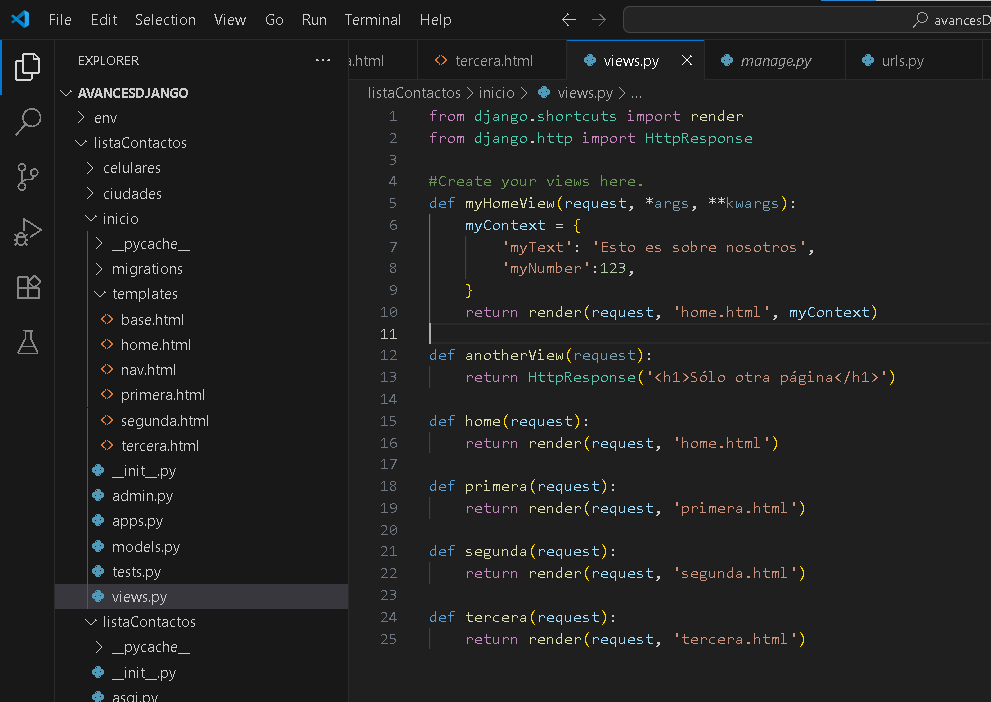
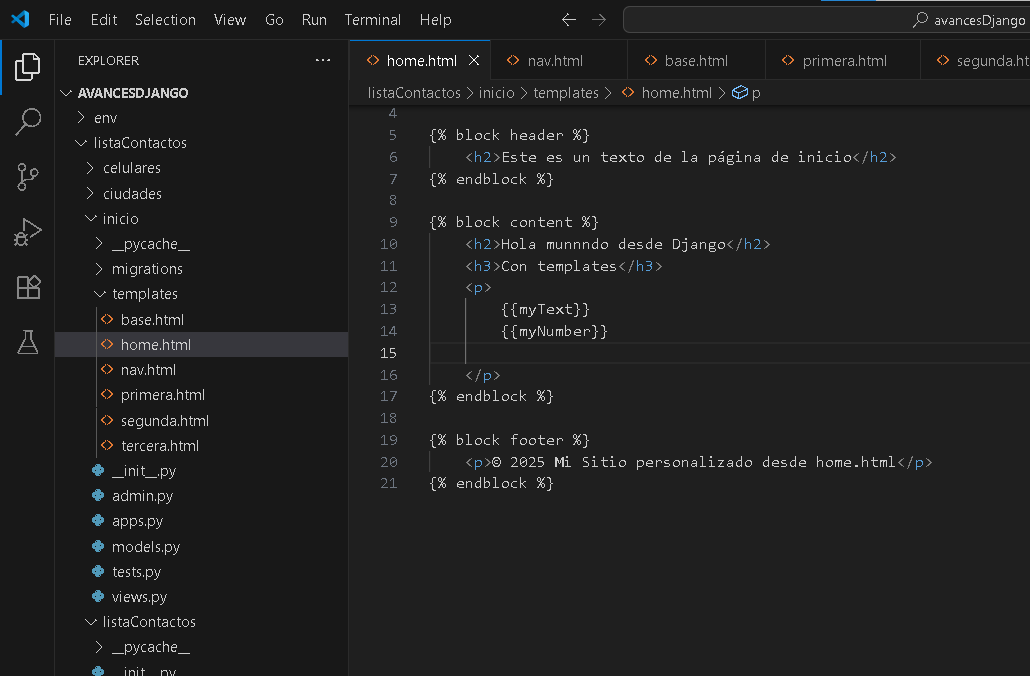
AVANCES GIT – DJANGO PARTE 3

