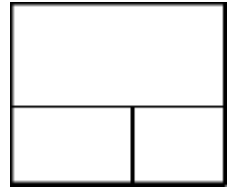


DWES Examen PHP 5-mar-2021



Se te habrá proporcionado un proyecto PhpStorm con este enunciado y tres carpetas, una por ejercicio. Las carpetas son la base sobre la que se recomienda realizar los ejercicios. Incluyen una exportación de la BD cuando es necesario.

[1] Editor online (3,00p)

(Solo PHP, sin DAO, sin Ajax...)

Programa una web que contenga un input en el que el usuario pueda introducir texto.

Al abrirse la web, si hay algún texto guardado de antes, se cargará en el input.

Habrà un botón que guardará el texto. Se pide que el texto perdure si se cierra y vuelve a abrir el navegador. Es imprescindible este comportamiento en el ejercicio. Utilizar un mecanismo adecuado para ello, cookies y/o sesiones.

Además, en CADA guardado, el servidor añadirá (anexará) un carácter “#” al final del texto que ya haya, como para confirmar el guardado.

Si al pulsar el botón el texto estaba vacío, la web debe volver a mostrar el formulario inicial con un mensaje de “Por favor, introduzca algo de texto”. No se pide controlar otro tipo de situaciones (han metido solo espacios, hay espacios a los lados, etc.)

Además, habrá un botón de Reiniciar que volverá al estado inicial, vacío.

Crea el fichero/ficheros que necesites. Si tu solución tiene copiapaga de código (por ejemplo, copiapaga del formulario en dos PHP separados), aspiras a 2 puntos. Si la haces monopágina, o bipágina pero evitando ese copiapaga, aspiras a la puntuación máxima.

[2] Concesionario DAO (3,50p)

(Sí se pide “DAO”, no se pide inicio de sesión ni nada de eso, ni se pide AJAX.)

Importa la BD incluida en el ZIP o, si tienes problemas al hacerlo, crea una nueva BD (schema) y una tabla en una base de datos: BD/schema: “concesionario”. Tabla: “coche”.

Ponle estos campos:

- id: INT, autoincrementable, no nulo, clave primaria
- matricula: VARCHAR(7)
- modelo: VARCHAR(30)
- precio: INT

Introduce algunos valores en la BD, o usa los de la BD incluida.

Se pide:

- Preparar toda la infraestructura que necesitaremos para crear los scripts/páginas que se piden. Esto incluye la(s) clase(s), el “DAO” y demás. Puedes utilizar, al igual que en el resto de ejercicios, material de clase para ello.
- (1,50p) Desarrollar un script para visualizar una tabla con un listado de todos los coches que hay en la BD.
 - La tabla incluirá la matrícula, el modelo y el precio.

- La tabla incluirá también un link para bajar el precio. El texto del link será “Rebajar”. Este link estará en una columna propia en la tabla.
- (No es necesario poner links para eliminar coches, ni dar la opción de editarlos como tal.)
- (2,00p) Desarrollar la funcionalidad para filtrar la tabla por precio:
 - Habrá un input text con placeholder “Ver menores a este precio” y al lado un botón “Filtrar”.
 - Al pulsar el botón de Filtrar se recargará la página del listado, visualizando solo los coches cuyo precio sea inferior o igual al indicado. El input text deberá volver a mostrar el precio con el que se filtró, de manera que si vuelven a pulsar el botón se vuelve a cargar el listado filtrado de la misma forma.
 - Habrá otro botón para quitar el filtrado (y, por tanto, mostrar ya todos los coches). Este botón provocará la recarga de la página con todos los coches y con el input text vacío.

Queremos utilizar el modelo DAO y queremos implementar las funciones de la manera más organizada posible, con el objetivo de crear una aplicación fácil de mantener y modificar, evitando copiapegas, etc.

Organiza el código lo mejor que puedas, dentro de estas premisas. Un objetivo en este ejercicio es demostrar que sabes organizar código. Por tanto, conseguir la funcionalidad indicada sin organización no es valorable.

[3] **AJAX (3,50p)**

Se proporciona la aplicación funcionando según lo visto en clase.

Se pide implementar la siguiente función, de forma lo más “Ajax” posible, es decir, evitando uso de la red y distribuyendo las responsabilidades entre los distintos agentes de la forma más útil.

Tal y como está la Agenda, si se elimina una categoría que tenga asignadas personas, se eliminan también dichas personas de la BD por el ON DELETE CASCADE (pero no del DOM, con lo que se produce una situación extraña).

Queremos:

- Que cuando el usuario pulse el botón de eliminar una categoría se realice una comprobación: se mirará a ver si esa categoría tiene personas asignadas o no.
- Si no tiene a nadie asignado, se eliminará la categoría y ya está. Como hasta ahora.
- Sin embargo, si tiene al menos una persona asignada, NO se procederá a eliminar la categoría. En este caso, se deberá informar al usuario de lo que ocurre.
- Se pide añadir lo necesario en HTML/JS/PHP/DAO, creando funciones, métodos y ficheros que sean adecuados.
- Puedes, por ejemplo, devolverá al cliente un valor de control (un false, un string “PERSONAS_ASIGNADAS”... Decídelo tú) que permita al cliente detectar esta situación e informar de ello al usuario de alguna manera (un alert()) se considerará válido en ese sentido, en este ejercicio).

En general, para todo el examen: implementa primero lo básico y luego lo accesorio. Implementa primero los puntos que te resulten más rentables y luego los que menos. No te lées con detalles de control de errores si todavía no tienes la funcionalidad principal. Se pide explícitamente organizar el código según los modelos vistos en clase, u otros que sean objetivamente mejores en términos de calidad del código, simplicidad, genericidad y facilidad para su modificación. Dicho de otro modo, “que funcione” no es el objetivo principal de los ejercicios.