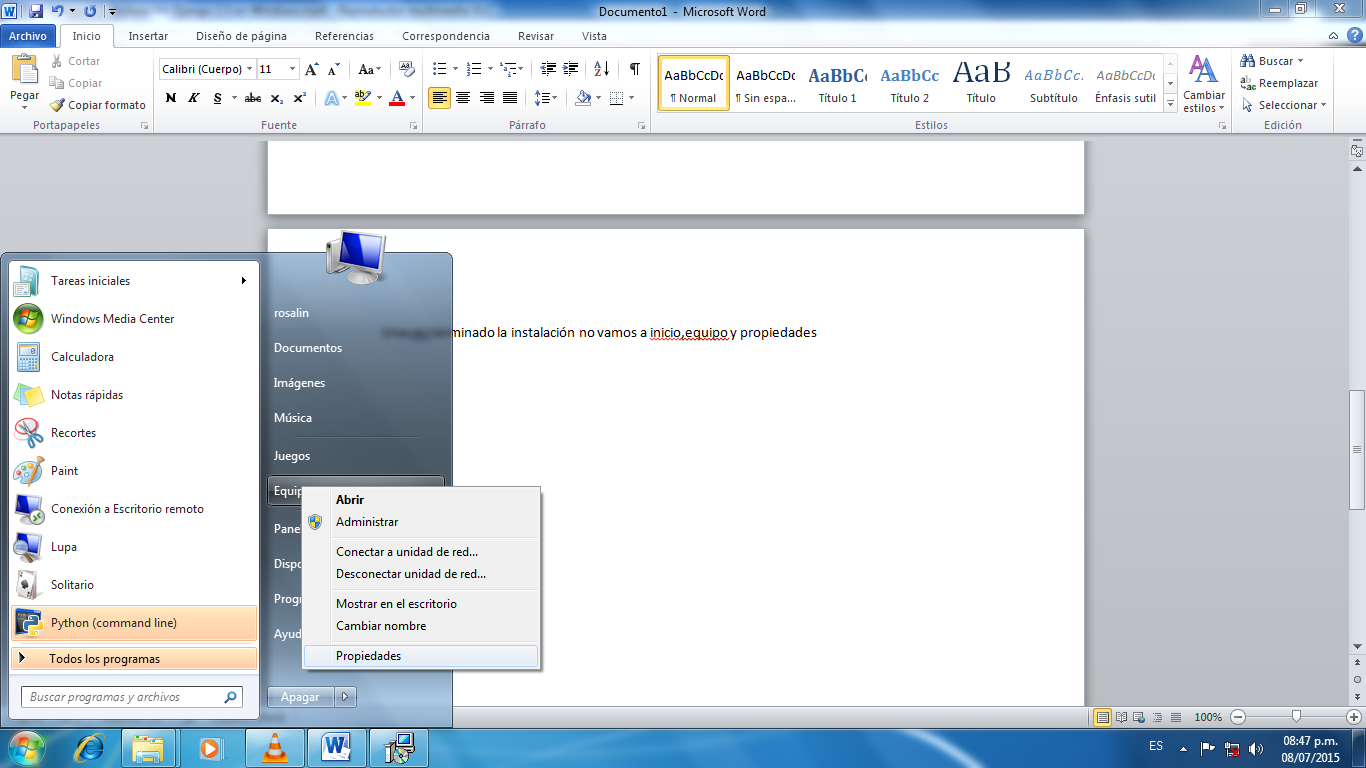
**MANUAL DEL PROGRAMADOR**

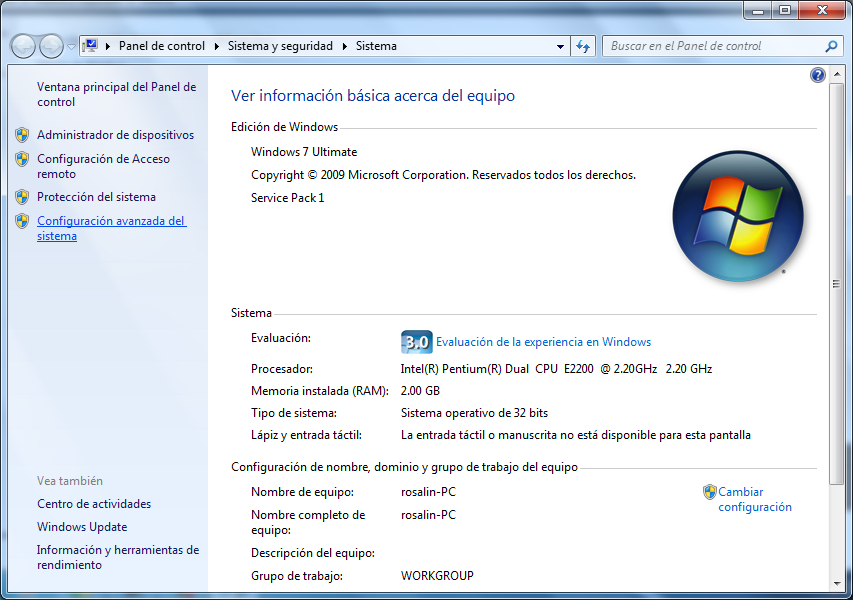
Instalación de Python 2.7.10



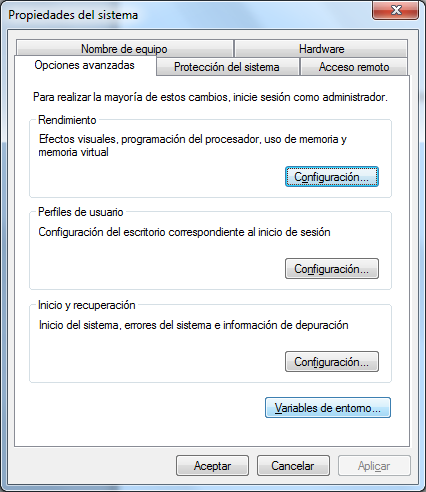
Una vez terminado la instalación no vamos a inicio, equipo y propiedades

