Programación Avanzada
Ingeniería Informática en Sistemas de Información - Curso 2018/2019
ENSEÑANZAS PRÁCTICAS Y DE DESARROLLO
Convocatoria de Recuperación - Evaluación 2 - PHP

Nombre:_____ **Equipo:**_____

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 MariaDB como base datos, que permitirá gestionar un sitio web de gestión de contactos.

Los **usuarios** pueden **registrarse** indicando su nombre, contraseña y email. Tras esto, pueden autenticarse y, una vez dentro, pueden ver los contactos, en una única **página de listado de contactos**, donde se mostrarán todos los contactos en una tabla (nombre, teléfono, email y foto). En esta misma página se pueden borrar contactos marcando las filas de la tabla que deseen y pulsando en el botón para borrar. Además, pueden añadir contactos con un botón de añadir que lleva a un formulario de alta de contacto que permite introducir los datos del nuevo contacto. Los datos que almacena el sistema de un contacto son los siguientes: nombre, teléfono, email y una foto.

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 ser ***protegidas*** *mediante hashing*. Se utilizará una **variable se sesión** para controlar que el usuario está autenticado en todas las páginas de la aplicación web. Se permitirá hacer **logout** desde cualquiera de ellas. Se guardará una **cookie** que contendrá el nombre del usuario que se ha autenticado, para que éste aparezca relleno automáticamente en la página de login en próximas ocasiones.

Encontrará en el material adicional la base de datos exportada (con datos de ejemplo), ficheros de imágenes de pruebas 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. **[3,5 puntos]** El sistema dispone de una página inicial con un formulario de autenticación (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: autenticación y registro **(0.75 puntos)**. Como parte del proceso de registro de usuario, se debe crear una carpeta para las fotos del usuario dentro de la carpeta "fotos" 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 hacer el *hash* y comprobar el mismo, 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 **(0.25 puntos)**.

Una vez realizada con éxito la autenticación, se definirá una variable de sesión para conocer en el resto de páginas que el usuario se ha autenticado con éxito. En la sesión se almacenará cuál es su nombre de usuario y su identificador de usuario (campos *nombre* e *id* de la tabla usuarios) **(0.5 puntos).** Se mostrará el contenido de la misma en el saludo al usuario en las diferentes páginas **(0.25 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 autenticación **(0.5 puntos)**. Se establecerá una *cookie* con duración de 30 días que recordará el usuario con el que se autenticó y lo mostrará prerelleno en el formulario de autenticación **(0.5 puntos)**.

2. **[2 puntos]** Una vez realizada la autenticación, la aplicación redirigirá al usuario a la página de listado de contactos **(0.25 puntos)**. En esta página se mostrará un listado en forma de tabla con todas los contactos del usuario autenticado de la tabla *contactos* de la BD (respetando el formato definido en la vista: *listado_contactos.html*) **(1 punto)**. Tenga en cuenta que cada contacto puede tener una foto que se mostrará **(0.25 puntos)**. Recuerde que si el usuario accediera a esta página o cualquier otra directamente (por ejemplo, escribiendo la URL) y no estuviera autenticado, redirigirá al usuario a la página de autenticación **(0.5 puntos)**.

3. **[4.5 puntos]** Amplíe el punto anterior, teniendo en cuenta que:
   - La página mostrará un *checkbox* en cada fila de la tabla de contactos para marcar contactos y un botón al final de la página para borrar los contactos marcados. Implementar el borrado de contactos como acción del botón. Se borrarán los registros de la base de datos **(0.75 puntos)** y el fichero físico de la foto si la tuviera **(0.75 puntos)**. Tras borrar los contactos, se mostrará de nuevo el listado actualizado con los contactos existentes **(0.25 puntos)**.
   - Al final de la página habrá un botón para añadir un contacto nuevo. Al pulsar sobre él nos mostrará una página con un formulario, con los siguientes campos y sus respectivas restricciones (vista: *alta_contacto.html*) **(1,5 puntos)**:
     - Nombre: es obligatorio.
     - Teléfono: es obligatorio y ha de tener 9 dígitos y empezar por 6, 7 o 9.
     - Email: es obligatoria y ha de cumplir el formato correcto de un email.
     - Foto: debe ser un archivo de tipo PNG o JPEG (compruebe el tipo MIME del mismo) y su tamaño máximo será de 500 KB. No es obligatorio adjuntar una foto, en cuyo caso el valor del campo que contiene el nombre del fichero en la base de datos quedaría como nulo. Tenga también en cuenta que hay que evitar colisiones de nombres al almacenar el fichero en el sistema, por lo que habrá que tomar las precauciones necesarias para ello.

   Si se cumplen todos los criterios, se insertará el contacto **(1 punto)** y se mostrará de nuevo el listado de contactos **(0.25 puntos)**.

Material suministrado

Como material adicional dispondrá del fichero "Material Adicional.7zip" que contendrá:

- Carpeta **BBDD**, fichero *agenda.sql*: Script de creación de la base de datos. Existe ya creado un usuario "pepe" con clave "pepe".
- Carpeta **Vistas**, ficheros *login.html*, *registro_usuario.html*, *listado_contactos.html* y *alta_contacto.html*, con el código HTML de cada vista. Carpeta "fotos" con las fotografías de los contactos de ejemplo.
- Carpeta **Ficheros de prueba**: donde encontrará ficheros *png* y *jpeg* que cumplen las validaciones para ser usados y otros que no para probar.
- Carpeta **Documentación**, fichero *Manual_password_hash_verify.pdf:* con información sobre las funciones *password_hash* y *password_verify*.

Datos de la prueba

Autor del documento: Daniel Prieto Tagua (Mayo 2019).

Revisiones del documento

1. Carlos D. Barranco (Mayo 2019). Correcciones menores.