

Binding de parámetros

Utilizando el método de PDO [bindValue](#) o enviando los parámetros dentro del método **execute** realizaremos los siguientes ejercicios:

1. Partiendo del archivo `agregarPelicula.php` dado en la clase de ayer (y también subido en la clase de hoy). Modificarlo para que efectivamente agregue películas utilizando “bindeo” de parámetros.
2. Modificar una vez más `agregarPelicula.php` para que si el título de la película ya existe, solo actualice el resto de los datos.
3. En un nuevo archivo PHP
 - a. Abriremos una nueva conexión a la base de datos a través de PDO.
 - b. Definiremos la variable `$idSerie` con el valor 3.
 - c. Mostraremos los nombres de dicha serie.
 - d. Probar modificar `$idSerie` con los valores 1, 2 y 4.
 - e. Modificar el código para que el `$idSerie` sea tomado de los parámetros enviados por GET. En caso de que no hubiese, indicarlo.
 - f. Agregar abajo del título de la serie un listado con los nombres de las temporadas.
 - g. Agregar abajo de cada nombre de temporada un listado con los nombres de cada uno de los episodios.
 - h. Permitir que por GET también llegue un parámetro que indique la fecha de inicio de la serie. Así, este archivo solamente mostrará los datos de las series que hayan iniciado posterior a dicha fecha.
 - i. Permitir que por GET también llegue un parámetro que indique cuantas series se quieren mostrar en el listado.

SQL + Objetos

1. Modelar una clase **Actor** que contenga los datos que se ven en la tabla como atributos. Como métodos debe contar con los getters y setters por cada atributo y un método llamado **getNombreCompleto**.
2. En un archivo nuevo, utilizando PDO y la clase **Actor**, listar los nombres completos de los actores.
3. Modelar la clase **Pelicula** que contenga los datos que se ven en la tabla como atributos. Como métodos debe contar con los getters y setters por cada atributo.
4. Utilizando PDO y las clases creadas agregar dentro de la clase **Actor** un método que se llame **getFavoriteMovie** que devuelva la película preferida del actor o **null** en caso de que no tenga.
5. En el archivo que lista los actores agregar por cada actor su película preferida o el mensaje “No tiene” si no tiene película preferida.
6. Utilizando PDO y las clases creadas agregar dentro de la clase **Actor** un método que se llame **listMovies** que retorne un array con todas las películas en las que actúa dicho actor.

7. En el archivo que lista los actores agregar debajo un listado con las películas en las que actúa.

Transacciones

Para los siguientes ejercicios:

- Realizaremos [transacciones en PDO utilizando bloques try/catch](#)
 - Dentro del catch, además, imprimiremos la palabra "Error: " junto con **`$e->getMessage()`**
 - Siempre, tras ejecutar un ejercicio, observaremos si se insertaron o no cosas en la tabla **Pelicula**
1. Realizar una transacción que inserte 3 películas completando todos sus datos correctamente.
 2. Realizar una transacción que inserte 3 películas y que la primera tenga un `id_genero` que no exista.
 3. Realizar una transacción que inserte 3 películas y que la tercera tenga un `id_genero` que no exista.
 4. Realizar una transacción que realice una query con un error de sintaxis y luego inserte una película.
 5. Realizar una transacción que inserte una película y luego que realice una query con un error de sintaxis.