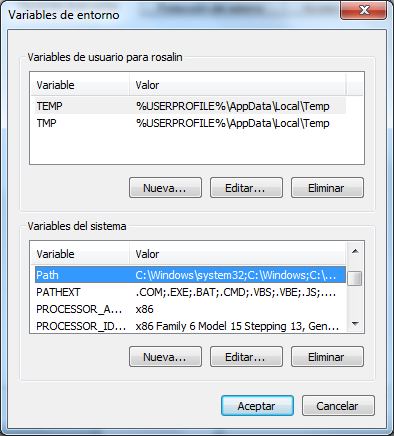


Después nos vamos a configuración avanzada del sistema

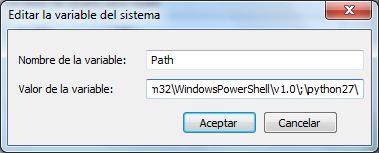


Y vamos abrir la variable de entorno aquí, vamos a buscar path



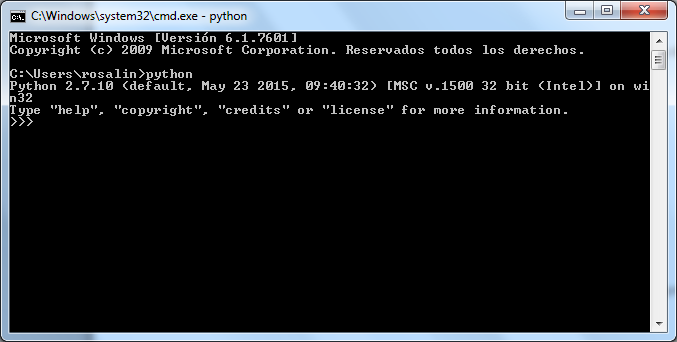


Doble clic en path nos vamos al último y ponemos lo siguiente **\python27\** y damos aceptar



Después para comprobar que estas instalado Python nos vamos a inicio cmd y escribimos Python en la consola.

Nos tiene que aparecer el siguiente mensaje:



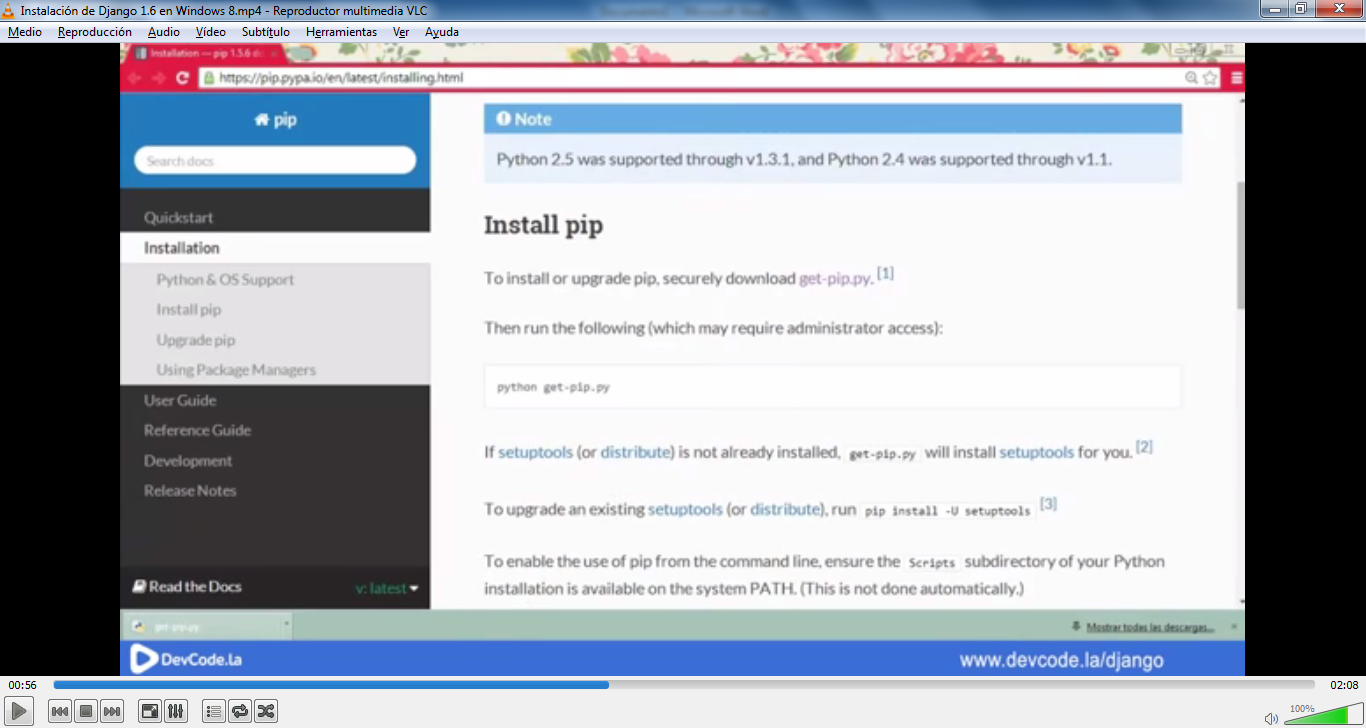
Instalación de **DJANGO 1.8**

Antes de instalar debemos tener instalado **Python 2.7** luego de esto debemos instalar los siguientes requisitos el primero es **setuptools** y el segundo es **pip**.

