

# Tarea para DWES06.

## Descripción de la tarea.

---

### Enunciado

Continuaremos utilizando la base de datos correspondiente a la tienda online. Primero, debes crear una base de datos llamada "tarea6", similar a la que usamos anteriormente. Puedes reutilizar el usuario "gestor" con su contraseña "secreto" para acceder.

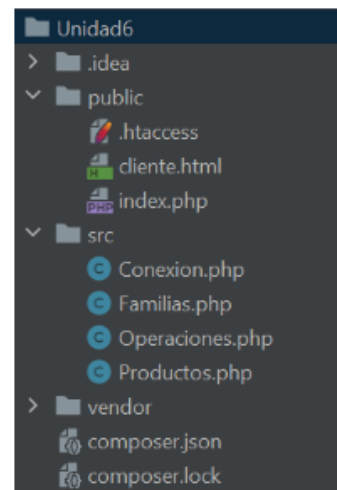
### ¿Qué te pedimos que hagas?

#### API REST

Vamos a utilizar PHP para crear un servicio web REST con varios endpoints que expongan información de la base de datos de la tienda online.

La estructura de las carpetas será la siguiente:

- **public:** donde almacenaremos los archivos para proveer los servicios que vamos a crear. (index.php)
- **src:** Contendrá todas nuestras clases, incluida la clase Operaciones que será la clase que recoja las peticiones Rest y utilizará las clases Producto y Familia, estas últimas contarán con métodos para manejar estas entidades e interactuar con la base de datos.
- **vendor:** Se generará automáticamente con Composer.



Para esta práctica, usaremos el "autoload" de las clases, instaladas con Composer. Dentro de la carpeta "src", crearemos el servicio REST operaciones.php. Este servicio ofrecerá los siguientes endpoints:

- **GET:**
  - /productos/{codigo}/pvp: Este endpoint recibirá el código de un producto y devolverá el PVP correspondiente al mismo como JSON.
  - /familias: este endpoint devolverá un array JSON con los códigos de todas las familias existentes.
  - /familias/{codigo}/productos: Recibirá el código de una familia y devolverá un array JSON con los códigos de todos los productos pertenecientes a esa familia.
  - /productos/{codigo}/tiendas/{tienda}/stock: Recibirá dos parámetros en la URL: el código de un producto y el código de una tienda, y devolverá el stock existente de ese producto en esa tienda como JSON.
- **PUT:**

- /familias/: recibirá los datos para la inserción de una familia en la bbdd en el body de la llamada:
  - Por ejemplo un Json: { "cod": "Beacons", "nombre": "Balizas Bluetooth" }
- **DELETE:**
  - /familias/{código}: recibirá el cod de la familia en la url y si existe procederá a su borrado.

## CLIENTE

Para verificar la correcta ejecución del servicio REST, también crea un cliente llamado **cliente.php** en la carpeta public que realice una llamada HTTP a cada uno de los endpoints y muestre los resultados obtenidos.

Maneja los posibles errores en los requests, por ejemplo, un producto o familia que no exista.

Utiliza **Guzzle** para realizar las peticiones: <https://docs.guzzlephp.org/en/stable/>

En la imagen adjunta se muestra una propuesta de salida donde se prueban los distintos endpoints del API Rest.

```
Testing GET /productos/{codigo}/pvp
--- Response Start ---
HTTP Status Code: 200
Array ( [0] => Array ( [pvp] => 270.00 ) )
--- Response End ---

Testing GET /familias
--- Response Start ---
HTTP Status Code: 200
Array ( [0] => Array ( [cod] => CAMARA [nombre] => Cámaras [5] => Array ( [cod] => MP3 [nombre] => Reproductores MP3 [10] => Array ( [cod] => ROUTER [nombre] => Routers ) [11] ) ) )
--- Response End ---

Testing GET /familias/{codigo}/productos
--- Response Start ---
HTTP Status Code: 200
Array ( [0] => Array ( [id] => 7 [nombre] => Canon Ixus 115H [pvp] => 196.70 [familia] => CAMARA ) [1] => Array ( [id] => sensor CCD de 14,1 megapíxeles y un objetivo gran angular de Canon en un cuerpo compacto y ligero para capturar fotografías ) )
--- Response End ---

Testing GET /productos/{codigo}/tiendas/{tienda}/stock
--- Response Start ---
HTTP Status Code: 200
Array ( [0] => Array ( [unidades] => 1 ) )
--- Response End ---

Testing PUT /familias/
--- Response Start ---
HTTP Status Code: 201
Response: Familia creada con éxito
--- Response End ---

Testing DELETE /familias/{codigo}
--- Response Start ---
HTTP Status Code: 200
Response: Familia borrada con éxito
--- Response End ---
```

## Información de interés.

## Recursos necesarios

## Consejos y recomendaciones

### Indicaciones de entrega

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

**apellido1\_apellido2\_nombre\_SIGxx\_Tarea**

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas para la sexta unidad del MP de DWES, debería nombrar esta tarea como...

**sanchez\_manas\_begona\_DWES06\_Tarea**

## Evaluación de la tarea.

---

### Criterios de evaluación implicados

### ¿Cómo valoramos y puntuamos tu tarea?

Crear la estructura del proyecto e instalar todas las dependencias necesarias con Composer.	1 punto.
Programar correctamente el index.php	2 puntos.
Crear y gestionar los endpoint en Operaciones.php	0.50 puntos.
Conexión con la bbdd en fichero independiente.	2 puntos.
Crear de forma correcta productos.php y familias.php	0.5 puntos.
Crear la estructura de directorios según lo pedido en el enunciado.	1 punto.
Control adecuado de errores.	1 punto.
Crear de forma correcta cliente.php para testar la ejecución del servicio REST.	1 punto.
El diseño del código es adecuado y el código está comentado.	1 punto.