

# FLICKPICK



Hecho por: Iván Ríos

# Índice

- Introducción
- Donde está almacenado el proyecto
- Diseño y arquitectura
- Tecnología usada
- Creación de la página

## **1- Introducción**

El principal objetivo de mi proyecto es el de ofrecer un servicio al público similar al que te ofrecían en los antiguos videoclubs.

Es por eso que cree una página web llamada FlickPick para recrear los videoclubs antiguos pero modernizado, ya que no tendrás que bajar al videoclub para coger la película que quieres y acordarte de devolverla sino que podrás con un click alquilar la película y cuando se te acabe el plazo desaparecerá de tu cuenta o si lo quieres podrás comprarla y tenerla para siempre en tu biblioteca.

Se diferencia de las demás plataformas de streaming en que no ofertamos un catálogo de películas al que suscribirse al mes por X€ sino que ofrecemos un catálogo de películas y usted decide cual quiere alquilar o comprar.

### **Información detallada**

Cada contenido audiovisual viene con su propia ficha detallada, que incluye una sinopsis, un tráiler y un sistema de calificación para poder calificar a la película si te gusta, permitiéndote tomar decisiones informadas sobre lo que quieres ver.

Aunque cualquier persona puede explorar y visualizar las fichas de contenido en FlickPick, para alquilar, comprar o calificar el contenido, necesitarás registrarte y crear tu perfil personal. Este perfil no solo te permite acceder a tu contenido, sino que también viene con un monedero virtual que puedes recargar y usar para comprar o alquilar contenido en la plataforma. También tienes tu ficha personal en la cual podrás cambiar tu foto de perfil, cambiar tus datos o borrar tu cuenta. También en la ficha podrás ver las películas adquiridas recientemente.

Al adquirir un contenido podrás descargarlo y se guardará en tu biblioteca, y al alquilarlo lo tendrás en tu biblioteca durante 30 días. Además, al alquilar o comprar contenido, serás redirigido a una página dedicada donde podrás ver el contenido directamente desde la plataforma.

También tiene un apartado de administrador, en el cual podrás insertar películas nuevas a la página, crear códigos promocionales para que los usuarios lo puedan usar para meter dinero en la hucha, ver el apartado de AWX en el que monitorear las copias de seguridad de la base de datos y unas estadísticas que te permiten ver cuántas películas han sido alquiladas y adquiridas.

## 2. Donde está almacenado el proyecto

Mi proyecto está almacenado en dos docker: uno llamado flickpick\_page donde estan las paginas y otro llamado flickpick\_bdd donde estan las bases de datos.

<input type="checkbox"/>	Name	Image	Status	Port(s)
<input type="checkbox"/>	<a href="#">flickpick_page</a> c842803f742c	<a href="#">php:7.4-apc</a>	Running	<a href="#">8080:80</a>
<input type="checkbox"/>	<a href="#">flickpick_bdd</a> 5c4c97fc290b	<a href="#">mysql:5.7</a>	Running	<a href="#">3306:3306</a>

A su vez el ansible está en una máquina virtual llamada flickpick\_AWX

flickpick\_AWX (Docker instalado)  
Corriendo

### General

Nombre: flickpick\_AWX  
Sistema operativo: Ubuntu (64-bit)

### Sistema

Memoria base: 3624 MB  
Procesadores: 2  
Orden de arranque: Disquete, Óptica, Disco duro  
Aceleración: Paginación anidada, Paravirtualización KVM

### Pantalla

Memoria de video: 16 MB  
Controlador gráfico: VMSVGA  
Servidor de escritorio remoto: Inhabilitado  
Grabación: Inhabilitado

### Almacenamiento

Controlador: IDE  
Dispositivo IDE secundario 0: [Unidad óptica] Vació  
Controlador: SATA  
Puerto SATA 0: flickpick\_AWX.vdi (Normal, 60,00 GB)

### Audio

Controlador de anfitrión: Predeterminado  
Controlador: ICH AC97

### Red

Adaptador 1: Intel PRO/1000 MT Desktop (Adaptador puente, «Realtek RTL8852AE WIFI 6 802.11ax PCIe Adapter»)

