**Trabajo Final Practica profesionalizante 1**

Alumna: Alem Federica

Github:

https://github.com/Federicaalem/Practica1\_final

**Informe**

* Instalación Django y Entornos virtuales

Instalacion de herramienta de entorno Virtual:

* Pip3 install virtualvenv

Crear una carpeta donde estará el Entorno virtual:

* Mkdir EntornoVirtual
* cd EntornoVirtual
* Py –m venv django (creamos el entorno virtual dentro de la carpeta)

Activar entorno virtual:

* Cd django
* . \Scripts\activate

Instalacion de django dentro del entorno virtual:

* (django)>pip install Django == 3.2.7

Abrimos consola de Python para verificar:

>Python

>>import django

>>django.get\_version()

“3.2.7”

>>quit()

Comenzamos nuestro proyecto

* django-admin startproject establecimiento

Corremos el servidor para verificar que todo este yendo bien

* python manage.py runserver

**instalación Base de datos postgres**

Ahora si instalamos postgres para su conexión con la base de datos

* pip install psycopg2

**Agregamos carpetas y archivos a nuestro proyecto por única vez**

Creamos una carpeta llamada “applications” donde estara nuestra aplicación.

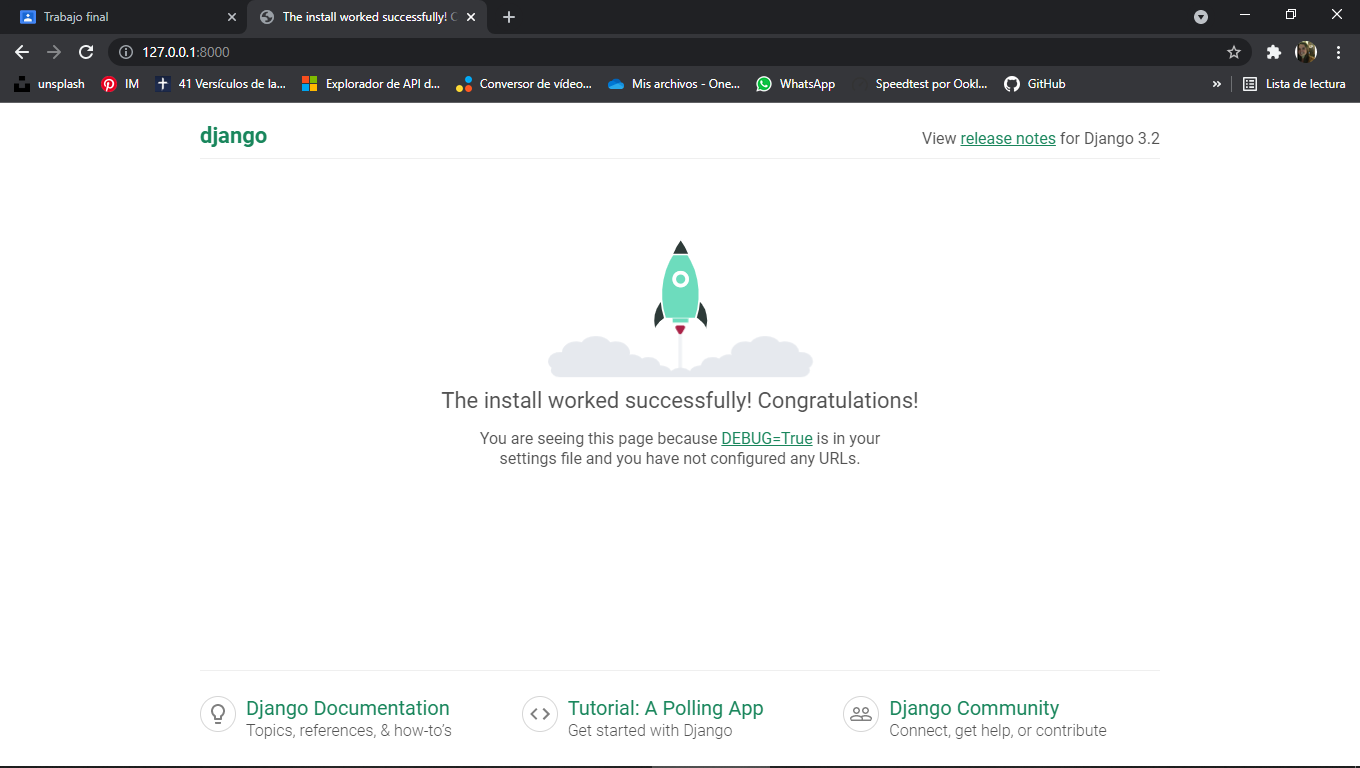
Creamos un archivo “\_\_init\_\_.py”

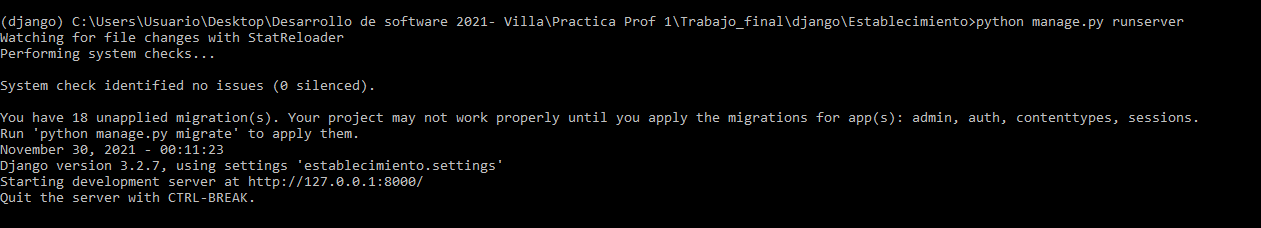
E inicializamos nuestra app:

* django-admin startapp registro

Agregamos la carpeta “templates” donde estarán nuestras vistas html

Corremos el proyecto en el servidor local para probar que todo siga bien.



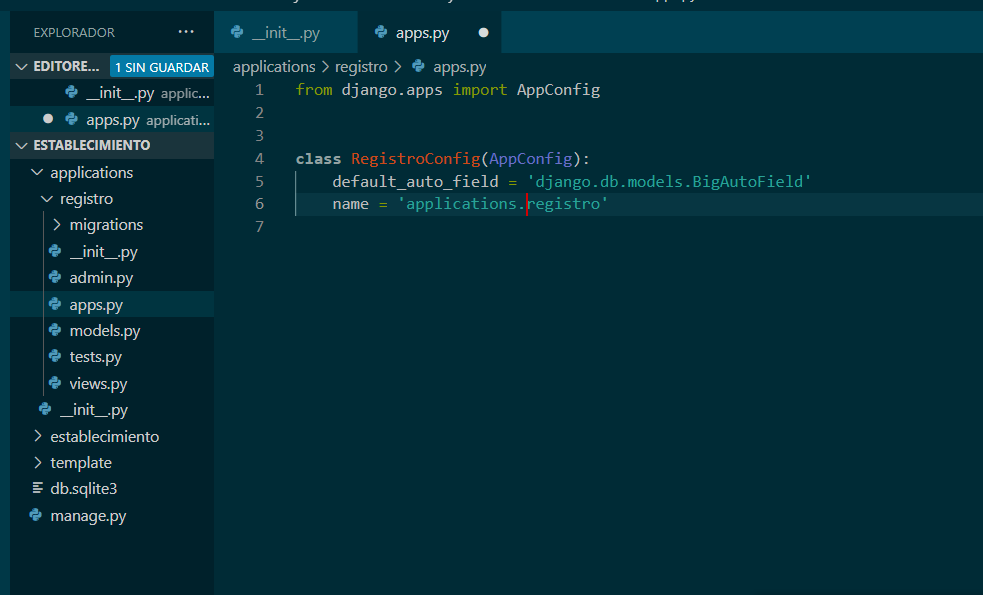


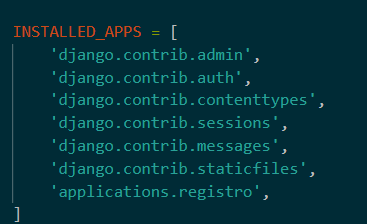
(instalamos django y djaneiro en el editor de código visualStudio)

