CFGS DESARROLLO APLICACIONES WEB **CURSO 2024/25**
DESARROLLO WEB EN ENTORNO SERVIDOR **12/12/2024**

NOMBRE: _____

A partir de la estructura usada en clase, crear un proyecto para el examen. Se mantendrán las clases creadas anteriormente en la carpeta /scripts/clases y las clases que creemos en /scripts/examen. Se debe garantizar la autocarga en ambas carpetas. Se tendrá también la librería validaciones.php en su ubicación correcta.
Siempre que tengamos un objeto, función o método que nos haga algo se usará.
Se tendrán en cuenta todas las consideraciones que ya conocemos como modelo/vista/controlador, definir tipo de propiedades y parámetros, uso de funciones mb_, librería validaciones, etc
Se podrán definir métodos/procedimientos adicionales a los pedidos si entendemos que son necesarios.

Una empresa de desarrollo piensa en crear una nueva aplicación de gestión de proyectos. Para ello tras el análisis de requisitos decide crear:
1.- (2.4 ptos) Crear la clase Proyecto que contendrá los datos para un nuevo proyecto. La clase **Proyecto** tendrá las siguientes características:

- Constante publica TIPOPROYECTO. Es un array con valores 10: Solo software, 20: Solo hardware, 30: Software/Hardware.
- Propiedades: **Nombre** (cadena 40 caracteres, obligatoria), **Empresa** (cadena de 35 caracteres, obligatorio), **fecha_inicio** (fecha, no puede ser posterior al día de hoy, por defecto hoy), **fecha_fin** (fecha, no puede ser anterior a fecha_inicio, por defecto dentro de 6 meses), **duracion** (entero, representa el número de días entre fecha_inicio y fecha_fin, calculado) y **tipo** (entero, uno de los valores indicados en TIPOPROYECTO, por defecto 10) y **tipo_descripcion** (cadena, descripción del tipo a partir de TIPOPROYECTO, calculado). Estas propiedades no pueden ser accedidas directamente desde el exterior aunque si desde las clases descendientes.
- Se tendrá el constructor al que se le pasan parámetros para rellenar las propiedades (solo los no calculados, aquellos que tienen valores por defecto, pueden indicarse en la llamada y se rellenarán con los valores por defecto). Se deben comprobar las restricciones y en caso de que no las cumplan se asigna el valor por defecto. Si no se cumple la obligatoriedad se lanzará una excepción. Para comprobar/asignar valores se usarán los métodos setXXX y se tendrán en cuenta los valores devueltos.
- Se tienen métodos **getXXXX** para todas las propiedades.
- Se tienen métodos **setXXXX** para las propiedades NOMBRE, EMPRESA, FECHA_INICIO, FECHA_FIN y TIPO. Se deben comprobar las restricciones. Si no cumple las restricciones no se asignará el valor a la propiedad correspondiente. Devuelve 0 si todo es correcto o un entero negativo que identificará el tipo de error. Por ejemplo para setNombre, devolverá 0 si es correcto, -1 si no cumple la obligatoriedad y -2 si tiene más de 40 caracteres.
- No se permite la sobrecarga dinámica.
- Cuando se convierte en cadena devuelve el texto "proyecto ***NOMBRE*** para ***EMPRESA*** que durará ***DURACION*** días entre FECHA***_INI y FECHA_FIN de tipo TIPO_DESCRIPCION"***

Como vemos que se realizarán proyectos para la administración decidimos crear la clase **Proyecto_Admin** que hereda de Proyecto, no permite clases heredadas y que tiene el siguiente comportamiento adicional:

- Se tiene la propiedad **Expediente** (cadena 10 caracteres, de la forma 9999/99999, con valor por defecto 2024/00001)
- En el constructor incrementa automáticamente la fecha_fin en 20 dias.
- Se tiene método getExpediente y setExpediente con el mismo funcionamiento para estos métodos de lo indicado en la clase Proyecto
- Cuando se convierte en cadena, aparece además de la descripcion de proyecto el texto " con expediente ***EXPEDIENTE***" .

2.- (1.5 ptos) Se quieren controlar propiedades adicionales como importe_presupuestado o provincia para lo que creamos la clase **Otras** que permitirá definirlas de forma dinámica. La clase Otras tendrá las siguientes características:

- Las propiedades se deben guardar internamente en mayúscula (se pueden llamar con cualquier combinación de mayúscula y minúscula) Por ejemplo, la propiedad importe_presupuestado, se almacenará internamente como IMPORTE_PRESUPUESTADO y puede llamarse como ->imPOrte_PREsupuestado.