DIAPOSITIVA 3: Las variables del “contexto” en un template

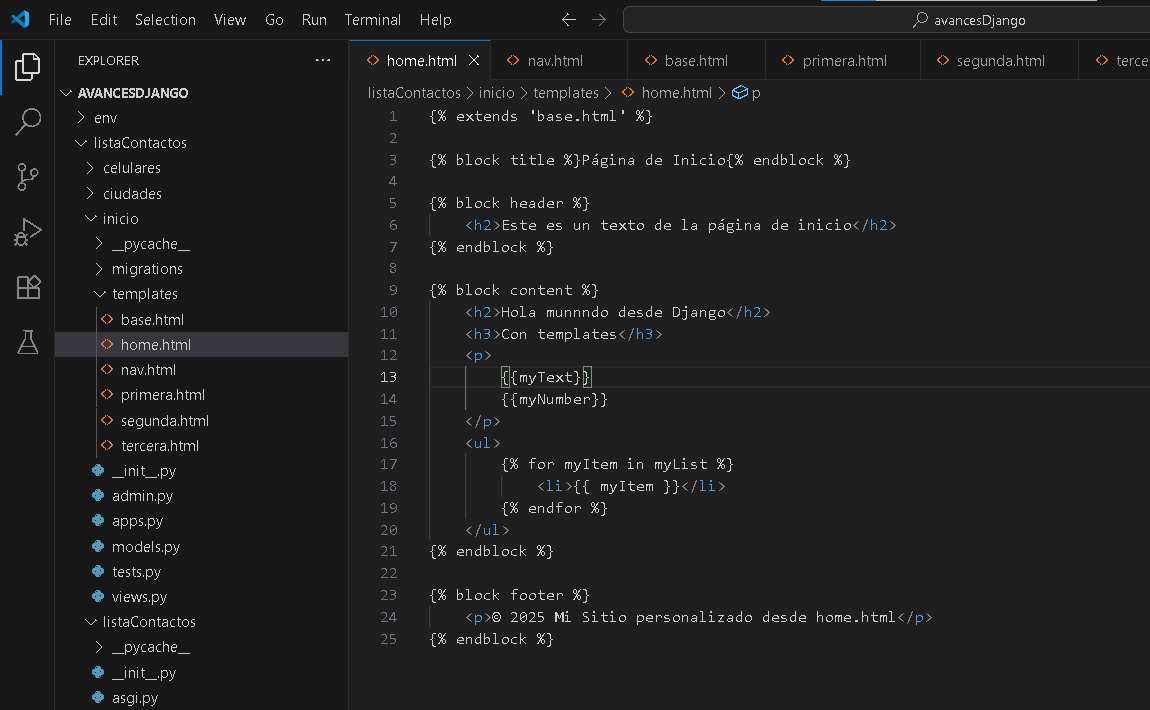


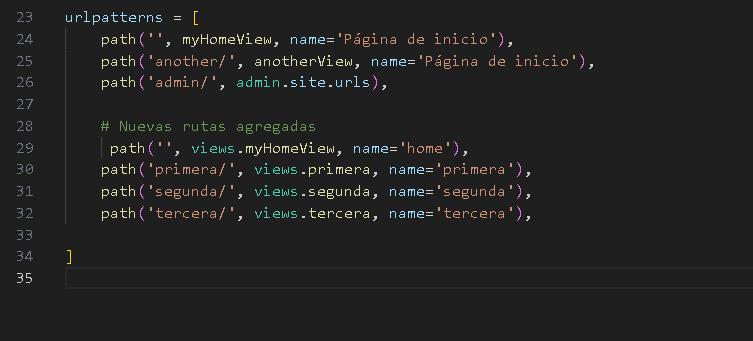


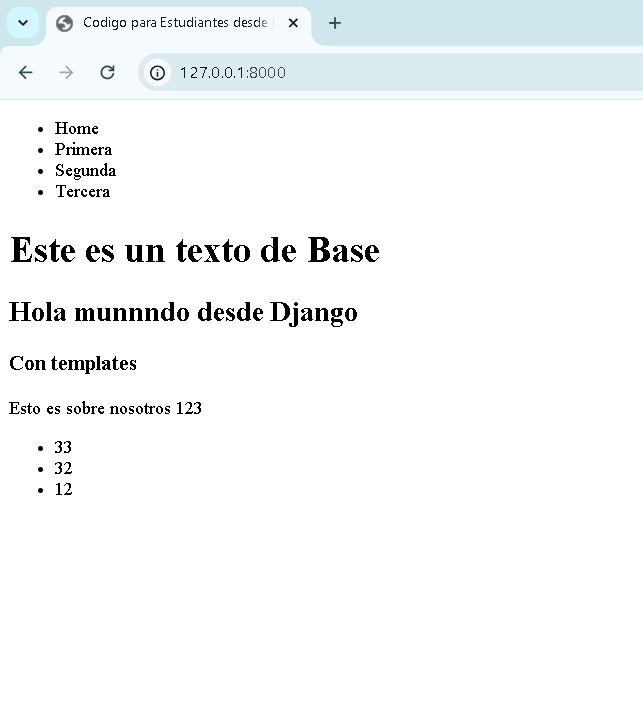
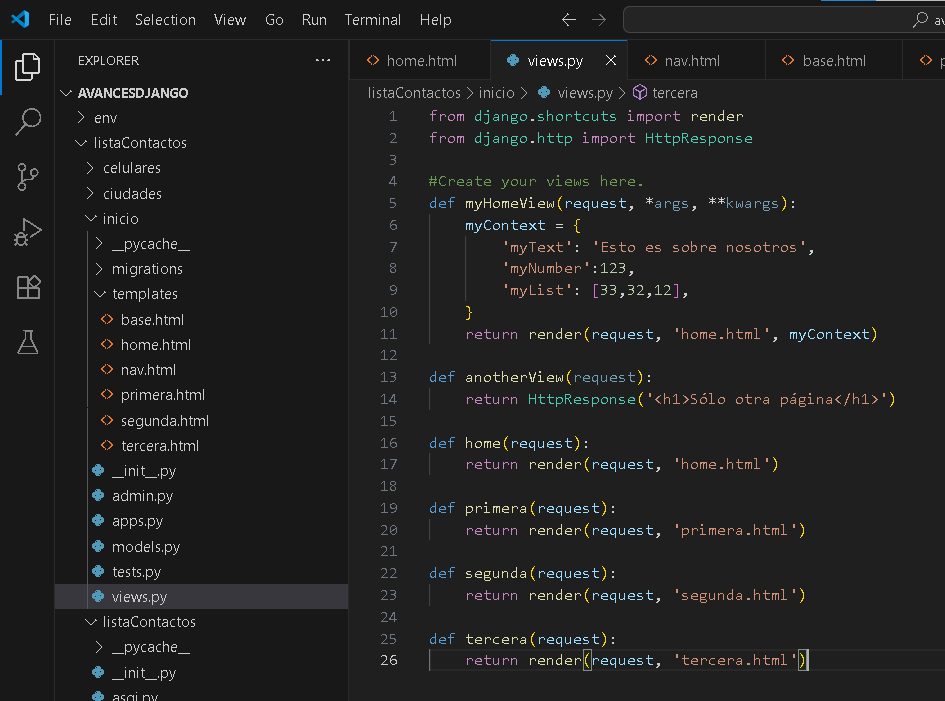


Usando variables desde views.py para enviar datos al template y mostrarlo en plantillas.

