

Instrucciones

- Crear una carpeta que se llamará “Apellidos_Nombre”, donde guardaréis las soluciones de los problemas
- Para cada problema crearéis una carpeta que llamaréis “Problema1”, “Problema2”, etc, donde guardaréis las soluciones.
- Una vez hayáis finalizado debéis subir el examen al aula virtual comprimido en un fichero con el formato “Apellidos_Nombre”.
- El examen puntúa sobre 10, selecciona los ejercicios como quieras.

Consejos

- Guardar cada poco el trabajo que estáis realizando.
- Mirar la puntuación de los problemas y centraros primero en el aprobado. Luego incluíis mejoras.
- El programa es personal e intransferible, por lo que no deberíais tener que mirar para nada al monitor del compañero.
- Escoger en primer lugar los problemas que tienen fácil solución, asegurad el aprobado lo primero.
- Recordad que podéis hacer uso de código que ya tenéis creado

Problema 1 – Operaciones CRUD – 6,5 Ptos

El nombre del fichero con el que arranca el programa será **index.php**. En este fichero incluiréis al comienzo un comentario en el que indicaréis para que sirven el resto de ficheros utilizados en la solución.

Dada la siguiente definición de la tabla “diario”, ubicada en la base de datos “diario” cuya descripción tenéis en el fichero “diario.sql” de la carpeta “Ej1” del examen.:

```
CREATE TABLE `diario` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `fecha` date DEFAULT NULL,
  `anotacion` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Menú inicio

- [Inicializar año 2015](#)
 - [Añadir entrada de diario](#)
 - [Ver todas las entradas](#)
 - Ver entradas de mes:
- [Ver por mes]**

Se desea crear una aplicación en PHP que se inicie mostrando el menú de la izquierda y que permitirá realizar las siguientes operaciones sobre dicha tabla:

Inicializar año 2015

Esta opción automáticamente borrará todas las entradas que contiene la tabla e insertará **una entrada para cada día** con los siguientes datos en los campos:

- *fecha*: el día (yyyy-mm-dd)
- *anotacion*: “anotación del día DD de MMM”

Donde DD es el día del mes y MMM la abreviatura en español del mes (Ene, Feb, ...).

Añadir entrada de diario**Añadir**

Fecha (aaaa-mm-dd) ****Error****

Anotación ****Error****

[Guardar]

***** Se ha guardado la anotación enviada *****

[Volver al menú](#)

Esa opción nos permitirá añadir información a nuestro diario. Se deberán filtrar los campos para evitar que se meta información incorrecta. El filtro a aplicar será:

- Fecha: Debe tener un valor y un formato válido. Utilizaremos un campo de tipo `<input type="text"/>`.
- Anotación: Debe tener un valor

En caso de error se mostrarán de nuevo el valor de los campos y un mensaje de error junto al campo. Una vez guardada una anotación se

mostrará un mensaje indicando este hecho debajo del botón guardar y se permitirá seguir añadiendo más entradas al diario, dejando los campos vacíos.

Ver todas las entradas

Esta acción nos mostrará la lista de todas las entradas existentes ordenada por fecha paginando los resultados.

Ver entradas de mes **[Ver por mes]**

Esta opción se disparará al pulsar el botón [Ver por mes] y mostrará las entradas que se hayan realizado en el mes seleccionado en el campo de tipo select, que contendrá el nombre o abreviatura de los meses (Ene, Feb, Mar.....)

Dentro de cada opción se incluirá un mensaje que me permitirá volver al menú.

Puntuación sobre 10:

- Inicializar año 2015 – 2,5 pts – Inicialización con valor incorrecto fecha, o anotación 0,5-1,5 pts
- Añadir entrada – 1,5 pts
- Añadir entrada. Filtrar entrada – 1,5 pts
- Ver todas las entradas – 1 pts
- Ver todas las entradas. Pagar – 1,5 pts
- Ver entradas por mes – 1 pts
- Valoración global aplicación completa – 1 pto

Notas:

- Las siguientes funciones pueden resultaros interesantes *mktime*, *checkdate*, *explode* en PHP y *month* en mysql
- El contenido del fichero *meses.php* situado en la carpeta ej1 puede resultaros útil

Problema 2 – Operaciones B.D. con MVC y Controlador frontal (4 pts)

El nombre del fichero con el que arranca el programa será **index.php**. En este fichero incluiréis al comienzo un comentario en el que indicaréis para que sirven el resto de ficheros utilizados en la solución.

Utilizando como base el ejemplo de programa con controlador frontal con POO que hemos visto en clase y que tenéis en la carpeta "ej2" del examen, se desea que ampliéis/modifiquéis dicho programa para que permita realizar las siguientes operaciones adicionales.

