

Documentación



Pro Gol App

v 1.0

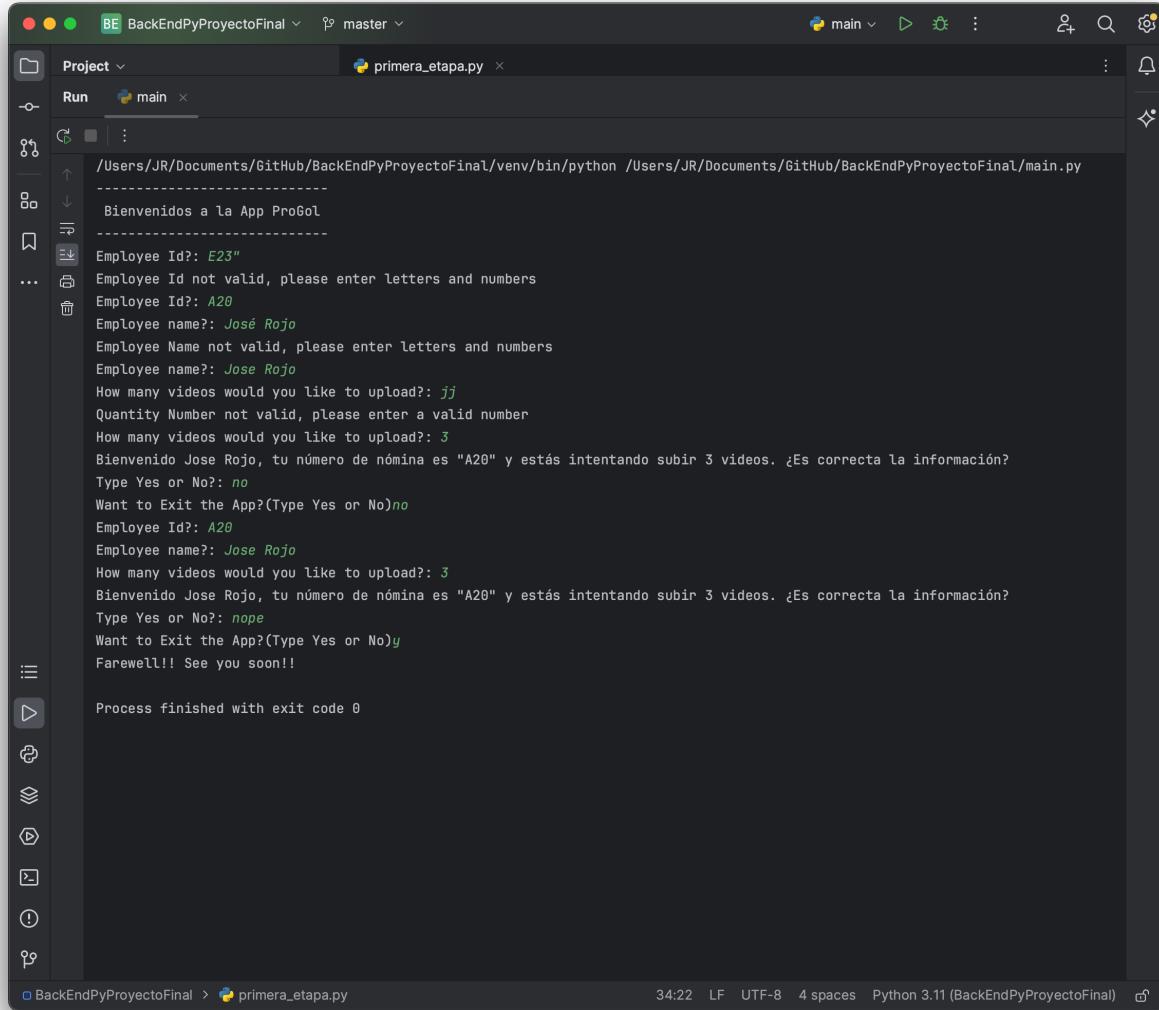
BACKEND PYTHON-DJANGO

José Arnoldo Rojo Borquez
Profesor Freddy Romero
TecMilenio
Noviembre 2023

Avance 1. Pro Gol App Terminal - Python

En este avance, tenemos una App que corre en Terminal, la cual ejecuta validaciones y condicionales, conduciéndote intuitivamente a través de sus campos inputs, confirmando pues, que se ingresen los datos correctos, evitando que ocasionen conflictos en las bases de datos.

Para esto, se creó una carpeta donde descansa todo el proyecto final, creando el environment de Python, que después se utilizará en las demás etapas del proyecto.



```
BE BackEndPyProyectoFinal master
Project Run main primera_etapa.py
-----
/Users/JR/Documents/GitHub/BackEndPyProyectoFinal/venv/bin/python /Users/JR/Documents/GitHub/BackEndPyProyectoFinal/main.py
-----
Bienvenidos a la App ProGol
-----
Employee Id?: E23"
Employee Id not valid, please enter letters and numbers
Employee Id?: A20
Employee name?: José Rojo
Employee Name not valid, please enter letters and numbers
Employee name?: Jose Rojo
How many videos would you like to upload?: jj
Quantity Number not valid, please enter a valid number
How many videos would you like to upload?: 3
Bienvenido Jose Rojo, tu número de nómina es "A20" y estás intentando subir 3 videos. ¿Es correcta la información?
Type Yes or No?: no
Want to Exit the App?(Type Yes or No)no
Employee Id?: A20
Employee name?: Jose Rojo
How many videos would you like to upload?: 3
Bienvenido Jose Rojo, tu número de nómina es "A20" y estás intentando subir 3 videos. ¿Es correcta la información?
Type Yes or No?: nope
Want to Exit the App?(Type Yes or No)y
Farewell!! See you soon!!

Process finished with exit code 0
34:22 LF UTF-8 4 spaces Python 3.11 (BackEndPyProyectoFinal)
```

En esta imagen mostramos el uso de la App ProGol versión Terminal, donde mostramos el uso de las validaciones, además de la ruta en el formulario de SI/NO, de las opciones que nos muestran.