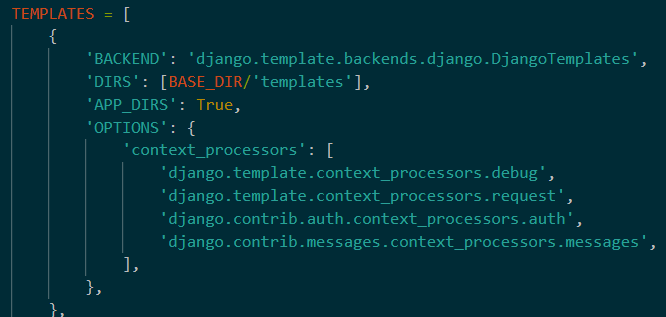
**Creamos la base de datos**

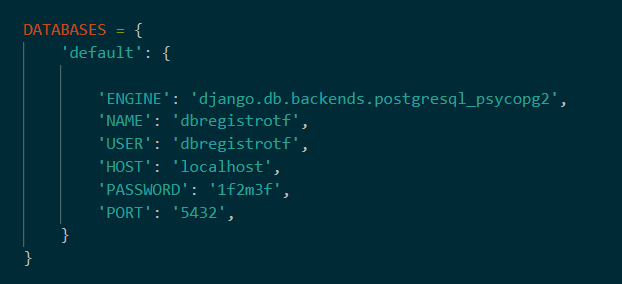
Con la consola postgres (en mi caso abrí la consola de postgres aparte)