DIAPOSITIVA 4: El tag FOR

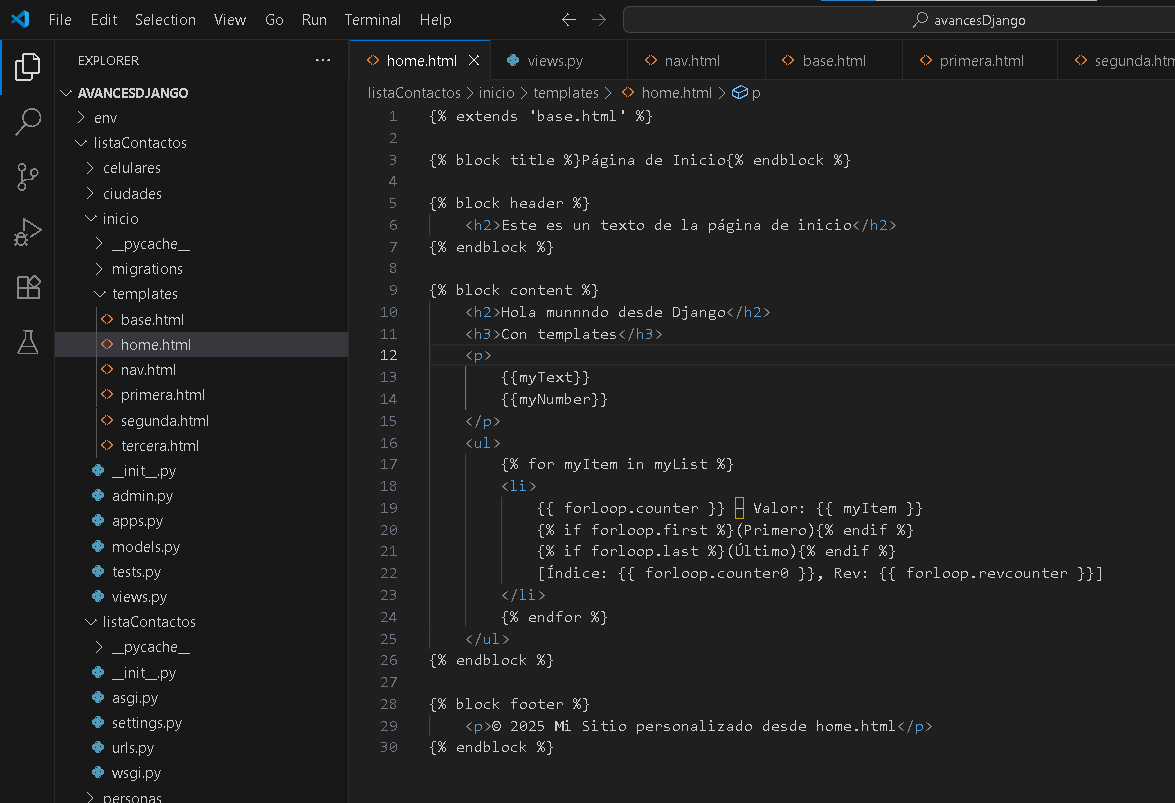


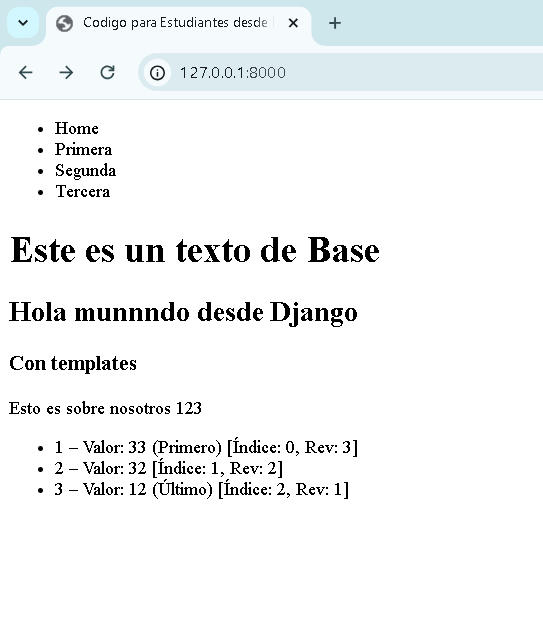




Añadiendo variables-listas a views.py para que sea recibido en el template con variación tag For.

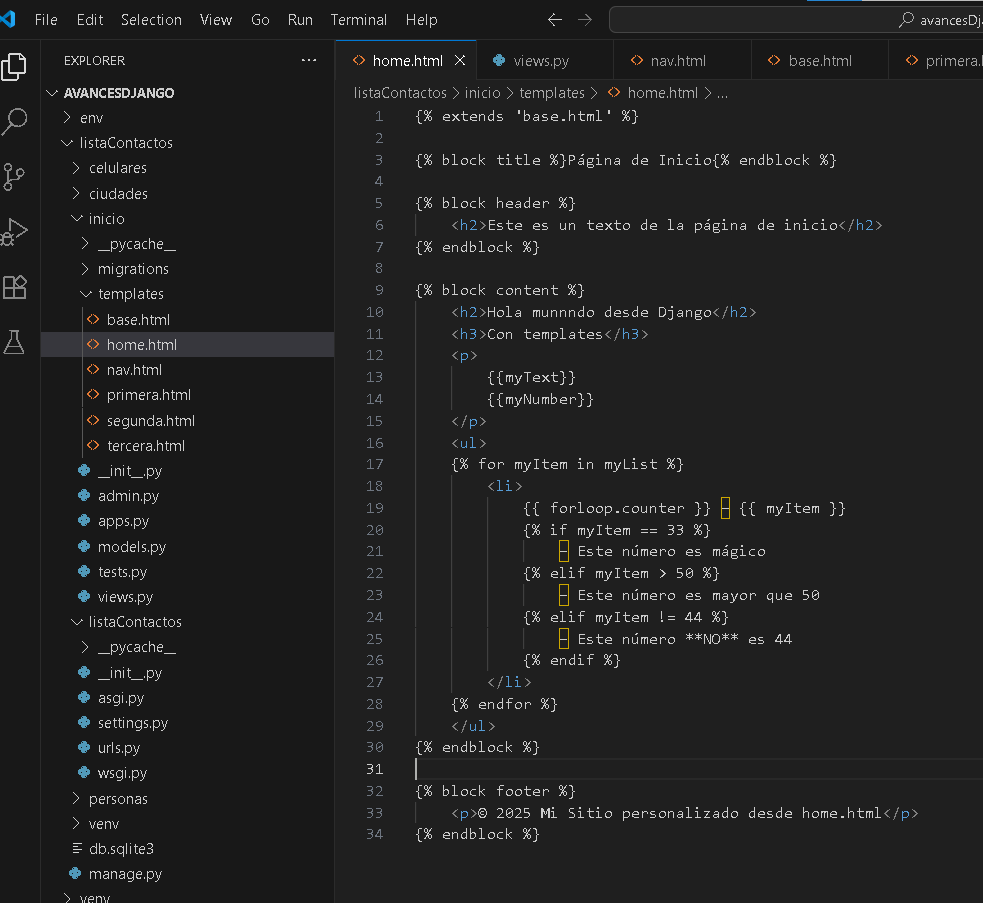
DIAPOSITIVA 5: El contador de ciclos

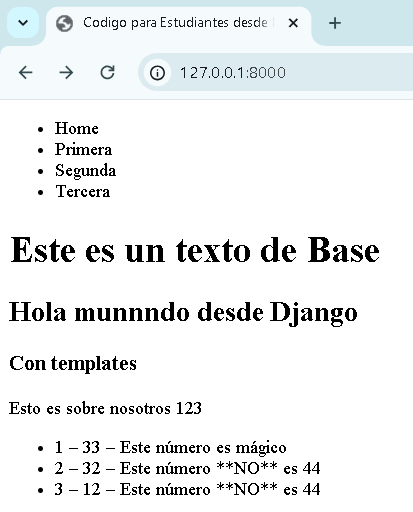
Revisar el objeto forloop en la documentación oficial y hacer experimentos



Añadiendo forloop al template home.html y algunas variables predefinidas como forloop.first y forloop.last, haciendo pruebas en el template

