PATRONES DE DISEÑO

Son estándares que hay que seguir para que nuestro código sea más entendible y abierto a mejorar.

El patrón base es:

MVC:

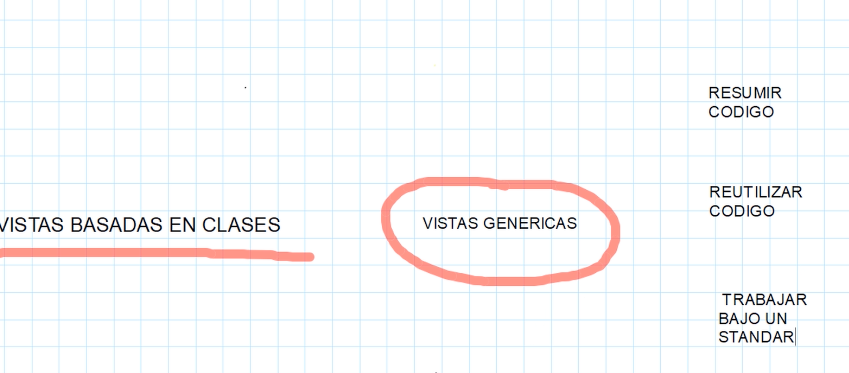
* Modelo
* Vista
* Controlador

Django tiene su propio patrón:

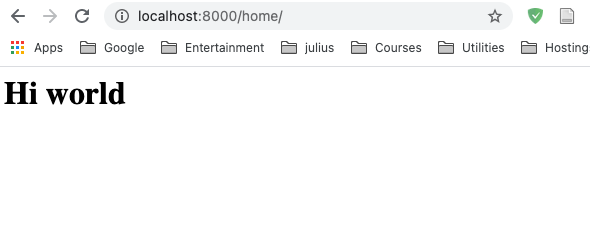
MVT:

* Modelo: Consulta a base de datos.
* Vista: Se ecarga de la lógica del negocio y tener todo preparado para que el template se encargue solo de pintar algo en pantalla.
* Template: Solo dibuja en pantalla, solo tiene html, no tiene o no debe tener nada de lógica.

Django trae unas vistas genéricas

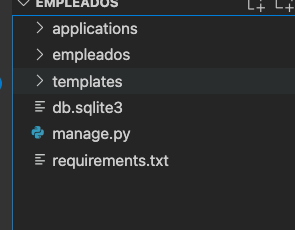


Seguir con el ejemplo, ahora crearemos una app home



Ahora creamos template genéricos. En otras palabras noc reaer la carpeta template para cada una de las aplicaciones.

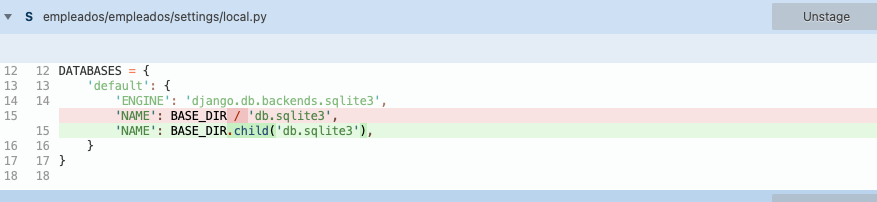
Creamoos una carpeta templates

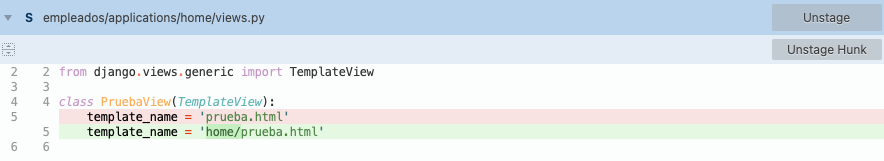


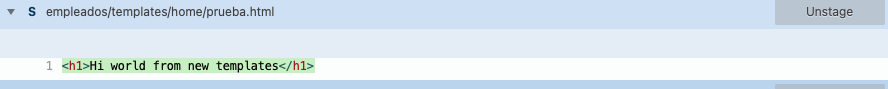
Para que django reconozca esa caperta necesitamos un paquete llamado unipath







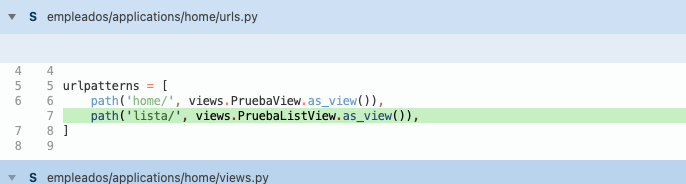


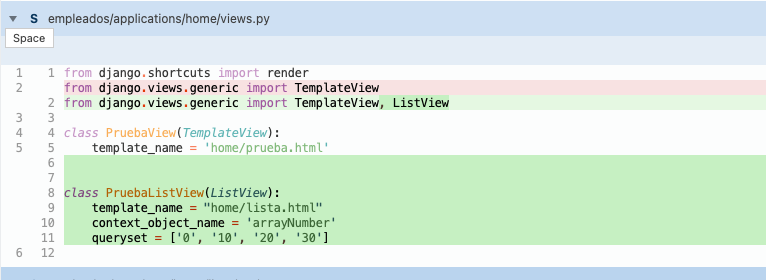


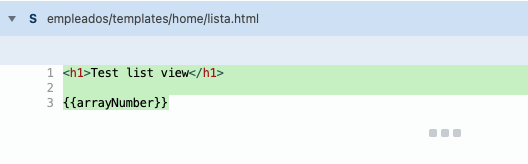
Conociendo más sobre las vistas genéricas.