1. El programa deberá almacenar los datos en la base de datos "taresexamen" cuya estructura se describe en el fichero "tareas_examen.sql".
2. Obligatoriamente se utilizará una capa de abstracción de base de datos para acceder a la base de datos.
3. Se desea implementar una operación de borrado, con confirmación en el lado servidor, que permita borrar una tarea. Después de borrarla se mostrará la lista.
4. Se desea modificar las operaciones "Alta" y modificar de forma que después de completar la operación nos muestre la lista, y no un mensaje de confirmación como hace actualmente.
5. Se desea modificar el filtro de la operación "Alta" (**solo en el alta**) de forma que no permita introducir dos tareas con el mismo nombre.

Puntuación:

- Realizar cambios para que lea y almacene en la base de datos (1 pto)
 - Listar 0,5 pts.
 - Alta: 0,25 pts
 - Modificar 0,25 pts
- Implementar operación de borrar (1,5 pts)
 - Sin confirmación 0,5 pts.
 - Con confirmación 1,5 pts.
- Mostrar lista en operaciones Alta y Modificar después de insertar: 0,5 pts
- Implementar filtro que evite nombres tareas repetidos en inserción 1 pto.

Problema 3 – Sesiones 2,5 pts.

El nombre del fichero con el que arranca el programa será **index.php**

Nos han encargado que realicemos una aplicación web que **sin utilizar bases de datos**, tenga la siguiente presentación y realice lo siguiente.

Cuando se escriba una URL en el campo de texto, y se pulse el botón "Añadir nuevo diario a lista" se almacenará la URL en una lista que se mostrará debajo separada por comas (Diarios almacenados).

Cuando se pulse el botón "Pulse aquí para seleccionar un diario aleatorio" se seleccionará de forma aleatoria, alguno de los diarios

que hemos guardado en la lista y se mostrará como enlace detrás del texto "Diario seleccionado", dicha selección deberá permanecer aunque añadamos nuevos diarios y solamente cambiará al pulsar de nuevo el botón "Pulse aquí para seleccionar un diario aleatorio".

Además, incluiremos un **enlace Borrar todo** que nos borra toda lista de diarios seleccionados, y la selección realizada.

Notas: Puede que las sesiones os resulten útiles para resolver el problema. La función *rand* de php permite generar números aleatorios

Puntuación (sobre 2,5 pts):

- Guardar diario en lista y mostrarlo: 0,75 pts
- Seleccionar diario de forma aleatoria y mostrar enlace: 0,75 pto.
- Recordar diario aunque se añadan nuevos elementos: 0,5 pts.
- Implementar funcionalidad de borrar todo 0,5 pts.

Problema 4 – Formularios 1,5 pts.

El nombre del fichero con el que arranca el programa será **index.php** En este fichero incluiréis al comienzo un comentario en el que indicaréis para que sirven el resto de ficheros utilizados en la solución.

MemorizadorAnotar nombre:

Nombres anotados: * Ninguno * / José A., Manuel, Maria

[Borrar todo](#)

Sin utilizar sesiones, cookies, base de datos, ficheros o javascript se desea realizar una página web que realice y muestre lo siguiente.

La aplicación permitirá ir almacenando nombres que se introducen en un campo de texto. Cada vez que se pulse el botón anotar se almacenará el nombre y se mostrará en una lista debajo separados por comas. Si la lista está vacía se mostrará el valor "* Ninguno*". Al pulsar el enlace "[Borrar todo](#)" se perderán todos los enlaces

almacenados.

Nota: Se pretende saber si sabéis manejar los campos en un formulario

Puntuación:

- Almacenar y mostrar nombres 1 pts.
- Borrar todo 0,5 pts.

Problema 5 – 1,5 pts.

Apartado 1 (1 pto) - Deberéis modificar el fichero "ej5/apartado_1.php" para que al ejecutarlo los vehículos no superen los siguientes límites de velocidad:

Solamente se podrá modificar el código que existe entre las etiquetas **<Modificar>** y **</Modificar>**. Cualquier otro tipo de modificación adicional puntuará con 0 en el ejercicio.

Tipo Vehículo	Vel. Máx
Camión	100
Moto	120
Deportivo	150

Apartado 2. (0,5ptos) Deberéis modificar el fichero "ej5/apartado_2" para conseguir lo mismo que en el apartado 1. El fichero tan solo cambia respecto al apartado 1 en el modificador de una variable.

Un buen planteamiento en el apartado 1 resuelve también el dos. No tienen por que ser diferentes.

Nota: Este ejercicio tiene como propósito evaluar vuestra comprensión de la herencia, el polimorfismo, los modificadores de acceso y la invocación de métodos de la clase padre.