



Para la realización de la prueba debes descargar los ficheros asociados. Se trata de una **prueba individual y no se permite la copia**. Puedes hacer uso de cuantos recursos necesites, siempre y cuando entiendas aquello que realizas. Añade todos los **comentarios** que consideres oportunos para explicar tu código, y usa la notación camelCase para los identificadores. Sube todo a GitHub con el nombre del repositorio público ex3inicial+apellido (en mi caso sería ex3vmonfort).

La prueba está estructurada en dos partes:

1ªPARTE: Con esta parte tienes suficiente para aprobar. La puntuación máxima es de 6 sobre 10.

2ªPARTE: Con esta parte podrás alcanzar la excelencia. La puntuación máxima será de 4 sobre 10.

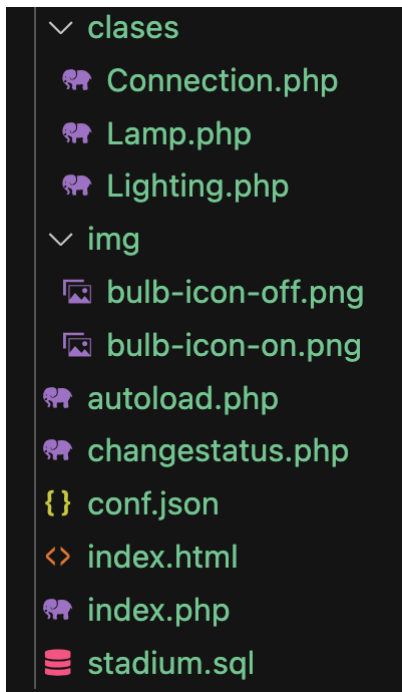
Entrega:

- La URL del repositorio
- El zip descargado



1ª PARTE

Ejercicio 1.- (1 punto) Crea la estructura necesaria para la aplicación en php, según muestra la imagen:



- Habilita un mecanismo de autoloading. Guarda los ficheros de clase en un directorio llamado "clases". El resto de ficheros php se ubicarán sobre la raíz. Existe un archivo conf.json, que debe contener las credenciales de la BBDD.
- Crea una clase Connection que contenga toda la lógica de conexión a la base de datos "stadium". Toda clase que tenga uso con la base de datos debe heredar de esta clase. Debes importar la base de datos desde el fichero sql proporcionado (hazlo ejecutando la consulta en phpmyadmin directamente).

Ejercicio 2.- (1 punto) La base de datos contiene información sobre los focos que iluminan un estadio. Se pretende modelar cada uno de estos dispositivos de iluminación:

- crea una clase "Lamp", cuyos atributos deben ser almacenar el identificador de la lámpara, su nombre, si se encuentra encendida o no, la denominación del modelo, la potencia en vatios y la zona del estadio donde está ubicada.
- Crea los getter y el constructor. Crea los setter solo en caso de necesitarlos.



Nombre

Ejercicio 3.- (4 puntos) Como se pretende controlar la iluminación del estadio, debemos desarrollar la lógica necesaria para listar e interactuar con las lámparas, y para ello debes crear una clase "Lighting" (Model), con los siguientes métodos:

- **(2 puntos) getAllLamps()** -> que devolverá todas las filas de la tabla "lamps" en un array, encapsulando cada fila en un objeto de tipo "Lamp".

Ayuda:

```
$sql = "SELECT lamps.lamp_id, lamps.lamp_name, lamp_on,
lamp_models.model_part_number, lamp_models.model_wattage,
zones.zone_name FROM lamps INNER JOIN lamp_models ON
lamps.lamp_model=lamp_models.model_id INNER JOIN zones ON
lamps.lamp_zone = zones.zone_id ORDER BY lamps.lamp_id;"
```

...

```
$lamps = []; //Array de objetos Lamp
```

```
//hacemos objetos Lamp y los añadimos al array
```

- **(1 punto) drawLampsList()** -> que mostrará, según el fichero index.html de muestra proporcionado, el listado de lámparas que proporciona getAllLamps(). Debes modificar **index.html** y convertirlo en el script **index.php** capaz de hacer uso de este método.
- **(1 punto) Potencia por zona** -> deberás de mostrar el total de la potencia por zona (solo de las lámparas encendidas). Muestra esa información en el lugar que desees del index. Deberás de hacer uno o varios métodos que retornen esos valores.



Ayuda:

```
$sql = "SELECT SUM(lamp_models.model_wattage) as power FROM
`lamps` INNER JOIN lamp_models on
lamp_model=lamp_models.model_id WHERE lamp_on = 1 ;"
```



2ª PARTE

Ejercicio 4.- (2 puntos) El listado creado con anterioridad necesita tener cierta interactividad. Para ello:

- **(1 punto)** Convierte los iconos  o  en enlaces, de forma que sean capaces de cambiar el estado de encendido a apagado o viceversa.
- **(1 punto)** En la clase Lighting crea el método **changeStatus(\$id, \$status)** que debe ser capaz de actualizar esta información en la base de datos.

Ayuda: Crea un script **changestatus.php** capaz de hacer uso de este método y re-dirigir de nuevo a la página que contiene el listado.

Ejercicio 5.- (2 puntos) Dado que son muchas las lámparas a tener en cuenta, se propone un formulario para realizar el filtrado por zona de las lámparas. Para ello:

- **(1 punto)** Crea el método **drawZonesOptions()** en la clase Lighting que poblará el desplegable con la información de las zonas disponibles, quedando seleccionada la opción elegida en el desplegable.
- **(1 punto)** al enviar el formulario, el listado se debe actualizar solo con las lámparas de la zona elegida. Para ello tan solo modifica el método **getAllLamps()** de manera que lo que devuelva sean las lámparas de la zona seleccionada. Puedes hacer un método nuevo para no tocar **getAllLamps** e ir probando.

Ayuda: Vas a necesitar crear una propiedad más en la clase Lighting (por ejemplo `currentFilter`) que almacene la zona. Por defecto podremos decirle que almacene "all" para todas las zonas.