Programación Avanzada
Grado en Ingeniería Informática en Sistemas de Información - Curso 2017/2018

EPD 6: Acceso a bases de datos en PHP

La entrega del trabajo se hará a través de la tarea correspondiente en el Campus Virtual. Pasado el límite de entrega se aceptará el envío del trabajo, con una penalización de 2 puntos sobre 10 de la calificación por cada hora o fracción de retraso. La entrega consistirá en un único fichero comprimido en formato ZIP cuyo nombre deberá ser de la forma *equipoXX.zip*, donde *XX* serán dos cifras que indicará el número del equipo. Por ejemplo, equipo07.zip. Este fichero contendrá una serie de carpetas cuyo nombre deberá ser de la forma *ejY* o *pZ*, donde *Y* y *Z* representan, respectivamente, el número de cada ejercicio o problema del presente guión. Dentro de dichas carpetas se incluirán exclusivamente los archivos necesarios en la resolución del correspondiente ejercicio o problema. Las rutas de los ficheros empleados serán relativas, a fin de que las resoluciones a los ejercicios y problemas puedan ser examinadas en cualquier equipo. Cualquier entrega que no cumpla las reglas de nombrado, el formato de compresión del archivo o el contenido de los archivos del mismo, será penalizada con 2 puntos sobre 10 por cada incumplimiento.

Objetivos

- Crear aplicaciones PHP que accedan a bases de datos.
- Familiarizarse con la administración de MySQL a través de PHPMyAdmin.

Conceptos

1. Acceso a bases de datos en PHP

En PHP hay disponible una gran cantidad de funciones para acceder a bases de datos dentro de Sistemas de Gestión de Bases de Datos (SGBD) de distintos fabricantes. Según el fabricante del SGBD, deberá usar unas funciones u otras. Durante el desarrollo de las prácticas nos centraremos en las funciones de acceso a MySQL como caso genérico. Puede encontrar una introducción al acceso a bases de datos en PHP en [1].

2. MySQL y PHPMyAdmin

MySQL es un SGBD libre y gratuito que está incluido en el paquete XAMPP, por lo que si tiene instalado dicho paquete no tendrá que instalar nada adicionalmente. Puede encontrar más información acerca de MySQL en [2].

Para gestionar las bases de datos MySQL que necesitaremos durante el desarrollo de esta práctica emplearemos una interfaz web, muy sencilla e intuitiva, llamada PHPMyAdmin. Esta interfaz permite administrar un servidor de bases de datos creando (o eliminando) usuarios, bases de datos, tablas, insertando y editando filas, etc. Es recomendable que consulte [3] para familiarizarse con el funcionamiento de PHPMyAdmin. Es especialmente importante dominar correctamente las operaciones de exportación e importación de bases de datos ya que permitirán llevarse, en forma de fichero SQL, a otra máquina las bases de datos desarrolladas durante la sesión de prácticas (éste será el mecanismo que se empleará para incluir la base de datos en las entregas de ejercicios y problemas que realice) y realizar ejercicios o pruebas de evaluación en las que se suministre una base de datos, previamente creada, que deberá importar.

Bibliografía Básica

1. PHP 5: fast & easy web development. Julie Meloni. Thomson Course Technology, 2004. Parte VI.
   http://0-site.ebrary.com.athenea.upo.es/lib/bupo/Doc?id=10058862
2. Manual de Referencia de MySQL 5.0, Curso (tutorial) de MySQL.
   http://dev.mysql.com/doc/refman/5.5/en/tutorial.html
3. Mastering phpMyAdmin 3.4 for effective MySQL management. Marc Delisle. Packt Pub, 2012.
   http://site.ebrary.com/lib/bupo/Doc?id=10533703

Experimentos

E1. (30 mins.) Experimente con PhpMyAdmin. Para ello, realice la siguientes tareas:

1. Conéctese al servidor MySQL incluido en XAMPP.
2. Cree una base de datos.
3. Cree una tabla.
4. Inserte información en la tabla.
5. Exporte la tabla.
6. Elimine la tabla.
7. Importe la tabla desde los ficheros del paso 5.
8. Exporte la base datos.
9. Elimine la base de datos.
10. Importe la base de datos desde los ficheros del paso 8.

E2. (25 mins.) Cree una página PHP que permita un ataque de inyección SQL y ejecútelo satisfactoriamente. Posteriormente, analice las opciones existentes para evitar el ataque y aplique una.