DIAPOSITIVA 6: El tag IF

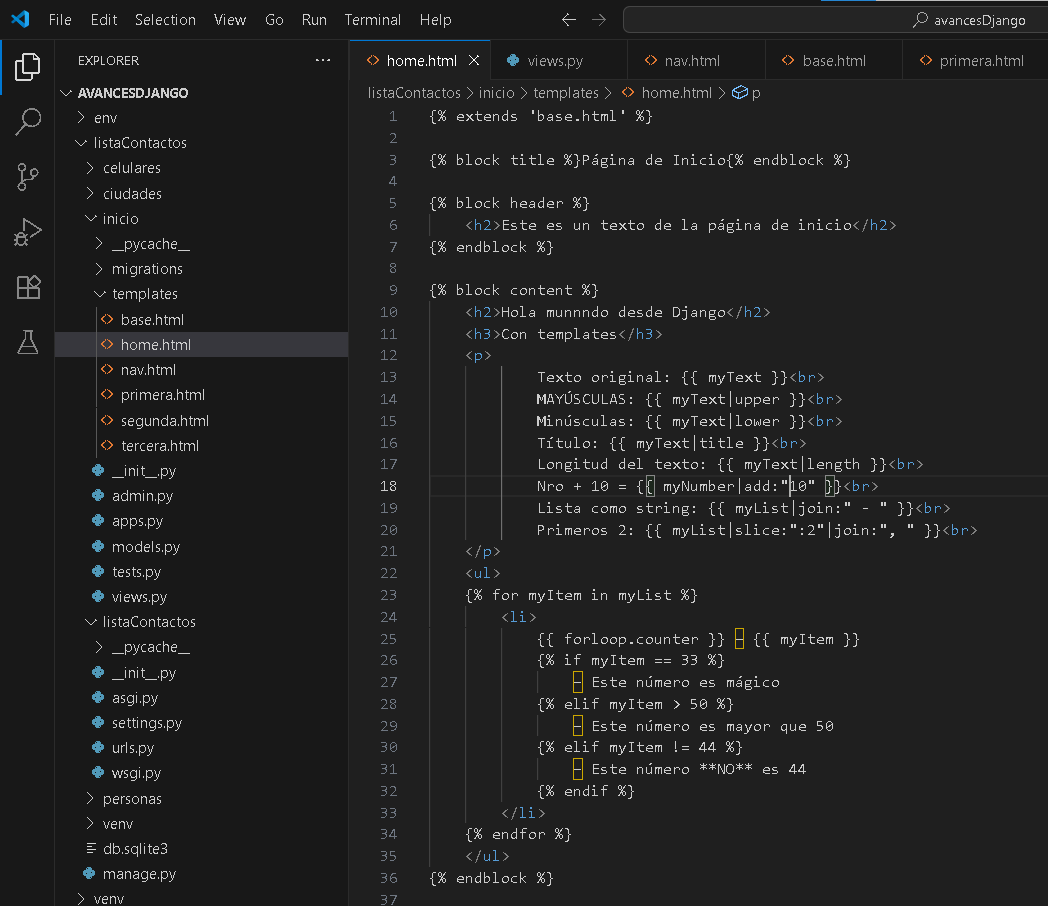
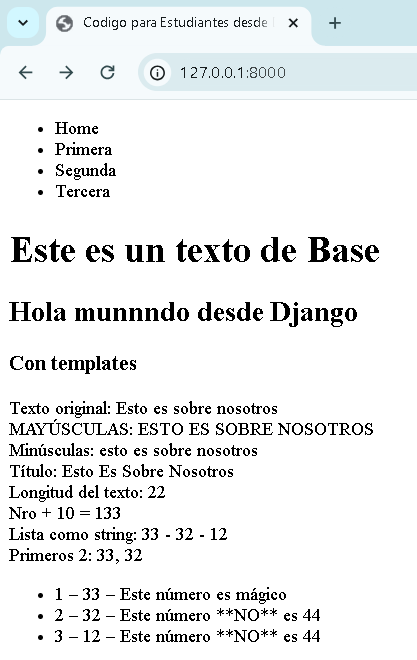
Revisar la documentación oficial sobre if, hacer experimentos con otros operadores de comparación



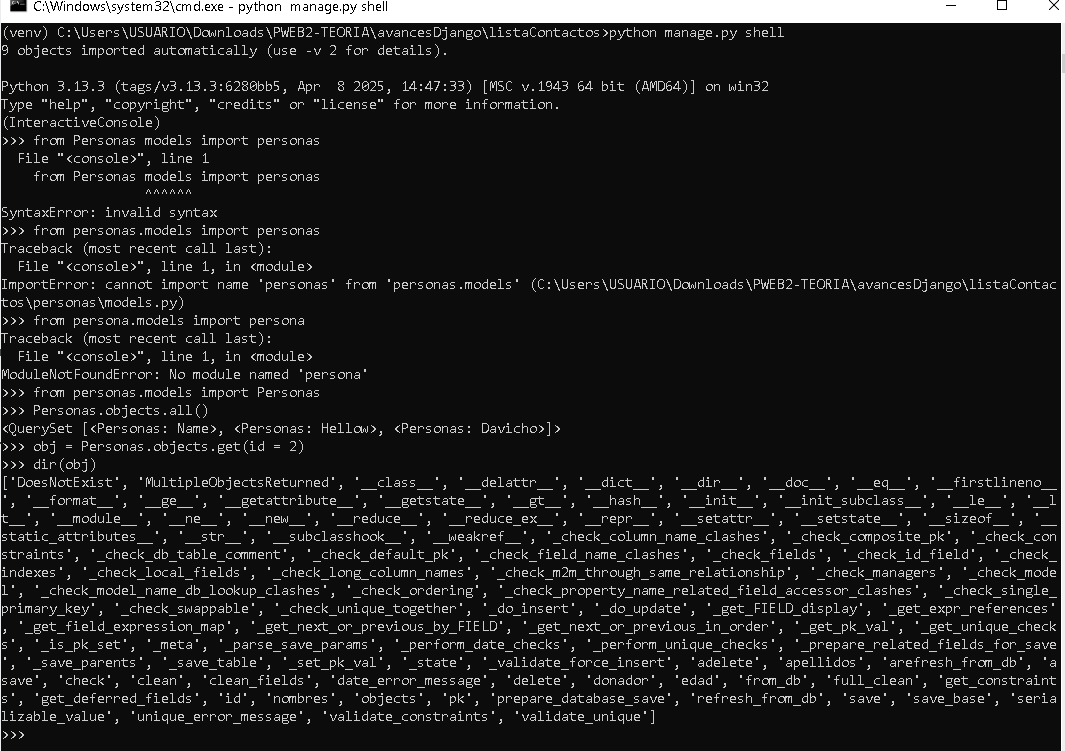
Añadiendo el tag if usando el condicional con los operadores igualdad, mayor que, y distinto de.

DIAPOSITIVA 7: Filtros

Probar la aplicación de varios otros filtros a variables y bloques, revisando la documentación oficial

