Clase 20 - Playground Intermedio (Parte 2)

Herencia de templates

¿En qué consiste?

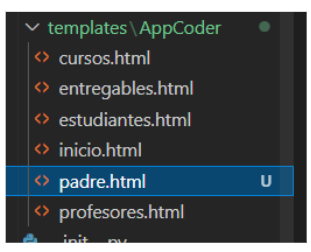
Consiste en usar de “base” determinadas cosas que estarán en todos nuestros templates. Suele ser la barra del menú principal, el pie de página, logo, etc. Son cosas que serán idénticas en cada página.

Con base en donde habíamos llegado antes, podemos definir que los siguientes bloques serán estáticos:



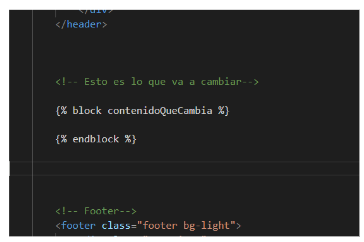
Esto se realiza dejando todo lo que no cambiará en un template “padre”, y haciendo que los templates que usan esas partes sean “hijos” que heredan esa estructura. Hacerlo es muy simple:

1. Crearemos un template “padre”, lo llamaremos padre.html, y ahí ponemos todo lo que teníamos en inicio.html.



Hasta el momento, en la plantilla padre.html solo tenemos una repetición de inicio.html.

1. Dentro de padre.html, creamos un segmento que será el que cambiará según la vista:



Es decir, todo será fijo, menos lo que esté dentro de este bloque. Este será el que se genera según la vista.

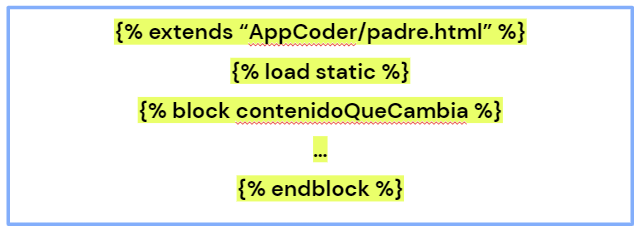
Notamos que aparece:

**{% block contenidoQueCambia %}**

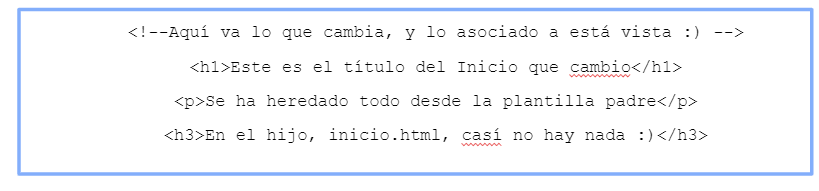
**…..**

**{% endblock %}**

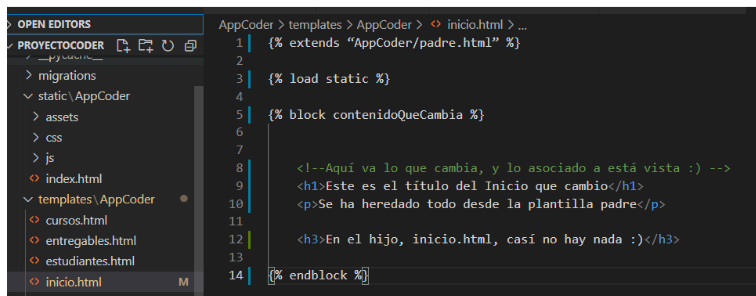
1. Vamos al template que no es base, es decir, a uno de sus hijos y borramos todo. Lo único que debemos dejar es:



1. Por último, nos resta colocar en los “…”, justamente el contenido que será dinámico:



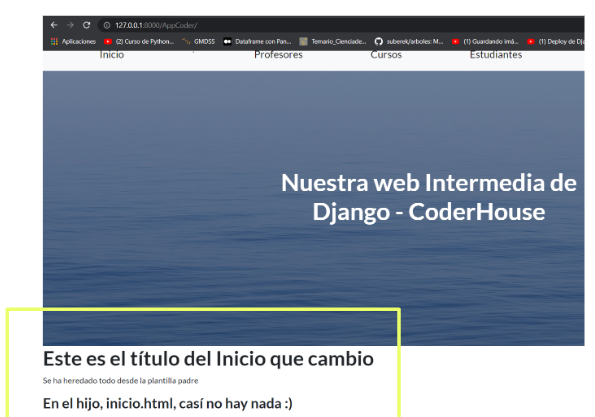
Así nos quedaría:



Y vemos que el inicio.html tiene el aspecto del padre, pero que agregó información pertinente a dicha plantilla:

Importante: En el caso de que no lo visualices, recuerda ejecutar nuevamente el siguiente comando:

**python manage.py runserver**



Navegar entre nuestros templates

¡Excelente! nuestro inicio se ve muy bien, tiene muy pocas líneas y agrega información a nuestro template padre, pero…

¿Cómo haríamos para ir desde este hijo a otros? ¿Cómo hacemos para que la barra principal funcione?

En el archivo padre.html reemplazamos el contenido de la etiqueta nav, como se muestra a continuación:

**<nav class="navbar navbar-light bg-light static-top">**

**<div class="container">**

**<a class="navbar-brand" href="{% url 'Inicio' %}">Inicio</a>**

**<a class="navbar-brand" href="{% url 'Profesores' %}">Profesores</a>**

**<a class="navbar-brand" href="{% url 'Cursos' %}">Cursos</a>**

**<a class="navbar-brand" href="{% url 'Estudiantes' %}">Estudiantes</a>**

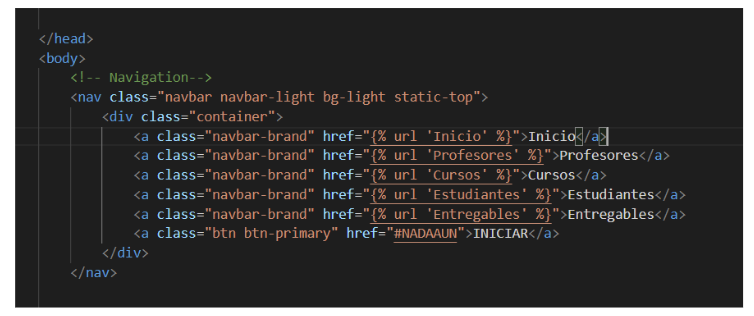
**<a class="navbar-brand" href="{% url 'Entregables' %}">Entregables</a>**

**<a class="btn btn-primary" href="#NADAAUN">INICIAR</a>**

**</div>**

**</nav>**

Así debería quedarnos:



Luego hay que modificar todos los html y quitar todo el código que no nos sirva. A continuación, mostraremos como realizar este paso para el archivo: cursos.html, pero recuerda que deberás repetir esta operación con el resto de plantillas como ser:

* entregables.html
* estudiantes.html
* profesores.html.

Script:

**{% extends "AppCoder/padre.html" %}**

**{% load static %}**

**{% block contenidoQueCambia %}**

**<!--Aquí va lo que cambia, y lo asociado a está vista :) -->**

**<h1>Cursos</h1>**

**<p>Acá va la info de los cursos</p>**

**{% endblock %}**

cursos.html:



Por ultimo! Modificamos el archivo: urls.py

**from django.urls import path**

**from AppCoder import views**

**urlpatterns = [**

**path('', views.inicio, name="Inicio"), #esta era nuestra primer view**

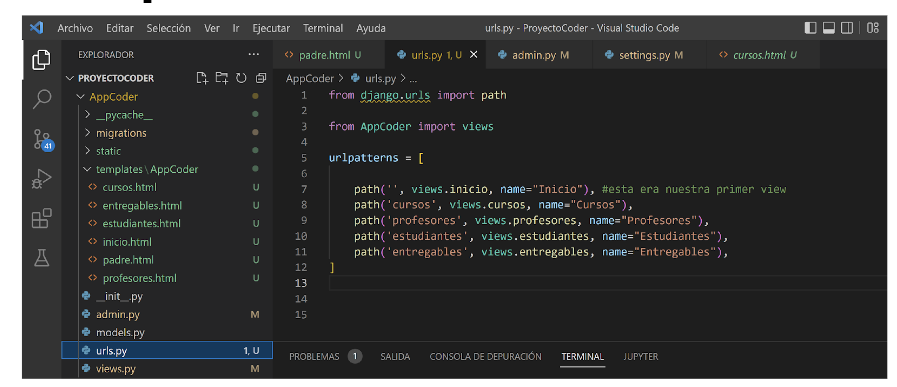
**path('cursos', views.cursos, name="Cursos"),**

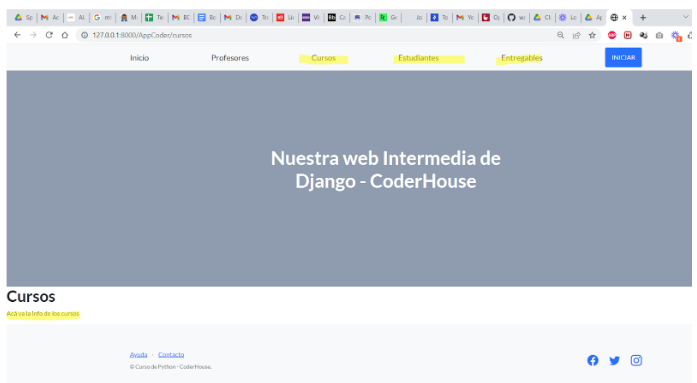
**path('profesores', views.profesores, name="Profesores"),**

**path('estudiantes', views.estudiantes, name="Estudiantes"),**

**path('entregables', views.entregables, name="Entregables"),**

**]**





¡Listo! De esta forma ya tenemos el menú que se hereda por todos los templates y que nos permite movernos de uno a otro.

Es importante ser consistentes con los nombres de los templates, de las urls y de los archivos .html

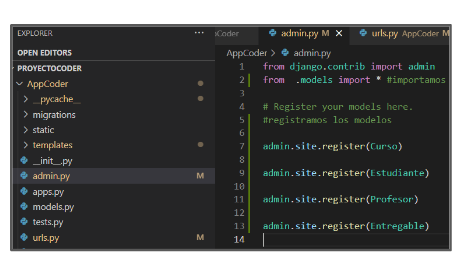
Panel de administración

Por ahora sabemos como guardar campos en una base de datos desde la consola, y hemos visto cómo hacerlo (muy a mano) desde un view. Pero por lo general, la información tiene que ser administrada por la misma web. Esto suele ser una tarea complicada, pero por fortuna, Django nos simplifica el manejo de la información por medio de un panel de administración ya preparado.

Dar de alta a nuestro panel

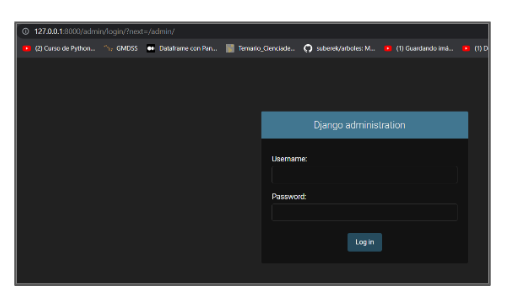
¿Cómo dar de alta a nuestro panel de administración?

1. Primero importar los modelos que queremos administrar y registrarlos en el archivo admin.py de la app. Importante: Si el archivo no existe, deberemos crearlo.



