

Proyecto Diseño de Aplicaciones Web



Grupo: 27

Integrantes:

Huang Xinying

Alfonso Sudara Padilla

```

root@localhost:~/mysite#
├── manage.py
├── myrestservice
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── models.py
│   ├── serializers.py
│   ├── tests.py
│   ├── urls.py
│   └── views.py
├── mysite
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── videoclub27
    ├── __init__.py
    ├── admin.py
    ├── apps.py
    ├── models.py
    ├── static
    ├── templates
    ├── tests.py
    ├── urls.py
    └── views.py

```

El contenido del proyecto se muestra en la figura :

1. myrestservice :

Se encarga de las operaciones CRUD (RESTful) de la aplicación web tanto en la administración de las “películas” como la de “usuarios” (Backend).

2. mysite:

Configuración general de la aplicación web.

3. videoclub27:

Esta carpeta contiene los templates (.html) y ficheros estáticos (scripts: .js y .css) del frontend .

```

videoclub27
├── __init__.py
├── admin.py
├── apps.py
├── models.py
├── static
│   └── videoclub27
│       ├── images
│       │   └── logo-video-club27-trans.png
│       ├── scripts
│       │   ├── administrator.js
│       │   ├── boton_eliminar.js
│       │   ├── forms.js
│       │   ├── home.js
│       │   ├── jquery.cookie.js
│       │   └── peli.js
│       └── styles
│           ├── administrator.css
│           ├── index.css
│           ├── logo.css
│           ├── styles-home.css
│           └── styles-login.css
├── templates
│   └── videoclub27
│       ├── administrator.html
│       ├── page-home.html
│       ├── page-login.html
│       └── page-peli.html
├── tests.py
├── urls.py
└── views.py

```

Pasos de despliegue en un entorno Máquina Virtual:

En la máquina donde arrancaremos el servidor debe tener los siguientes prerequisites:

1. Se recomienda tener instalado la versión **>= python 3**
2. Se recomienda tener instalado **django == 3.0.7**
3. Tener instalado el módulo **djangorestframework == 3.11.0**

Comandos de ejecución:

- Hay que dirigirse dentro de la carpeta del proyecto (creado con django-admin)

```
cd mysite
```

- Hay que hacer una migración de los modelos "Películas" y "Perfiles"

```
python3 manage.py makemigrations myrestservice  
python3 manage.py migrate
```

- Lo siguiente es crear un super-usuario para el manejo de la base de datos, del cual nos pedirá el nombre y contraseña de usuario

```
python3 manage.py createsuperuser  
>>username: root  
>>email: root@correo.com  
>>password: root
```

- Último paso, arrancar el servidor:

```
python3 manage.py runserver 0.0.0.0:8000
```

Salida de la ejecución de los comandos:

```
root@localhost:~/mysite# python3 manage.py makemigrations myrestservice
Migrations for 'myrestservice':
  myrestservice/migrations/0001_initial.py
    - Create model Peliculas
    - Create model Perfiles
root@localhost:~/mysite# python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, myrestservice, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying myrestservice.0001_initial... OK
  Applying sessions.0001_initial... OK
root@localhost:~/mysite# python3 manage.py createsuperuser
Username (leave blank to use 'root'): root
Email address: root@correo.com
Password:
Password (again):
Error: Your passwords didn't match.
Password:
Password (again):
Error: Your passwords didn't match.
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
root@localhost:~/mysite# python3 manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 06, 2020 - 22:46:26
Django version 3.0.7, using settings 'mysite.settings'
```

Se tiene las siguientes API REST (RESTful), el nombre base de la url tendrá valor distinto.

Method	PUT
Url	http://192.168.56.101:8000/myrestservice/perfiles
Content Type	application/json
Body	<pre>{ "name": "grupo27", "email": "grupo27@correo.com" "password": "*****", "tipo": "admin/user" }</pre>

Method	PUT
Url	http://192.168.56.101:8000/myrestservice/peliculas
Content Type	application/json
Body	<pre>{ "name": "Titanic", "year": 2020, "director": "nomredirector", "assessment": 7, "url": "http://...", "imagen": "http://...", "description": "descripcion de este video", "repartos": "nombres del reparto" }</pre>

Method	GET
Url	http://192.168.56.101:8000/myrestservice/peliculas/all http://192.168.56.101:8000/myrestservice/perfiles/all

Para el método/tipo POST se utiliza los mismos cuerpos descritos arriba según los correspondientes.

Method	GET, POST , DELETE
Url	<a href="http://192.168.56.101:8000/myrestservice/peliculas/<int:id_pelicula>">http://192.168.56.101:8000/myrestservice/peliculas/<int:id_pelicula> <a href="http://192.168.56.101:8000/myrestservice/perfiles/<str:nombre_usuario>">http://192.168.56.101:8000/myrestservice/perfiles/<str:nombre_usuario>

Method	GET
Url	<a href="http://192.168.56.101:8000/myrestservice/home/peliculas/search/<str:search_name>">http://192.168.56.101:8000/myrestservice/home/peliculas/search/<str:search_name>

URL de acceso, frontend, de la aplicación web:

Si no se ha dado de alta anteriormente o no ha accedido, todas las url redirige al login.

- LOGIN

<http://192.168.56.101:8000/videoclub27/login>

- HOME

<http://192.168.56.101:8000/videoclub27/home>

- ADMINISTRADOR

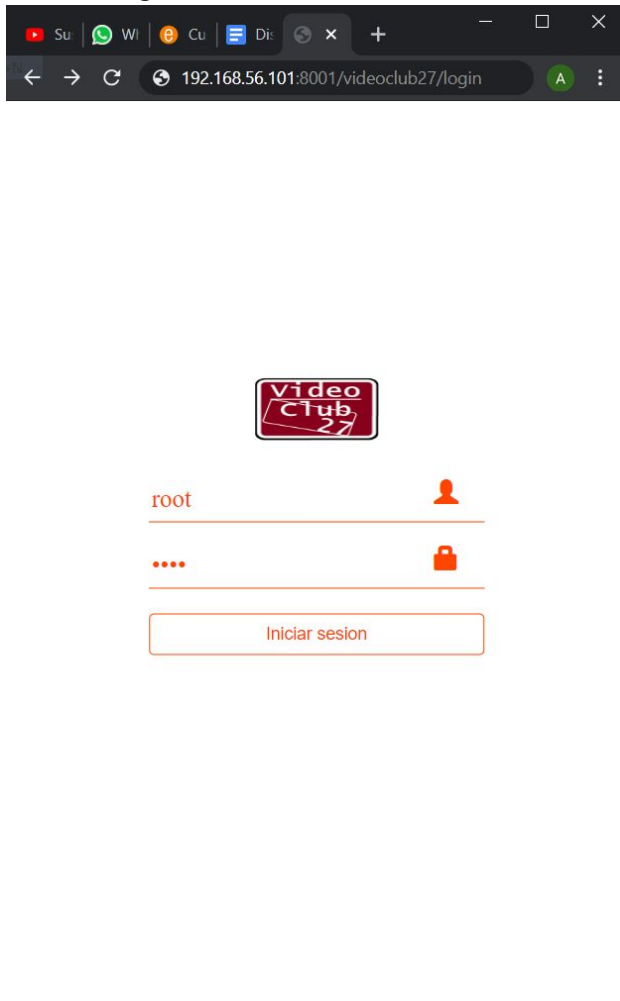
<http://192.168.56.101:8000/videoclub27/administrator>

DEMO

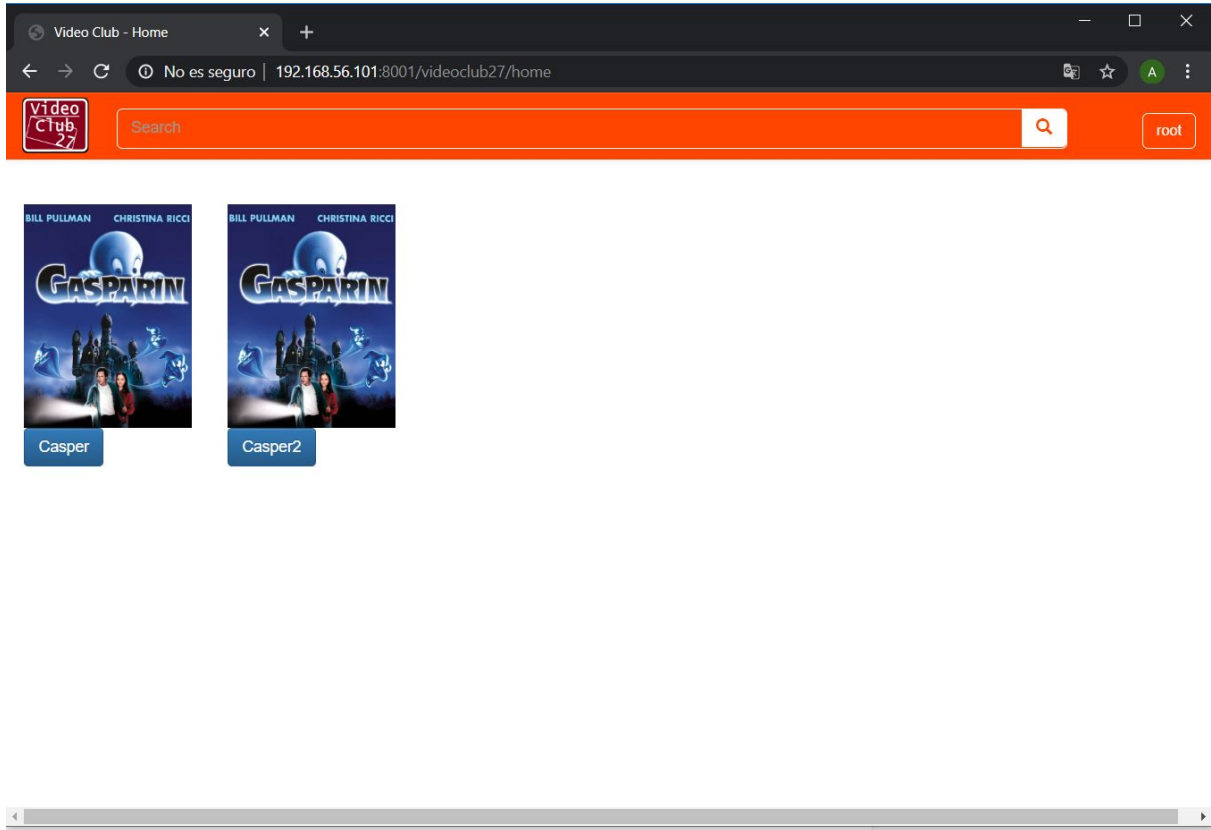
A continuación se muestra una secuencia básica de acceso y administración.

Si se ha conseguido crear el superuser en la sección de “ejecucion de comandos” , **root** es el usuario inicial que usaremos para crear lo usuarios ya sean de tipo administrador(*admin*) o usuario (*user*). Y los usuarios de tipo “admin” podrán hacer las mismas operaciones CRUD sobre los videos o usuarios.

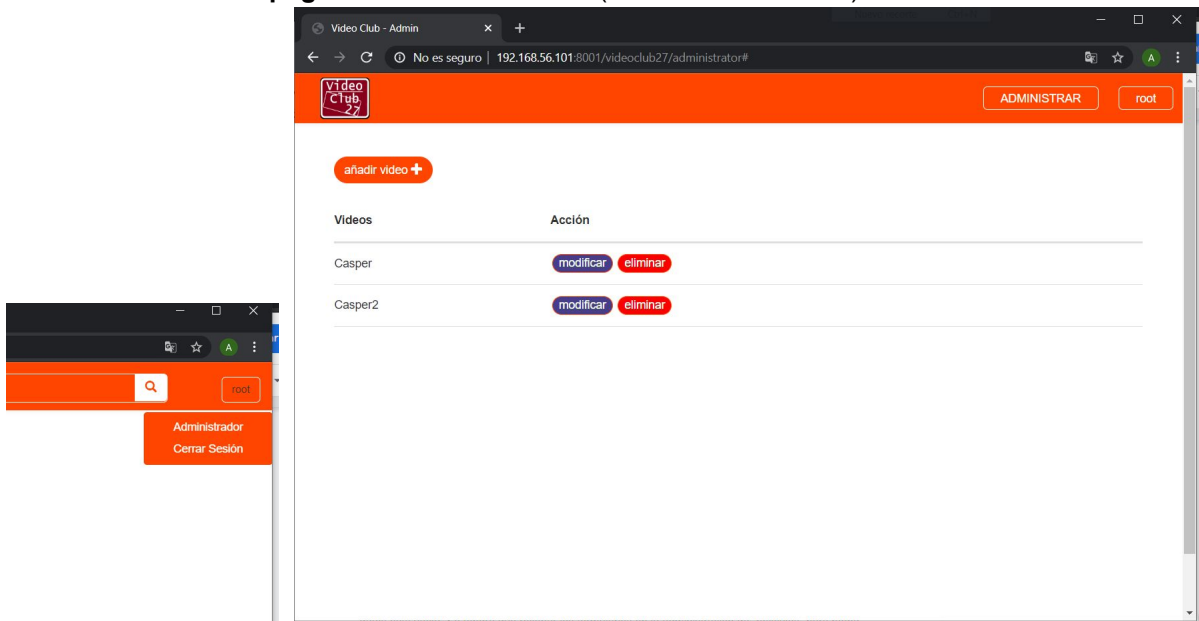
Paso 1: Login



Paso 2: Página inicial



Paso 3: Acceder a la página de administracion (click ADMINISTRAR)

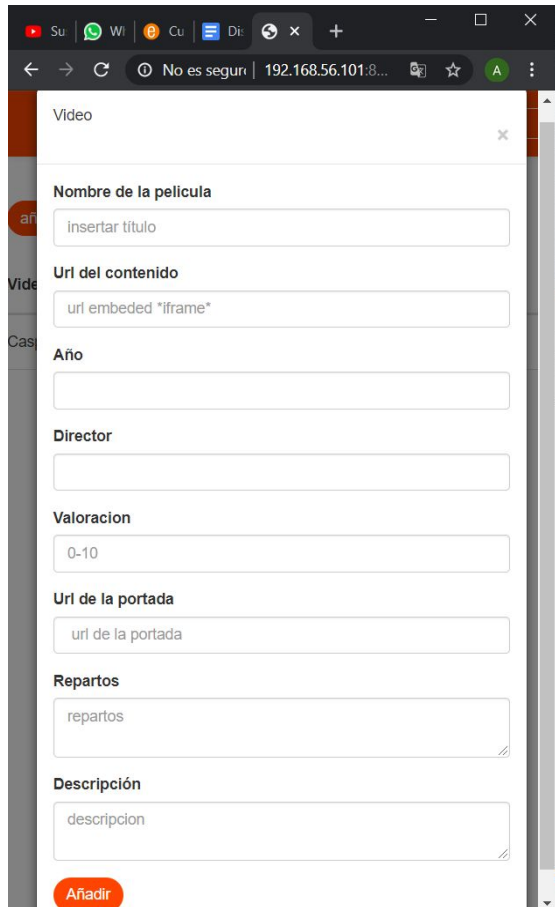


Observaciones:

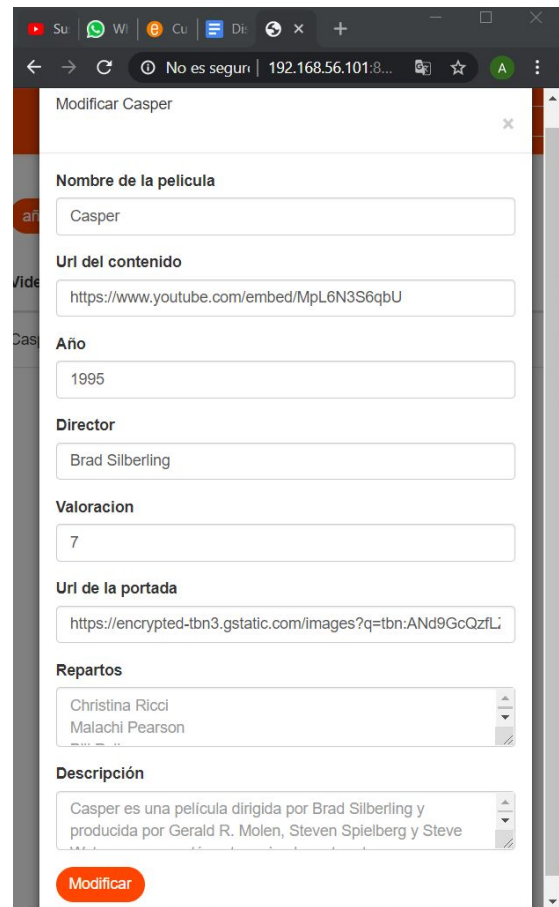
En la página home de la aplicación no se mostrará ningún video/pelicula inicialmente ya que la base de datos, si se arranca por primera vez, aún está vacía. Se tendrá que rellenar los formularios en la administración de "películas" para poder poblar la bbdd. Ídem para los usuarios.

Se puede observar que el superusuario creado no se puede eliminar o modificar los datos del mismo mediante una sesión de un cliente (frontend). Sin embargo, los usuarios que son administradores si se permite operar sobre ellos.

En el formulario de administración de película se debe poner la URL/src embeded" indicado:



A screenshot of a web browser showing a form titled 'Video' with a close button. The form contains several input fields: 'Nombre de la película' with placeholder text 'insertar título'; 'Url del contenido' with placeholder text 'url embeded "iframe"'; 'Año'; 'Director'; 'Valoracion' with a range '0-10'; 'Url de la portada' with placeholder text 'url de la portada'; 'Repartos' with placeholder text 'repartos'; and 'Descripción' with placeholder text 'descripcion'. At the bottom left is an orange button labeled 'Añadir'.



A screenshot of a web browser showing a form titled 'Modificar Casper' with a close button. The form contains several input fields: 'Nombre de la película' with the value 'Casper'; 'Url del contenido' with the value 'https://www.youtube.com/embed/MpL6N3S6qbU'; 'Año' with the value '1995'; 'Director' with the value 'Brad Silberling'; 'Valoracion' with the value '7'; 'Url de la portada' with the value 'https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcQzflZ57tTiollgClabK9DHWZZsw5adQTSv7_WQfaJg7uSy8qiP'; 'Repartos' with a list containing 'Christina Ricci' and 'Malachi Pearson'; and 'Descripción' with the text 'Casper es una película dirigida por Brad Silberling y producida por Gerald R. Molen, Steven Spielberg y Steve'. At the bottom left is an orange button labeled 'Modificar'.

Datos del formulario de la demo::

url del contenido: <https://www.youtube.com/embed/MpL6N3S6qbU>

url de la portada:

https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcQzflZ57tTiollgClabK9DHWZZsw5adQTSv7_WQfaJg7uSy8qiP

Otras Vistas:

