

# APLICACIONES EN INTERNET

## PROBLEMAS BLOQUES I Y II

**Se recomienda que los ejercicios no solo se resuelvan en papel si no que se implementen, cuando sea posible, en un servidor real, bien el del laboratorio, bien en el propio ordenador del alumno. De esta manera se puede además comprobar si la solución es correcta.**

- 1 Un grupo de estudiantes de telemática ha desarrollado un rudimentario foro web ([www.forosteleco.com](http://www.forosteleco.com)) para debatir diversos temas. La página principal muestra una lista de hiperenlaces a distintos temas ("política", "deportes", "fiesta", etc.). Dentro de cada tema hay una lista de mensajes dejados por los usuarios. Los usuarios deben iniciar una sesión en el foro introduciendo su clave y nombre de usuario. Una vez iniciada la sesión, ésta durará hasta UNA SEMANA, pasado ese tiempo el usuario tendrá que volver a iniciar sesión. Además, cuando una sesión está iniciada, el foro muestra al usuario los temas en los que hay mensajes sin leer cuando accede a la página principal. Implemente usando PHP y COOKIES:
  - a) (1 punto) Implemente la función *iniciarSesión()* que realiza las acciones necesarias para el establecimiento y control de sesiones. El nombre de usuario y la contraseña se comunica mediante el método POST en las variables *user* y *pass* respectivamente.
  - b) (1 punto) Implemente la función *temaLeido()*. Esta función se ejecuta al pinchar en un tema y permitirá que el servidor informe al usuario de los temas sin leer cuando acceda a la página principal.
  - c) (1 punto) Implemente la función *comprobarTemasConMensajesNoLeidos()*. Esta función se ejecuta en la página principal y comprueba los temas en los que hay mensajes sin leer. Como resultado mostrará el siguiente mensaje: "Hay mensajes sin leer en los temas: X, Y, Z, ..."

NOTA: las siguientes funciones se suponen ya programadas y disponibles para su uso.

- *bool comprobarClave(string usuario, string clave)*. Esta función devuelve "true" si la clave que se pasa como parámetro, coincide con la clave almacenada para el usuario que se pasa como parámetro.
- *\$nombreTema*. Esta variable almacena el nombre del tema que se está visitando en cada momento.
- *int numeroMensajes(string tema)*. Esta función devuelve el número de mensajes que se han escrito para el tema que se pasa como parámetro.

2. Una tienda online (*www.tienda.com*) ha implementado su servicio de ventas mediante el típico *carrito de la compra*. El usuario, a medida que navega por su catálogo, añade los objetos que desea comprar a su carrito de la compra. Cuando ha elegido todo lo que desea, efectúa la compra. Cada vez que el usuario selecciona productos, el carrito de la compra los incorpora. Además, la tienda ofrece descuentos del 10 % del precio total de la compra si: a) el usuario compra algún *producto estrella* o b) el usuario gasta más de 300 euros. Estos descuentos no son acumulables. Para simplificar, se supone que el usuario sólo puede seleccionar una unidad de un producto y no puede volver a seleccionar un producto ya seleccionado. Implemente con PHP Y COOKIES las funciones necesarias para ofrecer este servicio.

- a) (1 punto) Implemente la función *seleccionarProducto()* que realiza las acciones necesarias para actualizar el carrito del usuario. Esta función recibe el código del producto mediante el método GET, en la variable *id*. El carrito de la compra durará lo que dure la sesión. Cuando el usuario selecciona un nuevo producto recibirá el mensaje "Ha seleccionado el producto Y cuyo precio es Z".
- b) (1 punto) Implemente la función *eliminarProducto()*. Esta función elimina el producto que se indica mediante el método GET en la variable *id*, como en el caso anterior, y muestra el mensaje "Producto X eliminado".
- c) (1 punto) Implemente la función *validarCompra()*. Esta función se ejecuta cuando el usuario decide aceptar su carrito de la compra. Esta función mostrará un mensaje "Va a comprar los siguientes productos: X, Y, Z, etc., cuyo precio total es W". Es decir, una lista de los productos a comprar. Si el usuario se beneficia de algún descuento, mostrará además el mensaje "Dispone de un descuento del 10 %, con lo que el precio total es T". Donde debe sustituir las variables X,Y,Z,W,T por su valor correspondiente.

NOTA: Tenga en cuenta que el usuario puede seleccionar más de un producto estrella. Las siguientes funciones ESTÁN A DISPOSICION DEL PROGRAMADOR Y PERMITEN OBTENER INFORMACIÓN DEL PRODUCTO:

- `string nombreCompletoProducto($idProducto)`
- `double precioUnitario($idProducto)`
- `bool esEstrella($idProducto)`

3. Se desea desarrollar una aplicación para gestionar el trabajo en una empresa. La información se guardará en una BBDD MySQL. Se le pide que:
  - a) Diseñe una tabla con la información de los empleados: DNI (sin letra), nombre, apellidos, dirección, edad título.
  - b) Implemente 4 funciones en PHP: *insertar*, *eliminar*, *modificar* y *seleccionar*. Todas reciben la información necesaria mediante el método GET (elija los nombres de variables que considere adecuados). La función *seleccionar* recibe una cadena de búsqueda y el campo sobre el que se realiza la búsqueda o bien, si no se indica cadena de búsqueda, se muestra toda la tabla en la BBDD.
  - c) Diseñe una tabla adicional de tareas: nombre, contratista, tiempo y coste.
  - d) Diseñe una tabla que relacione tareas con empleados. Un empleado puede tener más de una tarea asociada.
  - e) Repita el apartado b) para la tabla de tareas.
  - f) Implemente funciones que permitan mostrar las tareas asociadas a cada empleado o el empleado asociado a una tarea determinada.
4. Se desea implementar una tienda electrónica, mediante una aplicación en PHP. La tienda cuenta con un catálogo de productos diferenciados en 3 categorías (Literatura, Música y Video), que se ha implementado sobre una BBDD MySQL. Se dispone de una BBDD llamada "ai" y de un usuario llamado "aiuser" y clave "ai". Se pide que:
  - a) Diseñe una tabla para la BBDD que recoja la información del catálogo. Para cada producto se guardará la siguiente información: precio, nombre, descripción breve, unidades disponibles y categoría. Puede serle útil un campo adicional de identificación.
  - b) Cree un formulario HTML para introducir la información de cada producto. La información se enviará mediante el método POST a un script PHP llamado *procesarCat.php*. La categoría del producto se elige mediante una lista desplegable.
  - c) Implemente una función en PHP que inserta un nuevo elemento en el catálogo. La información se recoge mediante el formulario anterior. El script muestra el mensaje *Nuevo producto añadido*.
  - d) Cree un documento HTML que permite visualizar el catálogo. Se puede elegir la categoría que se quiere visualizar.
  - e) Implemente la función *mostrarCatalogoAdmin(\$cat)* que muestra el catálogo según la categoría. El catálogo se muestra en forma de tabla. Al final de la cada fila habrá dos hiperenlaces *Eliminar* y *Editar* para eliminar o editar los productos. Decida e incluya en los formularios el método que utilizará para enviar la información y los script que la procesarán.
  - f) Añada una hoja de estilo que haga que la tabla se muestre con bordes sólidos de 2 pixel de tamaño y color azul. Haga que las filas de la tabla se muestren alternativamente con fondo verde y gris, para ello se le aconseja que modifique la función anterior para que al mostrar los elementos de la tabla se asignen atributos apropiados para que la hoja de estilo pueda mostrarlos como se le pide.
  - g) Implemente un script en PHP que elimine un producto del catálogo y muestre *Producto eliminado*.
  - h) Implemente un formulario HTML que permita actualizar los campos de un

producto del catálogo. Este formulario se mostrará cuando el usuario pinche el botón *Editar* del formulario mostrado en el apartado e). Para ello, el script debe buscar en la BBDD la información del producto que se desea editar. He- cho esto, se utiliza esa información para rellenar los campos que se le muestran al usuario en el formulario. Además se incluye un botón *Guardar*.

- i) Implemente una función PHP que actualiza el valor de los campos de un pro- ducto según la información recibida del formulario del apartado anterior.
- j) Implemente una formulario HTML y un script PHP que permita buscar y mostrar al usuario todos los productos de una determinada categoría cuyo precio sea inferior o superior a uno dado.
- k) Diseñe una tabla adicional con información extra asociada a cada producto: Autor, Género y año de publicación.
- l) Implemente una función que, dado un producto, muestre su información extra asociada.

5. Un sitio de Internet (*www.widgets.com*) ofrece a sus usuarios una serie de pequeños programas para su descarga. Si el usuario se registra, le ofrece un servicio adicional de actualizaciones: cada vez que el usuario accede al sitio web, es informado sobre actualizaciones disponibles para los programas que el usuario ha descargado.

Implemente en PHP las acciones necesarias para que este mecanismo funcione, sin que sea necesario almacenar información en el servidor de *www.widgets.com*. Para ello:

- a) Implemente un script PHP llamado *registro.php*. Este script se encarga de procesar los datos del usuario cuando rellena el formulario de registro, que se envían mediante el método GET. El nombre de usuario se envía en un campo llamado *name*. Es decir, realiza las acciones para que el servidor pueda saber que un usuario se ha registrado e imprime un mensaje de bienvenida al usuario.
- b) Implemente la función *descarga(string programa, int version)*. Esta función PHP es invocada cada vez que el usuario descarga un nuevo programa. Recibe como parámetros el nombre del programa descargado y la versión del mismo. Esta función toma las acciones necesarias para que el servidor pueda saber los programas que el usuario ha descargado e imprime un mensaje por pantalla “Aquí tiene su programa”. Observe que realmente no tiene que implementar la descarga en sí.
- c) Implemente la función *actualizaciones()*. Esta función PHP comprueba si existen actualizaciones de los programas que el usuario ha descargado. Para ello hace uso de otra función *int versión(programa, versionprevia)*, que recibe como parámetros el nombre del programa y la versión descargada por el usuario y devuelve 0 si no hay una versión nueva o un entero con el nuevo número de versión. NO DEBE IMPLEMENTAR ESTA ULTIMA FUNCION, SOLO UTILIZARLA SI LE ES NECESARIO. Por cada programa en el que exista una nueva versión, se escribe un mensaje que dice “La versión X del programa Y está disponible para descargar”.

6. Una web comercial (*www.musicapami.com*) permite descargar gratuitamente hasta 10 Mbytes durante una semana para que los usuarios comprueben la calidad de sus productos. Si el usuario compra algún producto de la web, se le permite comenzar a descargar desde cero. Implemente en PHP las acciones necesarias para que este mecanismo funcione **sin que sea necesario usar una base de datos**. Para ello:

- a) Implemente la función *descarga(string file, int kbytes)*. Esta función se encuentra en la página *www.musicapami.com/descarga.php* a la cual es dirigido un usuario cuando quiere descargar un archivo. La función recibe como parámetro el nombre y el nombre y el tamaño en Kbytes del archivo que pretende descargar el usuario. Esta función comprueba si el usuario ha

superado el límite de descargas, en cuyo caso muestra un mensaje del tipo “*Lo siento. Ha superado el límite. Espere un tiempo o compre uno de nuestros productos*”. En caso contrario simplemente muestra un mensaje “*Su archivo. Ya lleva descargados X bytes*”.

b) Implemente la función *comprar()*. Esta función se encuentra en la página *www.musicapami.com/compra.php*. Esta función le muestra al usuario un mensaje “*Gracias por la compra. Ahora puede disfrutar de otros 10 Mbytes de descargas*”. Y toma las acciones necesarias para que el usuario pueda seguir descargando.

7. Una web de viajes mantiene una BBDD con información actualizada de vuelos ofertados por distintas compañías aéreas y el precio del mismo. El portal ofrece un servicio de notificaciones que funciona de la siguiente manera: (1) el usuario selecciona el vuelo en el que está interesado y el precio que está dispuesto a pagar por él; (2) cuando el usuario accede a la web del portal, se busca el precio de los vuelos por los que el usuario mostró interés y se le informa si el precio ofertado está por debajo del precio que el usuario seleccionó. Sabiendo que la web utiliza una tabla llamada *vuelos* con tres campos (*IdVuelo*, *nombre* y *precio*), implemente con COOKIES y PHP el servicio descrito. Para ello implemente las siguientes funciones:

a) *interes()*. Esta función recibe mediante el método GET el identificador (*IdVue- lo*) del vuelo en que el usuario está interesado y el precio máximo dispuesto a pagar (*max*) e implementa las acciones necesarias para que el usuario pueda ser notificado en sus próximas visitas (1 punto).

b) *notificar()*. Esta función accede a la BBDD y comprueba para cada vuelo en que el usuario está interesado si el precio es inferior al máximo dispuesto a pagar mostrando el siguiente mensaje *el vuelo X ya está disponible a un precio Y* o no mostrando nada en caso de que no esté disponible. Suponga que ya hay implementada una función *\$link=conectar()* que se conecta al servidor MySQL y la BBDD de vuelos. (1.5 puntos).

8. Dado el siguiente fragmento de código HTML.

```
<HTML>
<HEAD>
<TITLE>Session</TITLE>
</HEAD>
<BODY>
<div>

<h4>Session Manager Features</h4>

<ul>
<li>Encryption</li>
<li>Middle clicking the tab bar will reopen the last closed tab. This
is disabled if <a href="addon" alt="enlace">Tab Mix Plus</a></li>
</ul>
</div>
</BODY>
</HTML>
```

- Añada mediante HTML un evento para que al pasar el ratón por encima del elemento H4 se muestre en color rojo, usando para ello javascript. ¿Hay una forma alternativa de conseguir esto sólo mediante CSS?
- Añada un evento para que, al finalizar de cargar el documento se invoque una función javascript que asigne a todos los elementos LI el evento de pinchar con ratón. Implemente dicha función, que además debe estar declarada en un archivo externo que se debe asociar al documento. Para ello, añada los elementos necesarios al documento.
- La función javascript que se asociará al evento de los elementos LI anterior invocará mediante AJAX una petición tipo GET a un servidor. Cuando se reciba la respuesta del

servidor, se añadirá documento un nuevo elemento H1, a continuación del elemento H4 que contendrá el texto que devuelva el servidor.

9. Dado el siguiente fragmento de código.

```
<HTML>
<HEAD>
<TITLE>Session</TITLE>
</HEAD>
<BODY>
<SCRIPT>
var el=getElementById("titulo");
el.style='font-size:10px';
</SCRIPT>
<div>
<h4 id="titulo">Session Manager Features</h4>
</div>
</BODY>
</HTML>
```

- a) ¿Qué se pretende conseguir con el código javascript? ¿Funcionará? ¿Por qué? Indique dos formas de solucionarlo.

10. Una red social almacena en una BBDD MySQL la información de los usuarios registrados. Se almacena en la tabla *usuarios* la información de perfil del usuario: nombre y apellidos, país, URL a la foto del perfil y DNI (sin letra). El DNI se utiliza como identificador de usuario y es un campo único en la tabla (no puede contener entradas repetidas). Además, se utiliza la tabla *fotos* para almacenar las URL de las fotografías que los usuarios suben al portal para compartir con sus amigos y la fecha en la que se tomó. Por supuesto, cada foto está relacionada con un usuario mediante el DNI del usuario. El portal de la red social está implementado con PHP.

- a) Se desea implementar la funcionalidad típica de una red social, es decir, establecer relaciones entre usuarios. Cada usuario tendrá un conjunto *amigos* de entre el resto de usuarios. Indique las tablas adicionales que necesita para implementar esta funcionalidad, incluyendo y describiendo los campos que tendrían. Justifique su respuesta adecuadamente. (0.5 puntos)
- b) Implemente un script en PHP que, dado un usuario, muestre en una tabla HTML la información del perfil (nombre y apellidos, país y foto) de todos sus amigos. El DNI del usuario en cuestión es una variable recibida por el método GET en la variable *dni*. Además, esta información estará hiperenlazada, es decir, si el usuario pincha sobre el nombre de su amigo se abrirá una URL de la forma *http://social.es?dni=X* donde X es el DNI del amigo. (1 punto)
- c) Implemente un script PHP para que, dado un DNI mediante GET, se muestre una lista con todas las fotos de dicho usuario.
- d) Implemente el código que sea necesario (PHP, javascript, modificación de alguna de las funciones anteriores, CSS) para que, mediante AJAX, cuando el usuario pase el ratón por encima de una de las filas de la tabla del apto. b), se muestre bajo la tabla la lista de todas las fotos del usuario recogido en dicha fila.