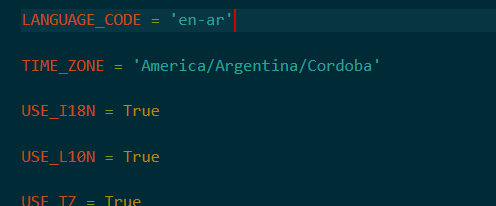


**Modificamos archivos por única vez**



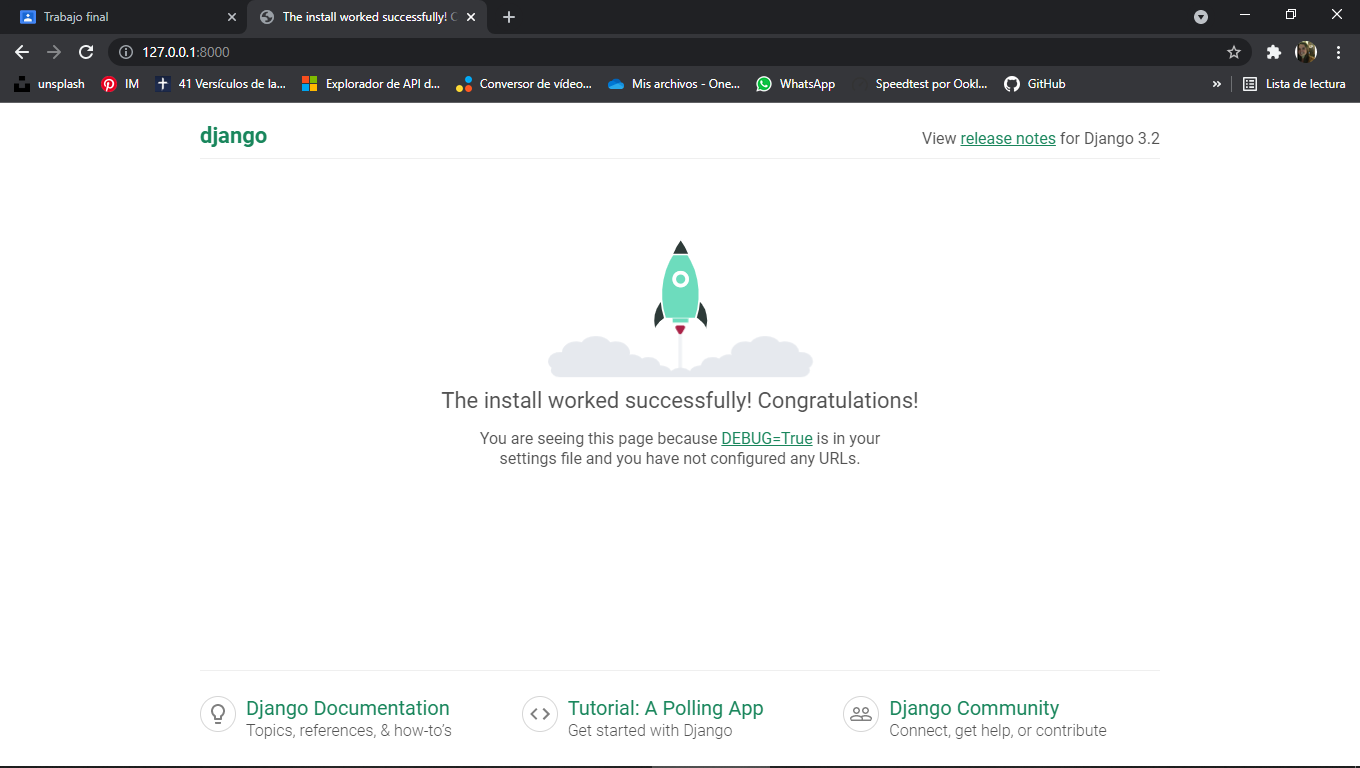


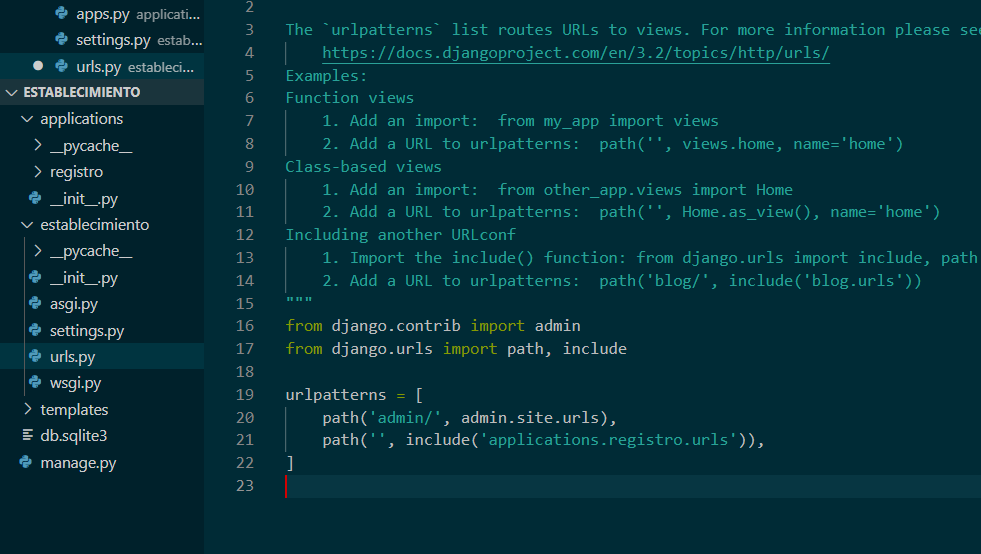




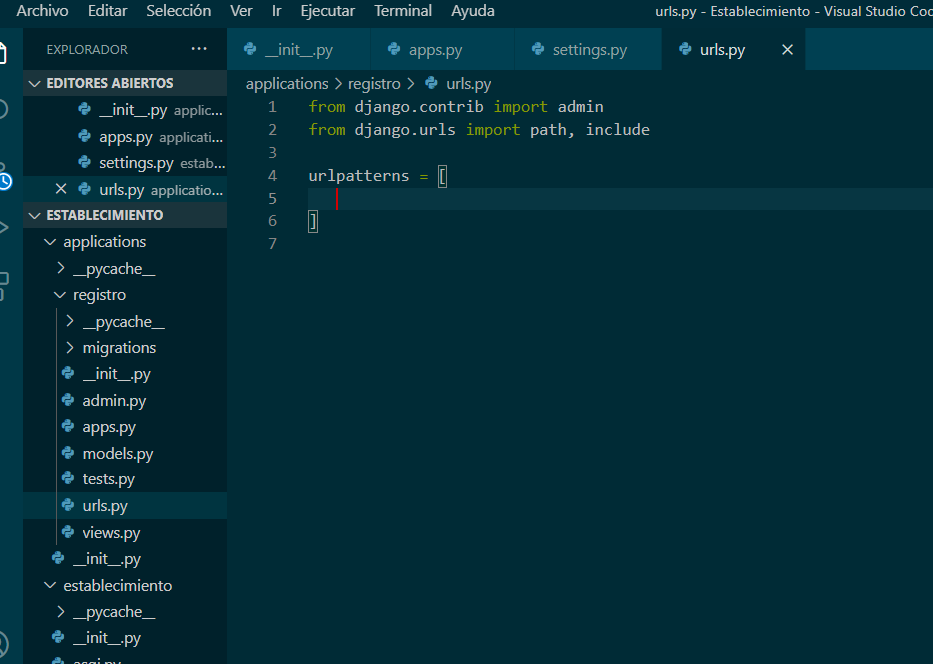
Corremos el proyecto nuevamente:

* Python manage.py runserver



