

Conectarse a una base de datos Mysql a través de PHP

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'
```

1. Generar una conexión a la última base de datos utilizada durante las clases de MySQL aclarando en el DSN únicamente **host** y **dbname**.
2. ¿Qué sucede si utilizamos un **host** o un **dbname** erróneo? ¿Qué sucede si utilizamos credenciales erróneas?
3. Agregar dentro del DSN el puerto deseado.
4. Modificar el código con una estructura **try/catch** para que presente errores en la conexión de forma más amigable.
5. Teniendo esta nueva estructura, ¿que podemos visualizar ahora dentro del **catch** si hay un error?
6. Optativo: Crear un archivo nuevo y allí una función para la conexión ¿que debe retornar la función?.
 - a. Luego incluir ese archivo.
 - b. probar la ejecución de la función de conexión.
7. (optativo) Pedirle a un docente que apague el servidor de MySQL. ¿Qué sucede en tal caso?

Prácticas con queries.

Utilizando el método **query** de la clase **PDO**:

1. Armar un archivo con nombre `generos.php`, donde:
 - a. Realice la conexión a la BD.
 - b. Ejecute un query y obtenga todos los géneros de nuestra base de datos.

- c. Imprime el resultado de la obtención de los géneros.
 - d. Luego arme un listado (div o ul) de los géneros.
2. Armar un archivo `peliculas.php` que arme un listado con todas las películas de nuestra base de datos.
 - a. Además que tenga el género de cada una de ellas usando joins.

Statements

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.
 - b. Hacer que el link del archivo `series.php` funcione para cada elemento y muestre los datos en `serie.php`.
3. Partiendo de **`agregarPelicula.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 **`agregarPelicula.php`** un `<select>` con todos los géneros existentes en nuestra DB, para poder elegir el mismo.

Ejercicios integradores

1. 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.
2. Armar un archivo `actores.php` que arme un `` con todos los actores de nuestra base de datos.
3. Modificar `generos.php` para que por cada uno de los géneros, además, arme un sector en donde indique todas las películas de dicho género.
4. 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.
5. Modificar `generos.php` para que los títulos de películas de los géneros solamente aparezcan luego de clicar en el nombre del género. Al clicar de nuevo, estos deberían ocultarse.
6. Realizar un sitio que se llame `favoritas.php` que liste los actores que tienen película preferidas e indique cuales son.

7. Modificar `favoritas.php` para que muestre todos los actores y si no tiene película preferida diga "No tiene".
8. Modificar `agregarPelicula.php` para que valide que el título junto con la fecha de estreno de la película no se encuentre repetido.
9. Agregar a `agregarPelicula.php` validación y persistencia de los datos (en caso de error)