# Playground Avanzado (Parte II)

**Login - Registro - Logout**

**Login y Logout**

Hemos logrado hacer **CRUD** con Profesores y con Cursos, profesores a mano y Cursos usando **ListView** y sus derivados.

Podríamos hacer lo mismo con todas las clases, pero….

¿Es adecuado que cualquiera con acceso a nuestra web pueda hacer ABM de nuestros modelos?

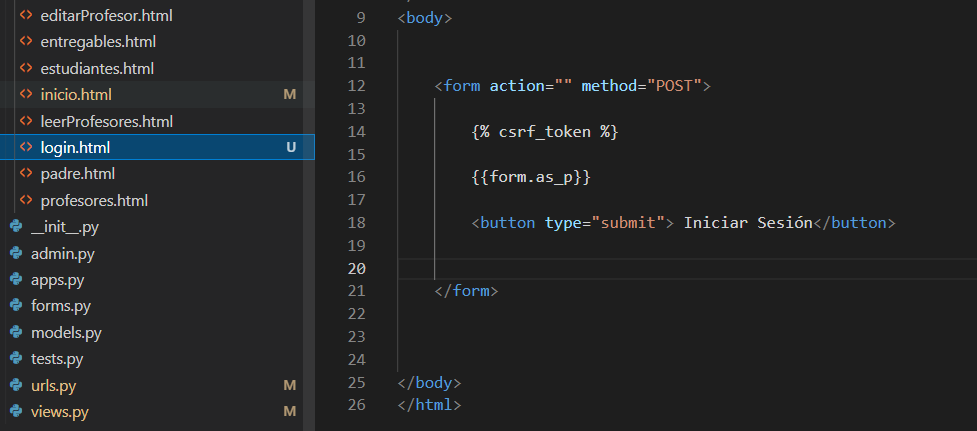
¡CLARO QUE NO!

**👉 Debemos limitar los permisos de los usuarios identificándolos con un login, similar al panel de admin.**

**Login**

Django nos provee de un Login y Logout muy sencillo, seguro y fácil de comprender.

👈 Arranquemos primero con un template simple para el login, login.html (form, automático de Django).

****

**Login - Script**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form action="" method="POST">

        {% csrf\_token %}

            {{ form.as\_p }}

        <button type="submit">Inciar sesion</button>

    </form>

</body>

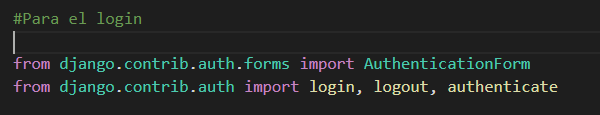
</html>

**Login**

Agregamos las urls.

****path('login', views.login\_request, name="Login")

Y en las views agregamos lo que necesitamos importar.

****from django.contrib.auth.forms import AuthenticationForm, UserCreationForm

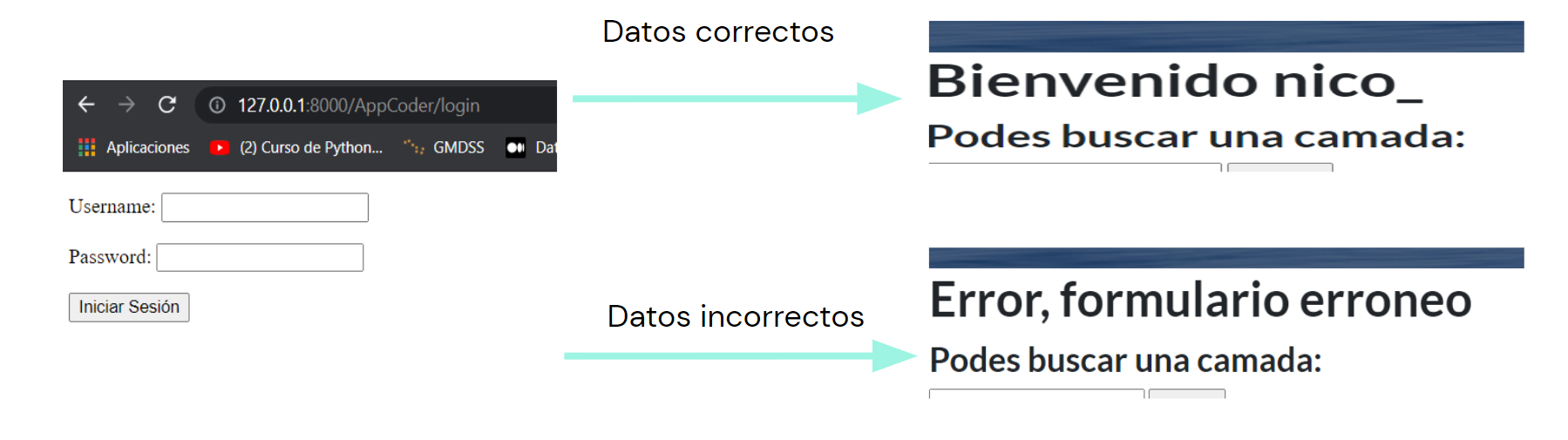
from django.contrib.auth import login, logout, authenticate