Incluimos la librería de urls de nuestro proyecto registro:

Y creamos nuestra urls.py para nuestro proyecto.



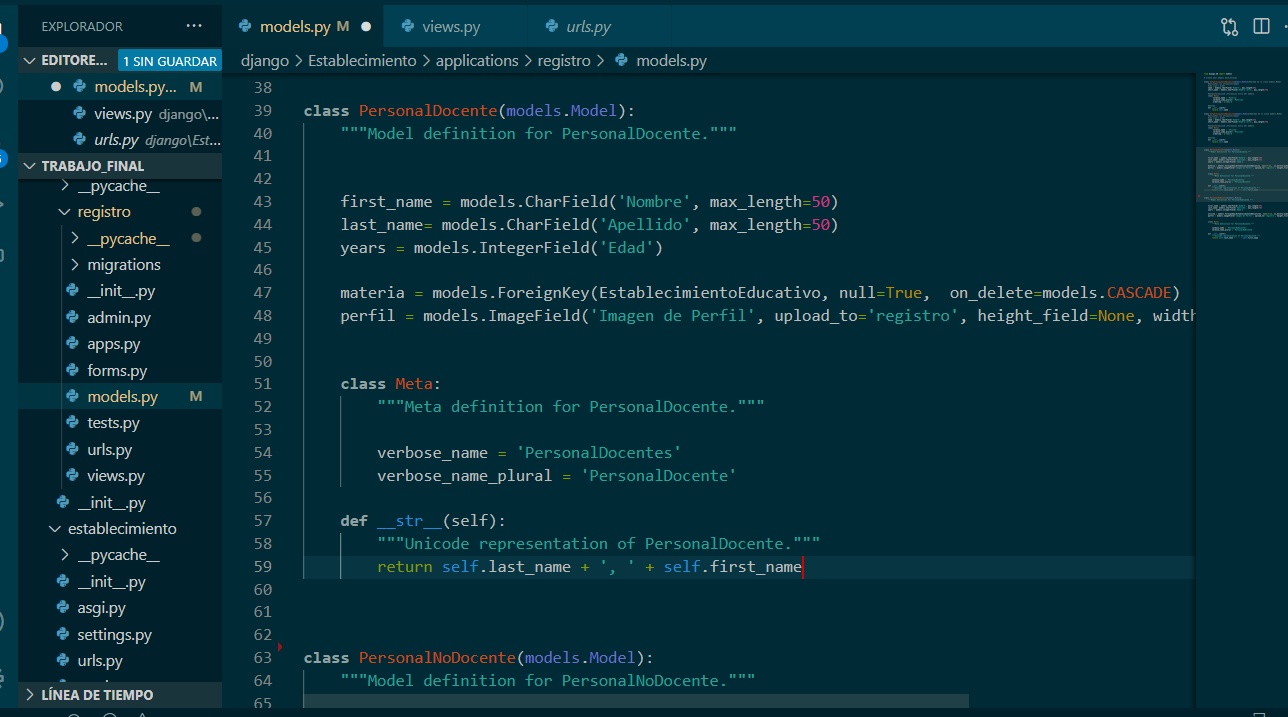
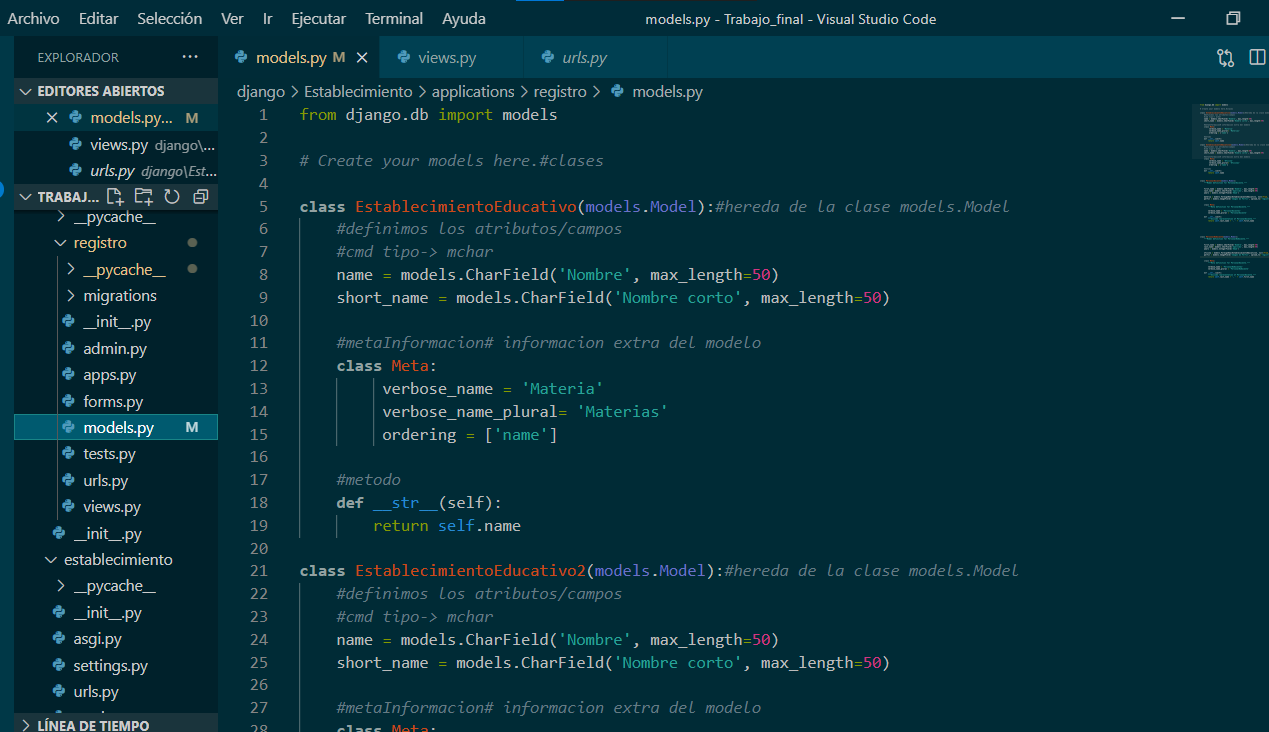
**DEFINIMOS LOS MODELS EN NUESTRA APPLICATIONS REGISTRO:**