- Las propiedades que empiecen por i son de tipo numérico y el valor asignado debe ser un real (en caso de que no sea poner el valor a 0). El resto de propiedades se consideran cadena y su valor se almacena como una cadena (aunque se indique un número).
- Se tiene la propiedad privada TOTAL_IMPORTES que debe contener la suma de todas las propiedades de tipo entero. Se tendrá la propiedad dinámica tot_imp para acceder a la propiedad privada. Al acceder a objeto->tot_imp se debe devolver el valor de la propiedad privada TOTAL_IMPORTES
- Las propiedades se definirán como dinámicas y se guardarán en el array asociativo Propie.
- Se habilitará el recorrido con foreach de todas las propiedades dinámicas y tot_imp (TOTAL_IMPORTES) devolviendo como clave la propiedad con la primera y la última letra en mayúsculas y el resto en minúsculas.

3.- (1 pto) Añade a la clase Proyectos soporte para otras propiedades. Para ello:
- Se tendrá la propiedad privada **otras** que será un objeto de clase Otras.
- Tenemos el método **aniadeOtras** con al menos 3 parámetros: propiedad (cadena, nombre de la propiedad), valor (mixed) y n_prop (entero, se devolverá el número de propiedades que se asignan). En el momento de llamar a este método se pueden indicar mas propiedad-valor (como 2 parametros adicionales)
- Método **getDescripcionOtras**. Método que devuelve una cadena de todas las propiedades que tenemos en otras.

4.- (0.75 ptos) Para mantener los proyectos tendremos el array **$PRO** cuyos elementos serán del tipo Proyecto/Proyecto_admin (se definirá en un solo punto). Se debe garantizar que podamos guardar nuevos proyectos y tenerlos actualizados entre diferentes llamadas a páginas. Inicialmente el array tiene que tener al menos 2 proyectos (uno de cada tipo) con 3 propiedades adicionales (insertados con el método aniadeOtras).

5.- (0.75 ptos) Crear la función **cargarProyectosDesdeFicher**o que recibe como parámetros el nombre del fichero y un array donde guardar los proyectos y que se encargará de leer y analizar el fichero para crear tantos proyectos (se debe controlar posibles excepciones al crear el proyecto) como indique el fichero. Os adjunto el fichero pro.txt con datos que se deben poder incorporar. Para facilitar, todos los proyectos son de la clase Proyecto.
Ojo: puede que se de alguna excepción que hay que tratar.

6.- (3.6 ptos) Interfaz de la aplicación. La página /index.php tendrá:
- **Botón loguearse**: Se tendrá una cookie con valor inicial a 1 y que se increntará en 2 cada vez que se pulse estebotón. Si el valor es múltiplo de tres se registrará al usuario MULTIPLO, con permiso 1 y si no es se registrará al usuario NOMUL con el permiso 2. Junto al botón aparecerá si hay o no usuario registrado y el nombre del usuario y si esta logueado un botón para cerrar sesión (con funcionamiento).
- Párrafo (o textarea) **Mostrar Proyectos** en el que se mostrarán todos los proyectos que haya actualmente (información del proyecto). Además, si hay usuario logueado y tiene el permiso 1 se mostrarán las otras propiedades de cada proyecto usando un foreach y mostrando clave y valor Si no hay usuario o no tiene el permiso 1 aparecerá el texto “sin permiso para ver otros”
- Botón **Cargar Fichero**. Cuando se pulse este botón usará la función cargarProyectosDesdeFichero() para recoger los proyectos del fichero “pro.txt” que se adjunta y que situaremos en la carpeta /imágenes
- Formulario Acciones que tendrá un combo con todos los proyectos disponibles (debe tenerse una línea mas con un proyecto que no exista) y dos botones: modificar y exportar.
- Si se **pulsa modificar** abre la ventana /aplicación/proyectos/modificar.php. Esta ventana mostrará un formulario para modificar los datos del proyecto que se ha pasado como parámetro. Se debe verificar el proyecto. Una vez que se envíe el formulario se comprobarán los valores (la mitad usando verificación directa y la mitad usando la clase para verificar el valor). Si no son validos se mostrará un error y se volverá a mostrar el formulario con los datos indicados. Si son correctos se modificará el proyecto indicado y se hará permanente el cambio.
- Si se pulsa exportar, se llamará al archivo /aplicación/proyecto/datospro.php que se encargará de descargar un fichero de texto con los datos del proyecto/otras propiedades. Cada elemento en una línea distinta. Si no es válido el proyecto se devolverá un fichero en blanco.