La vista

Ver script: Vista.txt

****

Resultados ✋

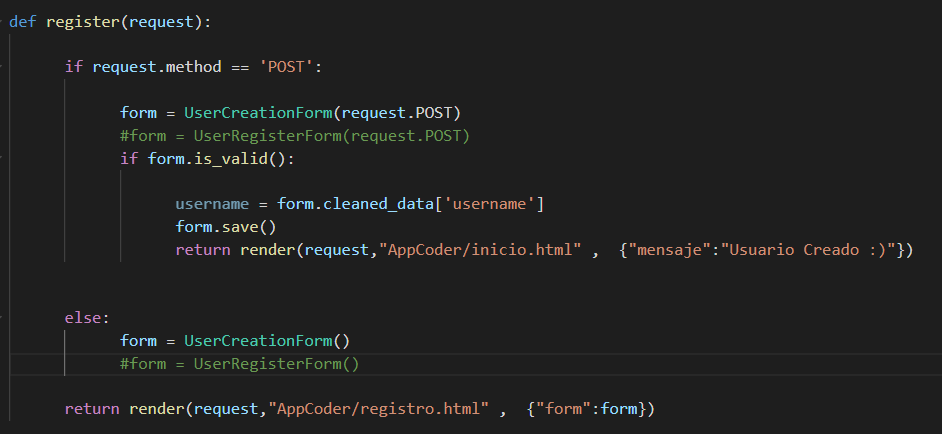
****

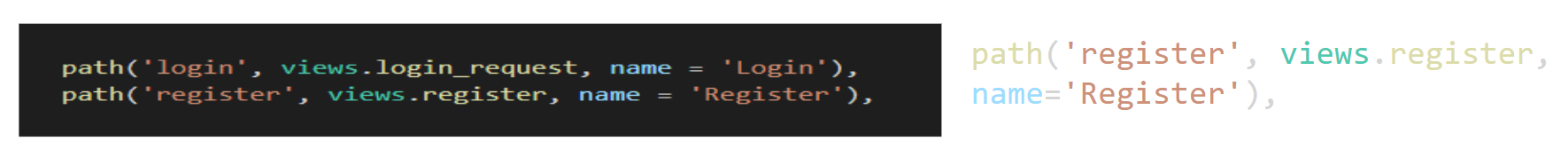
**Registro**

Lo anterior funciona si tenemos algún usuario creado por panel de administración, pero también podríamos registrar usuarios sin el panel.