Ejercicios

Ej1. (40 mins.) Cree un sistema de login para el problema 1 de la práctica anterior (aplicación de gestión de aerolíneas). Este sistema, antes de entrar en cualquier página de la aplicación, comprobará si el usuario está identificado. En caso de que lo esté, no se notará ninguna diferencia con la aplicación anterior (salvo la posibilidad de hacer *logout* comentada posteriormente). No obstante, en caso contrario no se mostrará al usuario el contenido de la página, sino que se le redirigirá a un formulario de *login*. En dicha página de *login*, el usuario dispondrá de dos posibilidades:

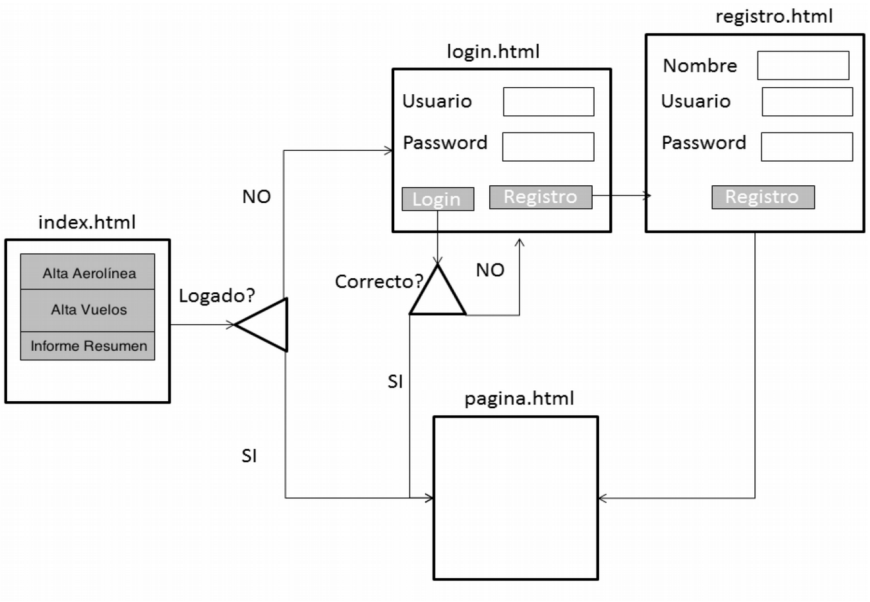
1. Logarse en el sistema introduciendo su usuario y contraseña. Una vez terminada la identificación, el usuario debe ser devuelto a la página que quería visitar originalmente (y no otra, por lo que deberá crear un mecanismo para recordar en qué página estaba). Tenga en cuenta que estos cambios de página se pueden conseguir gracias a la cabecera *location*.
2. Registrarse como nuevo usuario en la aplicación. De nuevo una vez terminada la identificación, el usuario debe ser devuelto a la página que quería visitar originalmente.

Para realizar la operación de login y registro, se empleará una base de datos en la que tendremos una tabla compuesta por:

- nombre: nombre del usuario
- usuario: usuario de acceso al sistema
- password: contraseña de acceso al sistema
- last_access: fecha y hora del último acceso

Con esta nueva tabla podremos comprobar las credenciales suministradas o registrar un usuario nuevo y se empleará una variable de sesión para conocer si el usuario está identificado.

Finalmente, proporcione en todas las páginas la opción de salir del sistema (es decir *logout*), haciendo que el usuario pierda su sesión y se requiera una nueva autenticación.



Ej2. (10 mins.) Añada al ejercicio anterior lo necesario para evitar que se puedan practicar inyecciones SQL.

Ej3. (10 mins.) Modifique el sistema anterior para evitar que las contraseñas de los usuarios puedan ser robadas por un acceso a la base de datos. Para garantizar la seguridad de las claves almacenadas en la base de datos procederemos a guardarlas protegidas, de tal forma que incluso alguien con acceso a la base de datos, no pueda conocerlas. Para ello haremos uso de las recomendaciones que PHP nos ofrece sobre almacenamiento de contraseñas. Se recomienda la lectura de [5] de la ampliación de la bibliografía, prestando especial atención a las funciones *password_hash()* y *password_verify()* que tendrá que utilizar.
De esta forma, bastará comparar la huella criptográfica de la contraseña del usuario (que previamente ha sido calculada mediante la función *password_hash()* y almacenada al darse de alta el usuario) con la función *password_verify()* con la contraseña introducida por el usuario en el momento de autenticarse en el sistema web. Gracias a ello, no tendremos almacenada la contraseña del usuario, sino su huella en la base de datos y, aunque podremos comprobar que se nos está suministrando en el momento de acceso una contraseña que genera la misma huella, no será posible descubrir sólo con la huella cuál es la contraseña del usuario (siempre que éstas sean fuertes, claro está).

Tenga en cuenta que, en lo que se refiere a funcionalidades críticas como es la gestión de contraseñas, es siempre recomendable verificar la documentación oficial de PHP para ver las últimas prácticas recomendadas. En particular, en lo que respecta a almacenamiento de contraseñas y conversiones de las mismas, algunas de las funcionalidades que se ofrecen en PHP 5.0 son ya consideradas débiles por la facilidad para descubrir la contraseña original y pronto serán obsoletas en PHP 7.0 (tales como la función *crypt()*).

Problemas

P1. (75 mins.) A partir del ejercicio 3 de esta sesión, adapte la aplicación de gestión de aerolíneas para que se empleen tablas en una base de datos, en lugar de ficheros.

P2. (75 mins.) Adapte el Problema 2 del guión de la sesión anterior para que se empleen tablas en una base de datos, en lugar de ficheros, para el sistema de imágenes basadas en categorías propuesto. No será necesario (ni recomendable) que los ficheros que contienen las imágenes se carguen en la base de datos. Sólo se utilizará ésta para almacenar los nombres de los archivos de las imágenes.