LISTVIEW

Para hacer listas

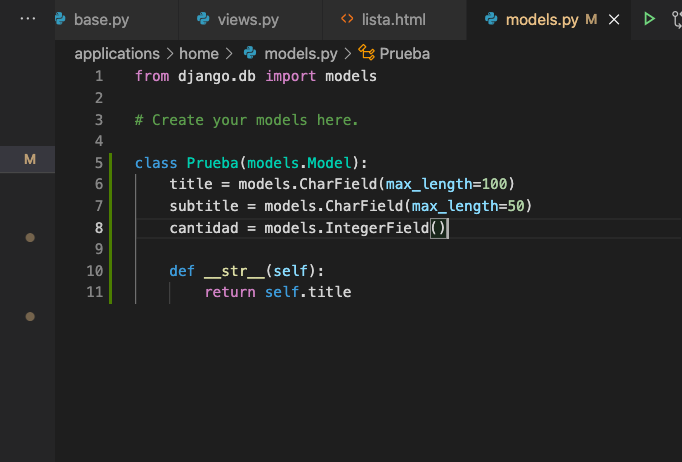






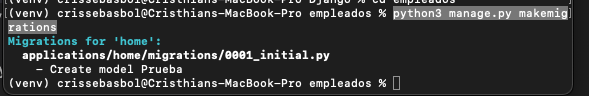
MODELS for VIEWS  
Django trabaja los modelos (bases de datos sql) con ORM

Creamos un modelo:



Para que la clase se plasme en. La base de datos

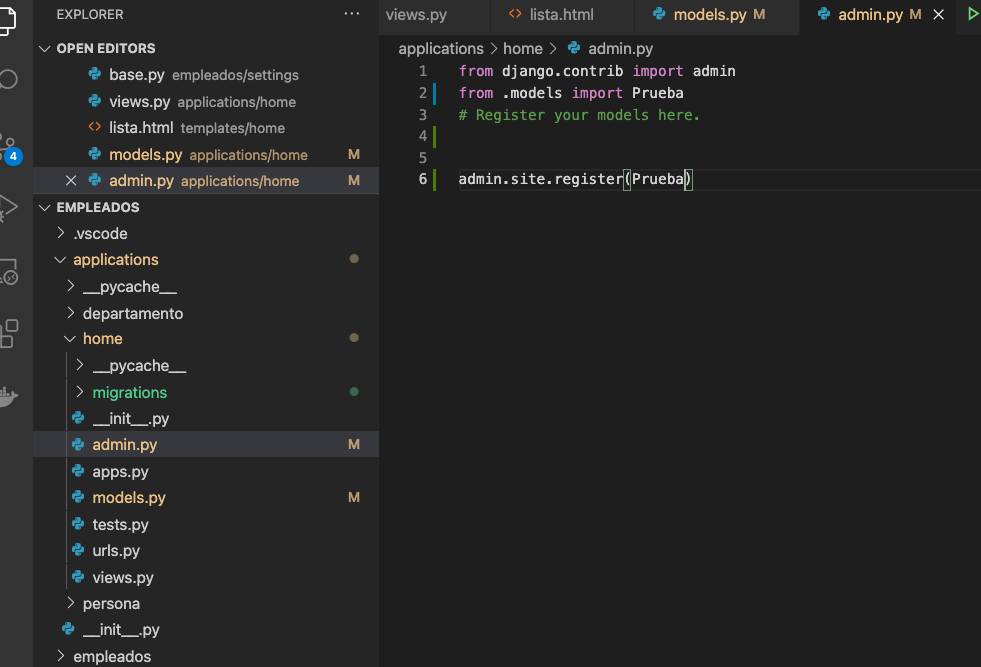
python3 manage.py makemigrations



Ahora subimos los cambios a la base de datos

python3 manage.py migrate

Para mirar que la table se ha plasmado debemos registrar en el adminsitrador el modelo que acabamos de creaer

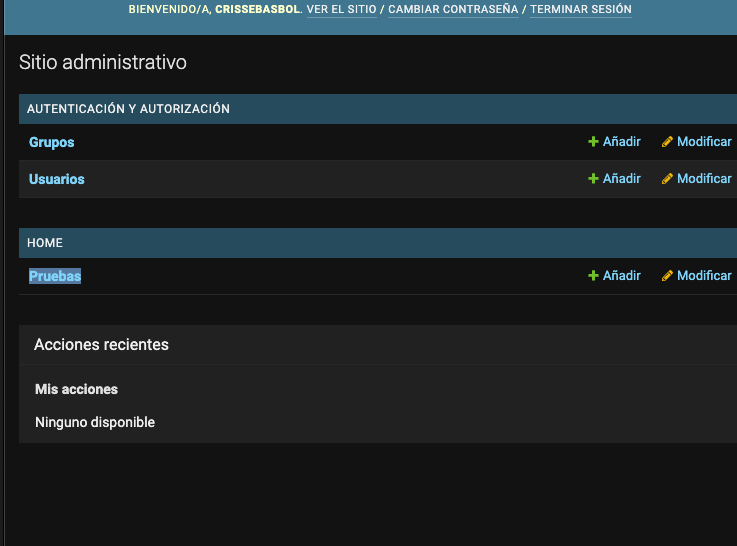


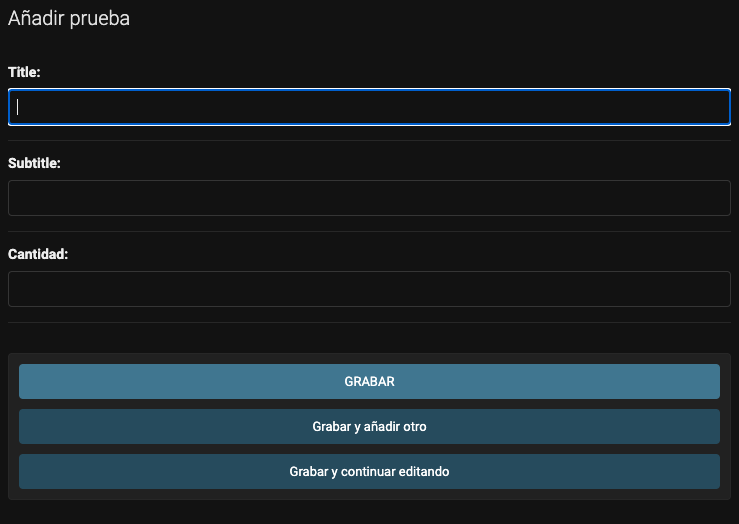
Para ver los registros entramos a la url de admin <http://localhost:8000/admin>

Pero esto nos pide un usuario, lo creamos con;

python3 manage.py createsuperuser

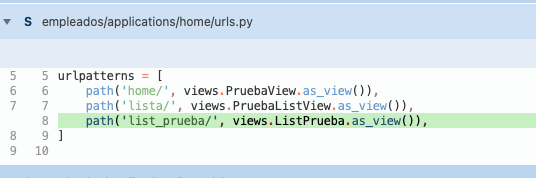
aquí Podemos ver que ya existe la table de prueba

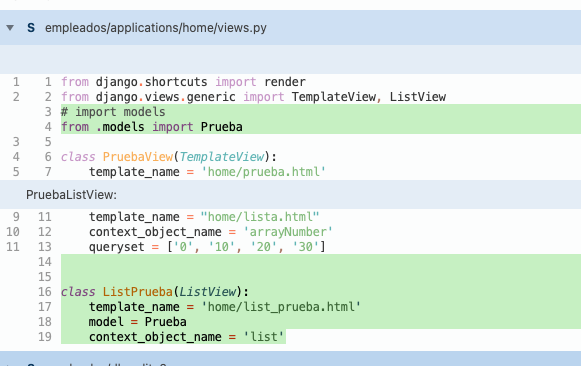


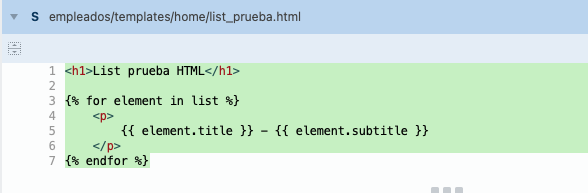


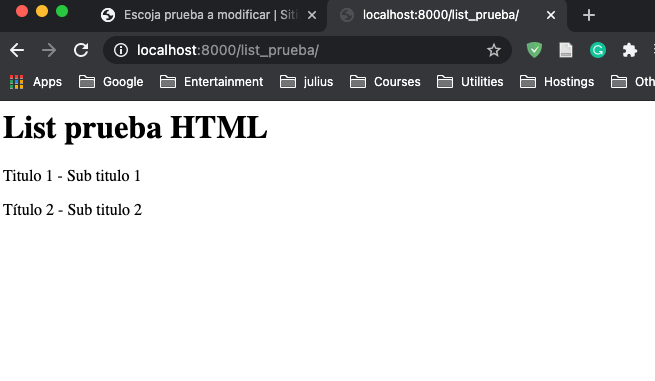
GENERIC VIEWS

LISTVIEW WITH DATA FROM DB









Cretae generic view