Los modelos a través de clases son:

Clase EstablecimientoEducativo: para materias

Clase EstablecimientoEducativo2: para oficinas

(En el examen explico porque tuve que crear con esos nombres, quería cambiar los nombre y no me dejaba hacer la migrations, seguro algún problema con las tablas, lo hice mucho dsp, cambie todo por eso)

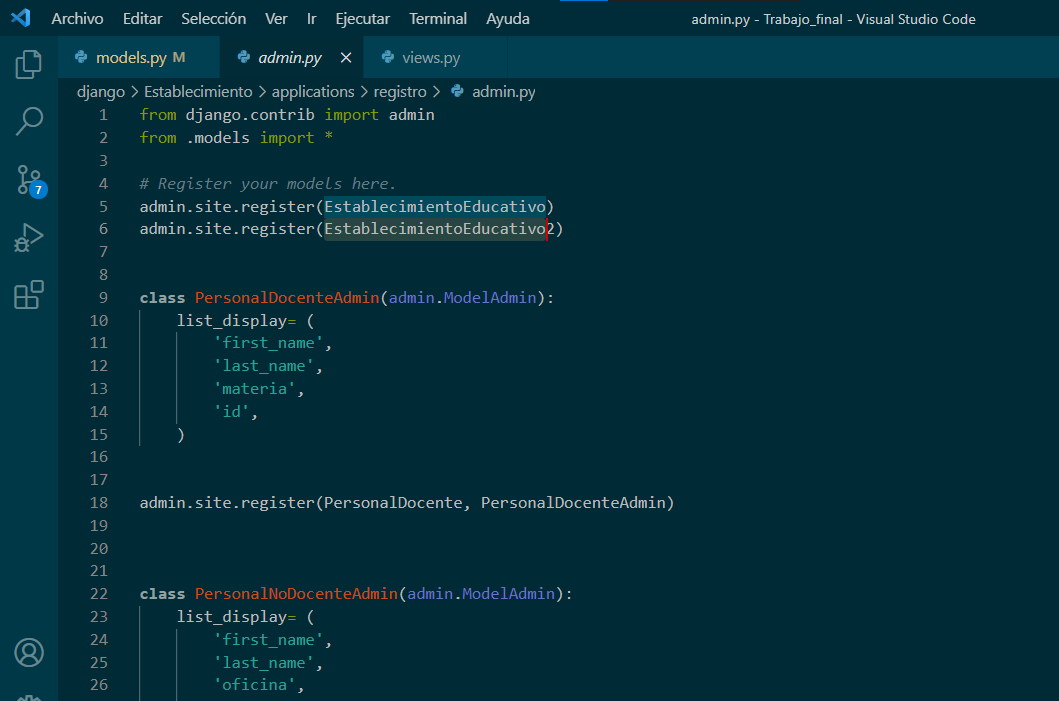
****

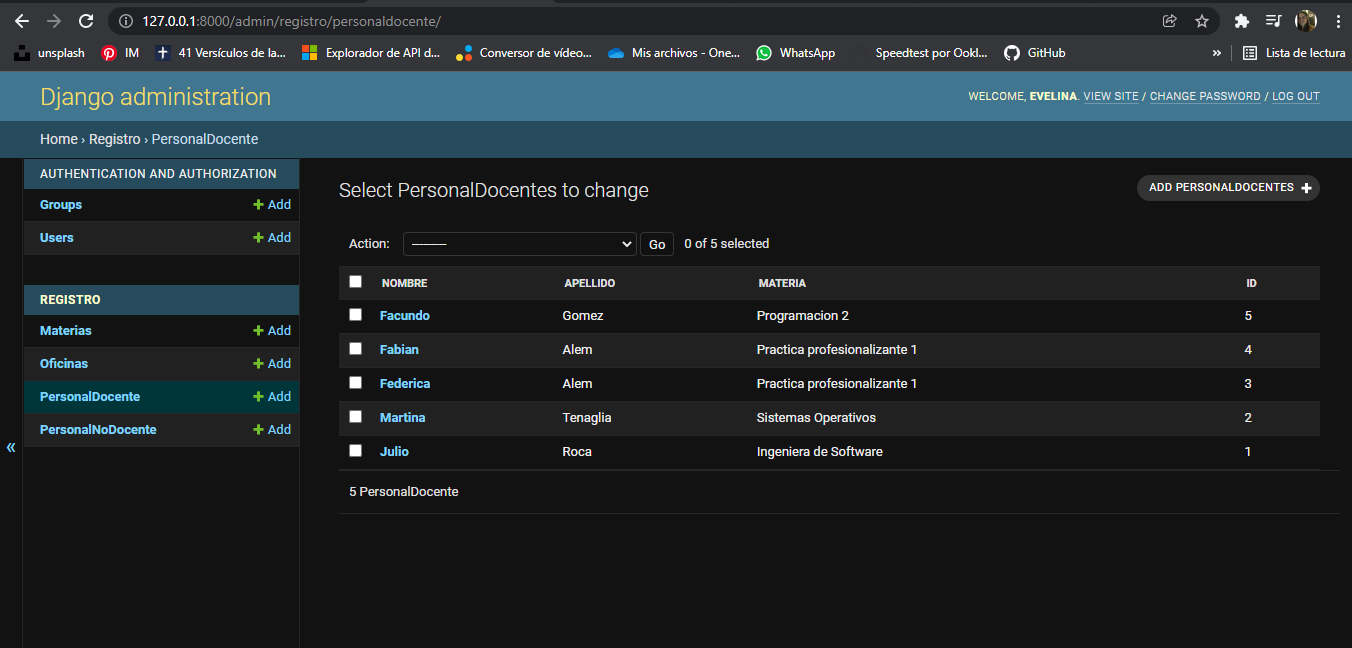
**Luego de definir las clases:**

Hago la migración para que me cree todas las tablas la base de datos

**Migrations: makemigrations y migrate(me olvide de sacarle fotos a makemigrations**

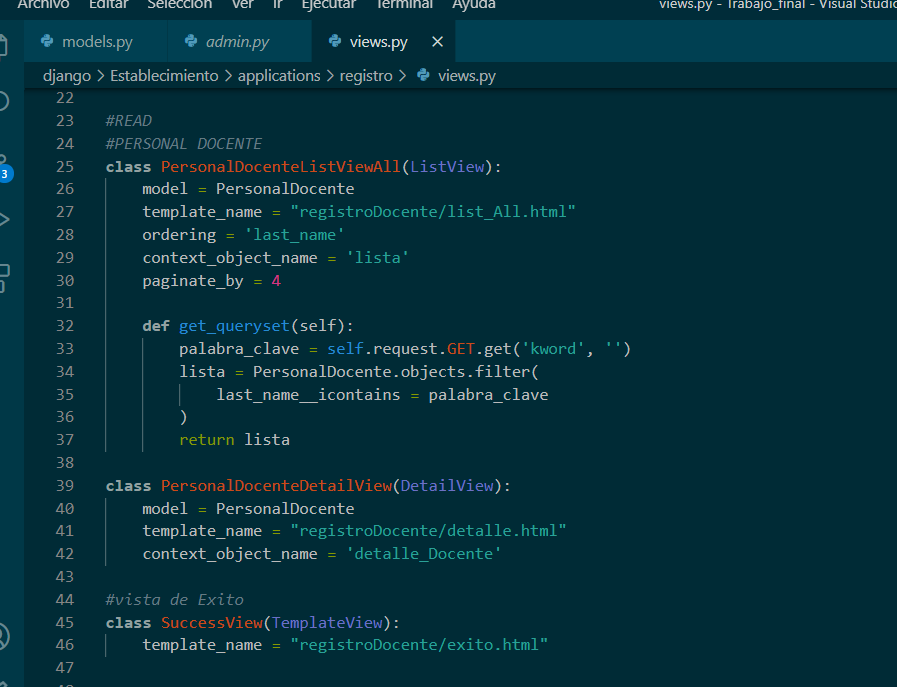
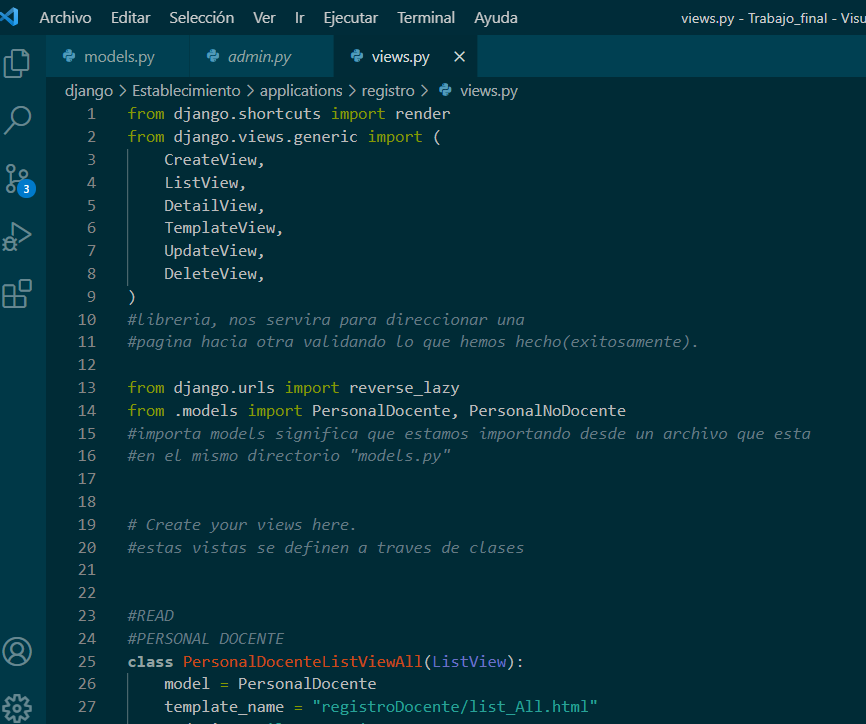
**Luego configuro a admin para definir como mostrara sus columnas**