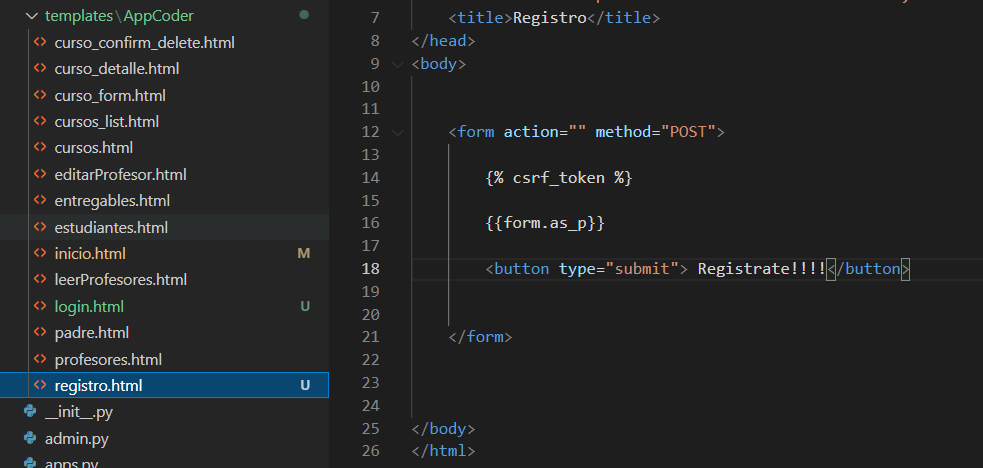
Veamos la vista.

**Ver: Registro.txt**

****

****

Modificamos las urls y creamos el template



Script

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form action="" method="POST">

