Programación Avanzada
Ingeniería Informática en Sistemas de Información - Curso 2018/2019
ENSEÑANZAS PRÁCTICAS Y DE DESARROLLO
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 tareas diarias. Existen dos tipos de usuarios que accederán a la web: desarrolladores y jefes.

Los **desarrolladores** pueden **registrarse** indicando su nombre, contraseña, email y perfil (P: programador, AP: analista-programador, A: analista). Tras esto, pueden autenticarse y, una vez dentro, pueden seleccionar qué tareas van a resolver, en una única **página de listado de tareas**, donde se mostrarán todas las tareas en una tabla (título, descripción, enlace fichero anexo, tiempo y nombre del usuario que tiene asignada la tarea). Mediante *checkboxes*, el desarrollador elige las tareas que va a realizar en el día y pulsa un botón. El sistema le muestra cómo quedarían sus tareas en una **página de resumen tareas seleccionadas**, con detalle de cada tarea seleccionada y el tiempo total. Ésta es solo una página informativa. Cada tarea puede tener un tiempo diferente según el perfil (los programadores pueden tardar más en realizar una tarea que los analistas). Por tanto, el sistema debe calcular el tiempo de cada tarea (en la tabla resumen) en función del perfil del desarrollador (se introdujo en su formulario de registro). Algunas tareas pueden no estar disponibles para todos los perfiles. En el caso de que alguna tarea elegida por el desarrollador no esté disponible para su perfil, se indicará en la web (con detalle de qué tarea no está disponible) y no se incluirá en el tiempo total.

Los jefes, una vez se autentiquen como tales, acceden a la página de listado de tareas para su perfil. Éstos pueden borrar tareas marcando las filas de la tabla que deseen y pulsando en un botón para borrar. Además, pueden añadir tareas con un botón de añadir que lleva a un formulario de alta de tarea que permite introducir los datos de la nueva tarea. Los datos que almacena el sistema de las tareas son los siguientes: título, descripción, perfiles, tiempos, nombre del fichero adjunto y usuario que tiene asignada una tarea. En concreto, estos dos últimos campos son cadenas de caracteres con los diferentes perfiles y sus tiempos asociados separados por punto y coma. Tenga en cuenta que los tiempos se guardarán en minutos. Por ejemplo, una tarea podría ser {“Autenticación”, “Sistema de autenticación en la web para desarrolladores”, “(nombre fichero anexo)”, “P;A”, “120;90”} (por ejemplo, esta tarea no puede ser realizada por un AP). El formulario de alta de tarea incluirá mediante *checkboxes* los diferentes perfiles posibles (P, AP, A) y un cuadro de texto junto a cada uno para incluir un tiempo. El jefe marcará las tareas que estén disponibles según el perfil y escribirá el tiempo de cada una.

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 pdf para anexar a una tarea 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. **[2.25 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 **(0.5 puntos)**. Para ello, tenga en consideración que no se permiten registros de usuarios *jefes*, solo de usuarios tipo *desarrollador*. 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 **(0.1 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, cuál es su nombre de usuario y el tipo de usuario (campo *nombre* y *tipo* de la tabla usuarios) **(0.4 puntos)**, mostrándose el contenido de la misma en el saludo al usuario en las diferentes páginas **(0.2 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 **(0.4 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 **(0.4 puntos)**.

2. **[1.5 puntos]** Una vez realizada la autenticación, la aplicación redirigirá al usuario a la página de listado de tareas **(0.25 puntos)**. En esta página se mostrará un listado en forma de tabla con todas las tareas de la tabla *tareas* de la BD (respetando el formato definido en la vista: *listado_tareas.html*) **(1 punto)**. Tenga en cuenta que cada tarea puede tener un anexo que se mostrará como un enlace, que al pulsar en él abra el fichero asociado **(0.15 puntos)**. Recuerde que si el usuario accediera a esta página directamente (por ejemplo, escribiendo la URL) y no estuviera identificado, redirigirá al usuario a la página de autenticación **(0.1 puntos)**.

3. **[4 puntos]** Amplíe el punto anterior, teniendo en cuenta que la página mostrará una información diferente dependiendo del tipo de usuario que haya accedido a la página, teniendo en cuenta las siguientes consideraciones por tipo de usuario:
   1. Usuarios ***jefes*** (el campo *tipo* de la tabla *usuarios* de la BD es *"jefe"*), se mostrará:
      - Un *checkbox* en cada fila de la tabla de tareas para marcar una tarea y un botón al final de la página para borrar las tareas marcadas. Implementar el borrado de tareas como acción del botón. Se borrarán los registros de la base de datos **(0.75 puntos)** y los ficheros físicos anexos a una tarea si los hubiera **(0.75 puntos)**. Tras borrar las tareas, se mostrará de nuevo el listado actualizado con las tareas existentes **(0.25 puntos)**.
      - Al final de la página habrá un botón para añadir una tarea nueva. Al pulsar sobre él nos mostrará una página con un formulario, con los siguientes campos y sus respectivas restricciones (vista: *alta_tarea.html*) **(1 punto)**:
        - Titulo: no podrá dejarse en blanco.
        - Descripción: no podrá dejarse en blanco.
        - Perfiles: *checkboxes* con las etiquetas P, AP y A.
        - Tiempo: ninguna tarea puede ser superior a 480 minutos (8 horas). Son obligatorios solo aquellos tiempos cuyo *checkbox* de perfil esté marcado. Por tanto, dará error si hay tiempo y no está marcado el checkbox.
        - Anexo: debe ser un archivo de tipo PDF (compruebe el tipo MIME del mismo) y su tamaño máximo será de 500 KB. No es obligatorio adjuntar un fichero anexo, en cuyo caso el valor del campo que contiene el nombre del fichero en la base de datos quedaría como null. Tenga también en cuenta que hay que evitar colisiones de nombre al almacenar el fichero en el sistema, por lo que tomar las precauciones necesarias para ello.

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

2. Usuarios ***desarrollador*** (el campo *tipo* de la tabla *usuarios* es *"desarrollador"):* se mostrará en el listado de tareas un *checkbox,* en las filas de las tareas que no tengan usuario asignado*,* para marcar tareas y un botón al final de la página para mostrar un resumen de las tareas seleccionadas. Al pulsar sobre él nos redirigirá a la página de resumen de tareas (en la siguiente Actividad 4) **(0.5 puntos)**.

4. **[2.25 puntos]** Crear una página de resumen de tareas asignadas que se genere dinámicamente cuando el usuario pulse en el botón de seleccionar actividades (Actividad 3, apartado 2). Esta página de resumen mostrará una tabla con las tareas previamente elegidas por el usuario y el tiempo de cada una de ellas según el perfil del usuario que esté autenticado **(0.75 puntos)**. Al final de la página, debe aparecer el tiempo total de las tareas asignadas **(0.5 puntos)**. Al pulsar en un botón al final de la página las tareas asignadas se relacionarán con el usuario que las seleccionó. Para ello, actualice el nombre del usuario en el campo *nombreusuario* de la tabla *tareas*, si ya tuviera uno se sobrescribirá. **(0.8 puntos)**. Finalmente, el usuario será redirigido a la página de listado de tareas. **(0.2 puntos)**

Material suministrado

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

- Carpeta **BBDD**, fichero *gestortareas.sql*: Script de creación de la base de datos. Existe ya creado un usuario "admin" con clave "admin" de tipo jefe y un usuario de tipo desarrollador "pepe" con clave "pepe".
- Carpeta **Vistas**, ficheros *login.html*, *registro_usuario.html*, *listado_tareas_desarrollador.html, listado_tareas_jefe.html, resumen_tareas.html* y *alta_tarea.html*, con el código HTML de cada vista. Carpeta "anexos" con ficheros adjuntos a las tareas de ejemplo.
- Carpeta **Ficheros de prueba**: donde encontrará ficheros pdf que cumplen las validaciones para ser anexados 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 (Noviembre 2018). Basado en el examen de 2017.

Revisiones del documento

1. Carlos D. Barranco (Noviembre 2018): Correcciones menores de contenido y formato.