****

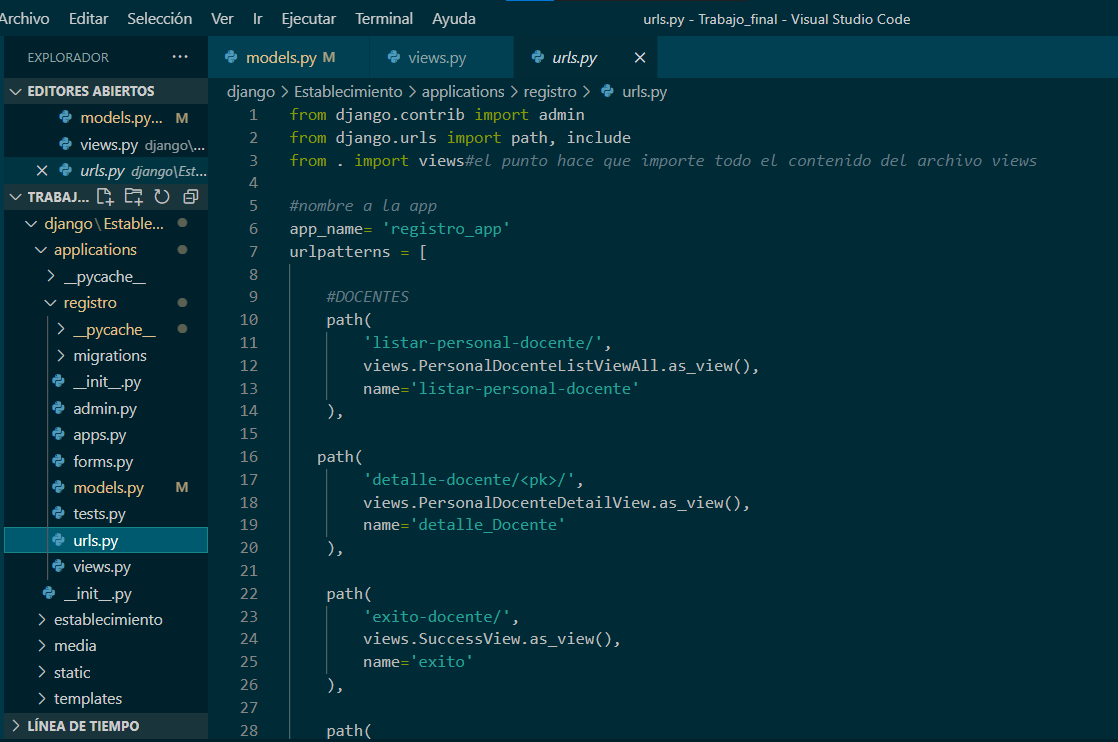
****

****

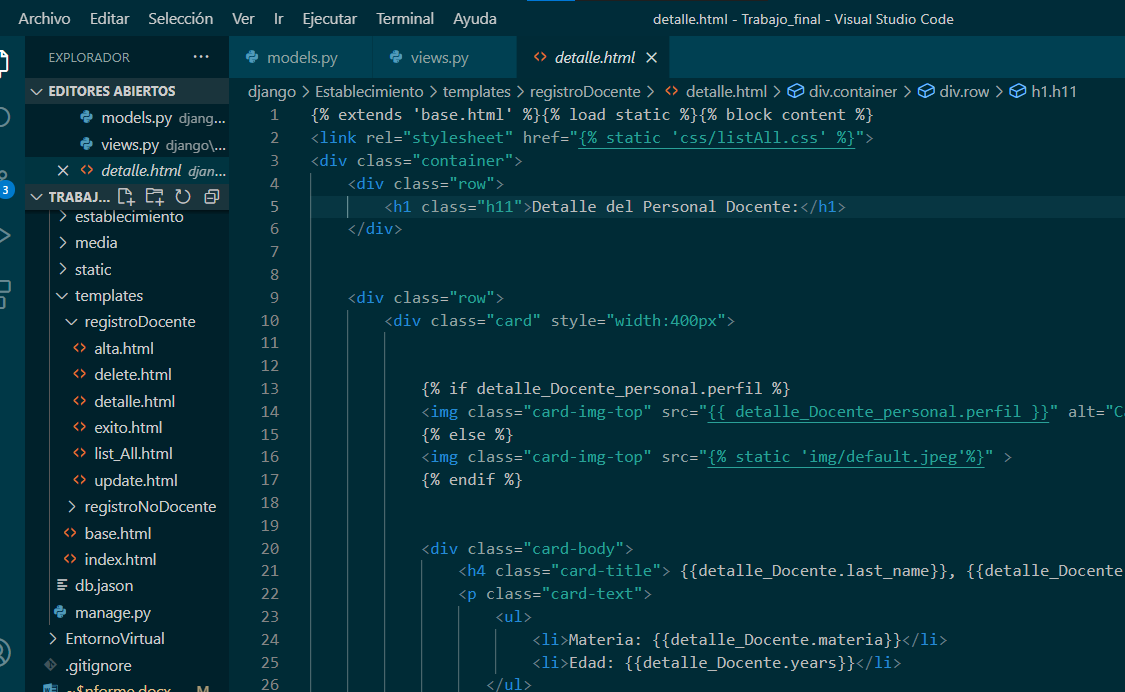
**DEFINIMOS LAS VISTAS EN NUESTRA APPLICATIONS REGISTRO JUNTO CON LAS URLS:**

**

*URLS*

**

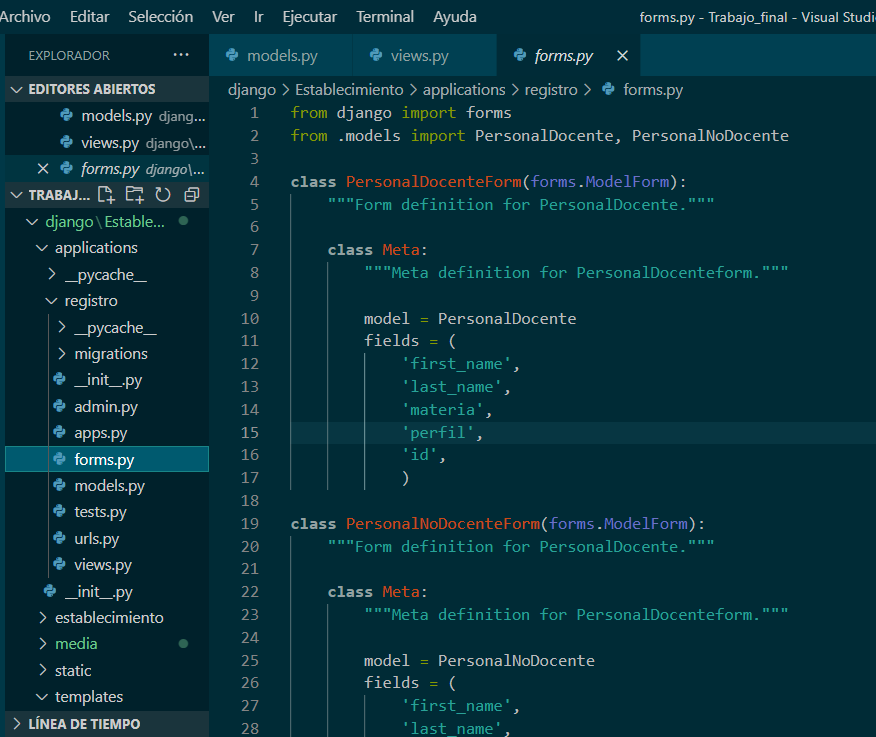
*Templates html*

**

Cree dos carpetas con sus vistar html, una para docentes y otra para No docentes



**Creamos el archivo form.py dentro de nuestra aplicattions registro:** para tener más ordenado nuestros formularios

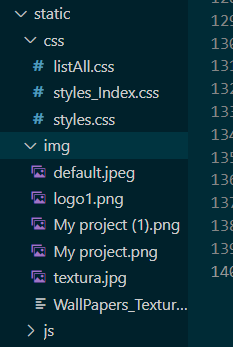
****

**Creamos nuestro html Base, que tendrá todas nuestras vistas/urls nuestra barra superior de navegación:**

Primero colocar un static url para que django pueda trabajar con archivos staticos



Y creamos nuestra carpeta static donde se guardaran los archivos estaticos(img, css y js)