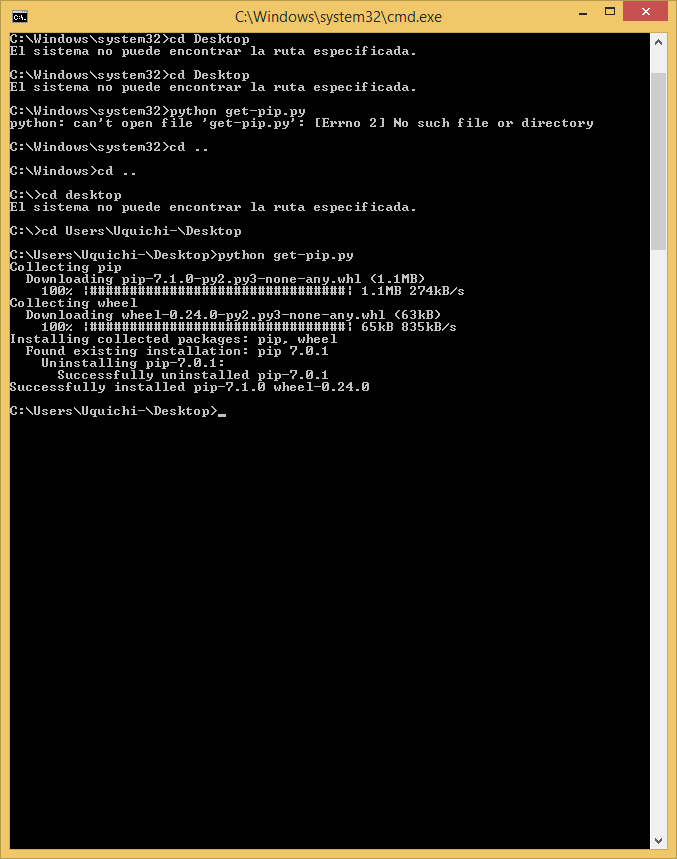
**Setuptools**: es una colección de programas para mejorar las herramientas de distribución de Python

**Pip**: es un administrador de paquetes pues estos nos van a permitir instalar paquetes de Python

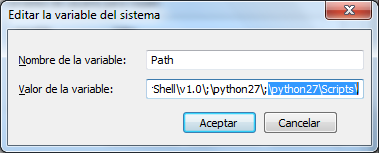
Nos vamos a su página y nos descargamos el **get-pip.py** que ya contiene el **setuptools**



Nos dirigimos a nuestro cmd y nos ubicamos nuestro directorio donde se bajó el **get-pip.py** e instalamos de la siguiente manera:



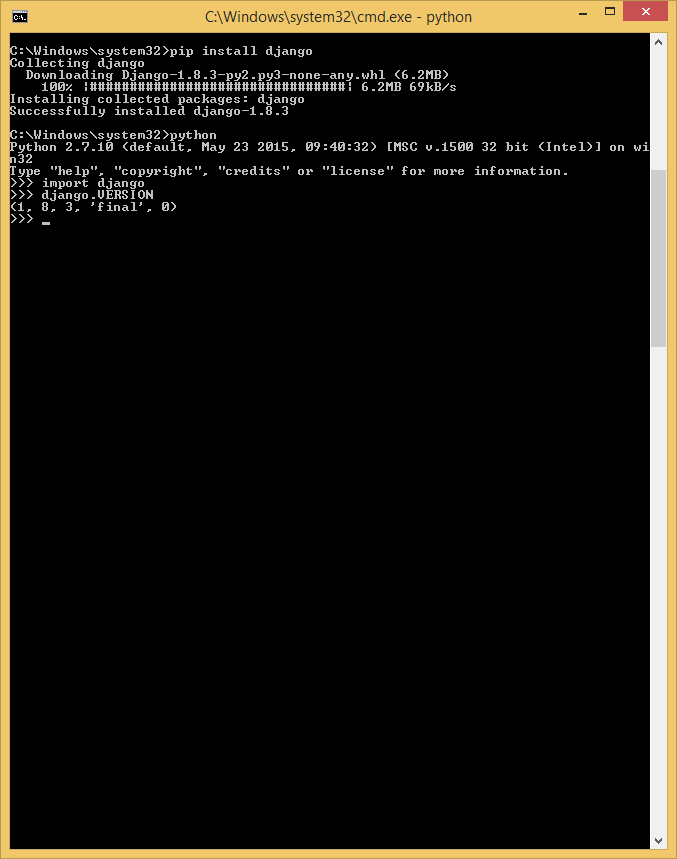
Luego tenemos que colocar en las variables de entorno **\python27\Scripts\** y aceptamos los cambios.



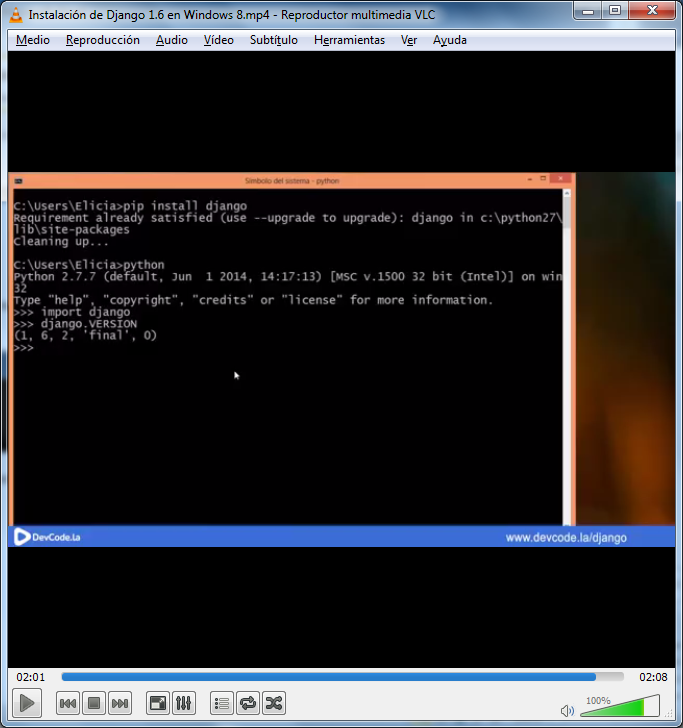
Como verificamos si pip se instaló correctamente en nuestro sistema en el cmd colocamos **pip** y nos debe mostrar el siguiente mensaje:



Ahora instalamos **Django**  utilizando el siguiente comando pip install django y esperamos que se instale.



