**Instalar python:**  
https://www.python.org/downloads/

Crear entorno virtual

**pip install virtualenv**

**Crear entorno:**

**python -m venv normbretenorno**

Activar entorno:

entorno-virtual-ecommerce\Scripts\activate.bat

**Instalar django:**

**pip install django**

**Crea proyecto con django**

**django-admin startproject nombre-del-proyecto**

**Arrancar proyecto:**

**py manage.py runserver 3000**

**Ver instalaciones que tiene el proyecto:**

**pip freeze**

**Muestra los comandos que hay disponibles:**

**python manage.py**

**todos los cambios que exista en el modelo los va a ejecutar:**

**Crear los archivos**

**python manage.py makemigrations**

**py manage.py makemigrations**

**Toma lo que hay pendiente de migrar y lo migra a la base de datos**

**ejecuta**

**python manage.py migrate**

**py manage.py migrate**

**Crear aplicación:**

**python manage.py startapp orders**

**py manage.py startapp orders**

**permite crear un superusuario**

**python manage.py createsuperuser**

**py manage.py createsuperuser**

**django-allauth**

provee un conjunto de aplicaciones modulares integradas que implementan los flujos comunes de registro y login, tanto locales como integrados con redes sociales. Se destaca especialmente al soportar de forma simultánea los flujos para login local y con redes social.

**pip install django-allauth**

**Tipos de datos - verificar documentación Django:**

**CharField: Almacena cadenas de caracteres de longitud limitada.**

**TextField: Almacena cadenas de caracteres de longitud variable.**

**IntegerField: Almacena números enteros.**

**FloatField: Almacena números en punto flotante.**

**BooleanField: Almacena valores booleanos (verdadero o falso).**

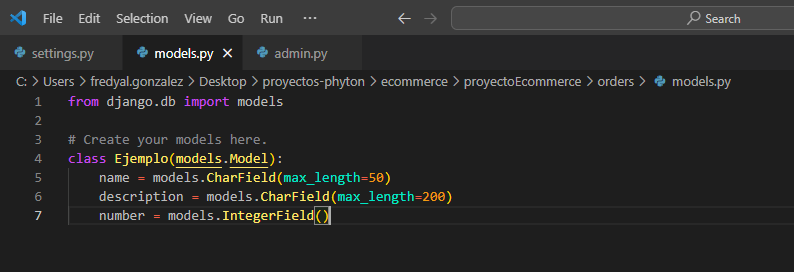
**DateField: Almacena fechas.**

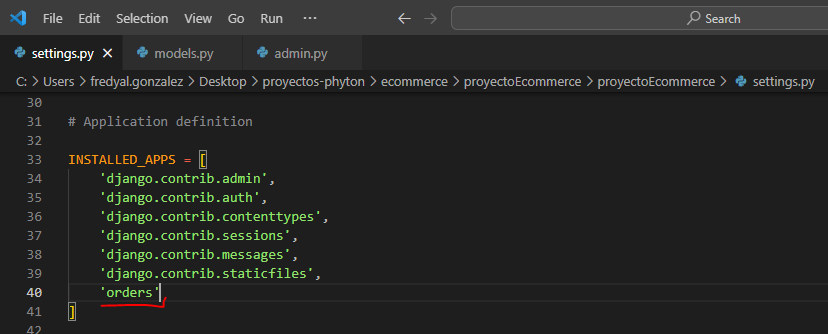
**DateTimeField: Almacena fechas y horas.**

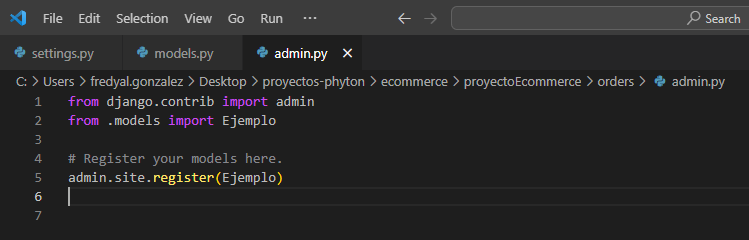
**ForeignKey: Crea una relación uno a muchos con otro modelo.**

**ManyToManyField: Crea una relación muchos a muchos con otro modelo.**

CharField: Campo de texto corto, se utiliza para almacenar cadenas de texto de longitud limitada.  
  
TextField: Campo de texto largo, se utiliza para almacenar cadenas de texto más largas.  
  
IntegerField: Campo para almacenar números enteros.  
  
FloatField: Campo para almacenar números de punto flotante.  
  
BooleanField: Campo para almacenar valores booleanos (verdadero/falso).  
  
DateField: Campo para almacenar fechas.  
  
DateTimeField: Campo para almacenar fechas y horas.  
  
EmailField: Campo para almacenar direcciones de correo electrónico.  
  
ForeignKey: Campo que define una relación de clave externa a otro modelo.  
  
ManyToManyField: Campo que define una relación de muchos a muchos con otro modelo.  
  
DecimalField: Campo para almacenar valores decimales.  
  
ImageField: Campo para almacenar imágenes.  
  
FileField: Campo para almacenar archivos.  
  
URLField: Campo para almacenar direcciones URL.  
  
SlugField: Campo para almacenar una cadena de texto que se utiliza para construir una URL amigable para el SEO.







Instalar framework

**pip install djangorestframework**

**para consumir api local**

instalamos

pip install django-cors-headers

**agregamos en settings.py**

en installed apps agregamos:

'corsheaders',

**en middleware agregamos:**

**'corsheaders.middleware.CorsMiddleware',**

Principio del formulario

Principio del formulario

Crear archivo serializer con:

from .models import Categoria,Producto,Historial,OrdenCompra

from rest\_framework.serializers import ModelSerializer

from rest\_framework.serializers import (

SerializerMethodField

)

from rest\_framework import serializers

class CategoriasSerializer(ModelSerializer):

    class Meta:

        model = Categoria

        fields = '\_\_all\_\_'

class ProductoSerializer(ModelSerializer):

    categoriaProducto = CategoriasSerializer(many=False)

    class Meta:

        model = Producto

        fields = '\_\_all\_\_'