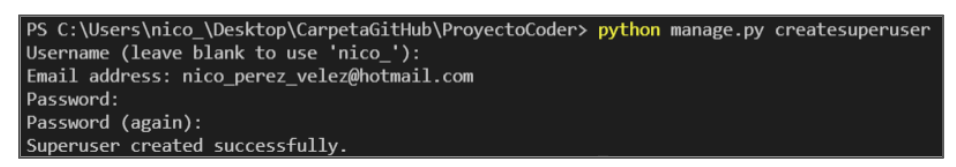
1. Notar que si entramos a web/admin/

Nos pide un usuario y contraseña.

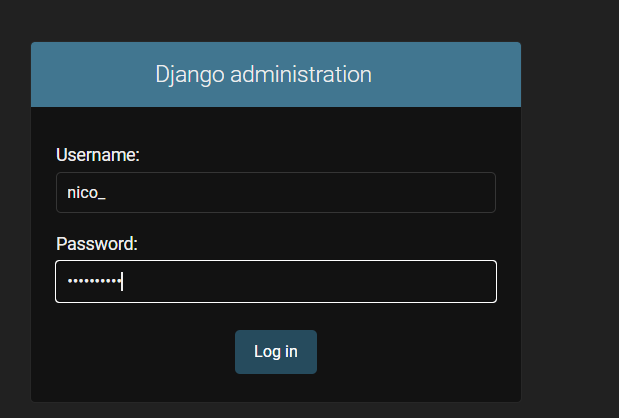
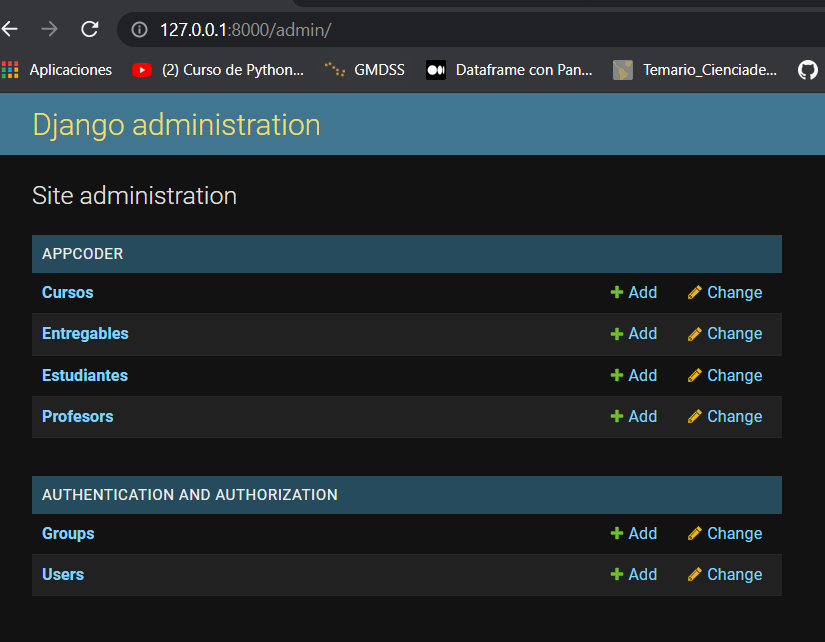


1. Entonces, hay que crear el usuario para administrar, lo que se llama un super usuario:

**python manage.py createsuperuser**



1. Ingresamos a /admin/ y probamos el super usuario:



1. Si el Site, no registra la app, deberemos copiar y reemplazar el codigo propuesto en el siguiente archivo, en settings.py.