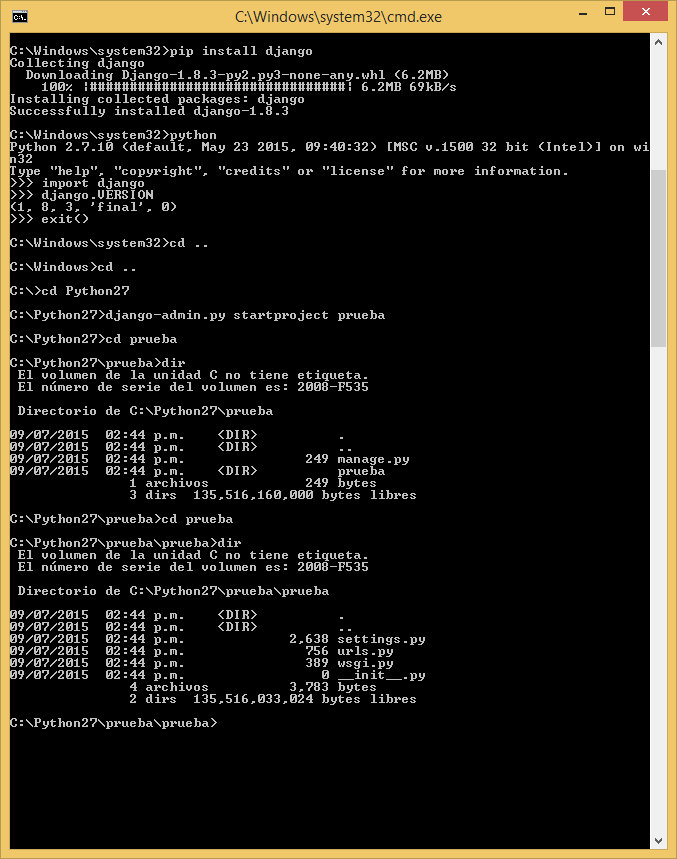
Como verificamos que django está instalando correctamente.



Con esto ya vamos a poder ejecutar los comandos de django en la consola vamos a ser la prueba:

Nos genera un archivo mange.py y una carpeta prueba.

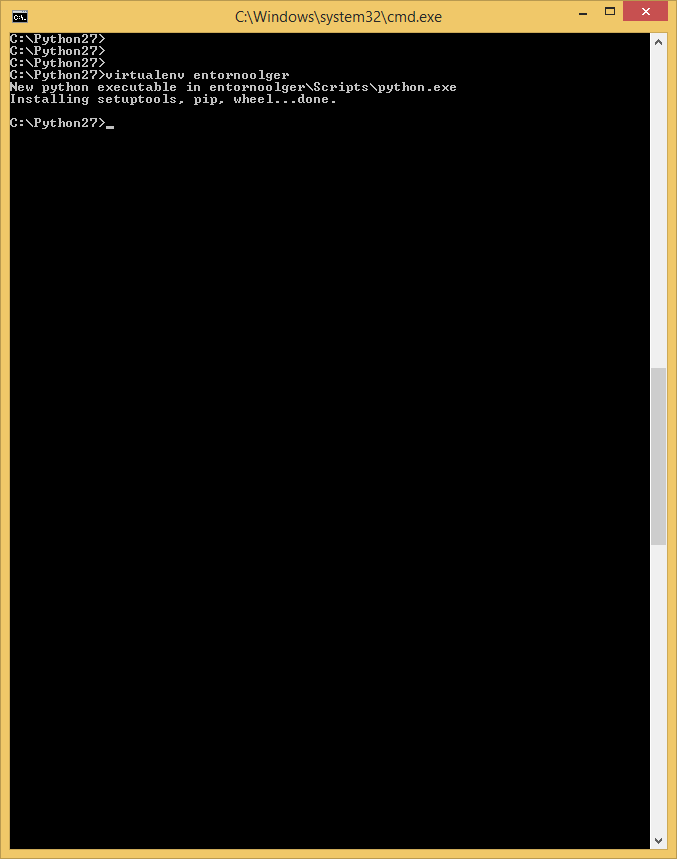
Entramos a la carpeta prueba dentro de ella tenemos settings.py, las urls.py, wsgi.py…



Bueno como ya tenemos django instalado ahora instalaremos el resto de cosas para trabajar correctamente con django instalaremos virtualenv.



Ya tenemos instalado el virtualenv y ahora vamos pasar a crear un entorno virtual



Ahora activaremos nuestro entorno virtual y nos vamos a la carpeta Scripts y ahí tenemos que ejecutar el archivo actívate.



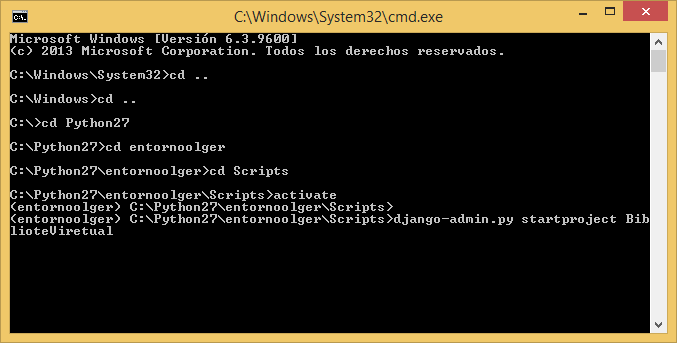
Dentro de ahí ya podemos instalar todo lo que vamos a necesitar como el Django y otros



Creación de nuestro sistema de Biblioteca Virtual

Habilitando el sistema de administrador por defecto Django

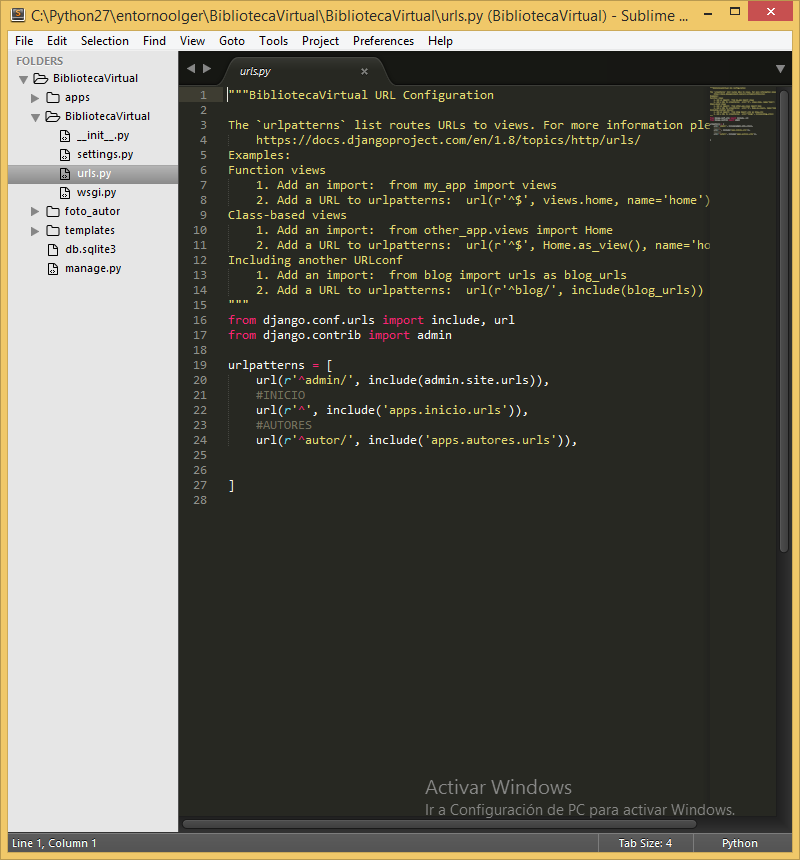
Nos vamos a nuestro proyecto que creamos



Nos crea barios archivos por defecto nos vamos a settings.py

Donde tenemos que declarar la base de datos

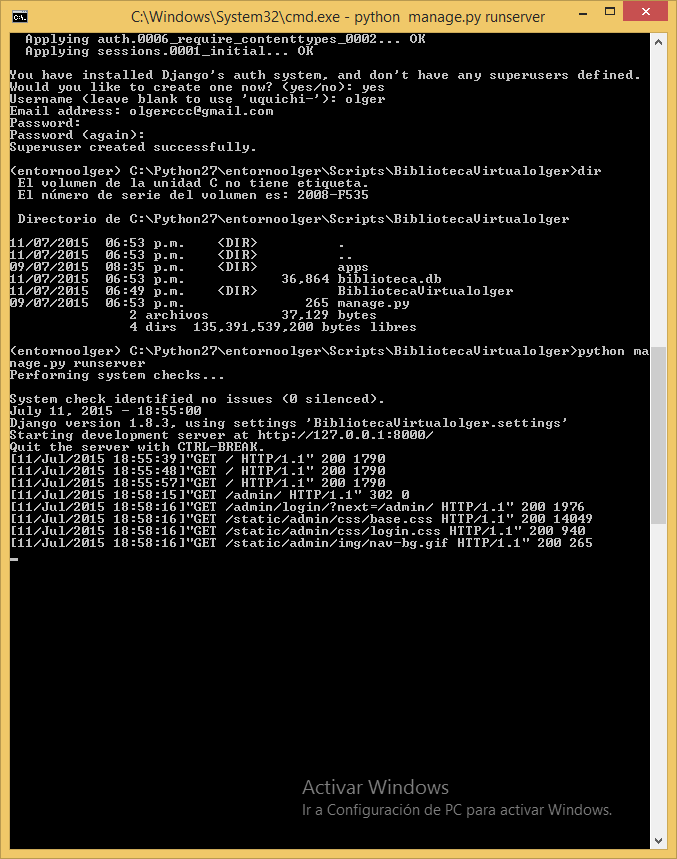
No vamos a los urls.py



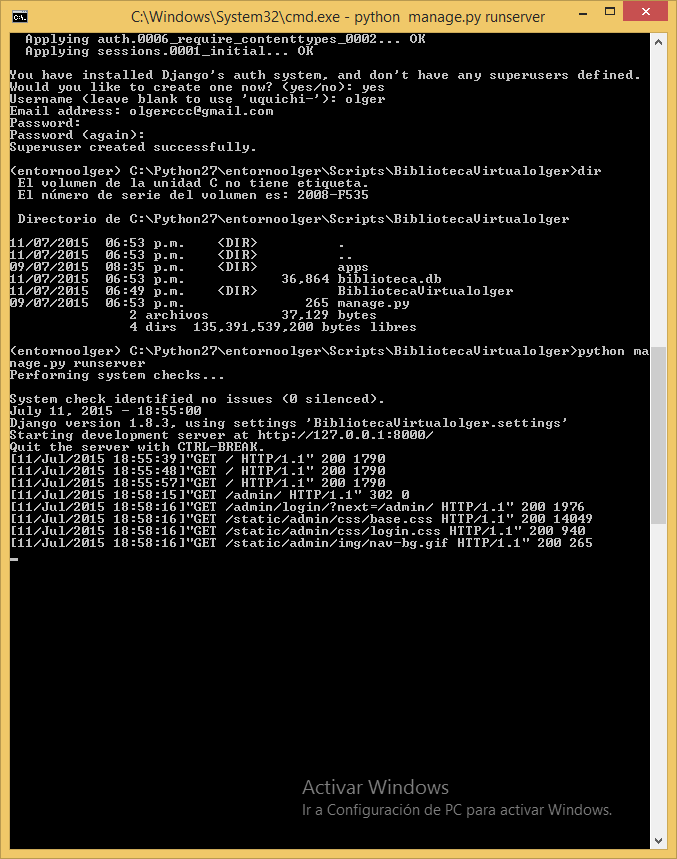
Creamos las tablas de la B.D y sincronizamos la base de datos



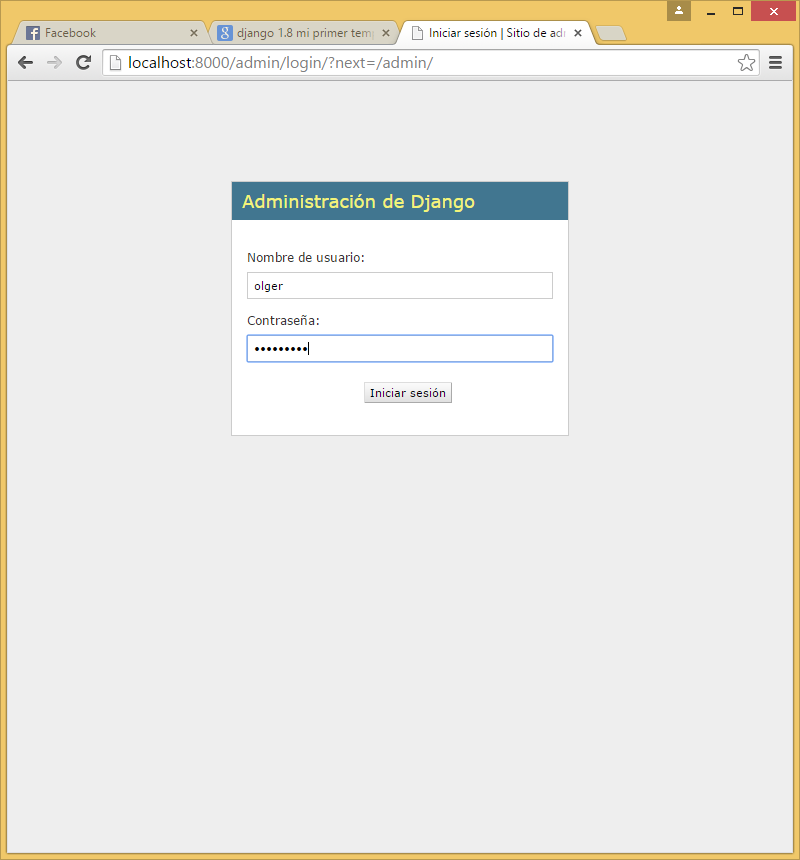
Vamos tener crear un súper usuario que Django trae por defecto para el administrador



Django trae su propio servidor por defecto así que lo corremos



Así que entramos a la url



Nuestra primera aplicación creamos apps y creamos una carpeta donde estará nuestra aplicaciones.

