	Programación Avanzada **Ingeniería Informática en Sistemas de Información - Curso 2017/2018** **ENSEÑANZAS PRÁCTICAS Y DE DESARROLLO** **Segunda Convocatoria. Prueba de evaluación: PHP**	

Nombre:_____

Importante: El resultado de esta prueba consistirá en un proyecto de NetBeans que será comprimido en formato ZIP conteniendo la aplicación. Emplee las credenciales por defecto de XAMPP para la conexión con la base de datos. El nombre del fichero tendrá un formato específico dictado por el nombre de cada alumno. Por ejemplo, para un alumno llamado "José María Núñez Pérez" el fichero se nombrará como NunyezPerezJM.zip. Obsérvese que las tildes son ignoradas y las eñes sustituidas. No se aceptará ningún envío que no cumpla con las anteriores especificaciones de formato y nombrado. **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. Pasado el límite de entrega se aceptará el envío del trabajo, con una penalización de 2 puntos sobre 10 de la calificación por cada minuto o fracción de retraso a partir de tercer minuto. No es necesario que incluya el script de la base de datos en el envío. Las imágenes sin embargo, sí deberá incluirlas.**

Objetivos

- Demostrar los conocimientos adquiridos en el desarrollo de aplicaciones empleando PHP.

Descripción del ejercicio propuesto

Crear una aplicación web, empleando PHP como lenguaje de programación y MySQL como gestor de base datos, que permitirá gestionar un sitio web de repositorio de archivos y versiones.

Los usuarios pueden **registrarse** indicando su nombre, email y contraseña. Los usuarios pueden hacer **login** y, una vez dentro, pueden subir archivos de cualquier tipo (salvo archivos .exe) y administrar versiones del mismo archivo. Tras hacer login, se muestra una **página de listado de archivos** en la que se pueden ver todos los archivos que ha subido el usuario, así como las versiones disponibles de cada uno de ellos. Las **versiones** de un archivo permiten al usuario tener organizados varios archivos con el mismo nombre y diferentes contenidos. Mediante *checkboxes*, el usuario puede **borrar archivos** marcando las filas de la tabla que desee y pulsando en un botón para borrar. Además, el usuario puede **añadir archivos** con un botón que lleva a un **formulario de alta de archivo** donde se permite introducir los datos del nuevo archivo. Los datos que almacena el sistema de los archivos son los siguientes: nombre, descripción, longitud total y versiones. Por ejemplo, un registro en la base de datos (BD) podría contener: {"trabajo.pdf", "Trabajo fin de curso", 768122, "1;2;3"} (por ejemplo, el archivo representado tiene tres versiones, numeradas del 1 al 3). Las versiones se almacenan en el servidor en una ruta del tipo: "archivos/userX/Y-Z.pdf", donde *X* es el identificador numérico del usuario en la BD, *Y* es el identificador numérico del archivo en la BD y *Z* el número de la versión (por ejemplo, "archivos/user1/17-1.pdf"). Los usuarios tienen un límite de almacenamiento de 10 MB y sólo pueden subir archivos de hasta 2 MB. El sistema debe gestionar en todo momento el espacio libre que le queda al usuario. Esto implica que, cuando se añaden archivos, se debe comprobar que el usuario tiene espacio suficiente y, en caso afirmativo, decrementar el espacio disponible en la medida del tamaño del archivo subido. Además, el valor indicado en el campo longitud de la tabla de archivos en la BD se refiere a la suma de tamaños de todos los ficheros que guardan las versiones. Análogamente, tras borrar archivos, el sistema restaura el espacio ocupado por los mismos. En la página de listado de archivos, se indica al usuario el estado actual de su espacio disponible.

Desde el punto de vista técnico, todas las entradas del usuario deben ser saneadas para **evitar inyecciones** tanto de HTML como de SQL, cuando proceda. Las contraseñas deben mantenerse cifradas. Se utilizará una **variable se sesión** para controlar que el usuario está previamente identificado en todas las páginas de la aplicación web. Se permitirá hacer **cerrar la sesión del usuario** desde cualquiera de ellas. Se guardará también una **cookie** que contendrá el nombre del usuario.

Encontrará en el material adicional la base de datos exportada (con datos de ejemplo), varios archivos de prueba y las vistas que se han de generar desde las que podrá extraer el código HTML a emplear (use dicho código para no perder tiempo en esto).

Actividades a realizar

Para implementar la aplicación realice cada una de las siguientes actividades:

1. **[2.5 puntos]** El sistema dispone de una página inicial con un formulario de identificación (vista: *login.html*) y un botón de registro que lleva a una página con el formulario de registro (vista: *registro_usuario.html*). Desarrolle ambas funcionalidades, identificación y registro **(0.5 puntos)**. Como parte del proceso de registro de usuario, se debe crear una carpeta para los archivos del usuario dentro de la carpeta "archivos" del servidor, cuyo nombre será "userX" (donde X es el identificador asignado de forma autonumérica por el sistema gestor de bases de datos) (puede utilizar la función mkdir() de PHP) **(0.5 puntos)**. Tenga en cuenta que se guarda el *hash* de las contraseñas en la BD, para ello utilice las funciones *password_hash()* y *password_verify()* para cifrar y comprobar las contraseñas, respectivamente. En el material adicional encontrará un documento, *Manual_password_hash_verify.pdf*, con información sobre como usar dicha función **(0.25 puntos)**. Se asegurará que el sistema no puede verse sometido a ataques de inyección de SQL **(0.1 puntos)**.