Codigo de registro.txt

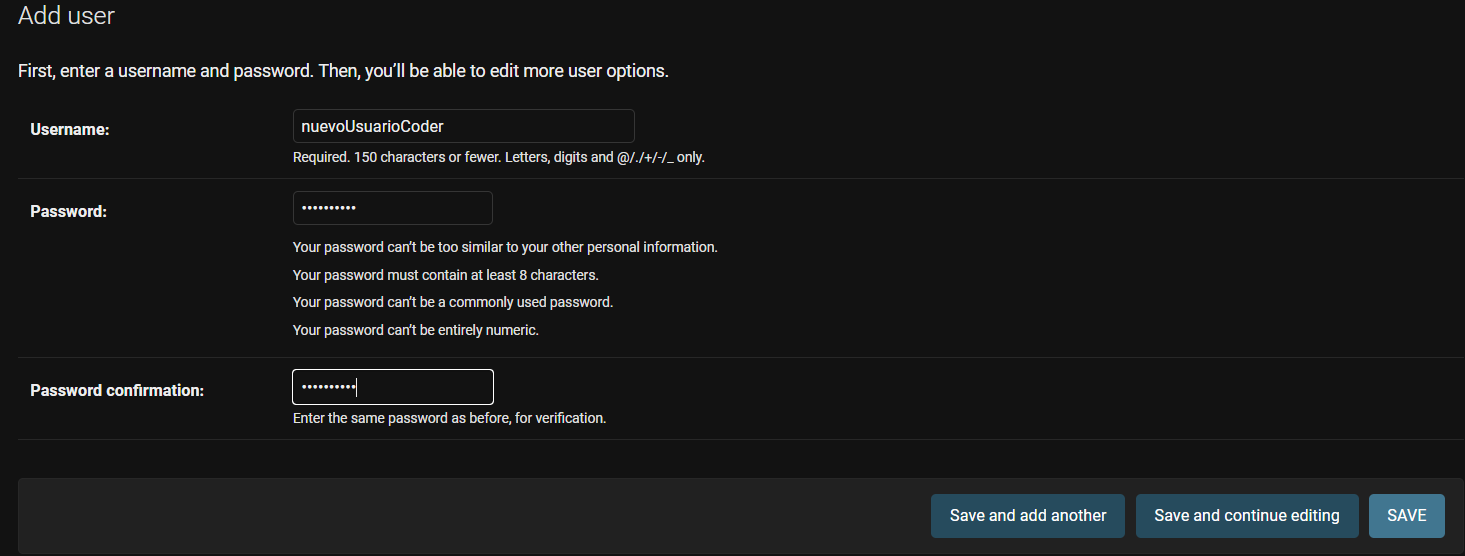
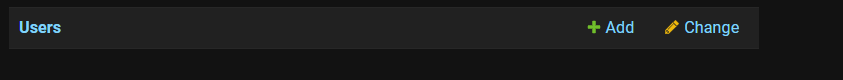
Recuerden guardar todos los cambios y lanzar nuevamente el servidor!.