Esta es la siguiente ruta de las opciones SI/NO, donde introducimos los videos, aplicando también las validaciones pertinentes, arrojándonos al final, una lista donde están almacenados los datos introducidos, a manera de confirmación. Hay que recordar, que dentro de sus funciones, itera tantas veces, como el número de videos que desea el usuario introducir.

```

BackEndPyProyectoFinal master
Run main
Project /Users/JR/Documents/GitHub/BackEndPyProyectoFinal/venv/bin/python /Users/JR/Documents/GitHub/BackEndPyProyectoFinal/main.py
-----
Bienvenidos a la App ProGol
-----
Employee Id?: A20
Employee name?: Jose Rojo
How many videos would you like to upload?: 3
Bienvenido Jose Rojo, tu número de nómina es "A20" y estás intentando subir 3 videos. ¿Es correcta la información?
Type Yes or No?: yes
Video Title: -2$ 
Title not valid, please enter letters and numbers
Video Title: Clase 2
Video Name: Proa-- 
Name not valid, please enter letters and numbers
Video Name: Programacion Orientada a Objetos
Video Extension: &/8
Extension not valid, please enter letters and numbers
Video Extension: mov
Video Size (Max 3MB): 8
;Maximum file size 3MB, Try again!
Video Size (Max 3MB): i
Only numbers are allowed, could not convert string to float: 'i'
Video Size (Max 3MB): 3
Video Title: Barrer
Video Name: Tecnicas de Limpieza
Video Extension: avi
Video Size (Max 3MB): 2.3
Video Title: Penales
Video Name: Los mejores penales de la historia
Video Extension: mov
Video Size (Max 3MB): 0.2
[['Clase 2', 'Programacion Orientada a Objetos', 'mov', 3.0], ['Barrer', 'Tecnicas de Limpieza', 'avi', 2.3], ['Penales', 'Los mejores penales de la historia', 'mov', 0.2]]
;;Saved Successfully!!

Process finished with exit code 0
  
```

The terminal window shows the execution of the script 'primera_etapa.py'. It prompts for employee information and video details. It handles validation errors for invalid titles and extensions. Finally, it prints a list of videos and their details, confirming successful saving.

Se crea un archivo de texto 'salida.txt', donde se exponen los datos que introdujimos en nuestra App, de los datos almacenados, mediante una función de open('salida.txt', 'a').

```

BackEndPyProyectoFinal master
Run main
Project /Users/JR/Documents/GitHub/BackEndPyProyectoFinal/venv/bin/python /Users/JR/Documents/GitHub/BackEndPyProyectoFinal/main.py
salida.txt
1 |N23 | Jose | 2 |
2 |Locos || Locos Adams || mov || 0.0 || Naranja || Naranja Mecanica || juj || 0.1 ||A20 | Jose Rojo | 3 |
3 | Clase 2 || Programacion Orientada a Objetos || mov || 3.0 || Barrer || Tecnicas de Limpieza || avi || 2.3 || Penales |
  
```

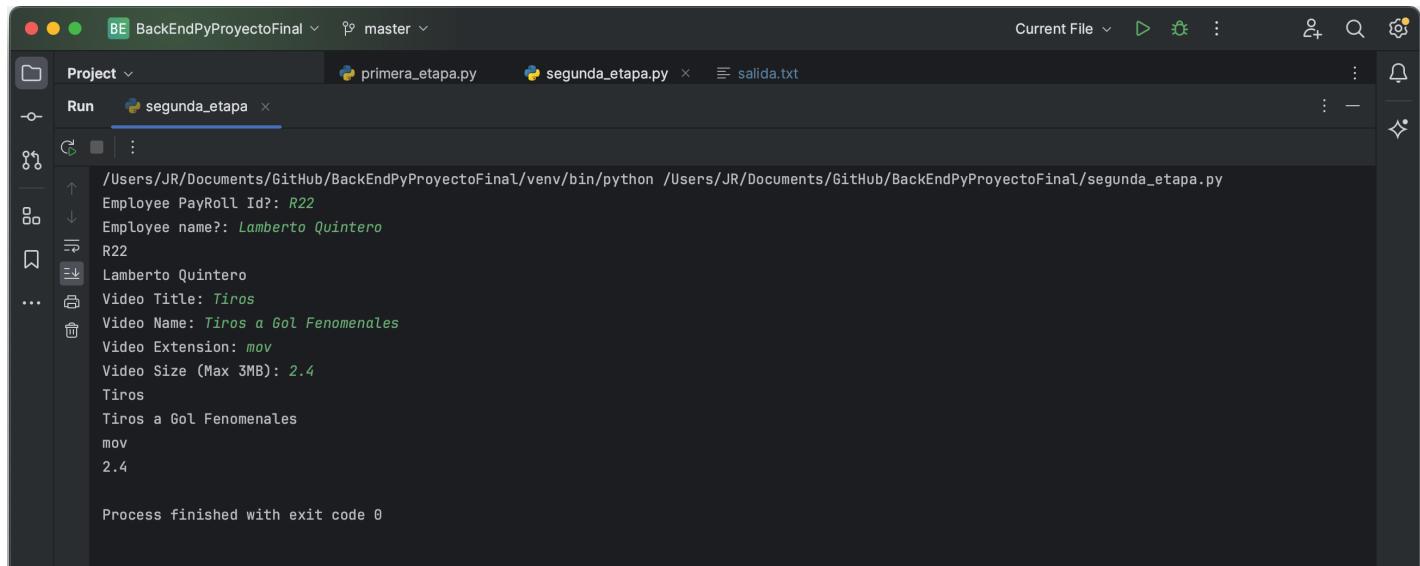
The terminal window shows the contents of the file 'salida.txt'. It contains three lines of data, each representing a video entry with fields separated by spaces.