P3. (75 mins.) Adapte el Problema 3 del guión de la sesión anterior para que se empleen tablas en una base de datos, en lugar de ficheros, para la aplicación de almacenamiento de artículos científicos. No será necesario (ni recomendable) que los artículos se carguen en la base de datos, sólo se utilizará ésta para almacenar los datos relativos a los ficheros registrados en la aplicación.

Ampliación de Bibliografía

1.PHP 5 Power Programming. Andi Gutmans, Stig Bakken, Derick Rethans. Prentice Hall, 2004. Capítulo 6.
http://www.informit.com/content/images/013147149X/downloads/013147149X_book.pdf
2.PHP 5/MySQL programming for the absolute beginner. Andy Harris, 2003. Capítulos 7 y 8.
http://0-site.ebrary.com.athenea.upo.es/lib/bupo/Doc?id=10069849
3.PHP Essentials. Julie Meloni. Course Technolgy, 2003. Capítulos 3 y 4.
http://0-site.ebrary.com.athenea.upo.es/lib/bupo/Doc?id=10065759
4.PHP 5 for dummies. Janet Valade. Wiley Pub., 2004. Capítulo 12.
http://0-site.ebrary.com.athenea.upo.es/lib/bupo/Doc?id=10114230
5.Funciones de *hashing* de contraseñas.
http://php.net/manual/es/ref.password.php

Datos de la Práctica

Autor del documento: Carlos D. Barranco González (Diciembre 2007).

Revisiones:

1. Carlos D. Barranco González (Febrero 2009).
2. Carlos D. Barranco González (Enero 2010). Adaptación a la nueva planificación, correcciones en el texto y enlaces.
3. Carlos D. Barranco González (Enero 2011). Revisión de texto y renovación del ejercicio 1 (por renovación del ejercicio al que hace referencia en la APD anterior).
4. Alejandro Gómez Morón (Enero 2012). Adaptación ejercicio 2 y problema 2 (por renovación del ejercicio que hace referencia en la APD anterior).
5. Carlos D. Barranco González (Enero 2012): Ajuste de formato.
6. Carlos D. Barranco González (Noviembre 2012): 6. Adaptación del formato a Programación Avanzada. Actualización de la referencia 2 del apartado de bibliografía básica. Adición de aclaración al enunciado del Ej2. Corrección del texto del Problema 2 y actualización del enlace a la referencia de MySQL. Sustitución de la referencia 3 de la ampliación de bibliografía al no estar ya disponible ésta en la biblioteca.
7. Miguel Montero (Noviembre 2013): Actualización de la referencia 2 y 3 del apartado ampliación de bibliografía. Renovación del problema 1 y 2.
8. Carlos D. Barranco (Noviembre 2013): Mejora de la redacción del problema 2.
9. Carlos D. Barranco (Noviembre 2014): Mejora de redacción de la parte de conceptos. Inclusión de experimento 2. Replanteamiento de tiempos de experimentos, ejercicios y problemas. Actualización de las referencias a ejercicios y del guión anterior.
10. Carlos D. Barranco González (Noviembre 2015): Mejora en objetivos y sustitución de la referencia de bibliografía 3. Cambio de ejercicio 1 a problema 1, y renumeración de los restantes. Eliminación del último problema. Creación de los ejercicios 1 a 3.
11. Ricardo – León Talavera Llames (Noviembre 2016): Correcciones en la redacción del guión. Especificación del ejercicio 1 para que se adapte a la ampliación de liga de fútbol en lugar de cualquiera como se recogía anteriormente. Además se añade dibujo explicativo. Se modifica el ejercicio 3 y se adapta el problema 3 en función de los cambios en los ejercicios de la anterior sesión. El problema 1 se adapta para que se siga trabajando con la misma aplicación de los ejercicio anterior y se amplía su explicación para que quede claro que debe realizarse un alta de usuarios.
12. Carlos D. Barranco González (Noviembre 2016): Mejoras en la sección de conceptos y en el enunciado del Ejercicio 3.
13. Ricardo – León Talavera Llames (Noviembre 2016): Añadido nuevo enlace a ampliación de bibliografía y redefinido Experimento 1 y Ejercicio 3.
14. Gualberto Asencio Cortés (Noviembre 2017): Corregidos pequeñas erratas en el texto de los ejercicios y problemas. Modificados ejercicio 1 y problemas 1 y 2. Trasvase de 5 minutos desde el tiempo para los experimentos hacia los ejercicios, para soportar una nueva pequeña funcionalidad en el ejercicio 1 (fecha y hora del último acceso). Corrección del hipervínculo en última referencia de la ampliación de la bibliografía.
15. Carlos D. Barranco González (Noviembre 2017): Correcciones de formato y solución de erratas.

Estimación temporal:

- Parte presencial: 120 minutos.
  - Explicación inicial: 5 minutos.
  - Experimentos: 55 minutos.
  - Ejercicios: 60 minutos.
- Parte no presencial: 270 minutos.
  - Lectura y estudio del guión y bibliografía básica: 45 minutos
  - Problemas: 225 minutos