### USB

Controlador USB: OHCI, EHCI  
Filtros de dispositivos: 0 (0 activo)

### Carpetas compartidas

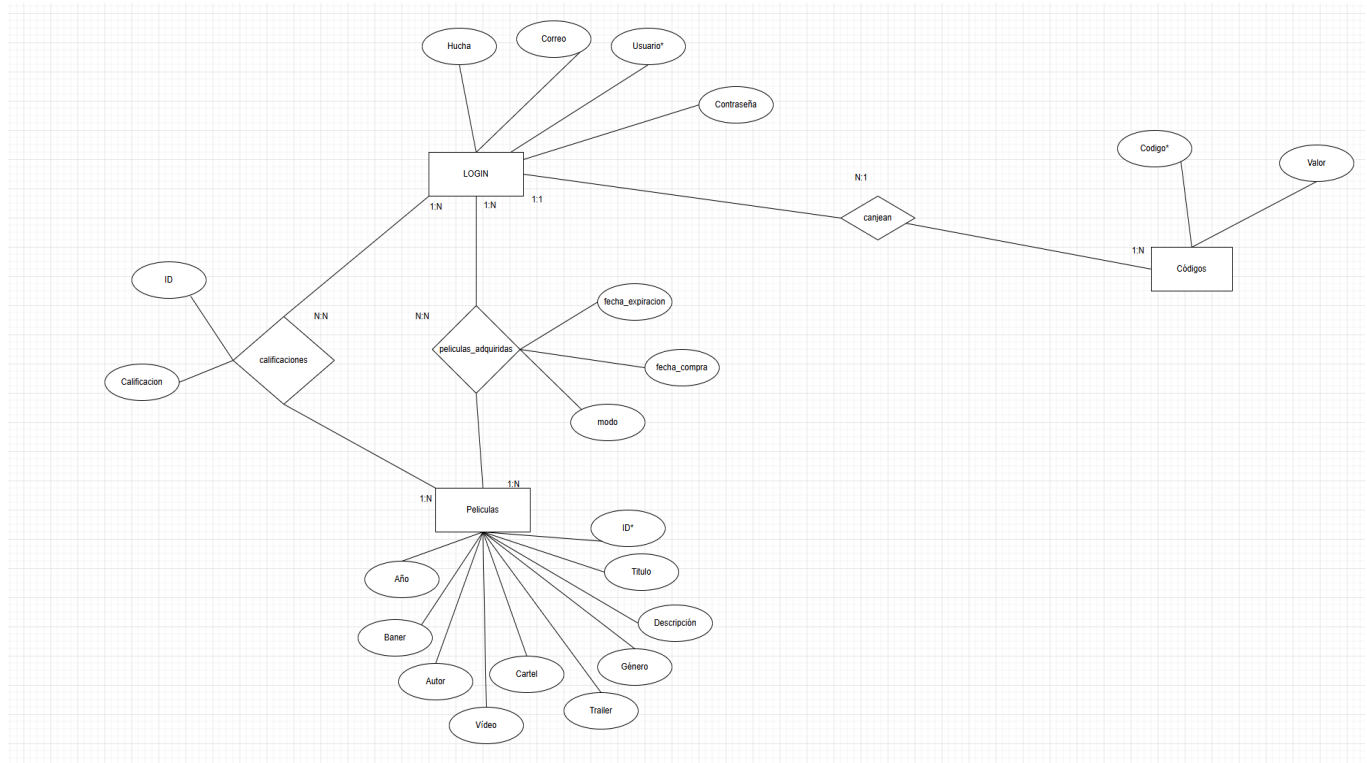
Ninguno

### Descripción

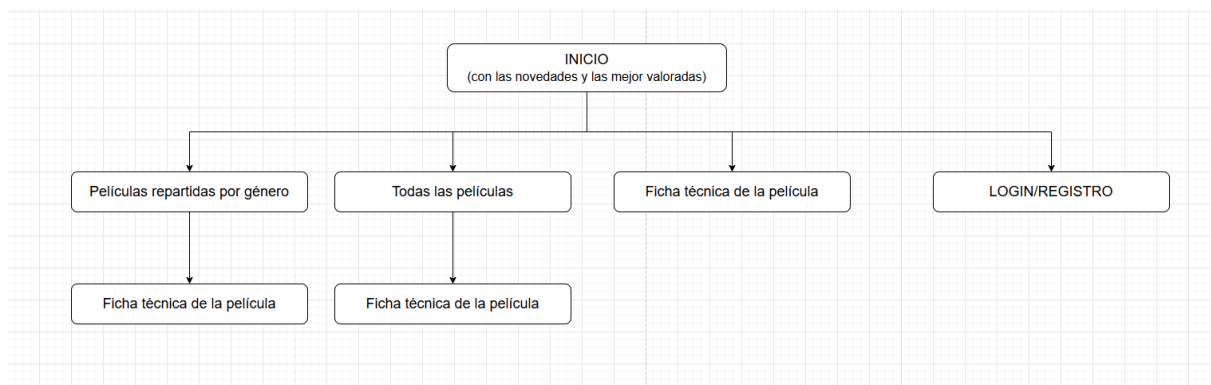
Ninguno

### 3. Diseño y arquitectura

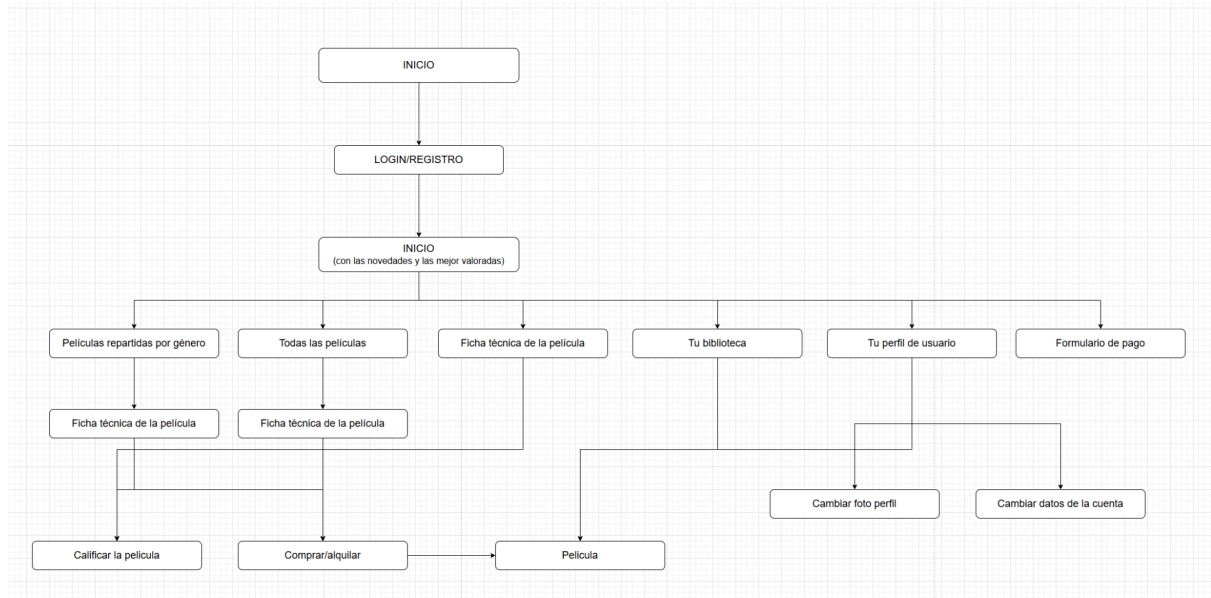
Modelo entidad-relación de la base de datos:



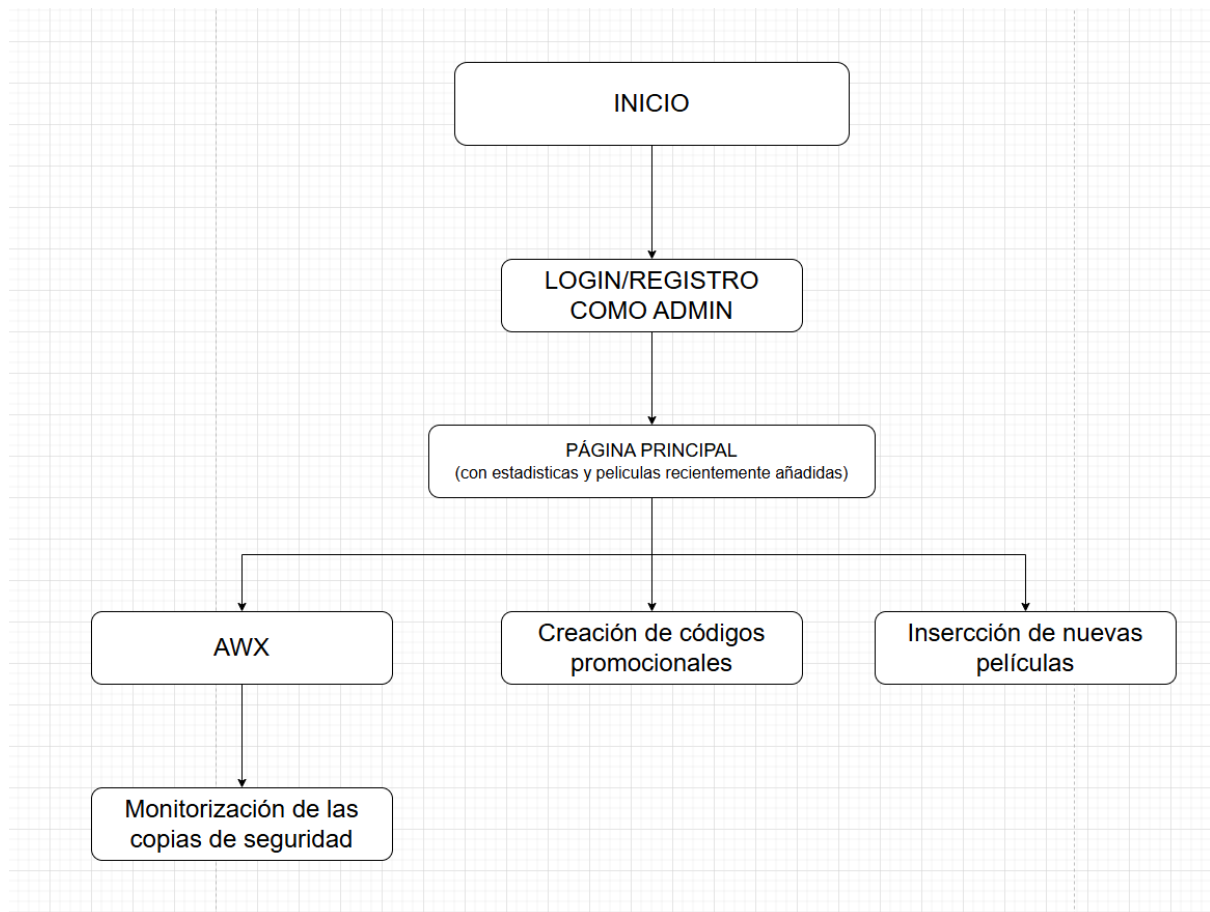
Diseño arquitectónico de la página si no estás registrado



## Diseño arquitectónico de la página si estás registrado



## Diseño arquitectónico si eres administrador



#### 4. Tecnología usada

Para la creación de mi página web he usado docker para la creación de un sistema apache donde guardar mis páginas web a la que le he puesto el puerto 8080 de escucha para acceder a mi página web.

También tengo otro docker para la creación de las bases de datos la cual he linkeado con el docker de apache.

También he usado una máquina virtual con la que he creado una torre AWX para hacer ansible sobre la base de datos y poder crear unos playbooks para hacer copias de seguridad sobre la base de datos.

Para la página web he usado HTML, CSS, PHP y JavaScript.

#### 5. Creación de la página

Para crear la pagina aparte de crear los php y html necesarios, tuve que crear una base de datos relacional, aqui dejo el codigo de la creación de la base de datos:

```
CREATE DATABASE FlickPick;
```

USE FlickPick;

```
CREATE TABLE login (  
    usuario VARCHAR(50) NOT NULL,  
    contraseña VARCHAR(50) NOT NULL,  
    correo VARCHAR(50) NOT NULL,  
    hucha int,  
    perfil VARCHAR(50),  
    PRIMARY KEY (usuario)  
);
```

```
CREATE TABLE peliculas (  
    id int auto_increment,  
    titulo VARCHAR(200) NOT NULL,  
    descripcion VARCHAR(600) NOT NULL,  
    video VARCHAR(250),  
    Año varchar(5),  
    trailer VARCHAR(250),  
    genero varchar(50),  
    autor varchar (50),  
    cartel varchar(50),  
    baner varchar(50),  
    precio_alquilar DECIMAL(5, 2),  
    precio_comprar decimal(5,2),  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE codigos(  
    codigo varchar(50) NOT NULL,  
    valor numeric(3) NOT NULL,  
    Primary key (codigo)  
);
```

```
CREATE TABLE calificaciones (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    id_usuario varchar(50),  
    id_pelicula INT,  
    calificacion INT,  
    FOREIGN KEY (id_usuario) REFERENCES login(usuario),  
    FOREIGN KEY (id_pelicula) REFERENCES peliculas(id)  
);
```

```
CREATE TABLE peliculas_adquiridas (  
    usuario VARCHAR(50) NOT NULL,  
    id_pelicula INT NOT NULL,  
    fecha_compra DATE NOT NULL,  
    fecha_expiracion DATE,  
    modo varchar(2),
```



```
PRIMARY KEY (usuario, id_pelicula),  
FOREIGN KEY (usuario) REFERENCES login(usuario),  
FOREIGN KEY (id_pelicula) REFERENCES peliculas(id)  
);
```

```
SET GLOBAL event_scheduler = ON;
```

```
CREATE EVENT IF NOT EXISTS eliminar_peliculas_expiradas  
ON SCHEDULE EVERY 1 DAY  
STARTS NOW()  
DO  
DELETE FROM peliculas_adquiridas  
WHERE fecha_expiracion < CURDATE();
```

La parte final es para que cuando expire la fecha\_expiracion de la tabla peliculas\_adquiridas se borre de la tabla el registro entero de esa fecha.

A su vez tuve que crear unos docker para guardar las páginas y la base de datos:

```
docker run -p 3306:3306 --name flickpick_bdd -v  
C:\Users\ivanr\OneDrive\Escritorio\tfg\ficheros:/var/lib/mysql -e  
MYSQL_ALLOW_EMPTY_PASSWORD=yes -d mysql:5.7
```

```
docker run -p 8080:80 --name flickpick_page -v  
C:\Users\ivanr\OneDrive\Escritorio\tfg\:/var/www/html -d --link flickpick_bdd php:7.4-apache
```

En el docker de flickpick\_page tuve que poner este comando para que se relacionase con el mysql:

```
docker-php-ext-install mysqli
```

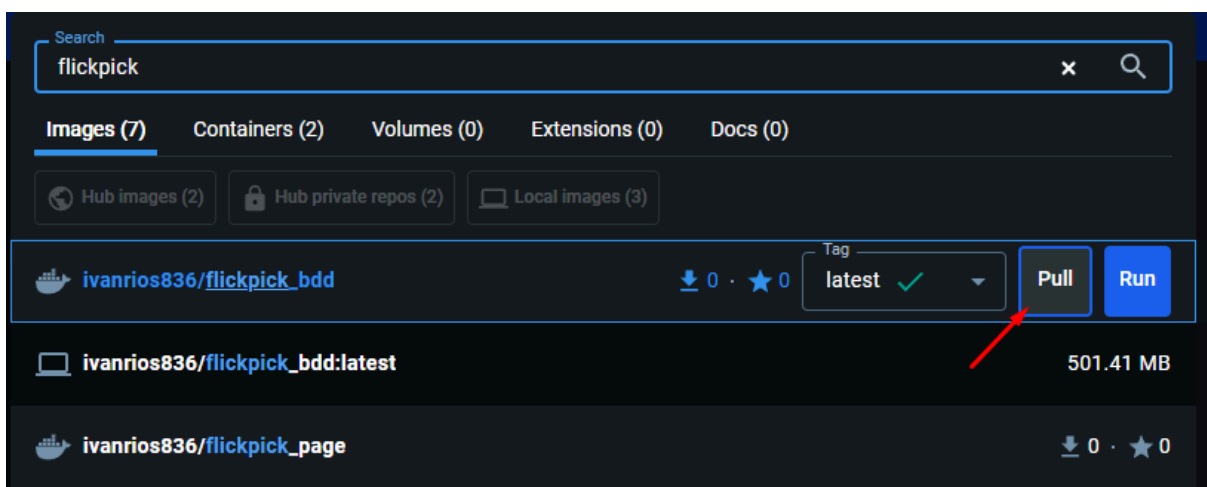
Y tambien cree un certificado autofirmado para usar https con estos comandos:

```
apt-get install openssl  
openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365
```

y tuve que copiar esto en el apartado Dynamic extensions en /usr/local/etc/php/php.ini:  
/usr/local/lib/php/extensions/no-debug-non-zts-20190902/mysqli.so

<input type="checkbox"/>	Name	Image	Status	Port(s)
<input type="checkbox"/>	<a href="#">flickpick_page</a> c842803f742c	<a href="#">php:7.4-apz</a>	Running	<a href="#">8080:80</a>
<input type="checkbox"/>	<a href="#">flickpick_bdd</a> 5c4c97fc290b	<a href="#">mysql:5.7</a>	Running	<a href="#">3306:3306</a>

Una vez hecho esto y reiniciado el docker, lo publique para que todo el mundo pudiese solamente hacerle un pull y tener la aplicación



Una vez hecho todo eso, cree la maquina virtual para AWX con Ubuntu 20.04  
Primero instale el docker y docker compose en esta máquina y una vez estuvo instalado el docker, procedí a instalar el AWX.  
Una vez instalado el AWX se vera como si le das a docker ps, te mostraran los 3 docker de AWX:

```
root@flickpick:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
01b966f8b205   ansible/awx:17.0.1                 "/usr/bin/tini -- /b..." 46 hours ago  Up 55 seconds  0.0.0.0:80->8052/tcp, :::80->8052/tcp
2963af6f00fe   postgres:12                         "docker-entrypoint.s..." 46 hours ago  Up 55 seconds  5432/tcp
e00b0caf16fe   redis                               "docker-entrypoint.s..." 46 hours ago  Up 55 seconds  6379/tcp
awx_web        awx_web                             "docker-entrypoint.s..."
awx_postgres   awx_postgres                        "docker-entrypoint.s..."
awx_redis      awx_redis                           "docker-entrypoint.s..."
```

Una vez hecho esto pasamos a crear unos servicios para cada uno de los docker para que cuando la máquina se encienda se inicien los docker

[Unit]

Description=AWX Web Docker container

Requires=docker.service

After=docker.service

[Service]

Restart=always

ExecStart=/usr/bin/docker start -a awx\_web

ExecStop=/usr/bin/docker stop -t 2 awx\_web

[Install]

WantedBy=default.target

sudo cp docker-awx\_web.service /etc/systemd/system/

sudo systemctl enable docker-awx\_web.service

Una vez hecho esto cuando apaguemos la máquina y la volvamos a encender se encenderán los docker de AWX.

Ahora es el paso para crear el playbook de ansible para poder hacer las copias de seguridad

---

- hosts: 192.168.1.22

tasks:

- name: Eliminar la copia de seguridad existente

command: docker exec flickpick\_bdd rm -f /backup/backup.sql

- name: Hacer una copia de seguridad de la base de datos Docker

command: docker exec flickpick\_bdd /usr/bin/mysqldump --all-databases > /backup/backup.sql

Este es mi playbook para ejecutar las copias de seguridad, la IP es la de mi máquina host y dependiendo de qué IP tiene mi máquina habría que cambiarla.