Avance 2. Pro Gol App Terminal - Python POO

En el Avance 2, la App es una réplica de la App del Avance 1, la misma funcionalidad y las mismas validaciones, por lo cual mostraremos un ejemplo, para que puedan observar su funcionalidad.

La única diferencia, y lo podemos apreciar en el código, es, que esta App, instancia la información de 2 Clases, Persona y Videos, por lo que se vuelve más orientada a Objetos. Usando métodos para instanciar objetos.

Al final, se toman los atributos de estos objetos y los metemos a una lista, para después crear un archivo de texto 'salida2.txt' para poder mostrar esta información.



```
/Users/JR/Documents/GitHub/BackEndPyProyectoFinal/venv/bin/python /Users/JR/Documents/GitHub/BackEndPyProyectoFinal/segunda_etapa.py
Employee PayRoll Id?: R22
Employee name?: Lambert Quintero
R22
Lambert Quintero
Video Title: Tiros
Video Name: Tiros a Gol Fenomenales
Video Extension: mov
Video Size (Max 3MB): 2.4
Tiros
Tiros a Gol Fenomenales
mov
2.4

Process finished with exit code 0
```



```
|R22||Lambert Quintero||Tiros||Tiros a Gol Fenomenales||mov||2.4|
```

Avance 3. BD PostgreSQL

Para este avance, creamos la BD pro_gol en PostgreSQL, desde terminal, al igual que sus tablas y campos; mismas que tuvimos que eliminar, puesto que al usar Django en la etapa final, Django-Models, crea estos campos y tablas dentro de nuestra base de datos.

Mostraremos a continuación la estructura de nuestra base de datos usando Terminal:

```
JR — psql -U JR pro_gol — 105x63
Last login: Mon Nov  6 09:08:21 on ttys001
[JR@MacBook-Pro-JR ~ % psql -U JR pro_gol
psql (14.9 (Homebrew))
Type "help" for help.

[pro_gol=# \d
              List of relations
 Schema |           Name            |   Type   | Owner
        | auth_group          | table    | JR
        | auth_group_id_seq   | sequence | JR
        | auth_group_permissions | table   | JR
        | auth_group_permissions_id_seq | sequence | JR
        | auth_permission      | table    | JR
        | auth_permission_id_seq | sequence | JR
        | auth_user             | table    | JR
        | auth_user_groups     | table    | JR
        | auth_user_groups_id_seq | sequence | JR
        | auth_user_id_seq     | sequence | JR
        | auth_user_permissions | table    | JR
        | auth_user_user_permissions_id_seq | sequence | JR
        | django_admin_log     | table    | JR
        | django_admin_log_id_seq | sequence | JR
        | django_content_type  | table    | JR
        | django_content_type_id_seq | sequence | JR
        | django_migrations    | table    | JR
        | django_migrations_id_seq | sequence | JR
        | django_session        | table    | JR
        | videos_tbl_users      | table    | JR
        | videos_tbl_videos     | table    | JR
        | videos_tbl_videos_id_seq | sequence | JR
(22 rows)

[pro_gol=# \d videos_tbl_users;
               Table "public.videos_tbl_users"
 Column |          Type          | Collation | Nullable | Default
        | character varying(5) |           | not null |
 user_name | character varying(50) |           | not null |
Indexes:
    "videos_tbl_users_pkey" PRIMARY KEY, btree (user_id)
    "videos_tbl_users_user_id_68d9450e_like" btree (user_id varchar_pattern_ops)
Referenced by:
    TABLE "videos_tbl_videos" CONSTRAINT "videos_tbl_videos_user_id_id_49d0d190_fk_videos_tb" FOREIGN KEY
    (user_id_id) REFERENCES videos_tbl_users(user_id) DEFERRABLE INITIALLY DEFERRED

[pro_gol=# \d videos_tbl_videos;
               Table "public.videos_tbl_videos"
 Column |          Type          | Collation | Nullable | Default
        | integer               |           | not null | generated by default as identity
 video_title | character varying(50) |           | not null |
 video_name | character varying(50) |           | not null |
 video_ext | character varying(4) |           | not null |
 user_id_id | character varying(5) |           | not null |
Indexes:
    "videos_tbl_videos_pkey" PRIMARY KEY, btree (id)
    "videos_tbl_videos_user_id_id_49d0d190" btree (user_id_id)
    "videos_tbl_videos_user_id_id_49d0d190_like" btree (user_id_id varchar_pattern_ops)
Foreign-key constraints:
    "videos_tbl_videos_user_id_id_49d0d190_fk_videos_tb" FOREIGN KEY (user_id_id) REFERENCES videos_tbl_u
sers(user_id) DEFERRABLE INITIALLY DEFERRED
```

Dentro de la estructura de nuestra Base de Datos, en la tabla 'videos_tbl_videos' contamos con un campo con relación uno a muchos, para que almacene la información del usuario o empleado.

```
JR — psql -U JR pro_gol — 105x63

public | auth_permission_id_seq           | sequence | JR
public | auth_user                      | table    | JR
public | auth_user_groups               | table    | JR
public | auth_user_groups_id_seq        | sequence | JR
public | auth_user_id_seq                | sequence | JR
public | auth_user_permissions          | table    | JR
public | auth_user_user_permissions_id_seq | sequence | JR
public | django_admin_log               | table    | JR
public | django_admin_log_id_seq        | sequence | JR
public | django_content_type            | table    | JR
public | django_content_type_id_seq     | sequence | JR
public | django_migrations              | table    | JR
public | django_migrations_id_seq       | sequence | JR
public | django_session                 | table    | JR
public | videos_tbl_users               | table    | JR
public | videos_tbl_videos              | table    | JR
public | videos_tbl_videos_id_seq        | sequence | JR
(22 rows)

[pro_gol=# \d videos_tbl_users;
   Table "public.videos_tbl_users"
 Column |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 user_id | character varying(5) |           | not null |
 user_name | character varying(50) |           | not null |
Indexes:
 "videos_tbl_users_pkey" PRIMARY KEY, btree (user_id)
 "videos_tbl_users_user_id_68d9450e_like" btree (user_id varchar_pattern_ops)
Referenced by:
 TABLE "videos_tbl_videos" CONSTRAINT "videos_tbl_videos_user_id_id_49d0d190_fk_videos_tb" FOREIGN KEY
 (user_id_id) REFERENCES videos_tbl_users(user_id) DEFERRABLE INITIALLY DEFERRED

[pro_gol=# \d videos_tbl_videos;
   Table "public.videos_tbl_videos"
 Column |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 id    | integer        |           | not null | generated by default as identity
 video_title | character varying(50) |           | not null |
 video_name  | character varying(50) |           | not null |
 video_ext   | character varying(4)  |           | not null |
 user_id_id  | character varying(5) |           | not null |
Indexes:
 "videos_tbl_videos_pkey" PRIMARY KEY, btree (id)
 "videos_tbl_videos_user_id_id_49d0d190" btree (user_id_id)
 "videos_tbl_videos_user_id_id_49d0d190_like" btree (user_id_id varchar_pattern_ops)
Foreign-key constraints:
 "videos_tbl_videos_user_id_id_49d0d190_fk_videos_tb" FOREIGN KEY (user_id_id) REFERENCES videos_tbl_u
sers(user_id) DEFERRABLE INITIALLY DEFERRED

[pro_gol=# SELECT * FROM videos_tbl_users;
 user_id | user_name
-----+-----
 N90    | Emb 12
(1 row)

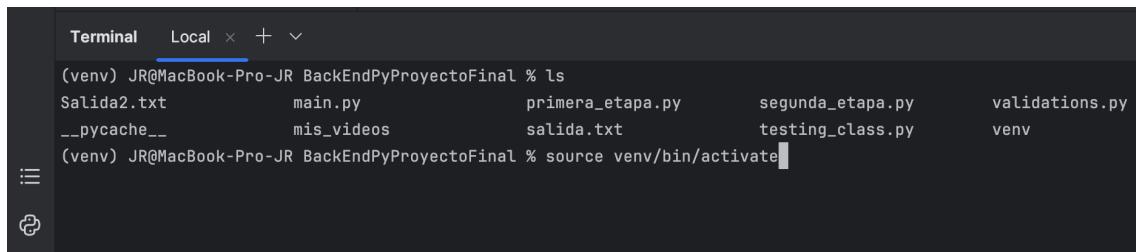
[pro_gol=# SELECT * FROM videos_tbl_videos;
 id | video_title | video_name    | video_ext | user_id_id
-----+-----+-----+-----+
 22 | Show        | The Business Stg | mov       | N90
 23 | Check        | Availability report | avi       | N90
(2 rows)

pro_gol=# ]
```

Avance 4. Pro Gol App Python - Django

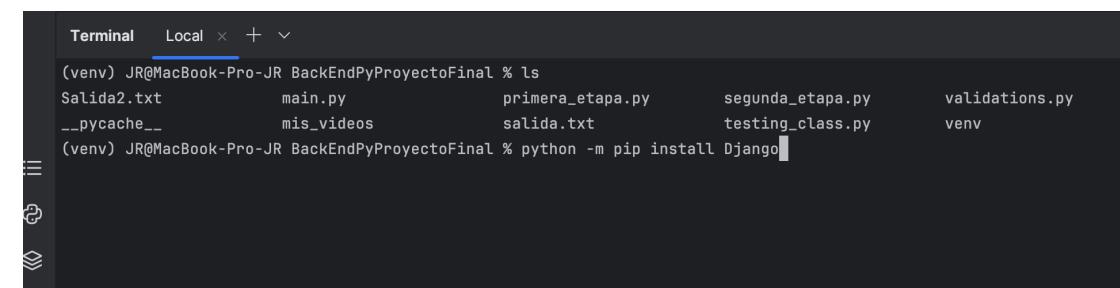
En el Avance 4, creamos la App con la ayuda del Framework Django, para una interfaz del usuario más agradable y más intuitiva, a continuación explicamos el proceso para crear nuestra App y su funcionamiento.

1. Activar el environment de Python, instalar Django, componentes e instalar la App:



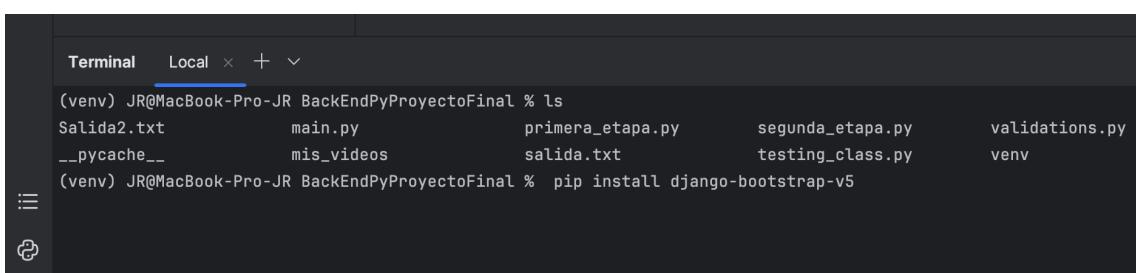
```
Terminal Local + 
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % ls
Salida2.txt      main.py      primera_etapa.py      segunda_etapa.py      validations.py
__pycache__      mis_videos   salida.txt       testing_class.py    venv
```

(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % source venv/bin/activate



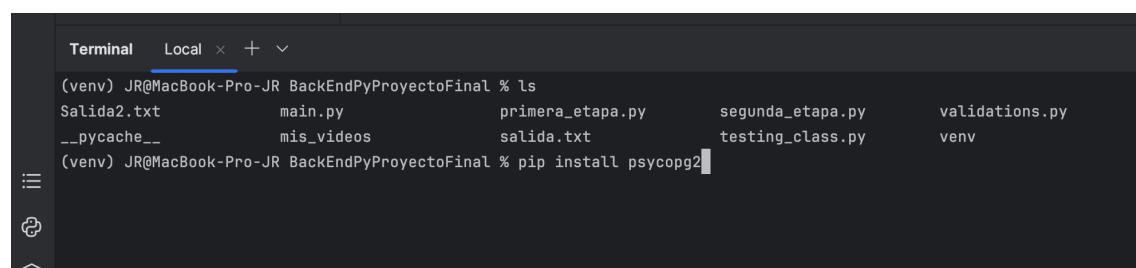
```
Terminal Local + 
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % ls
Salida2.txt      main.py      primera_etapa.py      segunda_etapa.py      validations.py
__pycache__      mis_videos   salida.txt       testing_class.py    venv
```

(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % python -m pip install Django



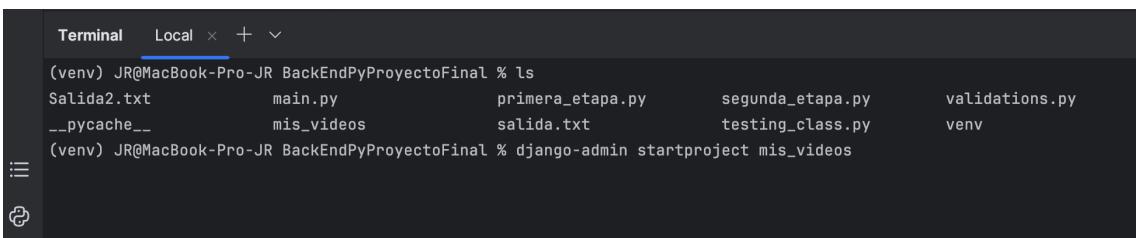
```
Terminal Local + 
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % ls
Salida2.txt      main.py      primera_etapa.py      segunda_etapa.py      validations.py
__pycache__      mis_videos   salida.txt       testing_class.py    venv
```

(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % pip install django-bootstrap-v5



```
Terminal Local + 
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % ls
Salida2.txt      main.py      primera_etapa.py      segunda_etapa.py      validations.py
__pycache__      mis_videos   salida.txt       testing_class.py    venv
```

(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % pip install psycopg2

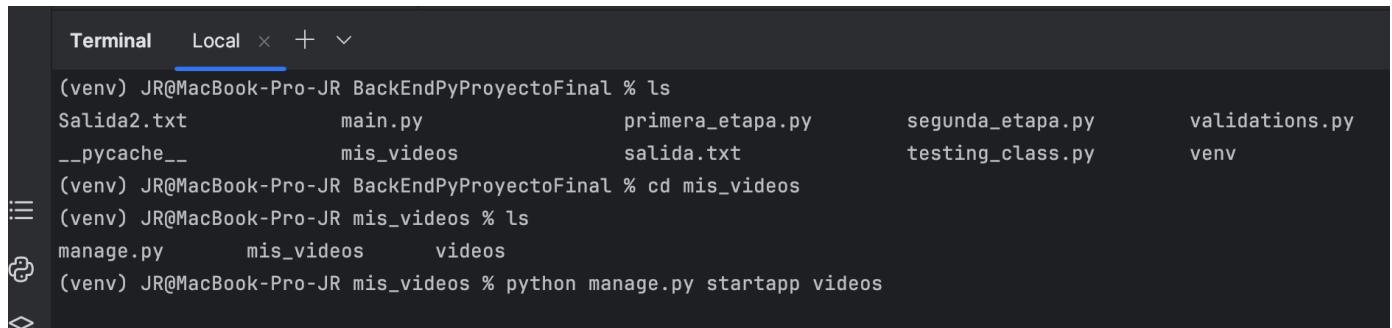


```
Terminal Local + 
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % ls
Salida2.txt      main.py      primera_etapa.py      segunda_etapa.py      validations.py
__pycache__      mis_videos   salida.txt       testing_class.py    venv
```

(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % django-admin startproject mis_videos

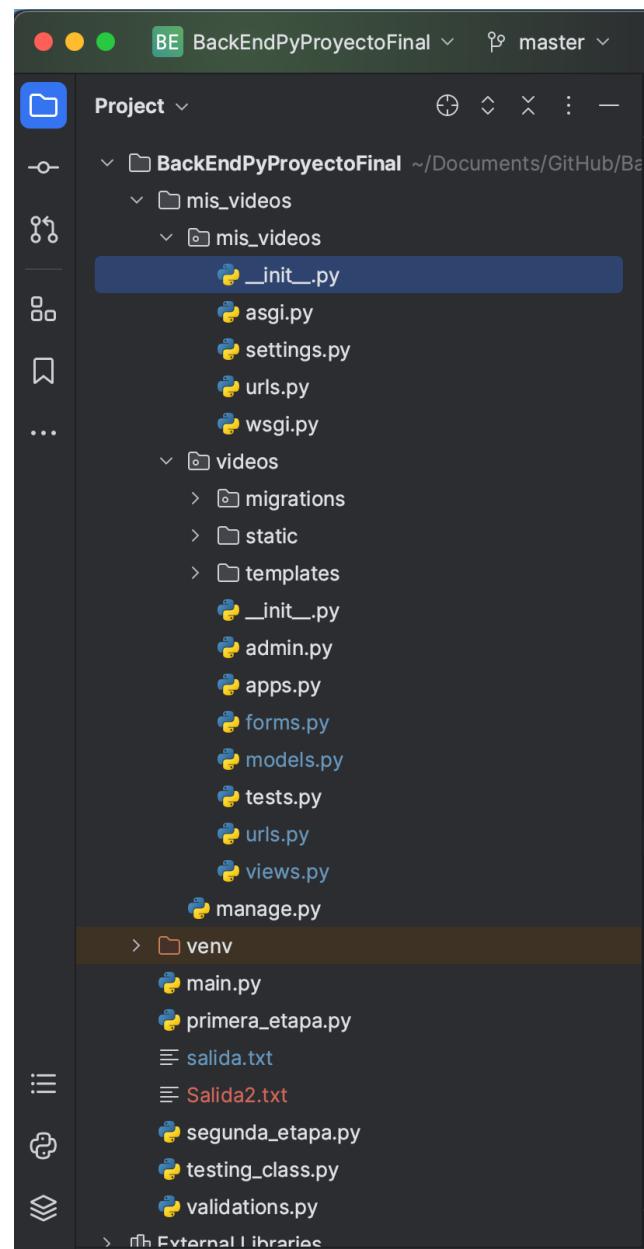
2. Nos queda esta estructura, nos disponemos a configurar nuestra App crear nuestra App Videos, crear nuestros Modelos-Vistas-Templates, de acuerdo con el approach Django:

Crear App Videos

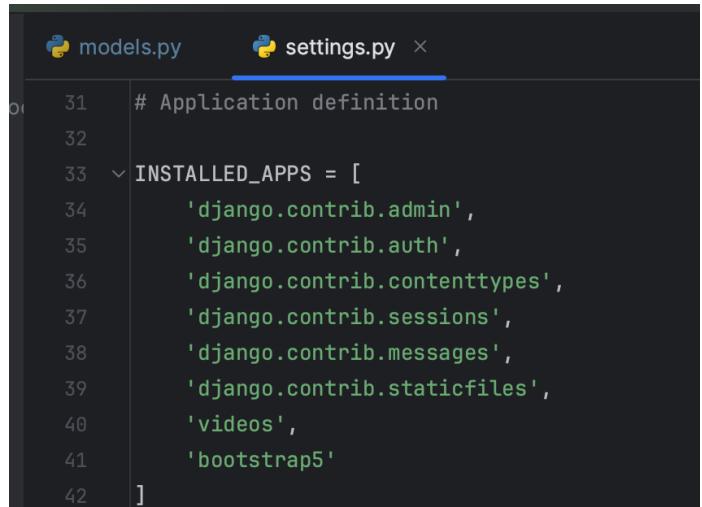


```
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % ls
Salida2.txt          main.py           primera_etapa.py      segunda_etapa.py    validations.py
__pycache__          mis_videos       salida.txt          testing_class.py  venv
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % cd mis_videos
(venv) JR@MacBook-Pro-JR mis_videos % ls
manage.py            mis_videos      videos
(venv) JR@MacBook-Pro-JR mis_videos % python manage.py startapp videos
```

Estructura de Archivos



Incluir la App en settings



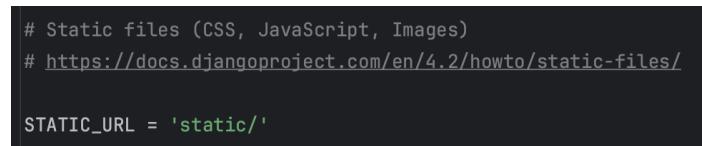
```
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'videos',
41     'bootstrap5'
42 ]
```

Configurar nuestra BD



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'pro_gol',
        'USER': 'JR',
        'PASSWORD': 'root',
        'HOST': 'localhost',
        'PORT': '5432'
    }
}
```

Incluir las páginas estáticas



```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/
STATIC_URL = 'static/'
```

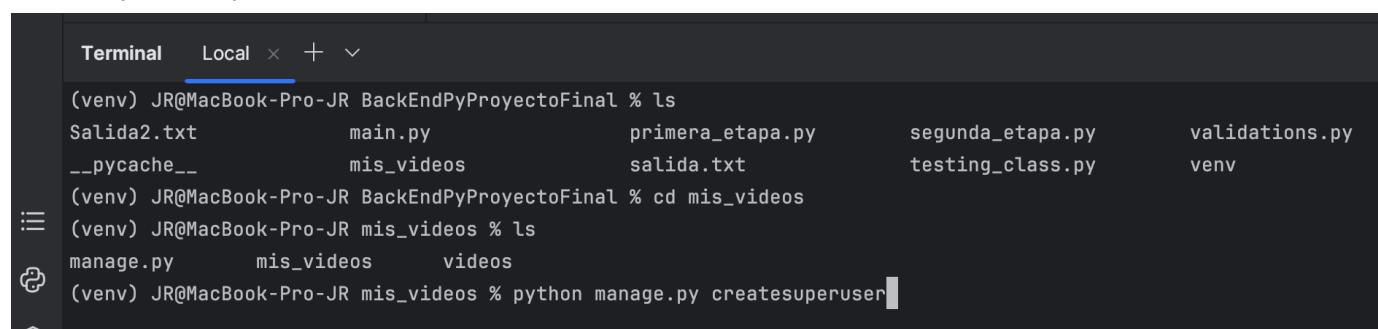
Incluir las urls de la app Videos

```
> import ...

urlpatterns = [
    path("", include("videos.urls")),
    path('admin/', admin.site.urls),
]
```

```
1 from django.urls import path
2 from . import views
3
4 #My paths to the end-points we made in Views
5 urlpatterns = [
6     path('', views.main, name='main'),
7     path('users/', views.users, name='users'),
8     path('users/details/<user_id>', views.details, name='details'),
9     path('videos/', views.videos, name='videos'),
10    path('videos/video_detail/<user_id>/', views.video_detail, name='video_detail'),
11    path('create_user/', views.create_user, name='create_user'),
12    path('edit_user/<user_id>/', views.edit_user, name='edit_user'),
13    path('delete_user/<user_id>/', views.delete_user, name='delete_user'),
14    path('add_video/', views.add_video, name='add_videos'),
15    path('edit_video/<id>/', views.edit_video, name='edit_video'),
16    path('delete_video/<id>/', views.delete_video, name='delete_video')
]
```

Crear SuperUser para la GUI de Admin



The screenshot shows a terminal window with the following command history:

```
Terminal Local × + ▾
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % ls
Salida2.txt          main.py          primera_etapa.py      segunda_etapa.py      validations.py
__pycache__          mis_videos       salida.txt        testing_class.py    venv
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % cd mis_videos
(venv) JR@MacBook-Pro-JR mis_videos % ls
manage.py      mis_videos      videos
(venv) JR@MacBook-Pro-JR mis_videos % python manage.py createsuperuser
```

3. Modelos

The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure for 'BackEndPyProyectoFinal'. The main editor window shows the contents of 'models.py'.

```

from django.db import models

# Create your models here.

class tbl_users(models.Model):
    user_id = models.CharField(max_length=5, primary_key=True)
    user_name = models.CharField(max_length=50, null=False)

    def __str__(self):
        return f'{self.user_id} {self.user_name}'

class tbl_videos(models.Model):
    id = models.AutoField(primary_key=True)
    user_id = models.ForeignKey(tbl_users, on_delete=models.CASCADE, default=None)
    video_title = models.CharField(max_length=50)
    video_name = models.CharField(max_length=50)
    video_ext = models.CharField(max_length=4)

    def __str__(self):
        return f'{self.user_id} {self.video_title}'

```

The terminal tab at the bottom shows the command-line history:

```

__pycache__ mis_videos salida.txt testing_class.py venv
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % cd mis_videos
(venv) JR@MacBook-Pro-JR mis_videos % ls
manage.py mis_videos videos
(venv) JR@MacBook-Pro-JR mis_videos % python manage.py startapp videos

```

Hacer mis migraciones

The terminal window shows the following commands being run:

```

Salida2.txt          main.py          primera_etapa.py      seg
__pycache__          mis_videos       salida.txt          tes
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % cd mis_videos
(venv) JR@MacBook-Pro-JR mis_videos % ls
manage.py           mis_videos       videos
(venv) JR@MacBook-Pro-JR mis_videos % python manage.py makemigrations
```
Salida2.txt main.py primera_etapa.py seg
__pycache__ mis_videos salida.txt tes
(venv) JR@MacBook-Pro-JR BackEndPyProyectoFinal % cd mis_videos
(venv) JR@MacBook-Pro-JR mis_videos % ls
manage.py mis_videos videos
(venv) JR@MacBook-Pro-JR mis_videos % python manage.py migrate
```

```

4. Vistas

```

1  from django.shortcuts import render, redirect
2  from django.http import HttpResponseRedirect
3  from django.template import loader
4  from .models import tbl_videos
5  from .models import tbl_users
6  from .forms import UserForm, VideoForm
7  from django.db import connections
8
9
10 # Create your views here.
11
12 #Main index page.
13 def main(request):
14     template = loader.get_template("main.html")
15     return HttpResponseRedirect(template.render())
16
17 #Shows a list of all users
18 def users(request):
19     users = tbl_users.objects.all().values()
20
21     return render(request, template_name='users.html', context={'users': users})
22
23 #Show details of each user in details.html
24 def details(request, user_id):
25     user = tbl_users.objects.get(user_id=user_id)
26
27     return render(request, template_name='details.html', context={'user': user})
28
29 delete_video() if request.method == 'POST'

```

The screenshot shows a code editor with the views.py file open. The file contains Python code for a Django application. It includes imports for django.shortcuts, django.http, django.template, and django.db. The code defines two views: 'main' which returns the main index page, and 'users' which returns a list of all users. It also defines a 'details' view to show details of each user. A 'delete_video' function is present but commented out. The code editor interface includes a sidebar with project files like migrations, static, templates, and tests, and a terminal window at the bottom showing command-line history.

5. Template

