

PROYECTO FINAL – Gestión de Videoclub / Biblioteca

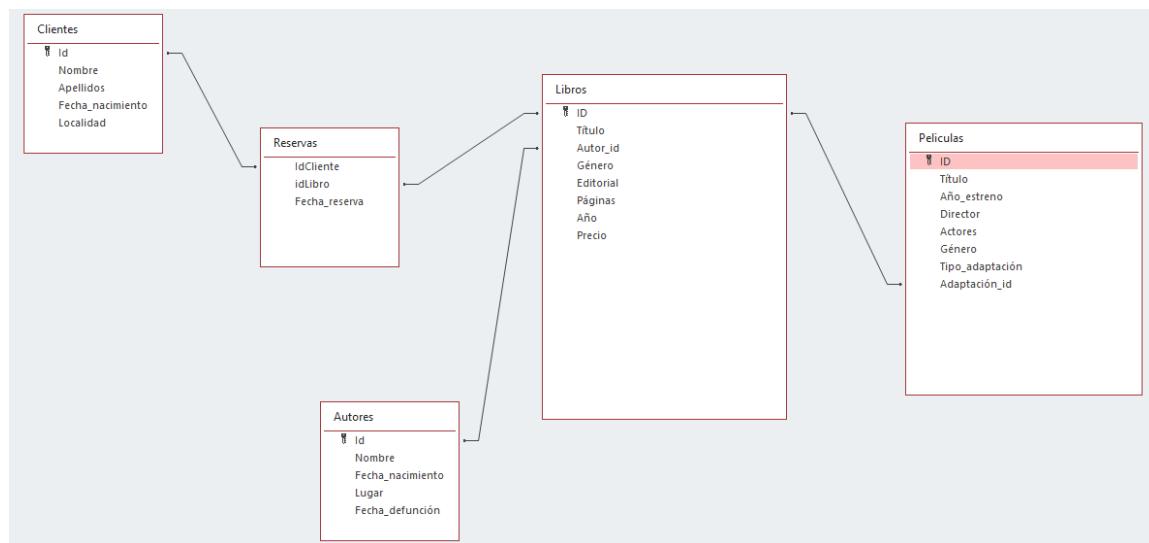
El objetivo del proyecto es partir de lo realizado hasta el momento y desarrollar un sistema web en PHP que permita a los clientes consultar, reservar y devolver libros y películas mediante conexión a la base de datos MySQL que desplegamos en nuestro entorno Docker.

Se hará en grupos de 2-3 personas (partiendo del código hasta el momento de uno de vosotros) y se trabajará obligatoriamente mediante repositorio compartido en GitHub.

Para desarrollar nuestro sistema, deberéis:

1. Base de datos MySQL:

- Crear BD con las tablas comentadas en el ejercicio anterior.



- Importar los CSV proporcionados usando phpMyAdmin.

2. Gestión de clientes

- Login por usuario y contraseña (contra BBDD).
 - Ampliación: Buscar una forma de cifrado (por ejemplo SHA-256) para que la información no se guarde en texto plano en la BBDD.
- Sesiones activas para navegación: Para poder navegar por las páginas el usuario tiene que haber iniciado sesión.
 - Pista: Se puede guardar el nombre de usuario en sesión y, si se intenta acceder a una URL que debería necesitar ese acceso (por ejemplo el catálogo), pero aún no lo ha hecho, se redirige a login.

3. Gestión de libros y películas:

- Sustituir los *arrays* de objetos por consultas MySQL que rellenen esos *arrays*. Es decir, no dar la información directamente si no consultarla y que dependa de lo que haya en BBDD.
 - Consejo: estaría bien tener la funcionalidad de manejo y conexiones con la BBDD en ficheros independientes que se puedan incorporar.
 - No os olvidéis tampoco de que alguna funcionalidad común a las clases (películas y libros) puede ir en *traits*.
- Filtros por título, género, autor/director, año.
- Mostrar estado disponible/reservado.

4. Reservas:

- Un cliente sólo puede reservar un libro o película si está disponible (suponemos que sólo hay una unidad de cada producto, aunque sea una librería/biblioteca bastante limitada...).
- Registrar reserva en BDD la reserva.
- Permitir devolución desde la web para que otro cliente pueda marcarlo como reservado.
 - La tabla de reservas funcionará a modo de histórico donde se guarden todas las reservas de libros y películas en una fecha.

5. Nuevas vistas:

- Vista de catálogo.
- Vista de registro de clientes.
- Vista de reservas del cliente autenticado.

Requisitos técnicos:

- Usar POO, modularización (dividir la clase en fragmentos que luego se puedan añadir con dependencias) y *traits*.
 - Libro, Película.
 - Intenta usar **herencia y polimorfismo** para ello.
 - Divide parte de la funcionalidad en traits.

Entrega:

- Enlace a GitHub compartido. +Defensa en el aula el 15/12.