

## Parte 02 (hasta un 6)

**Producto.php.** Crear, en `./clases`, la clase **Producto** con atributos públicos (codigoBarra, descripcion, precio y pathFoto), constructor (con todos sus parámetros opcionales), un método de instancia ToJSON(), que retornará los datos de la instancia (en una cadena con formato JSON).

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 **productos** (id, código\_barra, descripcion, precio, foto), de la base de datos **almacen\_bd**. Retorna **true**, si se pudo agregar, **false**, caso contrario.
- **Traer:** retorna un array de objetos de tipo Producto, recuperados de la base de datos.
- **CalcularIVA:** retorna el precio del producto más el 21%.
- **Existe:** retorna **true**, si la instancia actual está en el array de objetos de tipo Producto que recibe como parámetro. Caso contrario retorna **false**.
- **Modificar:** Modifica en la base de datos el registro coincidente con la instancia actual. Retorna **true**, si se pudo modificar, **false**, caso contrario.
- **Eliminar:** elimina de la base de datos el registro coincidente con la instancia actual. Retorna **true**, si se pudo eliminar, **false**, caso contrario.

Implementar la interface en la clase Producto.

**AgregarProducto.php:** Se recibe por POST un JSON con todos los valores (incluida una imagen) para registrar un producto en la base de datos. Se invocará al método Agregar.

La imagen guardarla en `"/productos/imagenes/"`, con el nombre formado por el **código de barra** punto **descripcion** punto hora, minutos y segundos del alta (**Ejemplo: 123.tomate.105905.jpg**).

Se retornará un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

**Listado.php:** (GET) Se mostrará el listado **completo** de los productos (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 incluidos.

**Nota:** preparar la tabla (HTML) para que muestre la imagen de la foto (si es que la tiene).

**VerificarProducto.php:** Se recibe por POST el objeto de tipo Producto (en formato de cadena JSON), si coincide con algún registro de la base de datos (invocar al método Traer) retornar los datos del objeto (invocar al ToJSON). Caso contrario informar lo acontecido.

**AgregarProductoVerificado.php:** Se recibe por POST un JSON con todos los valores (incluida una imagen) para registrar un producto en la base de datos.

Verificar la previa existencia del producto invocando al método Existe. Se le pasará como parámetro el array que retorna el método Traer.

Si el producto ya existe en la base de datos, se retornará un mensaje que indique lo acontecido.

Si el ovni no existe, se invocará al método Agregar.

Si se pudo agregar se redirigirá hacia Listado.php. Caso contrario, se mostrará un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

**ModificarProducto.php:** Se recibirán por POST todos los valores (incluida una imagen y el id) para modificar un producto 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

“./productosModificados/”, con el nombre formado por el **id** punto **descripción** punto '**modificado**' punto hora, minutos y segundos de la modificación (**Ejemplo: 52.tomate.modificado.105905.jpg**). Redirigir hacia Listado.php. Si no se pudo modificar, se mostrará un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

**EliminarProducto.php:** Si recibe un **código de barra** por GET, retorna si el producto está en la base o no (mostrar mensaje).

Si recibe el **código de barra** por POST, con el parámetro “**accion**” igual a “**borrar**”, se deberá borrar el producto (invocando al método Eliminar). Si se pudo borrar en la base de datos, redirigir hacia Listado.php.

Si no se pudo borrar, mostrar un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.