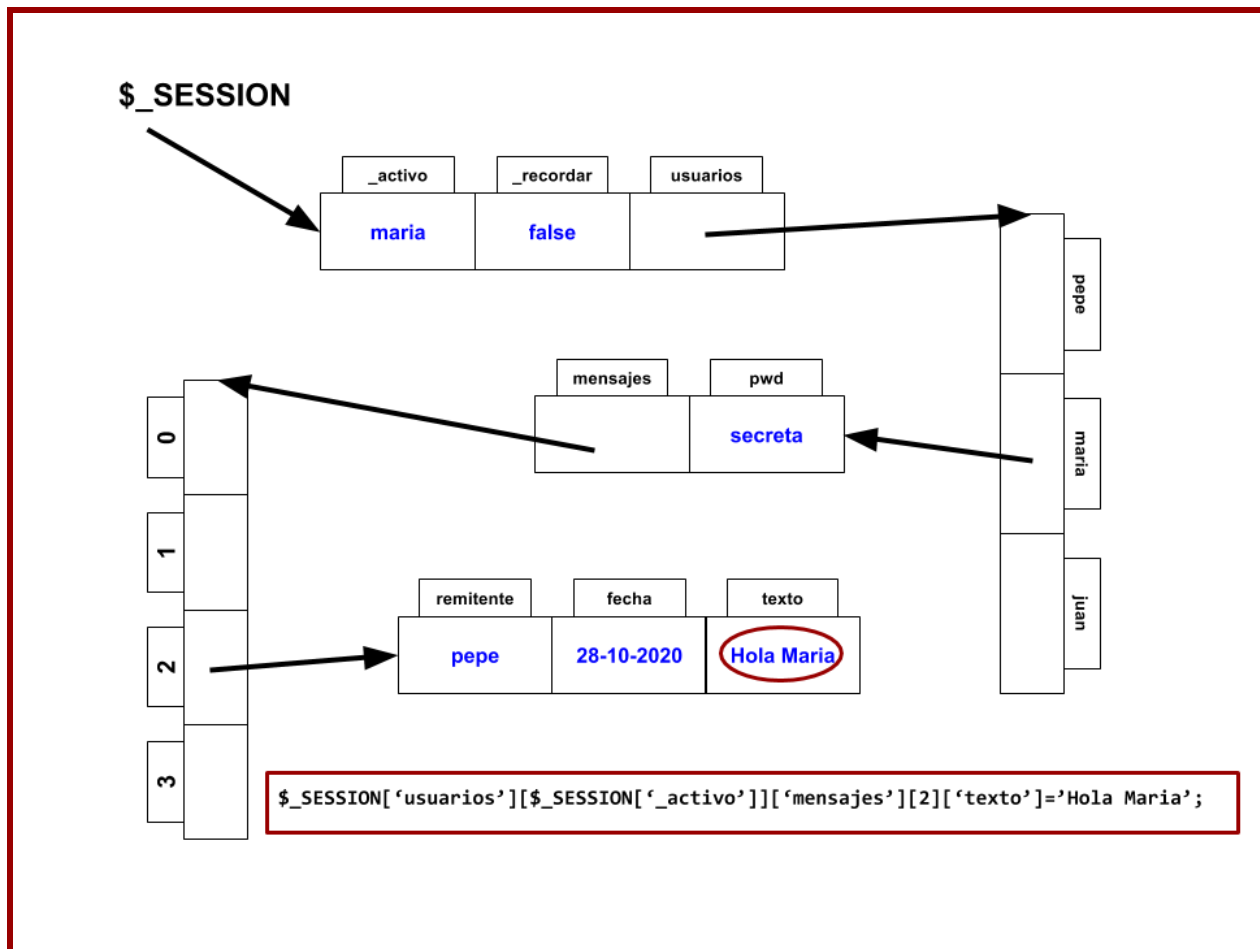
Para: manolo

Escribe el contenido del mensaje

Hola Manolo, cómo estás

Enviar

PISTA: Utilizar una estructura de datos similar a ésta para guardar la información.



10. Implementar una aplicación diseñada por un jefe de proyecto para gestionar las tareas para sus tres empleados “Pepe”, “Ana” y “Juan”.

## Gestión de tareas

Añadir tarea

Ver tareas

- Añadir tarea, mostrará un campo de texto con el título de la tarea que queremos crear, y un select en el que escogeremos (hard-coded) a qué empleado se la queremos asignar. Al asignarle la tarea la aplicación mostrará la lista de tareas asignada hasta el momento a ese empleado, destacando la añadida recientemente. Habrá un botón para volver a la lista de tareas

### Añadir tareas

Nombre de la tarea

Empleado Pepe ▾

Pepe
Pepe
Ana
Juan

### Lista de tareas de Pepe

- Estudiar PHP
- Descansar
- **Hacer ejercicios PHP (NUEVO)**

- El botón “Ver tareas” del menú principal, mostrará la lista de tareas de los tres empleados

### Lista de tareas

- Pepe
  - Estudiar PHP
  - Descansar
  - Hacer ejercicios PHP
- Ana
  - Estudiar JAVA
  - Instalar SPRING
- Juan
  - Arreglar despacho
  - Llamar clientes
  - Arreglar contratos

NOTA: Toda la información deberá mantenerse utilizando variables de sesión.

- Implementar el juego de las siete y media con el siguiente aspecto.

### Siete y media

**Jugada**

(no se ha jugado ninguna carta todavía)

- Al pulsar “Sacar carta” se irá mostrando el nombre de la carta (sacada al azar de un baraja española virtual), así como de la puntuación acumulada hasta el momento.

**Siete y media**  
**Jugada**

- Seis de oros (total: 6)
- As de copas (total: 7)

Sacar carta

Plantarse

Nuevo juego

- Si al sacar una nueva carta nos pasamos de siete y media, el programa informa de que has perdido y sólo muestra el botón de “nuevo juego”

**Siete y media**  
**Jugada**

- Seis de oros (total: 6)
- As de copas (total: 7)
- Tres de bastos (total: 10)

**:: Has perdido !!**

Nuevo juego

- Si al sacar una nueva carta llegamos a “siete y media” justo, el programa informa de que has ganado, y sólo muestra el botón de “nuevo juego”

**Siete y media**  
**Jugada**

- Seis de oros (total: 6)
- As de copas (total: 7)
- Rey de oros (total: 7,5)

**:: Has GANADO !!**

Nuevo juego

- Al pulsar “Plantarse”, el programa simulará la tirada de la banca que siempre jugará a igualar o superar la puntuación del jugador. Si la jugada final del jugador es igual o inferior a la banca, el jugador pierde. En caso contrario (la banca se pasa), el jugador gana.



## Siete y media

### Jugada

- Seis de oros (total: 6)
- As de copas (total: 7)

### Jugada de la banca

- Cinco de espadas (total: 5)
- Seis de copas (total: 11)

!! Has GANADO !!

Nuevo juego

**NOTAS:** La aplicación debe implementarse sin utilizar javascript y almacenando el estado que se considere necesario en variables de sesión.

**MEJORA 1:** Incorporar un contador de jugadas totales ganadas por ti y por la banca, y un botón de “Limpiar” para comenzar de nuevo el conteo.

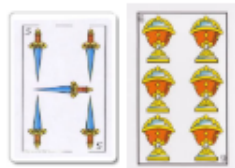
**MEJORA 2:** Mostrar una imagen de las cartas en lugar de su nombre con letra.

## Siete y media

### Jugada (Total: 7)



### Jugada de la banca (Total: 11)



!! Has GANADO !!

Nuevo juego

12. Realizar una aplicación que haga lo siguiente:

**Primer paso:** pida un número de sumandos

Indica el número de sumandos (2..15)

Enviar

**Pasos intermedios:** solicite el valor de cada uno de los sumandos indicados

Introduce el sumando número 1 (1..10)

Enviar

Introduce el sumando número 2 (1..10)

Enviar

Introduce el sumando número 3 (1..10)

Enviar

**Paso final:** Muestre el resultado de la suma

$10 + 9 + 5 = 24$

[Volver](#)

NOTA: No se pueden utilizar campos hidden.

Solución A: hacer el ejercicio utilizando sólo cookies.

Solución B: hacer el ejercicio utilizando variables de sesión.

13. ---