   Una vez realizada con éxito la identificación, se definirá una variable de sesión para conocer en el resto de páginas que el usuario se ha identificado con éxito y cuál es su nombre de usuario (campo *nombre* de la tabla usuarios) **(0.2 puntos)**, mostrándose el contenido de la misma en el saludo al usuario en las diferentes páginas **(0.15 puntos)**.

   Tenga en cuenta que, al estar el usuario identificado, cada página de la aplicación tendrá un botón que permitirá cerrar la sesión (*logout*), tras lo cual se volverá a mostrar la página de login **(0.2 puntos)**. Se establecerá una *cookie* con duración de 30 días que recordará el usuario con el que se identificó en el último uso y lo mostrará prerelleno en el formulario de identificación **(0.4 puntos)**.

2. **[2.5 puntos]** Una vez realizada la identificación, la aplicación redirigirá al usuario a la página de listado de archivos **(0.25 puntos)**. En esta página se mostrará un listado, en forma de tabla, con todos los registros de la tabla *archivos* (respetando el formato definido en la vista: *gestor_archivos.html*) **(0.75 puntos)**. Tenga en cuenta que cada archivo tiene asociada una serie de versiones que deberán mostrarse con hipervínculos a sus contenidos correspondientes **(0.7 puntos)**. Además, en la parte superior de la página, se deberá mostrar el espacio disponible actualizado que le queda al usuario (en megabytes (MB)) **(0.7 puntos)**. Recuerde que si el usuario accediera a esta página directamente (por ejemplo, escribiendo la URI) y no estuviera identificado, redirigirá al usuario a la página de identificación **(0.1 puntos)**.

3. **[2.25 puntos]** Amplíe el punto anterior para permitir el borrado de archivos. Para ello, añada un *checkbox* en cada fila de la tabla para marcar archivos y un botón al final de la página para borrar los archivos marcados (y todas sus versiones) **(0.5 puntos)**. Implementar el borrado como acción del botón **(0.75 puntos)**. Tras borrar los registros correspondientes de la tabla *archivos* de la BD (los archivos físicos también deben eliminarse), se actualizará el espacio disponible del usuario **(0.5 puntos).** Debe tener en cuenta que el espacio que se libera es el de todas las versiones **(0.25 puntos).** Tras el borrado, se mostrará de nuevo el listado actualizado con los archivos restantes **(0.25 puntos).**

4. **[2.75 puntos]** Agregue al final de la página de listado de archivos un botón para crear un nuevo archivo/versión. Al pulsar sobre él nos mostrará un formulario con los siguientes campos y sus respectivas restricciones (vista: *alta_archivo.html*) **(0.5 puntos)**:
   - Descripción: no podrá dejarse en blanco, debe ser de tipo cadena alfabética y se permiten espacios (utilice para ello la siguiente expresión regular: *'/^[[:alpha:]|[:space:]]+$/')*.
   - Archivo: no puede ser un archivo con extensión .EXE (compruebe el tipo MIME del mismo) y su tamaño máximo será de 2 MB.

   Si se cumplen todos los criterios anteriores, se procederá a comprobar el espacio disponible del usuario para saber si tiene espacio suficiente para el nuevo archivo **(0.5 puntos)**. Si no tiene espacio suficiente, se informará de tal circunstancia y se le permitirá regresar al formulario de alta de archivo **(0.25 puntos)**.

   Si tiene espacio suficiente, se comprobará si el archivo que se pretende almacenar es un archivo nuevo o una versión de uno ya existente con el mismo nombre. En función de esto, se creará un nuevo registro en la tabla *archivos* o se actualizará el campo *versiones* del registro correspondiente **(0.5 puntos)**. Además, si el archivo es una nueva versión de un archivo existente, se debe actualizar también el campo longitud del registro correspondiente, sumando la longitud existente con la del nuevo archivo subido (recuerde que el campo longitud contiene en todo momento la suma de los tamaños de todas las versiones) **(0.5 puntos)**.

   El archivo se guardará en la carpeta adecuada del servidor y con el nombre oportuno, según la nomenclatura especificada anteriormente ("archivos/userX/Y-Z.pdf") **(0.4 puntos).** Tras ello, se deberá reducir el espacio disponible del usuario en la misma medida del tamaño del archivo subido **(0.2 puntos)**. Finalmente, se redirigirá hacia la página del listado de archivos **(0.15 puntos)**.