



Alejandro Brugarolas Sánchez-Lidón

Curso 2º DAW

EFA El Campico

[alejandrobregarolas@elcampico.org](mailto:alejandrobregarolas@elcampico.org)

# PRÁCTICA DE PROYECTO | MYSQL

Realizar los ejercicios propuestos

## Objetivos

1. Recordar el funcionamiento de PDO.
2. Recordar la sintaxis de distintas consultas DML.
3. Aprender a enviar y recibir información mediante peticiones HTTP con parámetros GET y POST.

## Enunciado

Restaurar la base de datos “colegio.sql” que se encuentra subida en el Moodle.

Crear clases de conexión y gestión de consultas a la base de datos. Se pueden reutilizar y adaptar los distintos archivos que hay subidos en la carpeta “Ejemplos de conexión usando PDO” en el Moodle.

Añadir a la carpeta “ws” creada en la práctica anterior los siguientes archivos:

**getUsuario.php.** Este archivo tendrá dos funciones:

1. Si recibe un parámetro “id” por GET, el archivo deberá consultar en la base de datos el alumno con ese id y devolverlo en el formato de respuesta que se especifica en el siguiente apartado.
2. Si no recibe ningún parámetro, el archivo deberá devolver todos los alumnos de la base de datos en el formato de respuesta que se especifica en el siguiente apartado.

**deleteUsuario.php:** Este archivo recibirá un parámetro id por GET y deberá eliminar el usuario de la base de datos con ese id. Posteriormente devolverá los datos del alumno eliminado en el formato de respuesta que se especifica en el siguiente apartado.

**crearUsuario2.php:** Este archivo recibirá todos los parámetros enviados por el formulario HTML y creará el usuario en la base de datos (tabla alumno). Posteriormente devolverá los datos del alumno creado en el formato de respuesta que se especifica en el siguiente apartado.

**modificarUsuario.php:** Este archivo recibirá un parámetro “id” por GET y los mismos parámetros que el archivo “crearUsuario2.php” por POST. Si existe un usuario con el id especificado en la tabla alumno, se modificarán sus datos en base a lo especificado en los parámetros POST. Si algún parámetro POST no se envía o viene vacío, ese dato no deberá modificarse.

## Formato de la respuesta

La respuesta deberá ser un string en formato JSON con la siguiente estructura:

```
{
  success: boolean <- true si todo ha ido bien | false para cualquier fallo (id inválido, etc) o
  excepción
  message: string <- Mensaje informativo que acompaña a la respuesta
  data: [], {}, null <- Información a devolver (Usuarios obtenidos, creados, etc).
}
```

Ejemplos de respuesta:

```
{
  "success": true,
  "message": "Usuarios obtenido correctamente",
  "data": [
    {
      "nombre": "Pepe",
      "apellido": "Díaz",
    },
    {
      "nombre": "Juan",
      "apellido": "Martínez",
    }
  ]
}
```

```
{
  "success": true,
  "message": "Usuario creado con éxito",
  "data": {
    "nombre": "Pepe",
    "apellido": "Díaz",
  }
}
```

```
{
  "success": false,
  "message": "Usuario con id 938 no encontrado",
  "data": null
}
```

## Requisitos Obligatorios

- Trabajar con instancias de la clase User.
- No hacer echoes dentro de ningún método de clase.
- Utilizar PDO (mysqli\_etc no será admitido).
- Crear las clases que el alumno considere necesarias para la correcta reutilización del código.
- Limpieza del código y correcta estructuración en base a lo indicado en el estándar PSR-12 (<https://www.php-fig.org/psr/psr-12/>).

## Recomendaciones

Para poder hacer las pruebas pertinentes se recomienda el uso de clientes HTTP como Postman o Insomnia para hacer peticiones GET y POST con los parámetros que el alumno desee.

## Entrega

La entrega se realizará subiendo el proyecto a GitHub y compartiendo el repositorio al email [alejandrobregarolas@elcampico.org](mailto:alejandrobregarolas@elcampico.org). Posteriormente, se copiará el enlace **de la web del repositorio** en la entrega abierta en Moodle.

## Aclaración

Aunque se haya enviado el enlace al repositorio en otra tarea, será necesario enviarlo de nuevo para poder calificar la tarea y que el trabajo aparezca como enviado.

La URL de la web del repositorio **no** es la que termina en .git, si no la que aparece en el navegador cuando se accede al repositorio.