        {% csrf\_token %}

        {{ form.as\_p }}

        <button type="submit">Registrate!</button>

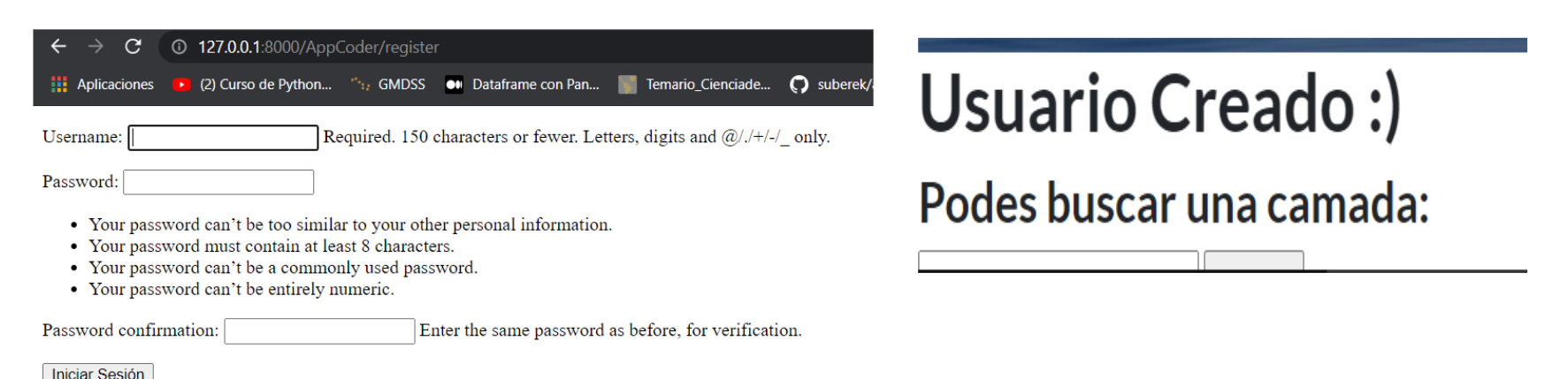
    </form>

</body>

</html>

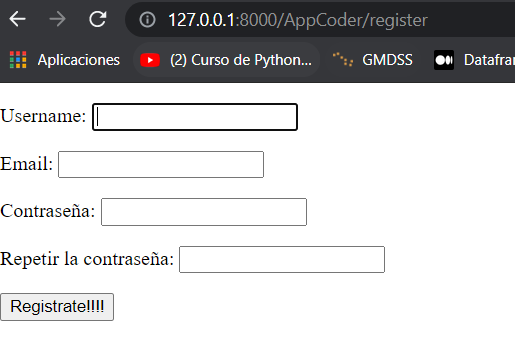
**Registro**

Nuestro resultado se verá así 😎

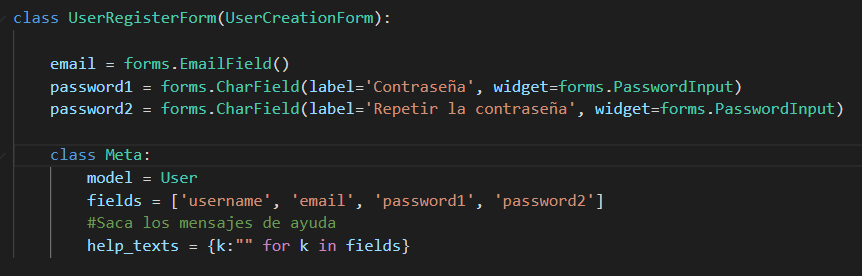


Funcionó, pero por defecto no se ve muy estético 😞

Lo arreglaremos para que se vea así



Para eso necesitamos heredar el formulario de Django y modificarlo a nuestro gusto, desde forms.py y luego usar “nuestro” form 👇



Script:

class UserRegisterForm(UserCreationForm):

    email = forms.EmailField()

    password1 = forms.CharField(label="Contraseña", widget=forms.PasswordInput)

    password2 = forms.CharField(label="Repetir contraseña", widget=forms.PasswordInput)

    class Meta:

        model = User

        fields = ['username', 'email', 'password1', 'password2']

        # Saca los mensajes de ayuda

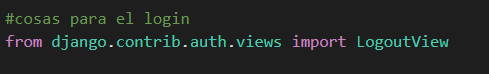
        help\_texts = {k:"" for k in fields}

Y en la vista usamos “nuestro” form 👇

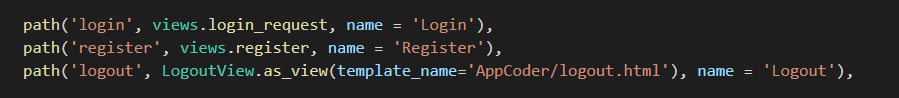


**Logout**

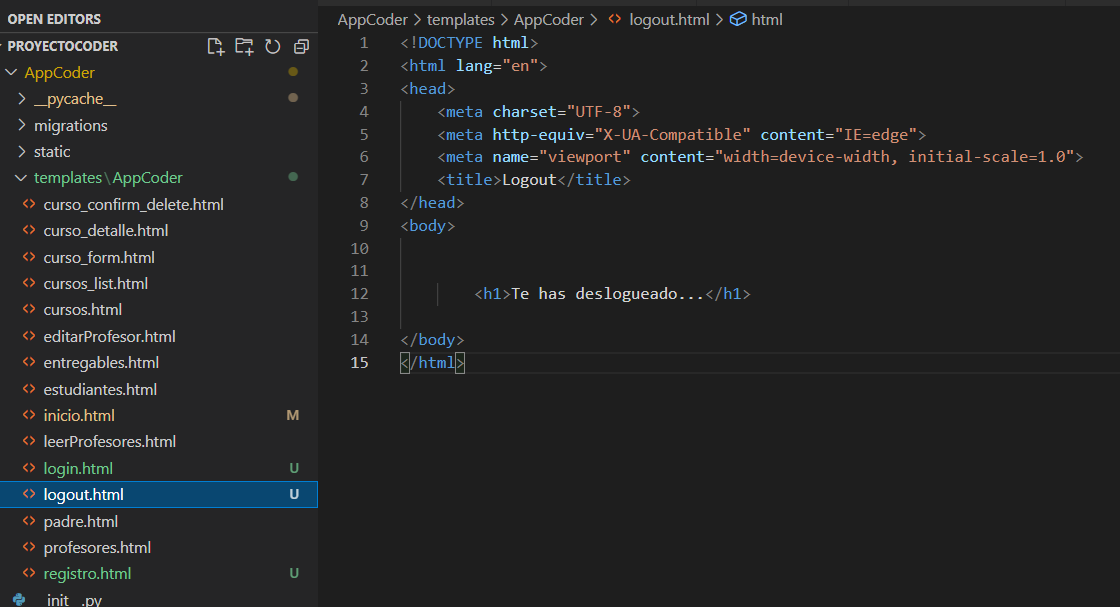
Django también tiene vistas listas para login y logout, usemos una vista predefinida para el Logout, en urls.py



from django.contrib.auth.views import LogoutView

path('logout', LogoutView.as\_view(template\_name='AppCoder/logout.html'), name='Logout'),

Creamos el logout.html 😮



Script

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1>Te has deslogueado...</h1>

</body>

</html>

**Mixin y Decoradores**

**Mixin**

Es posible que solo personas con login puedan ver nuestra página web, para ello existen los mixin y los decoradores que tienen como función validar lógica en nuestras views.

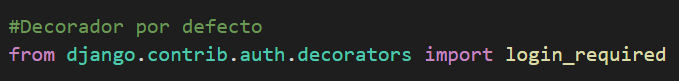
Si deseas que solo se pueda acceder a una Clase estando registrado podemos usar Mixin de Django:

**from django.contrib.auth.mixins import LoginRequiredMixin**

**class ClaseQueNecesitaLogin (LoginRequiredMixin):**

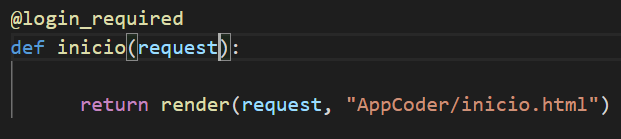
💃 ¡Listo!, solo podrán hacer uso de la clase los usuarios registrados.

**Decoradores**



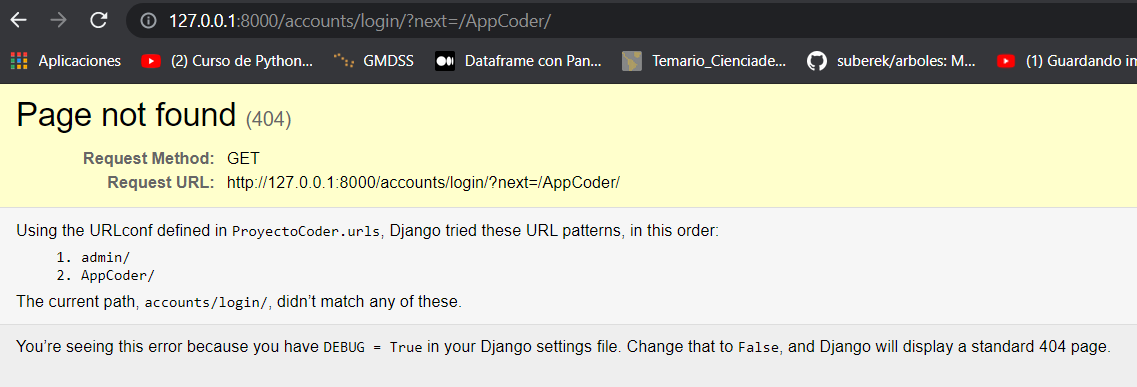
Nos sirven para validar identidad rápidamente, pero están más orientados a vistas y def.

👈 Primero usaremos el decorador más simple y ya dado por Django.



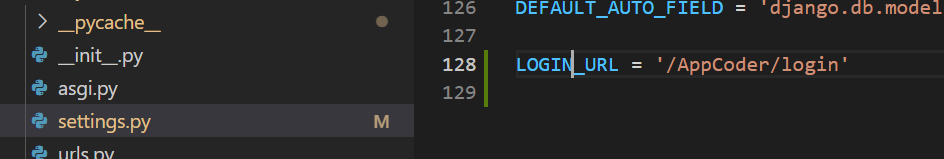
Al tratar entrar a la página de Inicio nos prohíbe el ingreso y nos sale lo siguiente

👇



Eso es porque no estamos logueados, entonces nos quiere redireccionar a un lugar predefinido que no existe.

Lo tenemos que modificar desde settings.py para que nos lleve a nuestro login



¡LISTO! 💃