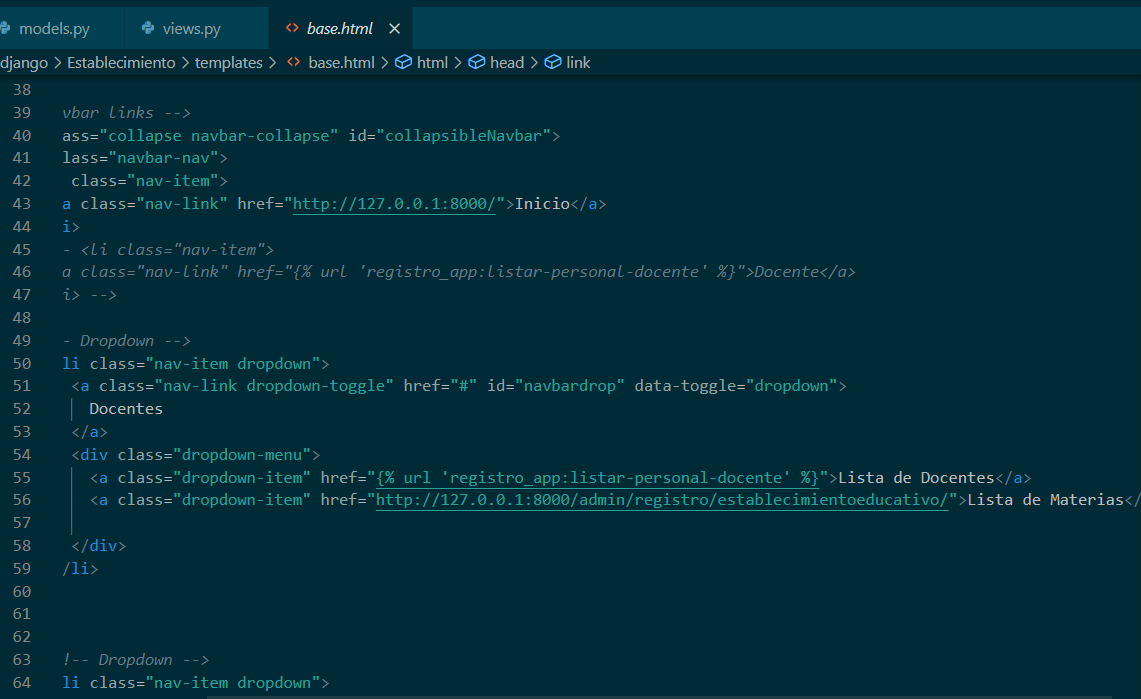


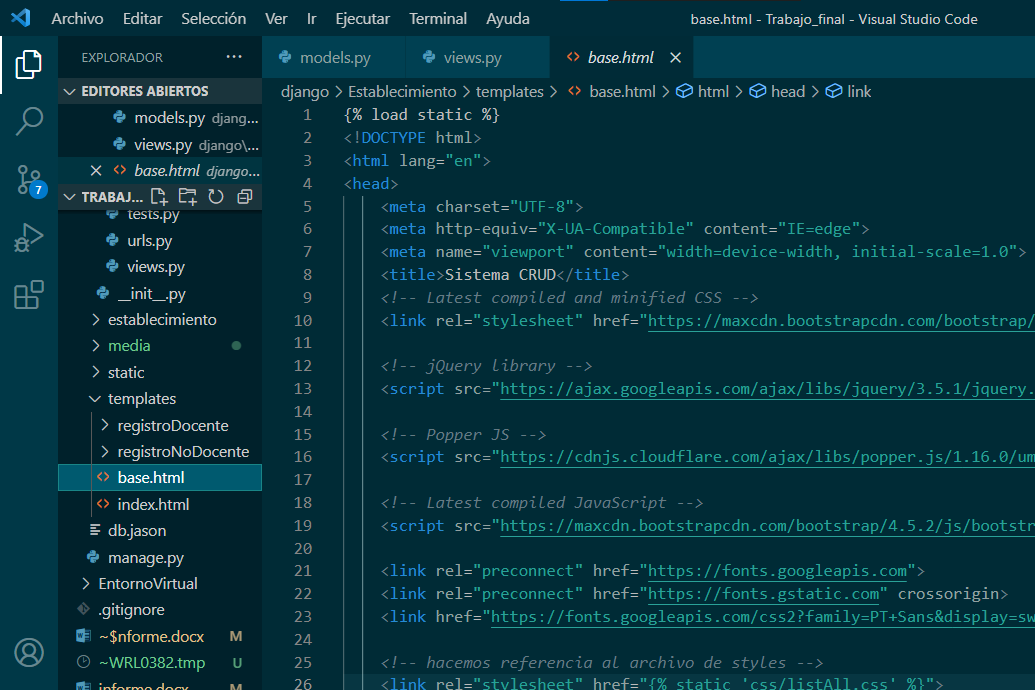
Y luego creamos nuestro archivo html base para nuestra navbar fija que tendremos en todas nuestras vistas.

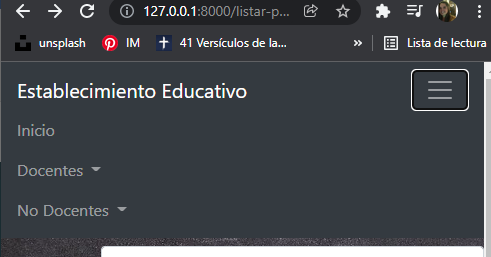
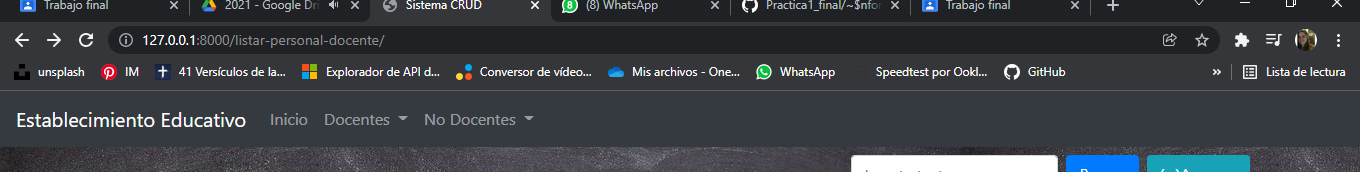
Tambien usamos nuestras etiquetas static para cargar la configuración.

**Explicacion de base.html:**

Aquí en el código creamos etiqueta url para direccionar cada uno de los dropdown (como href)

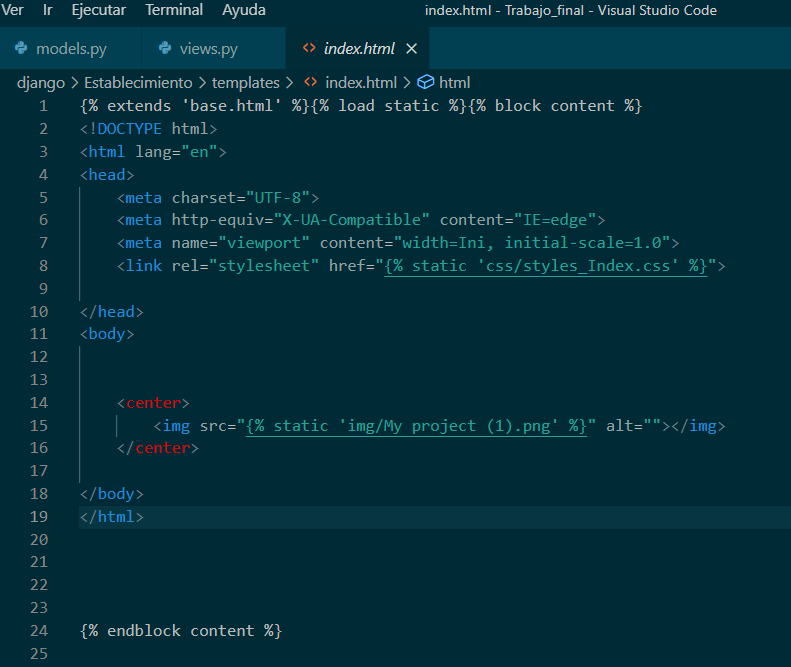






Heredamos todo lo de base.html para mi archivo index.html y cargamos la configuración static para todas mis vistas

Y a partir de la etiqueta block agrego cosas para esa vista especifica.

