

UT02 - Supuesto Práctico: Uso de los elementos del lenguaje de lado servidor PHP para programar aplicaciones web empleando formularios.

Se desea crear una aplicación de gestor de horarios para docentes, para ello disponemos de un formulario web con los siguientes campos **“curso”, “día”, “hora”, “materia”, “clase” y “color”**. Se proporcionan los enumerados con la información relativa a esos campos, esta información estará disponible en el fichero **“campos.php”**.

Desarrollar las siguientes clases que se proporcionan con el problema:

- La clase **“Modulo”** en el fichero **“modulo.php”** la componen los campos **“Curso”, “Clase” y “Materia”**.
- La clase **“FranjaHorario”** en el fichero **“FranjaHorario.php”**, contendrá todos los campos de la clase **“Modulo”** más los campos **“día”, “hora” y “color”**.
- La clase **“GestorHorario”** en el fichero **“gestorhorario.php”**, dispondrá de las siguientes funciones:
 - Función **“insertarHora”**, a esta función se le pasará un parámetro de tipo **“FranjaHorario”** que contendrá la información almacenada de todos los campos del formulario. La información contenida en este parámetro debe almacenarse en un fichero denominado **“horarios.dat”** en el directorio **“horarios” (Si no existe el directorio o el fichero se deben crear)**, cada **campo** se almacenará separado por **“;”** y cada **registro de franja horaria**, se diferenciará mediante el carácter **“@”**. En el fichero resultante debe quedar el siguiente formato para cada registro:

Curso;Dia;Hora;Materia;Clase;Color@Curso;Dia;Hora;Materia;Clase;Color@

Ejemplo)

1ADAW;L;1;PRO;R02;#FC9C1F@2ADAW;M;2:DSW:R02;#FF11CF@

Si la franja horaria existe (día y hora), se debe mostrar la siguiente excepción: **“Error Horario: La franja de hora ya existe, elige otra que esté disponible.”**.

- Función **“eliminarHora”** a la que se le pasará un parámetro de tipo **“FranjaHorario”**. Este parámetro contendrá la información relativa al **“día”** y la **“hora”**. Dada esta información se procederá a eliminar el registro correspondiente dentro del fichero de datos.

Si la franja horaria no existe (día y hora), se debe mostrar la siguiente excepción: **"Error Eliminar Hora: La hora y el día seleccionado no existe, no se puede eliminar."**

- Función **"mostrarHorario"** sin parámetros. Esta función generará una tabla en HTML de forma dinámica en **"horarios.php"**, con la información contenida en el fichero **"horarios.dat"**. Cada casilla del horario mostrará la información del **"Curso"**, **"Materia"**, **"Clase"** y además se le deberá aplicar el color de fondo correspondiente. Esta función se aplicará cada vez que se cargue el formulario en el fichero **"horarios.php"**. (Si no existen datos que cargar en la tabla, se genera la tabla en HTML con las horas y días disponibles y las casillas vacías).
- Función **"subirFichero"** a la que se le pasará un parámetro con la ruta del fichero subido denominado **"rutaFicheroSubido"**. La función sobrescribirá el contenido del fichero **"horarios.dat"** con el contenido del nuevo fichero subido.

Completar el fichero **"horarios.php"**, seguir la estructura establecida, el formulario dispone de varios selectores que se deberán rellenar con los datos de los enumeradores del fichero **"campos.php"** y tres botones:

- El botón **"Insertar Hora"** realiza una petición al módulo denominado **"horario.php"**, se deberán validar todos los campos, verificar que se envían los datos, y que no son vacíos. Los campos vacíos o no inicializados deben mostrar una excepción con el siguiente texto: **"Error Campos: El campo no puede estar vacío"**. Este botón insertará la información de la franja horaria en el cuadro horario. En el caso de que se ejecute todo correctamente, se deberá volver a la página de **"horarios.php"**.
- El botón **"Eliminar Hora"** realiza una petición al módulo denominado **"horario.php"**, se deben validar los datos de los campos **"día"** y **"hora"**, verificar que se envían los datos y que no son vacíos. Si se envían campos vacíos o no inicializados, se debe mostrar una excepción con el siguiente texto: **"Error Campos: El campo no puede ser vacío"**. Esta acción eliminará la franja horaria del cuadro horario. En el caso de que se ejecute todo correctamente, se deberá volver a la página de **"horarios.php"**.
- El botón **"Cargar Horario"**, realiza una petición al módulo denominado **"horario.php"**, se deberán validar si existe o no el fichero, verificar que se envía el fichero, y que no es vacío. El fichero subido se almacenará en el directorio **"datos_temporales"** y se denominará **"datos_temp.dat"**. Si no existe el directorio se debe crear. Una vez subido el fichero al servidor y guardado en el directorio correspondiente, se deberá realizar un volcado en el fichero de datos de horarios **"horarios.dat"**. En el caso de que se

ejecute todo correctamente, se deberá volver a la página de “**horarios.php**”.

Establecer un fichero denominado “**validaciones.php**” dónde se pondrán todas las validaciones referentes a los campos y a la subida de los ficheros.

Puntuaciones e Indicaciones:

Fichero “modulo.php” correctamente implementado. (Clases, Getter, Setter, Validaciones, etc.) **0,25 ptos**

Fichero “franjaHorario.php” correctamente implementado. (Clases, Getter, Setter, Validaciones, etc.) **0,25 ptos**

Fichero “gestorHorario.php” correctamente implementado: **(4,5 ptos)**

- Validaciones, constructores, excepciones, Getter, Setter, etc. **0,5 ptos**
- Función “insertarHora” correctamente implementada. **1 pto**
- Función “eliminarHora” correctamente implementada. **1 pto**
- Función “mostrarHorario” correctamente implementada. **1 pto**
- Función “subirFichero” correctamente implementada. **1 pto**

Fichero “validaciones.php” correctamente implementado, contemplando todas las validaciones del documento. **2 ptos**

Fichero “horario.php” correctamente implementado, contemplando todas las validaciones del documento, excepciones, validación de campos, etc. **1 pto**

Fichero “horarios.php” correctamente implementado, contemplando todos los campos y mostrando los datos correctamente. **1 pto.**

Indentación, comentarios y estructuración de código correctamente implementado. **1 pto.**

Indicaciones:

- No se podrán modificar la estructura de los ficheros proporcionados.
- Los elementos que se incluyen en los ficheros deben permanecer inalterados.
- No se pueden modificar parámetros, ni nombre de funciones, ni elementos HTML, etc.
- No se puede utilizar Javascript.
- No se pueden añadir nuevos ficheros.
- El uso inadecuado de elementos del código que no estén correctamente establecidos puede dar lugar a penalizaciones en la nota. (Nombre de variables que no corresponden conceptualmente con los datos, elementos agregados como HTML que no corresponden con la lógica del programa, etc.).
- Se pueden añadir funciones auxiliares dentro de las clases o ficheros.

- Pueden utilizar internet, documentos, manuales, etc, durante el desarrollo de la prueba.
- Si se detecta alguna copia. El alumno y/o alumnos implicados en dicha copia, tendrán un 0 en la nota del supuesto, lo que implica la suspensión de la unidad.
- El trabajo y entrega del supuesto es una tarea individual.

Indicaciones de la entrega:

La tarea se debe entregar en CAMPUS en tiempo y forma.

Entregar un fichero .zip con la estructura de ficheros proporcionada, el nombre del fichero se establece con el siguiente formato:

apellidos_nombre_dsw_ut02_Supuesto.zip