python manage.py runserver

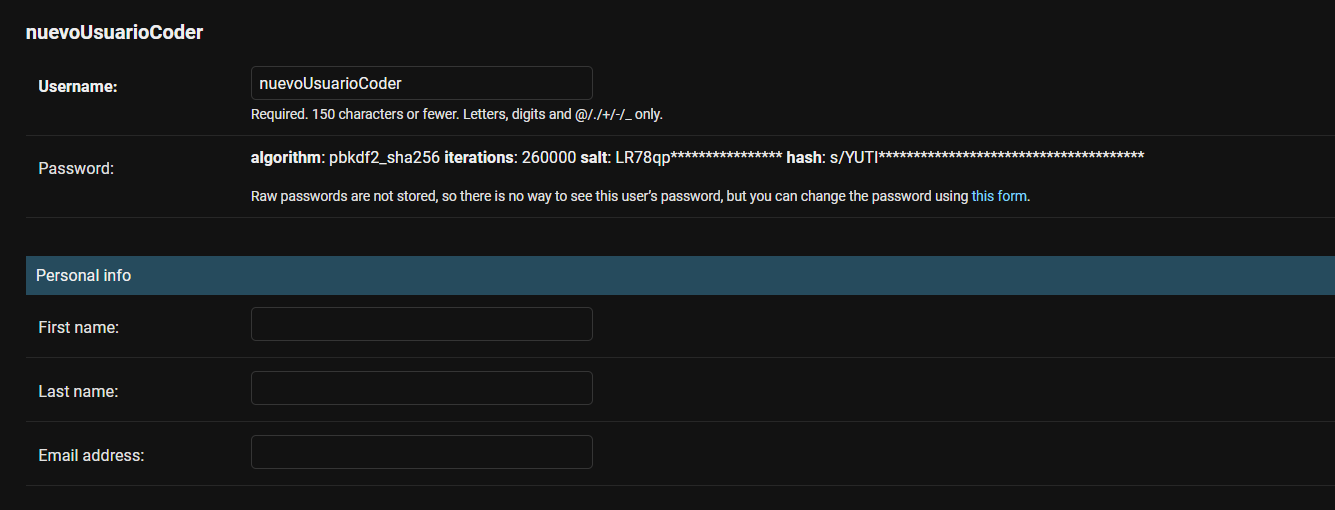
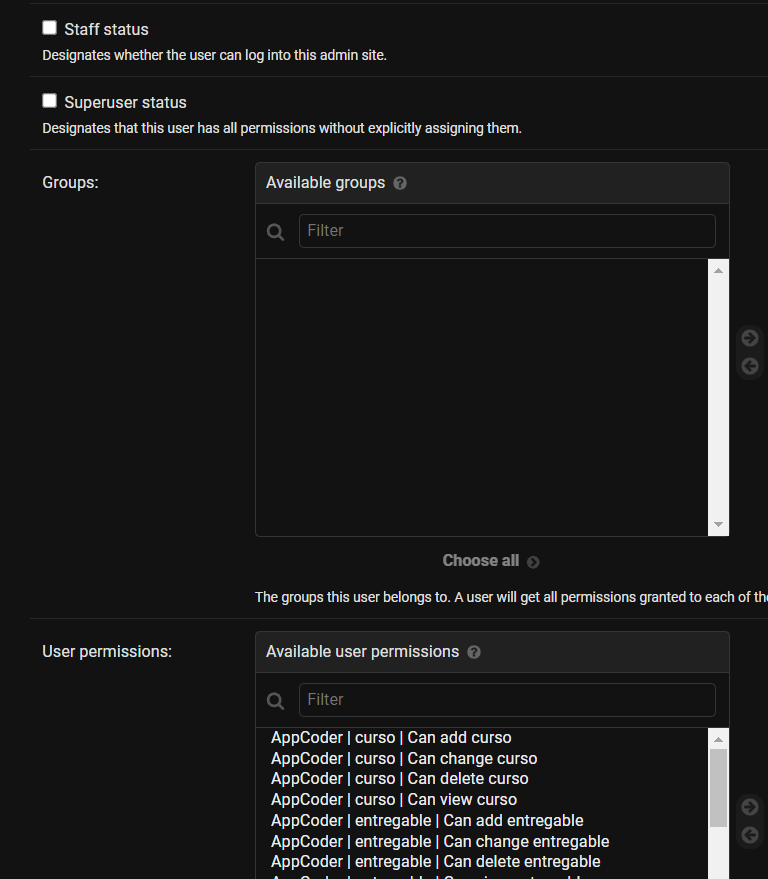
¡Increíble, ya tenemos acceso a nuestro panel de administración y a todas las clases del modelo de nuestra web!

1. Obviamente, seguramente no solo habrá un super usuario, sino también otros usuarios con menos permisos, podemos crear algunos muy simplemente.

Creación de usuarios



Pueden llenar otros campos no obligatorios:



Y establecer permisos que tendrá este usuario (por ahora no nos interesa):

Le damos SAVE y vemos que ya tenemos dos usuarios para usar el panel de administración, solo que el último no tiene permisos para nada:

