

EXAMEN 1ª EVALUACIÓN

Desarrollo de una Aplicación de Gestión de Tareas en PHP y MySQL

En este ejercicio, crearás una aplicación web de gestión de tareas que permite a los usuarios agregar, editar, eliminar y marcar tareas como completadas. A continuación, se detalla el enunciado:

- Crea una base de datos llamada “tareas”.

```
CREATE DATABASE IF NOT EXISTS tareas;
```

- Añade la tabla “usuarios” que tenga las siguientes columnas: id (clave primaria), nombre completo, correo, contraseña y la ruta a la imagen de perfil del usuario.

```
CREATE TABLE IF NOT EXISTS usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    correo VARCHAR(255) NOT NULL,
    contrasena VARCHAR(255) NOT NULL,
    ruta_img VARCHAR(255) NOT NULL
);
```

- Añade una tabla “tareas” que tenga las siguientes columnas: id (clave primaria), nombre, descripción, fecha creación, fecha modificación, fecha finalización, completada, usuario que la ha creado, modificado y/o completado.

```
CREATE TABLE IF NOT EXISTS tareas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    descripcion TEXT,
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    fecha_modificacion DATE,
    fecha_finalizacion DATE,
    completada BOOLEAN NOT NULL DEFAULT 0,
    id_usr_crea INT,
    id_usr_mod INT,
    id_usr_comp INT,
    FOREIGN KEY(id_usr_crea) REFERENCES usuarios(id),
    FOREIGN KEY(id_usr_mod) REFERENCES usuarios(id),
    FOREIGN KEY(id_usr_comp) REFERENCES usuarios(id)
);
```

- Crea un usuario “usr_tareas” (con contraseña “usr_tareas”) para conectar a la base de datos “tareas”.

```
CREATE USER 'usr_tareas'@'localhost'
IDENTIFIED BY 'usr_tareas';
GRANT ALL PRIVILEGES ON tareas.* TO 'usr_tareas'@'localhost' WITH
GRANT OPTION;
```

- Desarrolla la aplicación utilizando programación orientada a objetos, para ello crea al menos tres clases, “Tarea”, “Usuario” y “Conexion” (a base de datos) con sus respectivos atributos y métodos. Todas las operaciones que se realicen sobre tareas, usuarios o de conexión a la base de datos deberán ser gestionadas por métodos de las clases, si no es así se considerará que la clase está mal creada.
- Desarrolla una página principal que muestre la lista de tareas en formato de tabla, indicando en la cabecera de la tabla qué información contiene cada columna. Las tareas deben ordenarse por fecha de creación de forma descendente.
- Proporciona la posibilidad de añadir tareas, para ello crea un formulario que permita a los usuarios agregar nuevas tareas. El formulario debe incluir como obligatorios los campos para el nombre y la descripción de la tarea. La fecha de creación y el usuario que la ha creado los llenará la aplicación automáticamente.
- Implementa la funcionalidad para editar tareas existentes. Cada tarea que aparezca en la lista de la página principal, y que no esté completada, deberá tener un botón de "Editar" que permita a los usuarios modificar cualquier campo o completar la tarea. Cuando se modifique o complete una tarea, se deberá guardar también la fecha en que se ha hecho el cambio y el usuario que lo ha realizado.
- Implementa la capacidad de eliminar tareas. Agrega un botón "Eliminar" que permite a los usuarios eliminar tareas que no esté completadas.
- Crea una página en la que se pueda realizar una búsqueda de tareas por los campos nombre o descripción. En la misma página se mostrarán las cinco últimas tareas añadidas, modificadas o completadas indicando qué usuario ha realizado la acción.
- Proporciona una página de inicio de sesión que permita a los usuarios autenticarse, utilizando su correo electrónico, para poder realizar operaciones en la aplicación de gestión de tareas. Si un usuario no está autenticado no podrá acceder a la aplicación. Añade la opción de “recordar al usuario”, si un usuario se ha autenticado con éxito anteriormente, la próxima vez que acceda a la aplicación le debe preguntar si quiere continuar accediendo con ese usuario o con uno nuevo.
- Incluye una página que permita a los usuarios registrarse en el sistema, introduciendo su nombre y apellidos, dirección de correo, contraseña y una foto de perfil. Añade una página en la que un usuario autenticado pueda consultar su perfil con todos sus datos y modificarlos (imagen incluida).
- Incorpora una cabecera en la aplicación que contenga un menú de navegación con el que se pueda acceder a todas las páginas que hayas desarrollado para esta aplicación. La cabecera deberá ser visible en todas las páginas, además, deberá mostrar el nombre del usuario y la foto de perfil junto con la opción de cerrar sesión.
- Utiliza PDO para conectar a la base de datos y crea sentencias preparadas en todos los accesos que se realicen a base de datos con parámetros.
- Controla todos los posibles errores y excepciones en toda la aplicación.

PUNTUACIÓN: La puntuación se distribuirá de la siguiente manera:

Funcionalidad	Puntuación
Clases creadas	1 puntos
Lista tareas	1 punto
Añadir tareas	1 punto
Editar y completar tareas	1 punto
Eliminar tareas	0,5 puntos
Buscar tareas	1 punto
Página login	1 punto
Página registro	1 punto
Página perfil	0,5 puntos
Cabecera con menú y login/perfil/logout	1 punto
Uso de PDO y sentencias preparadas	1 punto

Si una funcionalidad está creada correctamente sumará la puntuación indicada. Para cada funcionalidad se restará 0,25 por cada excepción que aparezca y por cada fallo que se tenga, hasta el máximo de puntuación asignado a esa funcionalidad, es decir, si las clases creadas suman 1 punto, lo máximo que se puede restar con los errores en las clases es 1 punto.

ENTREGA: La entrega deberá consistir en un archivo zip con todo lo necesario para probar la aplicación. La aplicación deberá ser autocontenido, de manera que sea posible descomprimirla en un pc conxampp y no necesitar modificaciones del sistema y/o base de datos.