

Recordatorio: Una de las maneras de gestionar una conexión a una base de datos en PHP es a través de la clase **PDO**. En su constructor, puede recibir 4 parámetros.

- DSN. Obligatorio. Es un string que indica datos de la conexión.
- Username. Optativo. Un string que indica nombre de usuario con el que tratará de conectarse.
- Password. Optativo. Un string que indica la contraseña con la que tratará de conectarse.
- Options. Optativo. Es un array que puede recibir diversas de la conexión en un array asociativo. Importante para manejo de errores.

Dado esto, una conexión tendría la siguiente sintaxis:

```
$db = new PDO($dsn, $username, $password, $options);
```

El DSN es un string que podría tener por ejemplo la siguiente información

```
'mysql:host=localhost;dbname=mi_bd;port=3306'
```

Consultas SQL

Trabajando con **movies_db** (misma base de datos utilizada en la ejercitación de MySQL)

1. Crear el archivo **series.php** que con un statement consulte a la BD de todas la series y los muestre como un listado (div, li).
 - a. Adicionar que tenga un link que vaya a serie.php.
2. Realizar un archivo **serie.php** que muestre los datos de **una** serie (EJ: cantidad de episodios). Para esto, tendremos una variable **\$idSerie** que será modificada directamente en el código (ejemplo: **\$idSerie = 5;**). La consulta debe hacerse a través de un statement.
 - a. Modificar el archivo para que reciba un parámetro por GET **serie_id** y realice la consulta en la bd por ese parámetro. Para probar editaremos la URL modificando el **queryString**
 - b. Hacer que el link del archivo **series.php** funcione para cada elemento y muestre los datos en **serie.php**.
3. Partiendo de **agregarPelícula.php** modificarlo para que al enviarse el formulario inserte una nueva película en la base de datos utilizando statements. Sugerencia: crear un nuevo archivo (ej: **guardar.php**) en donde esté toda la lógica del proceso.
 - a. Sumarle a **agregarPelícula.php** un **<select>** con todos los géneros existentes en nuestra DB, para poder elegir el mismo.

4. Armar un buscador de SERIES o PELICULAS, el mismo archivo debe servir para buscar cualquier de las dos opciones, por ejemplo a través de un radio-button, preguntar al usuario qué desea buscar y luego de eso, realizar la consulta a la DB. Generar errores si no se seleccionó el tipo de búsqueda y así mismo si la búsqueda no arroja resultados
5. Armar un archivo actores.php que arme un con todos los actores de nuestra base de datos.
6. Armar un archivo generos.php para que liste todos los generos y por cada uno de los géneros, además, arme un sector en donde indique todas las películas de dicho género.
7. Modificar generos.php para que el título de cada película sea además un link a pelicula.php enviando por GET el id de dicha película. pelicula.php debe mostrar la información de dicha película.
8. Modificar agregarPelicula.php para que valide que el título no se encuentre repetido.
9. Modificar una vez más agregarPelicula.php para que si el título de la película ya existe, solo actualize el resto de los datos.

Proyecto Integrador

1. Modificar la registración y login para que guarde los usuarios en una base de datos MySQL

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.