PRIMER PARCIAL – PROGRAMACION III – 1 cuat. 2020

Aclaración:
Las partes se corregirán de manera secuencial (ascendentemente). Si están bien todos los puntos de una parte, habilita la corrección de la parte posterior.
Se debe crear un archivo por cada entidad de PHP. Todos los métodos deben estar declarados dentro de clases.
PDO es requerido para interactuar con la base de datos.
Se deben respetar los nombres de los archivos, de las clases y de los métodos.

Parte 1 (hasta un 4)

Usuario.php. Crear, en **./clases**, la clase **Usuario** con atributos privados (email y clave), constructor (que inicialice los atributos), un método de instancia ToString(), que retornará los datos de la instancia (en una cadena con formato email - clave).
Agregar:
Método de instancia GuardarEnArchivo(), que agregará al usuario en **./archivos**/usuarios.txt. Retornará un mensaje indicando lo acontecido.
Método de clase TraerTodos(), que retornará un array de objetos de tipo Usuario.
Método de clase VerificarExistencia($usuario), retornará true, si el usuario está registrado (invocar a TraerTodos), caso contrario retornará false.

UsuarioAlta.php: Se recibe por POST el email y la clave. Invocar al método GuardarEnArchivo.

VerificarUsuario.php: Se recibe por POST el email y la clave, si coinciden con algún registro del archivo de texto (VerificarExistencia), crear una COOKIE nombrada con el email del usuario que guardará la fecha actual (con horas, minutos y segundos). Luego ir a *ListadoUsuarios.php*.
Caso contrario, retornar un mensaje indicando lo acontecido.

ListadoUsuarios.php: (GET) Se mostrará el listado de todos los empleados.

MostrarCookie.php: Se recibe por GET el email del usuario y se verificará si existe una cookie con el mismo nombre, de ser así, retornará un mensaje, dónde se mostrará el contenido de la cookie. Caso contrario, un mensaje indicando lo acontecido.
Nota: *Reemplazar los puntos por guiones bajos en el email (en caso de ser necesario).*

Parte 2 (hasta un 6)

Televisor.php. Crear, en **./clases**, la clase **Televisor** con atributos públicos (tipo, precio, paisOrigen y path), constructor (con todos sus parámetros opcionales), un método de instancia ToString(), que retornará los datos de la instancia (en una cadena con formato tipo – precio – paisOrigen – path).

Crear, en **./clases**, la interface ***IParte2***. Esta interface poseerá los métodos:

- **Agregar**: agrega, a partir de la instancia actual, un nuevo registro en la tabla **televisores** (id, tipo, precio, país, foto), de la base de datos **productos_bd**. Retorna **true**, si se pudo agregar, **false**, caso contrario.
- **Traer**: retorna un array de objetos de tipo Televisor, recuperados de la **base de datos**.
- **CalcularIVA**: retorna el precio del televisor más el 21%.
- **Verificar**: retorna **true**, si la instancia actual no existe en el array de objetos de tipo Televisor que recibe como parámetro. Caso contrario retorna **false**. Se compara por tipo y pais.

Implementar la interface en la clase Televisor.

AgregarTelevisorSinFoto.php: Se recibe por POST el tipo, el precio y el paisOrigen.
Verificar la previa existencia del televisor invocando al método *Verificar*. Se le pasará como parámetro el array que retorna el método *Traer*.
Si el televisor ya existe en la base de datos, se retornará un mensaje por pantalla que indica lo acontecido.
Si el televisor no existe, se invocará al método *Agregar*.

Listado.php: (GET) Se mostrará el listado **completo** de los televisores (obtenidos de la base de datos) en una tabla (HTML con cabecera). Invocar al método Traer. Mostrar además, una columna extra con los precios con IVA incluido.
***Nota**: preparar la tabla (HTML) para que muestre la imagen de la foto (si es que la tiene).*

Parte 3 (hasta un 8)

Crear, en **./clases**, la interface ***IParte3***. Esta interface poseerá los métodos:

- **Modificar**: Modifica en la base de datos el registro coincidente con la instancia actual. Retorna **true**, si se pudo modificar, **false**, caso contrario.

Implementar la interface en la clase Televisor.

AgregarTelevisor.php: Se recibirán por POST todos los valores (incluida una imagen) para registrar un televisor en la base de datos.
Se invocará al método Agregar.
La imagen guardarla en "./televisores/imagenes/", con el nombre formado por el **tipo** punto **paisOrigen** punto hora, minutos y segundos del alta (***Ejemplo: led.china.105905.jpg***).
Si se pudo agregar se redirigirá hacia Listado.php. Caso contrario, se mostrará un mensaje indicando lo acontecido.

ModificarTelevisor.php: Se recibirán por POST todos los valores (incluida una imagen) para modificar un televisor en la base de datos. Invocar al método Modificar.
Si se pudo modificar en la base de datos, la foto modificada se moverá al subdirectorio "./televisoresModificados/", con el nombre formado por el **tipo** punto **paisOrigen** punto '***modificado***' punto hora, minutos y segundos de la modificación (***Ejemplo: lcd.taiwan.modificado.105905.jpg***). Redirigir hacia Listado.php.
Si no se pudo modificar, se mostrará un mensaje indicando lo acontecido.

Parte 4 (hasta un 10)

Crear, en **./clases**, la interface ***IParte4***. Esta interface poseerá los métodos:

- **Eliminar**: elimina de la base de datos el registro coincidente con la instancia actual. Retorna **true**, si se pudo eliminar, **false**, caso contrario.
- **GuardarEnArchivo**: escribirá en un archivo de texto (televisores_borrados.txt) toda la información del televisor más la nueva ubicación de la foto. La foto se moverá al subdirectorio "./televisoresBorrados/", con el nombre formado por el **id** punto **tipo** punto '***borrado***' punto hora, minutos y segundos del borrado (***Ejemplo: 688.4k.borrado.105905.jpg***).

Implementar la interface en la clase Televisor.

EliminarTelevisor.php: Si recibe un **tipo** por GET, retorna si el televisor está en la base o no (mostrar mensaje).

Si recibe el **tipo** por POST, con el parámetro “**accion**” igual a "**borrar**", se deberá borrar el televisor (invocando al método Eliminar). Si se pudo borrar en la base de datos, invocar al método GuardarEnArchivo y redirigir hacia Listado.php.
Si no se pudo borrar, mostrar un mensaje indicando lo acontecido.
Si recibe por GET (sin parámetros), se mostrarán en una tabla (HTML) la información de todos los televisores borrados y sus respectivas imagenes.

TelevisorFiltrado.php: Se recibe por POST el **tipo**, se mostrarán en una tabla (HTML) los televisores cuyo tipo coincidan con el pasado por parámetro.
Si se recibe por POST el **paisOrigen**, se mostrarán en una tabla (HTML) los televisores cuyo país de origen coincida con el pasado por parámetro.
Si se recibe por POST el **tipo** y el **paisOrigen**, se mostrarán en una tabla (HTML) los televisores cuyo tipo y país de origen coincidan con los pasados por parámetro.

MostrarBorrados.php: Muestra por pantalla todo lo registrado en el archivo de texto “televisores_borrados.txt”. Para ello, agregar un método estático (en Televisor), llamado **MostrarBorrados**.