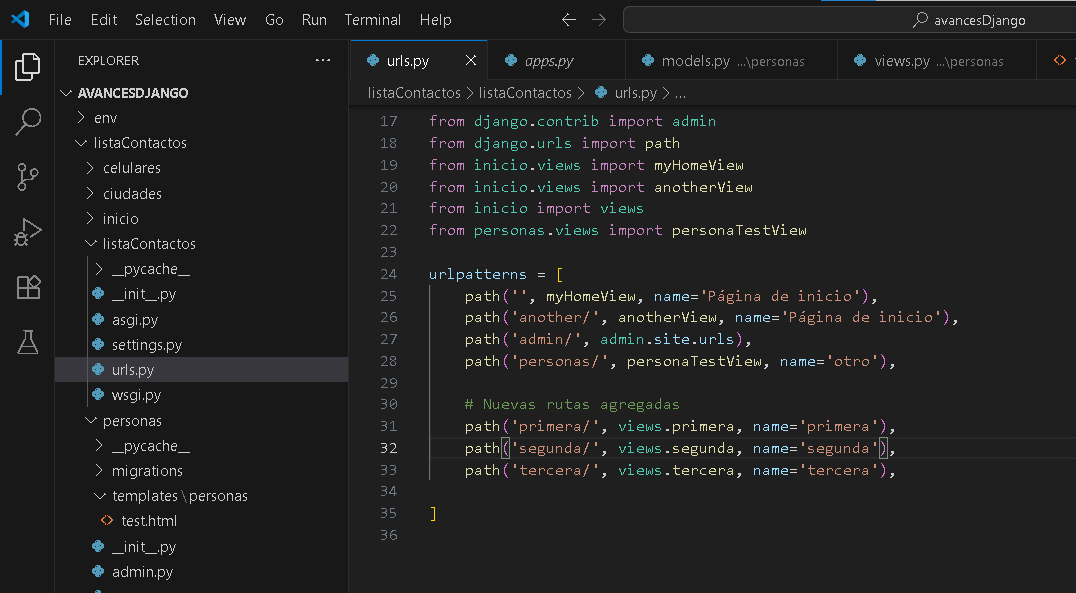


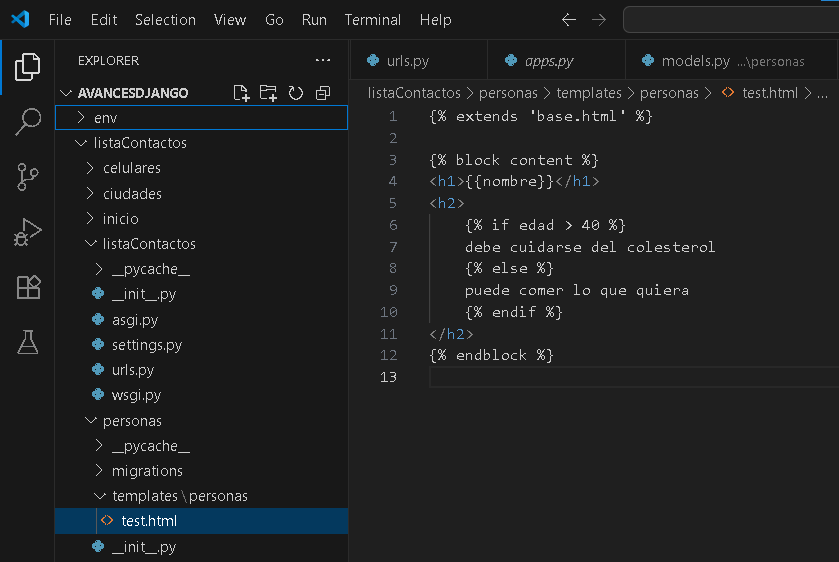
Añadiendo diferentes filtros a varias varibles como, texto en mayúscula, texto en minúscula, título, longitud de las posiciones totales del texto, etc dentro de bloque.

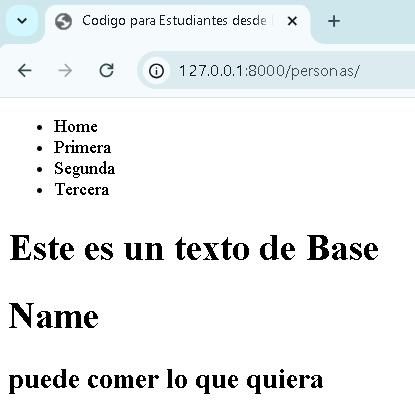
DIAPOSITIVA 9: Examinando los objetos del Modelo

Desde el Shell de Python en cmd, analizamos los diversos objetos de los modelos que hay en Django

DIAPOSITIVA 10: Renderizando una Persona de la app personas

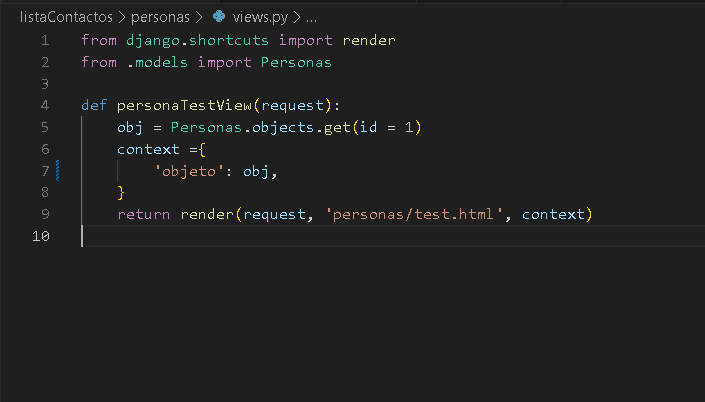


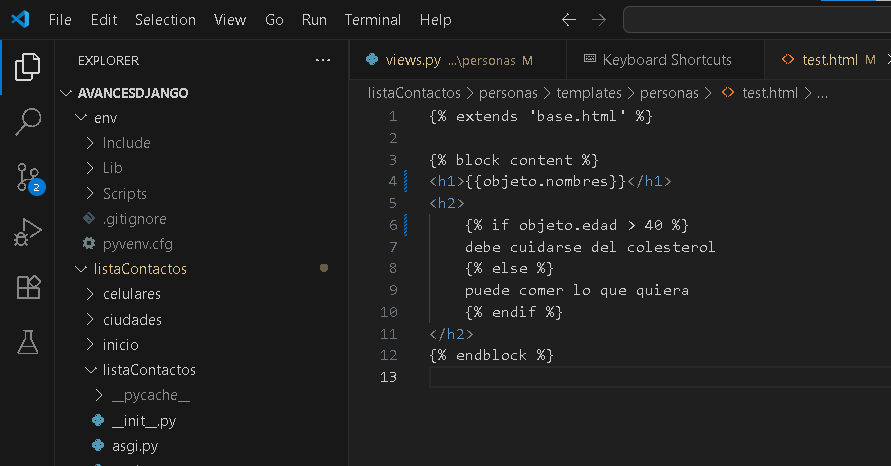




Añadiendo rutas, template en aplicación personas y renderizandola.

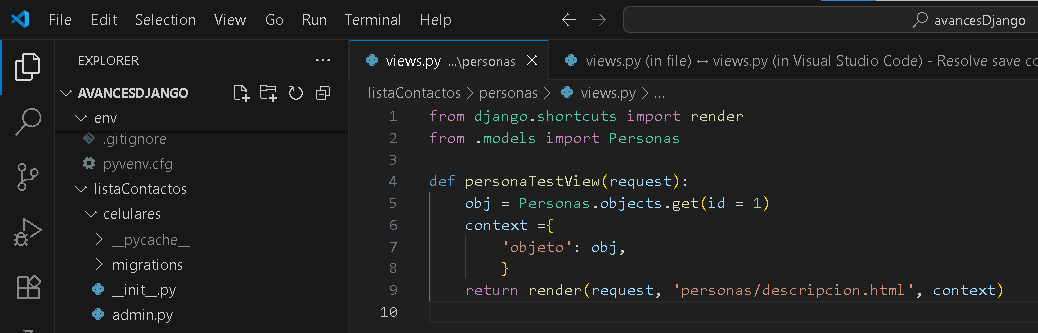
DIAPOSITIVA 11: Enviando el objeto entero en el contexto

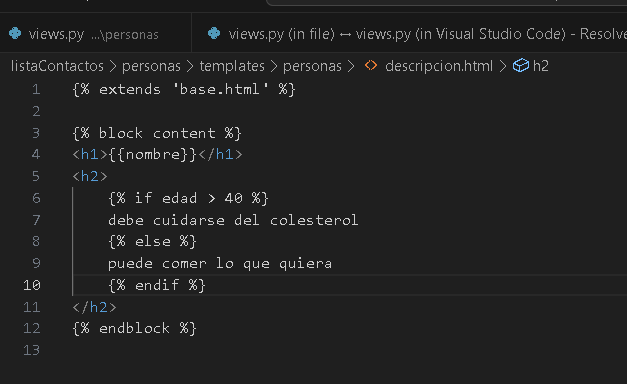


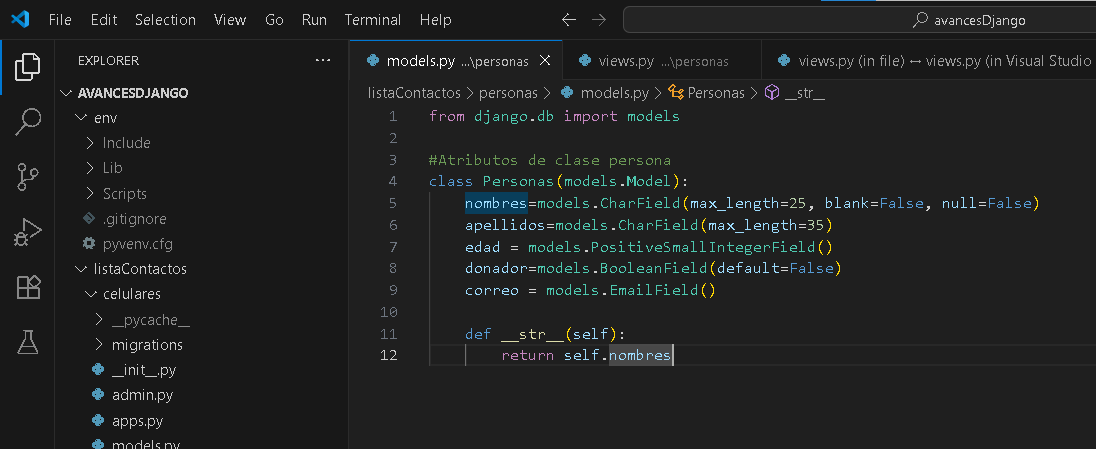


Enviando objeto entero al contexto

DIAPOSITIVA 13: Creando la plantilla dentro del app Persona

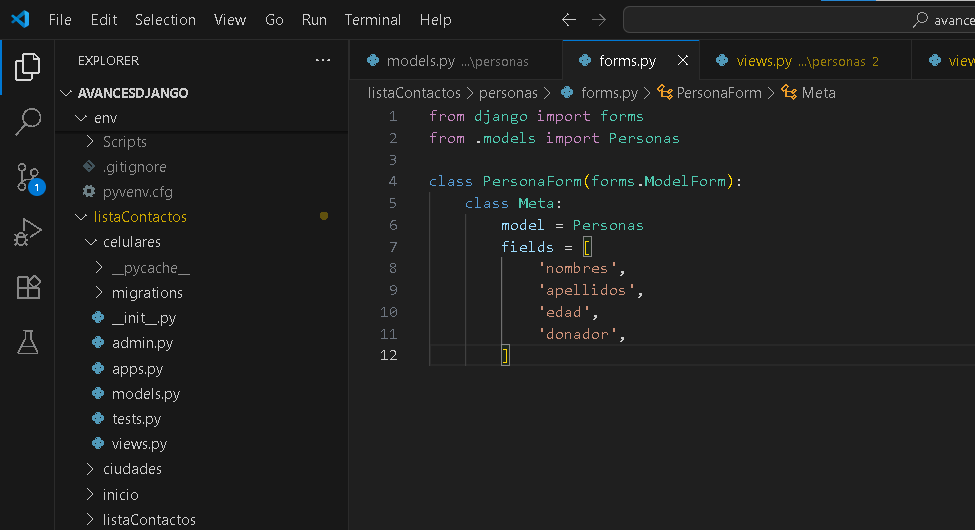


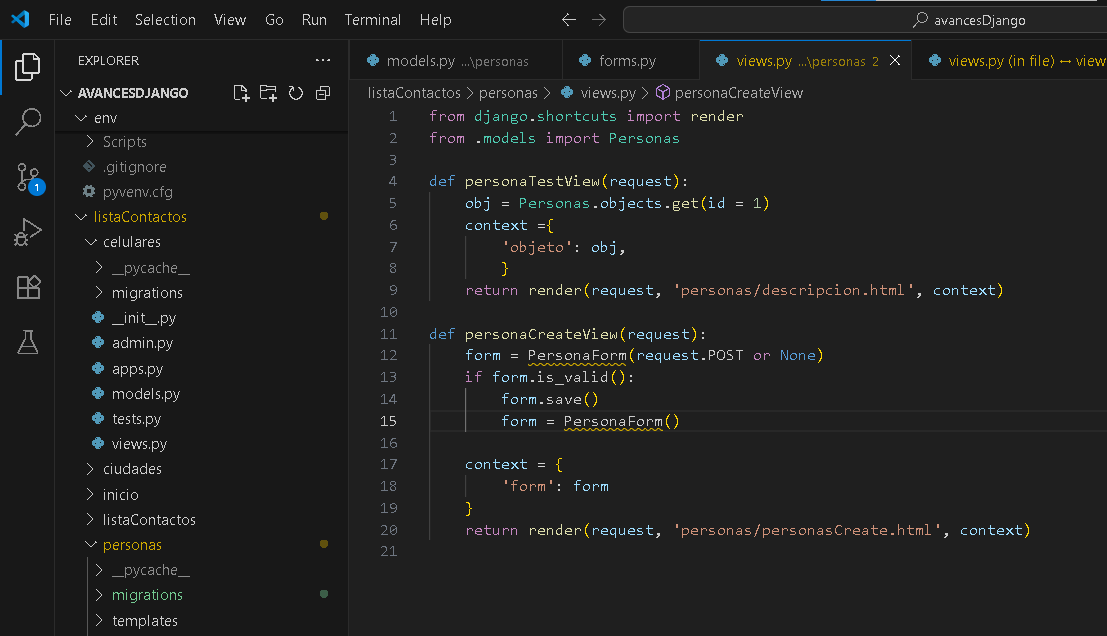




Plantilla creada para la aplicación Personas

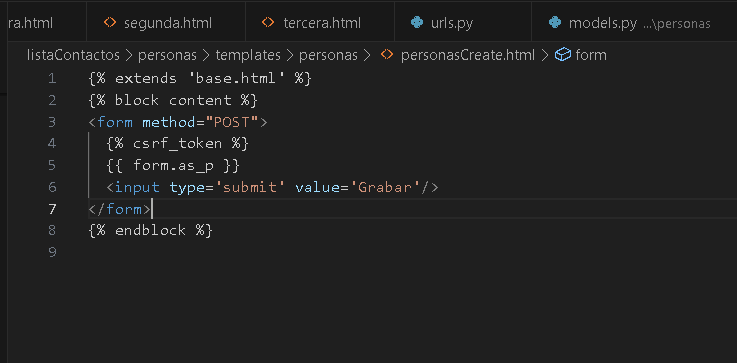
DIAPOSITIVA 15: Formulario sencillo para la app Personas

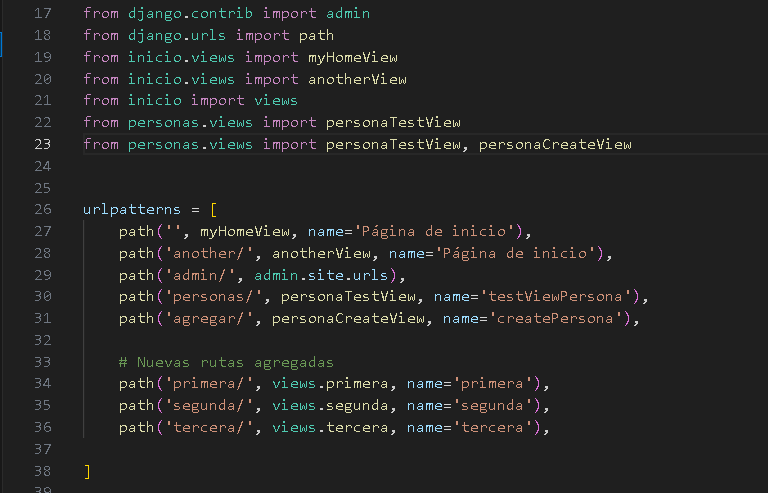




Añadiendo formulario a la aplicación personas

DIAPOSITIVA 16: Formulario sencillo para la app Personas (continuación)