```

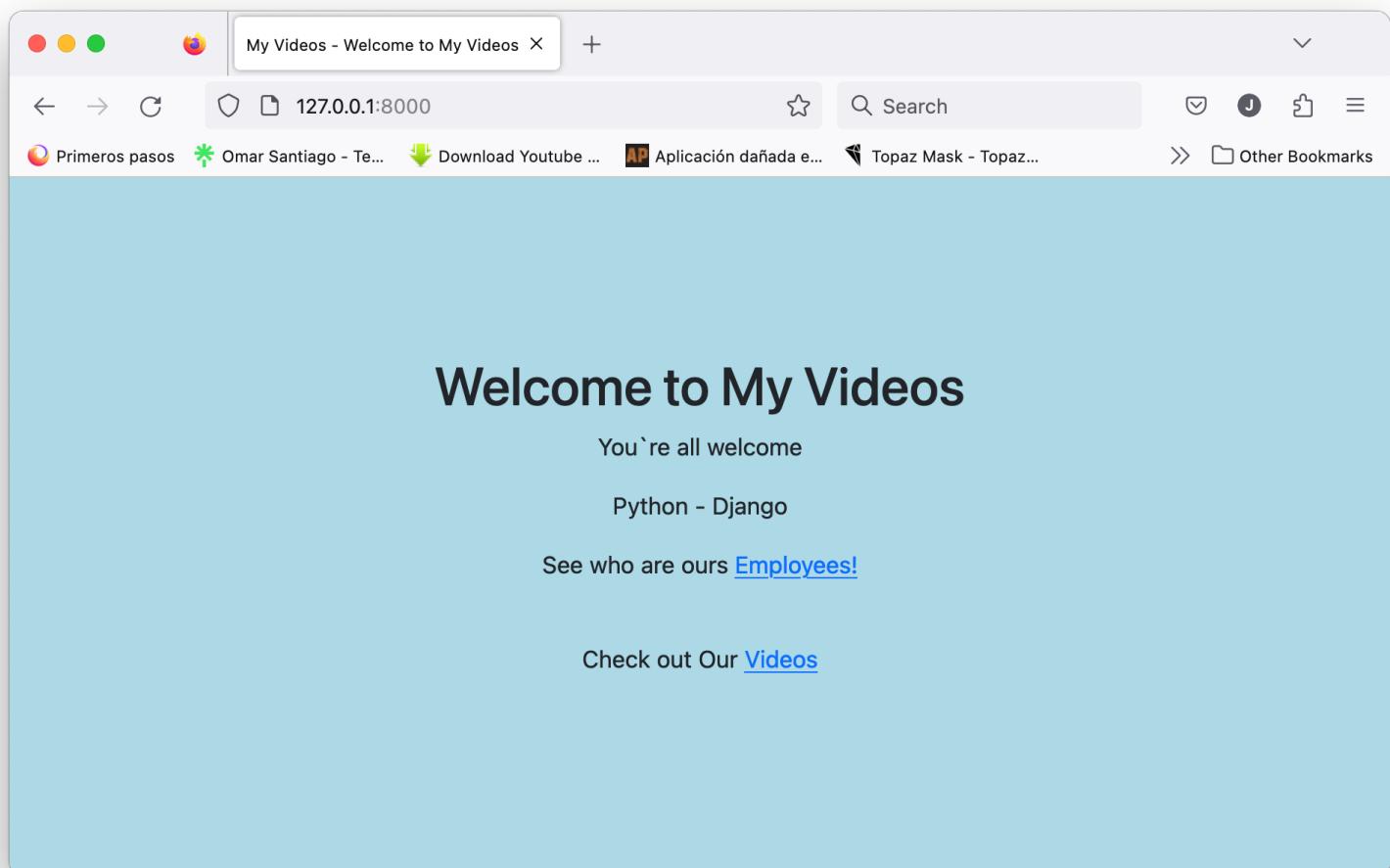
1  from django.shortcuts import render, redirect
2  from django.http import HttpResponseRedirect
3  from django.template import loader
4  from .models import tbl_videos
5  from .models import tbl_users
6  from .forms import UserForm, VideoForm
7  from django.db import connections
8
9
10 # Create your views here.
11
12 #Main index page.
13 def main(request):
14     template = loader.get_template("main.html")
15     return HttpResponseRedirect(template.render())
16
17 #Shows a list of all users
18 def users(request):
19     users = tbl_users.objects.all().values()
20
21     return render(request, template_name='users.html', context={'users': users})
22
23 #Show details of each user in details.html
24 def details(request, user_id):
25     user = tbl_users.objects.get(user_id=user_id)
26
27     return render(request, template_name='details.html', context={'user': user})
28
29 delete_video() if request.method == 'POST'

```

This screenshot is identical to the one above, showing the same views.py file and its contents. The only difference is the timestamp in the terminal window, which has changed from '10:34' to '10:35'.

Pro Gol App Python - Django

```
(venv) JR@MacBook-Pro-JR mis_videos % ls
manage.py      mis_videos      videos
(venv) JR@MacBook-Pro-JR mis_videos % python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
November 06, 2023 - 19:20:28
Django version 4.2.6, using settings 'mis_videos.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```



Lista de Empleados

Payroll Id	Employee Name	Edit	Delete
N90	Emb 12	Edit	Delete
R22	Lamberto Quintero	Edit	Delete

Add Employee
Back to [Home](#)

Añadir un Empleado

User id:
User name:

[Submit](#)
[Back to Employees List](#)

Lista de Empleados con Empleado Añadido

Payroll Id	Employee Name	Edit	Delete
N90	Emb 12	Edit	Delete
R22	Lamberto Quintero	Edit	Delete

Add [Employee](#)
Back to [Home](#)

Detalles de los empleados

Employee Info
Payroll Id: R22
Employee: Lamberto Quintero
[Back to Employees List](#)

Editar Registro de Empleado

User id:
User name:

[Submit](#)
[Back to Employees List](#)

Confirmar si desea borrar registro

Are you sure you want to delete "R22 Lamberto Quintero"

[Cancel](#) [Submit Query](#)

Lista de Videos

Payroll Id	Title	Name	Ext	Edit	Delete
N90 Elizabetha Ramos	Show	The Business Stg	mov	Edit	Delete
N90 Elizabetha Ramos	Check	Availability report	avi	Edit	Delete

Add a Video
Back to Home

Detalles de Video con campo de BD relacionando 1 a muchos

Payroll Id	Title	Name	Ext	Edit	Delete
N90 Elizabetha Ramos	Show	The Business Stg	mov	Edit	Delete
N90 Elizabetha Ramos	Check	Availability report	avi	Edit	Delete

Add a Video
Back to Videos
Back to Home

Añadir un Video

User id: N90 Elizabetha Ramos
Video title: Control
Video name: Total de Tu Vida
Video ext: mov
Submit
Back to [Videos List](#)

Editar un Video

User id: N90 Elizabetha Ramos
Video title: Check
Video name: Availability report
Video ext: avi
Submit Query
Back to [Videos List](#)

Lista después del CRUD

Payroll Id	Title	Name	Ext	Edit	Delete
N90 Elizabetha Ramos	Check	Availability report	avi	Edit	Delete
R22 Lamberto Quintero	James	James Bond 007	mpg4	Edit	Delete
R22 Lamberto Quintero	Tiros	Tiros Inigualables a Gol	mov	Edit	Delete
N90 Elizabetha Ramos	Control	Total de Tú Vida	mov	Edit	Delete

Add a Video
Back to Home

Mensaje para confirmar borrar registro

Are you sure you want to delete "N90 Elizabetha Ramos Check"
[Cancel](#) [Submit Query](#)

MUCHAS GRACIAS

Proyecto Final
BackEnd Python - Django

App Pro Gol

Pro Gol App

v 1.0

BACKEND PYTHON-DJANGO

José Arnoldo Rojo Borquez
Profesor Freddy Romero
TecMilenio
Noviembre 2023