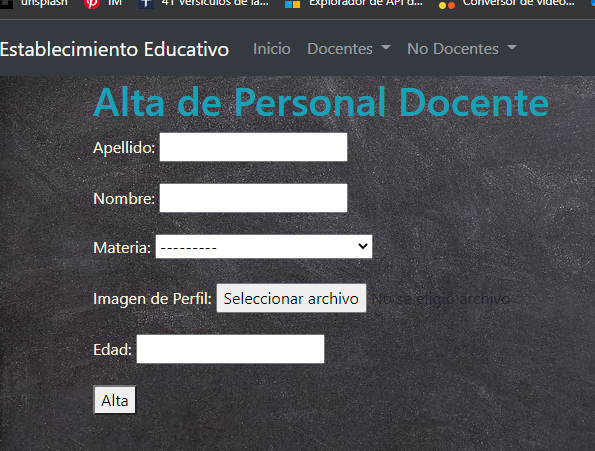


No quiero dejar de aclarar que, al crear las vistas con las urls, aplicamos Crud: vista para ALTA, AGREGAR, MODIFICAR Y ELIMINAR (créate, read, update y delete)

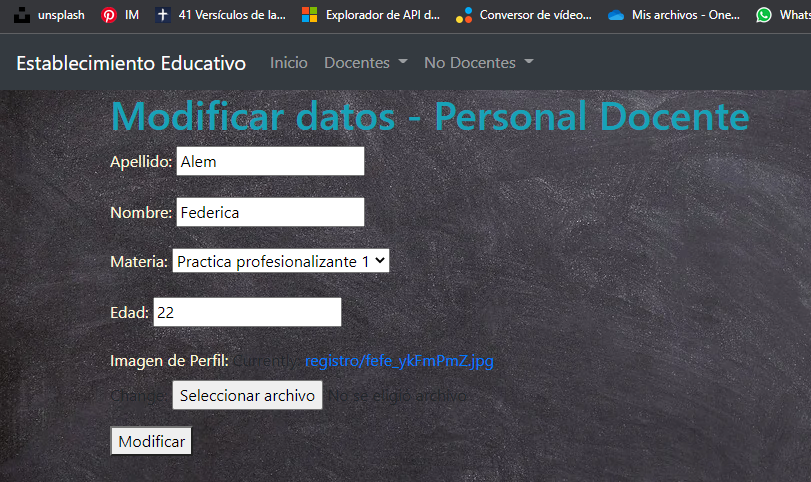
**READ:**



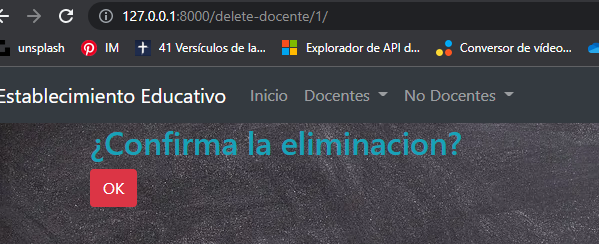
**ALTA:**

****

**MODIFICAR:**

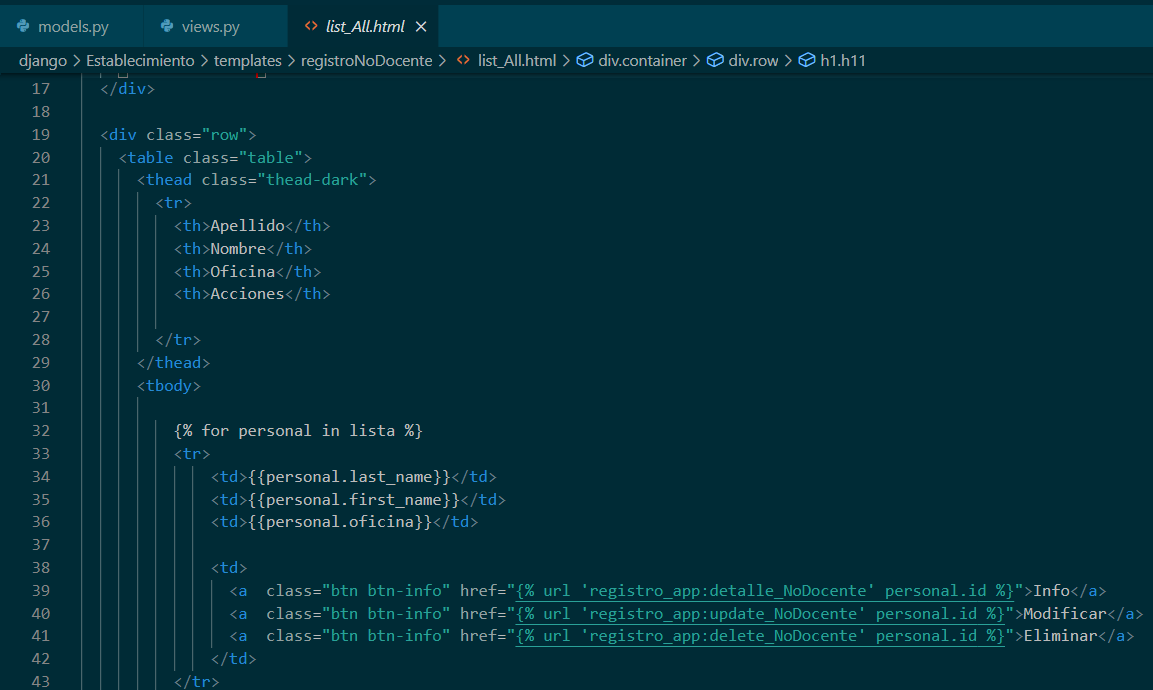
****

**ELIMINAR:**

****

**Aplique la ayuda de bostrapp para crear tablas con diseños bonitos**

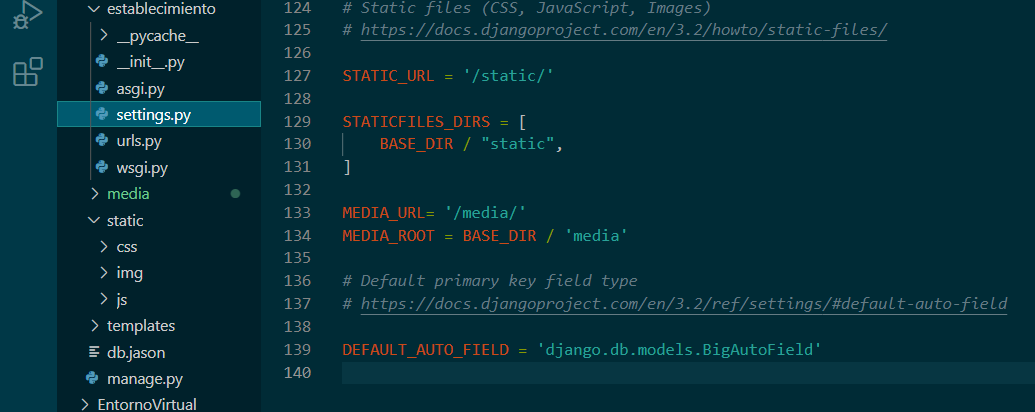
****

****

**Y como ve, agregue paginador a partir de 4 filas.**

****

**Luego agregue lo que es media en setting para que me tomara los archivos de multimedia y supiera la direccion de esa carpeta.**

****

Pido disculpas por el informe tan general y no detallado que he hecho y además por tanta desprolijidad.

Obvio que si tuve muchos inconvenientes, volver a crear una base de datos, temas de las vistas y demás que no lo he puesto. Pero paso a paso he seguido las clases y así he podido lograrlo…

Había empezado a hacer el informe, pero con tantos inconvenientes terminé siguiendo el práctico, pero no simultáneamente con el informe, por eso lo hice a último momento.

¡Saludos!