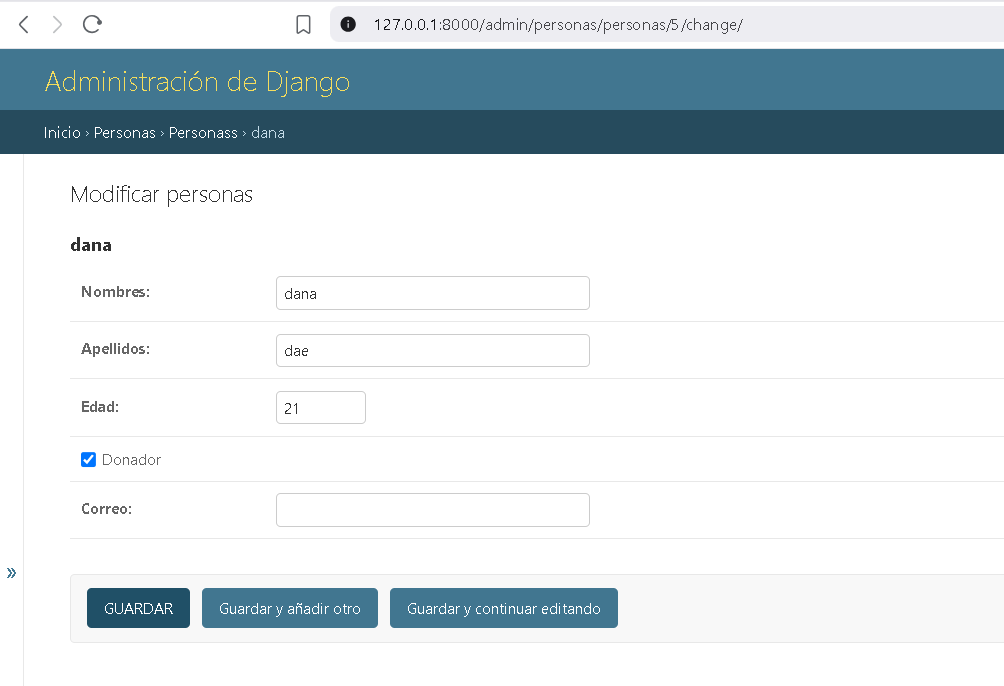


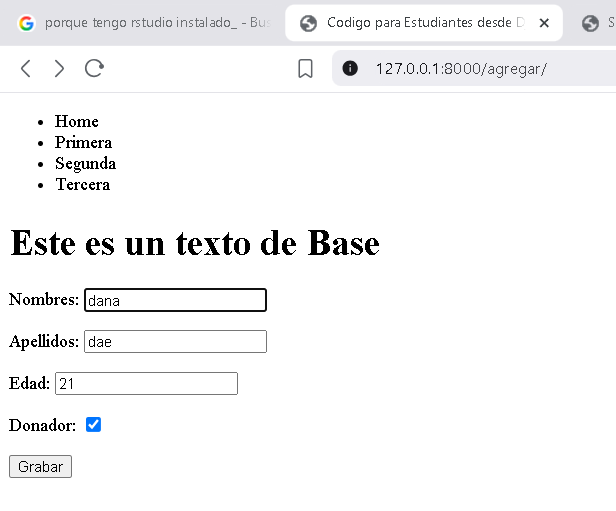
Añadiendo ruta y formulario.

DIAPOSITIVA 17: El formulario creado funcionando

Pruebe qué ocurre si no incluímos el campo “donador” en el formulario

¿De qué se trata este error?

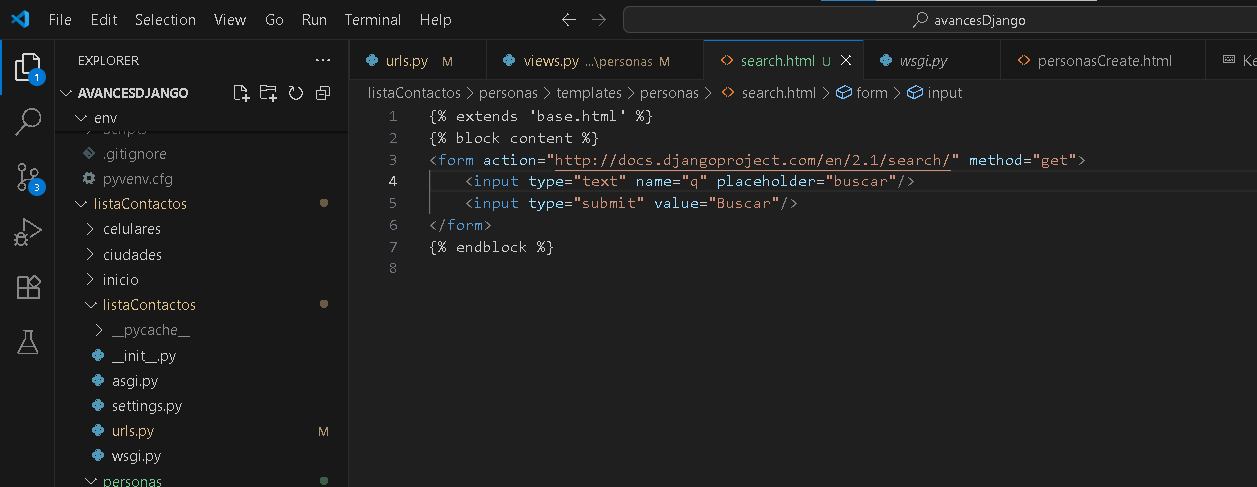
¿Cómo se puede solucionar?

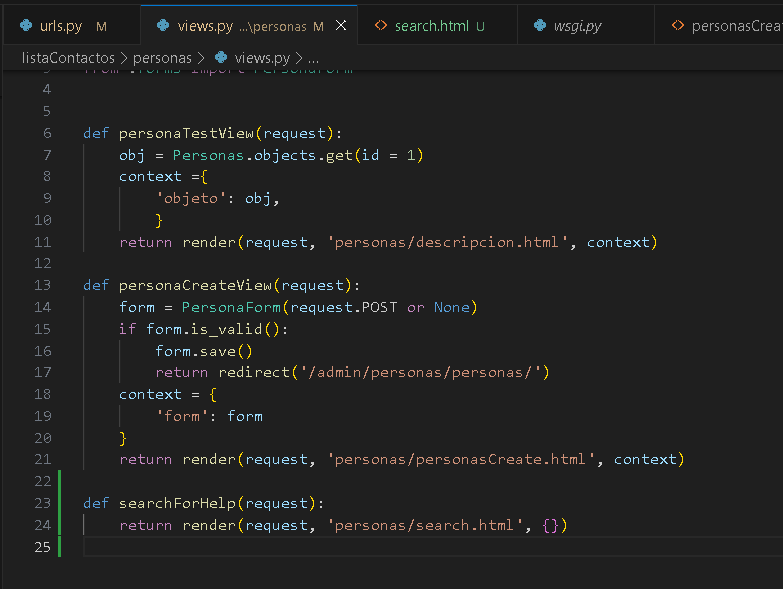


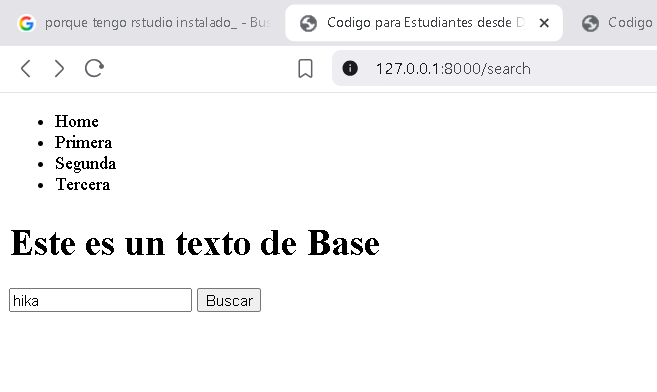
Hacemos correr el formulario ubicado en la url …/agregar y luego redireccionamos al administrador de django.

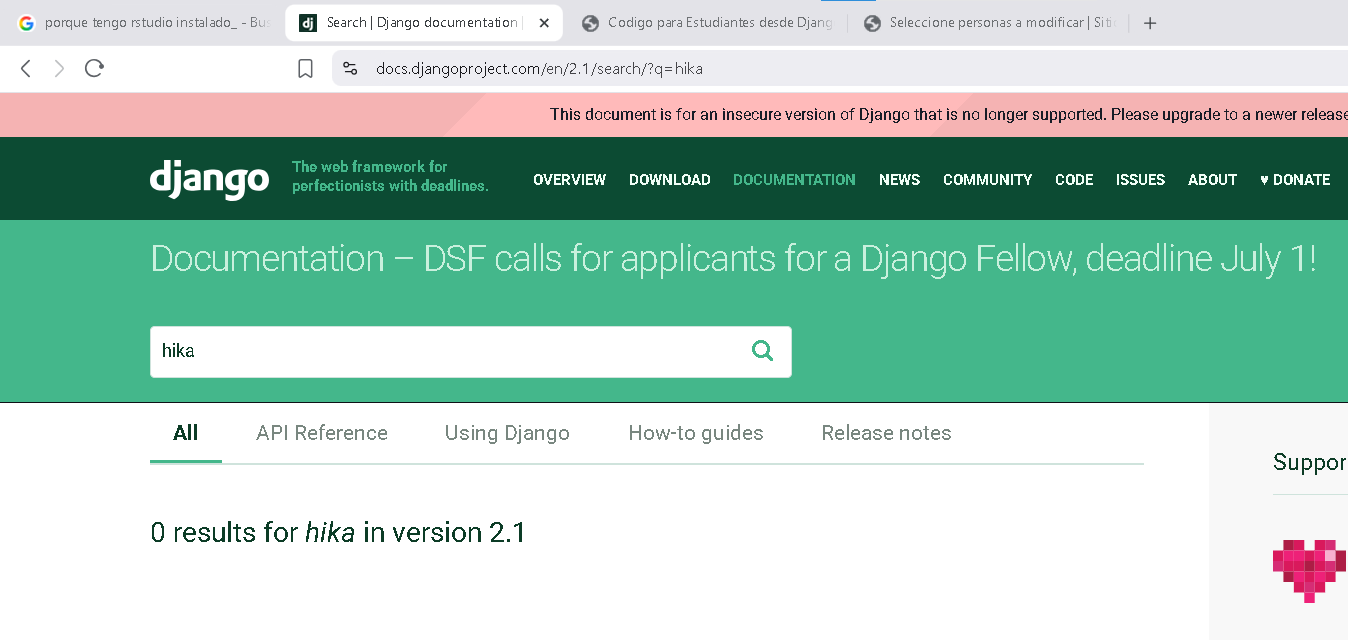
IntegrityError: NOT NULL constraint failed: personas\_personas.donador; Significa que el campo donador es obligatorio en la base de datos (es decir, null=False y no tiene un valor por defecto), pero como no está en el formulario no se envió ningún valor, y Django intentó guardarlo como NULL — lo cual está prohibido. En models :  
donador = models.BooleanField(null=True, blank=True)

DIAPOSITIVA 19: Templates con forms planos: GET



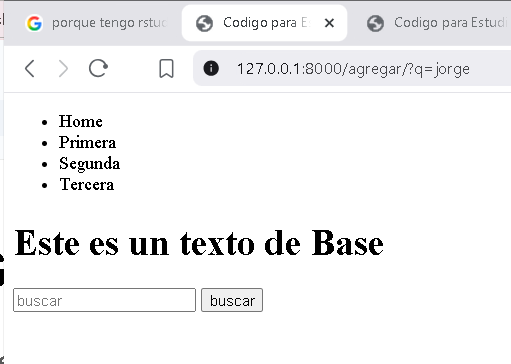


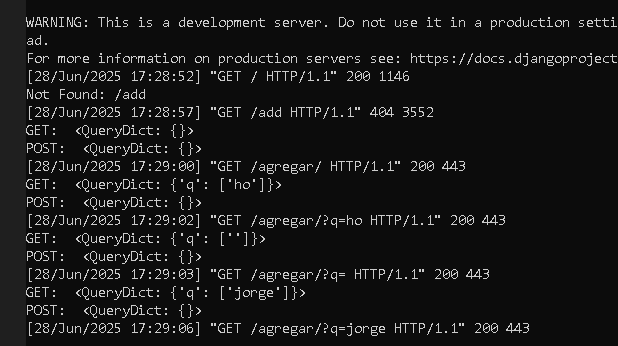




Usamos los método GET y POST para que nuestro proyecto en Django pueda mediante el llenado de formularios redireccionarnos a una URL externa y así hacerlo más interactivo.

DIAPOSITIVA 20: Datos de GET y POST

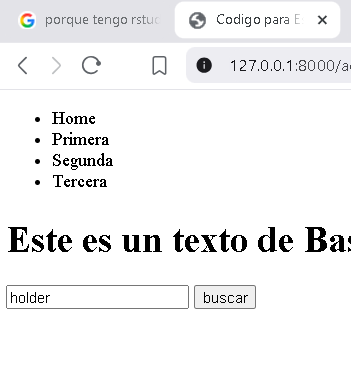


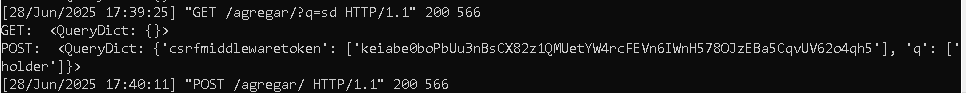


Creamos mediante el llenado de la cajita de texto y el botón buscar, envío y recepción de datos con GET Y POST

DIAPOSITIVA 21: Templates con forms planos: POST

¿En qué momento se hace llamada GET y en qué momento se hace la llamada POST?

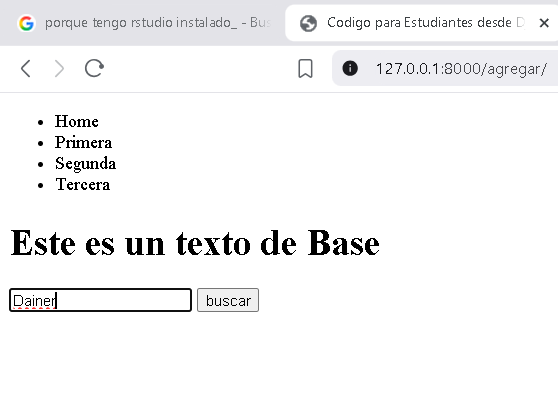




En el momento de tipear dentro de la cajita de texto se hace una llamada get y post cuando presionamos el botón buscar.

Haciendo el análisis de POST y GET y la validación de csrf\_token y e

DIAPOSITIVA 22: Acciones distintas para GET y POST





Envío de dato “Dainer” mediante POST en el formulario …/agregar usando la cajita de texto y la función del botón buscar. Análisis dentro del Shell de cmd, luego de las peticiones dentro del proyecto Django.