Después crearemos nuestra aplicación que se llamara inicio

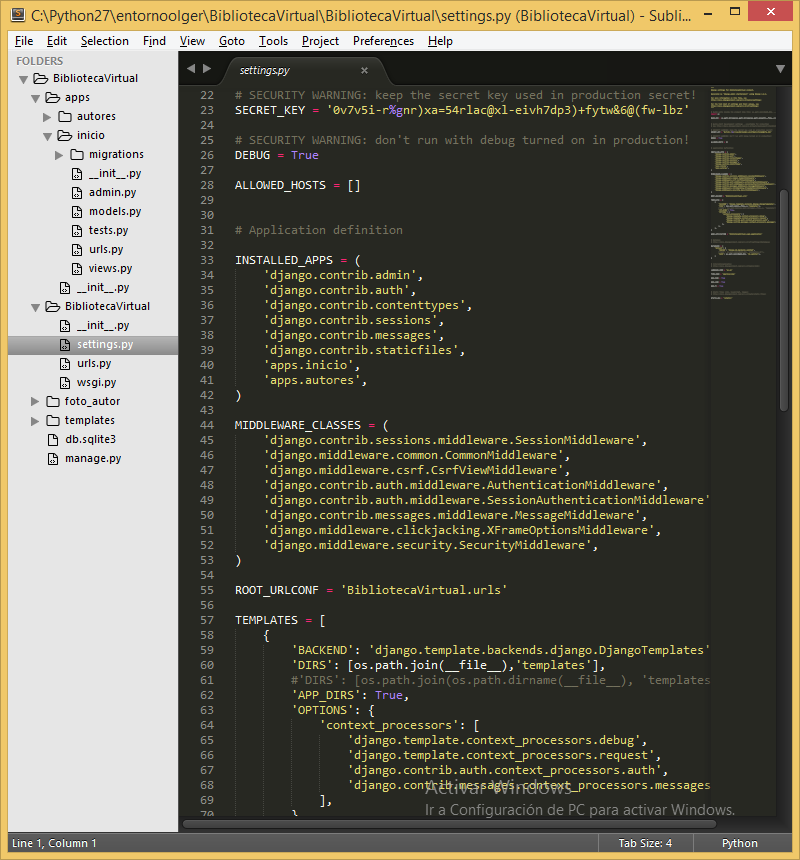


Toda aplicación que creemos nos va generar cuatro archivos

\_\_init\_\_.py, Models.py, Testes.py, Views.py

Donde se va traba mas es en models y views

Le vamos a dar conocer que hemos creado una nueva aplicación

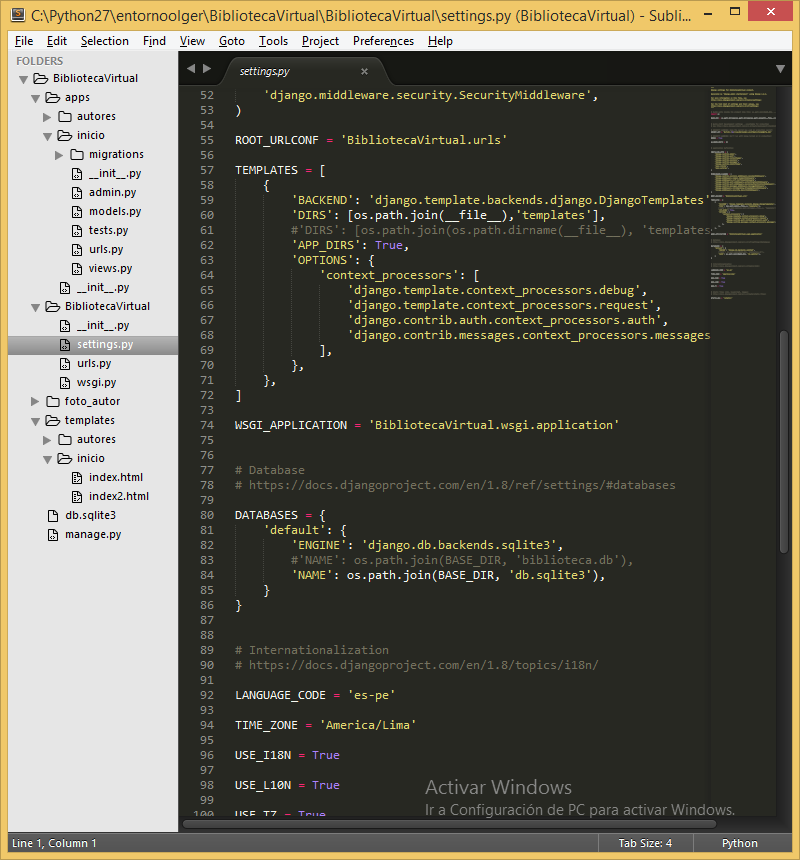


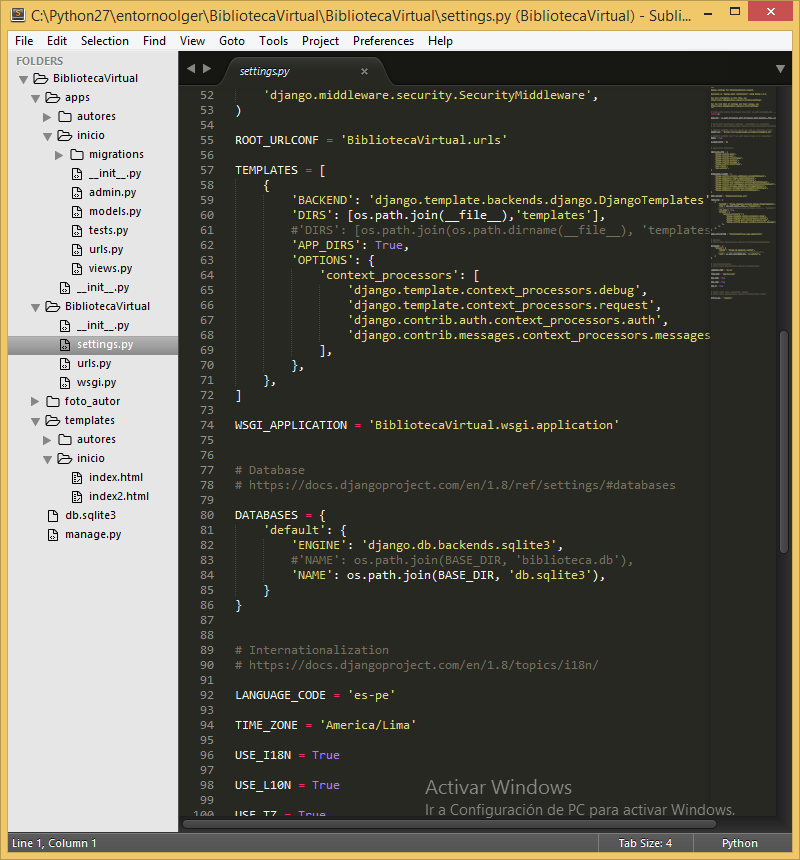
Creamos un archivo \_\_init\_\_py



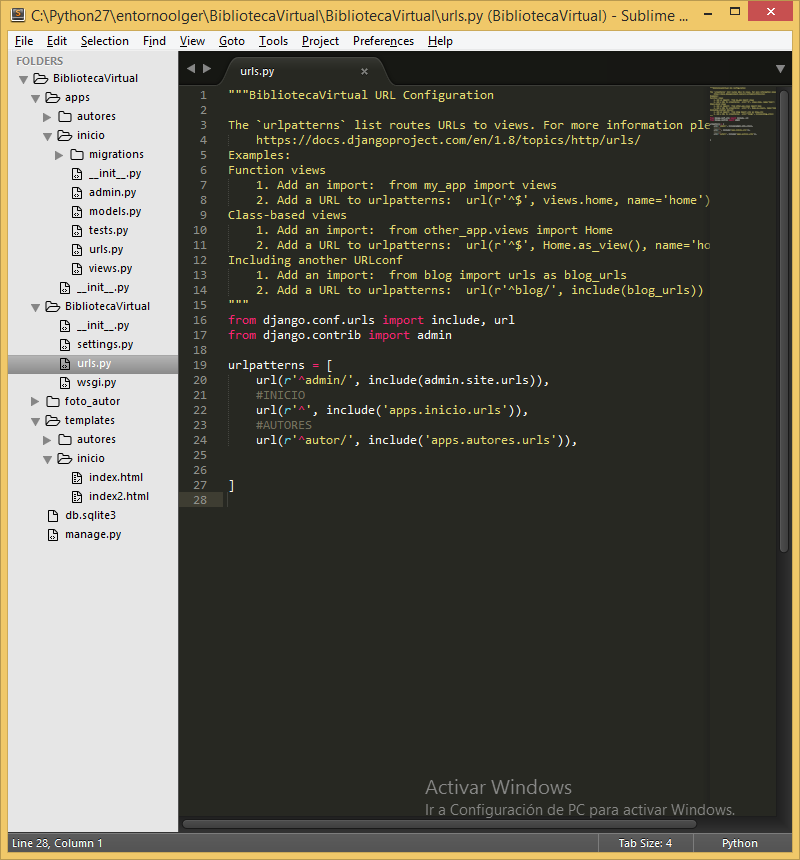
Nuestra primera vista crearemos una carpeta donde van estar todo nuestros archivos HTML llamada **templates** dentro de ellos van estar todo nuestros archivos html, dentro de ello crearemos una carpeta inicio porque asemos eso por hemos creado una apps llamado inicio

Toda aplicación que nosotros creemos va tener su propia carpeta de templates

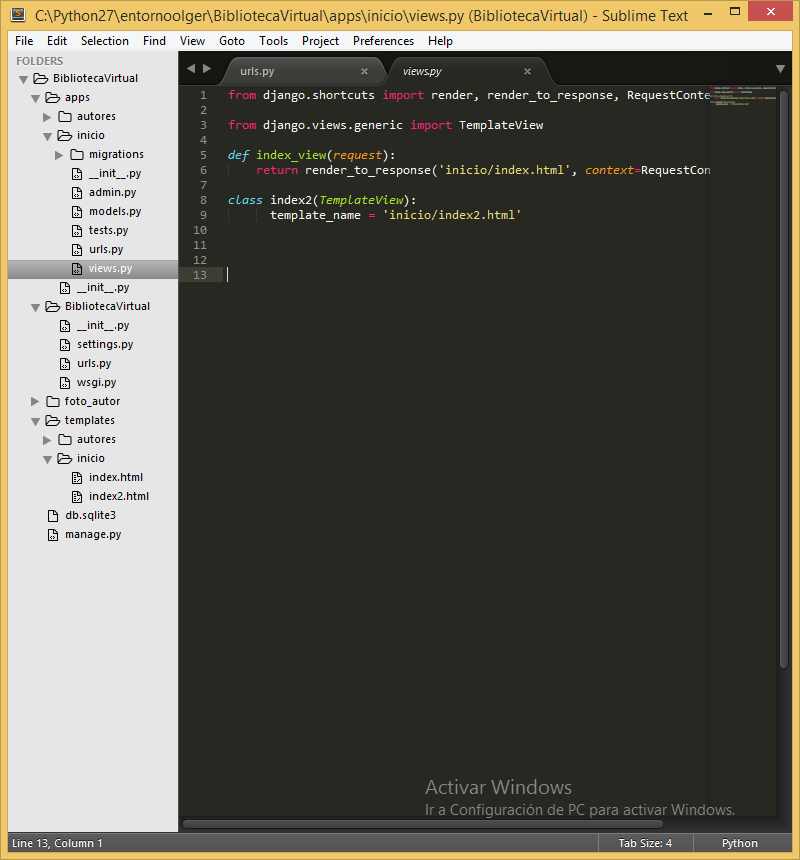




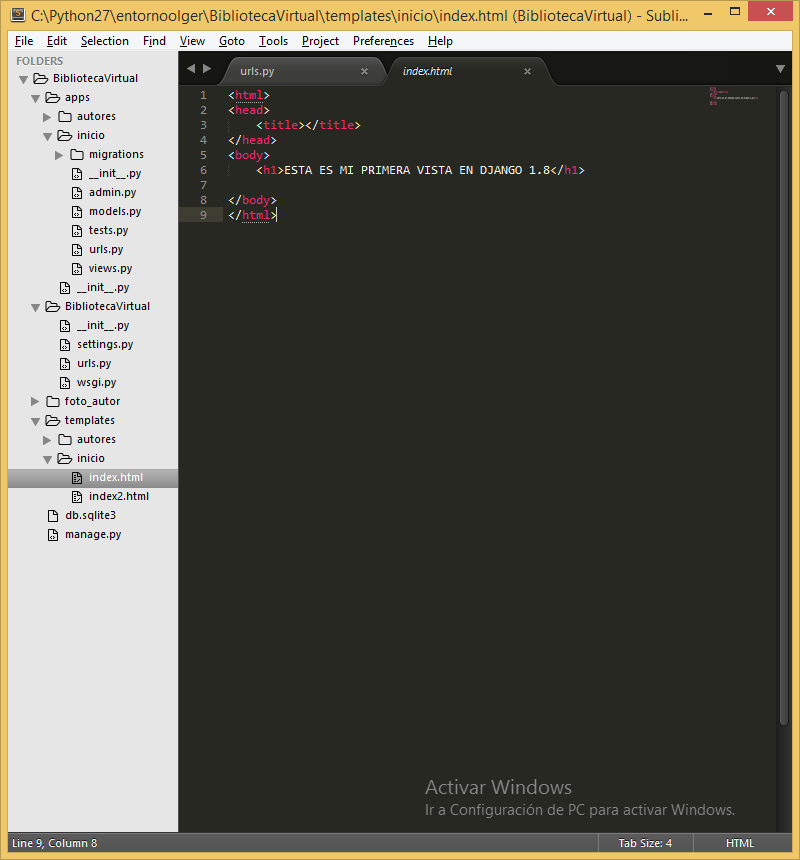
Creamos urls dentro del archivo inicio de nuestro apps y en la aplicación principal le tenemos que dar a conocer que voy a utilizar este archivo esto se hace en el archivo urls

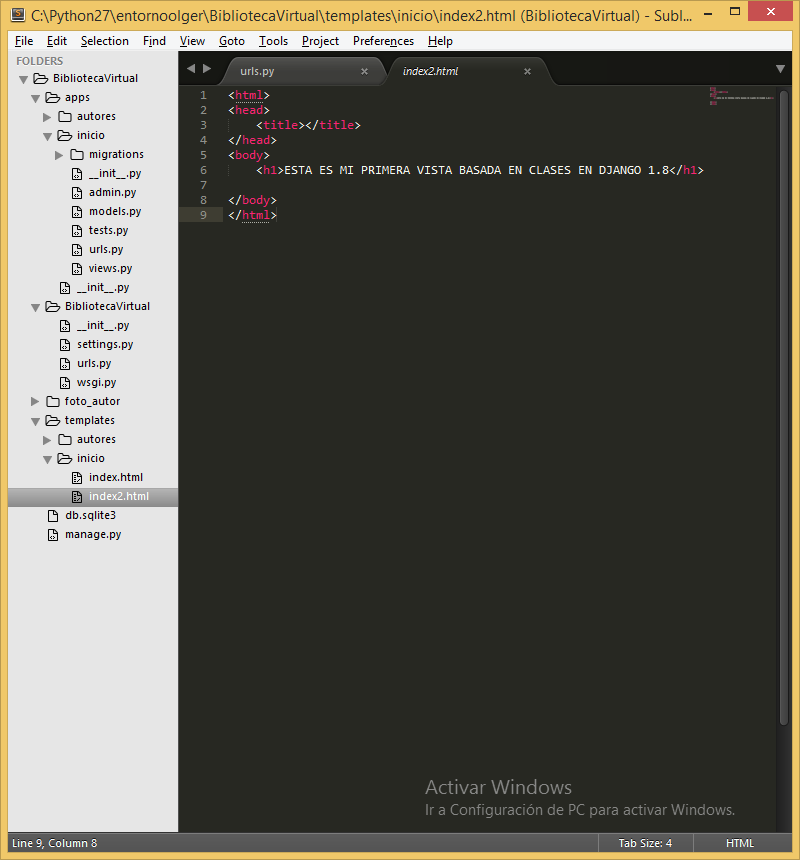


En le views

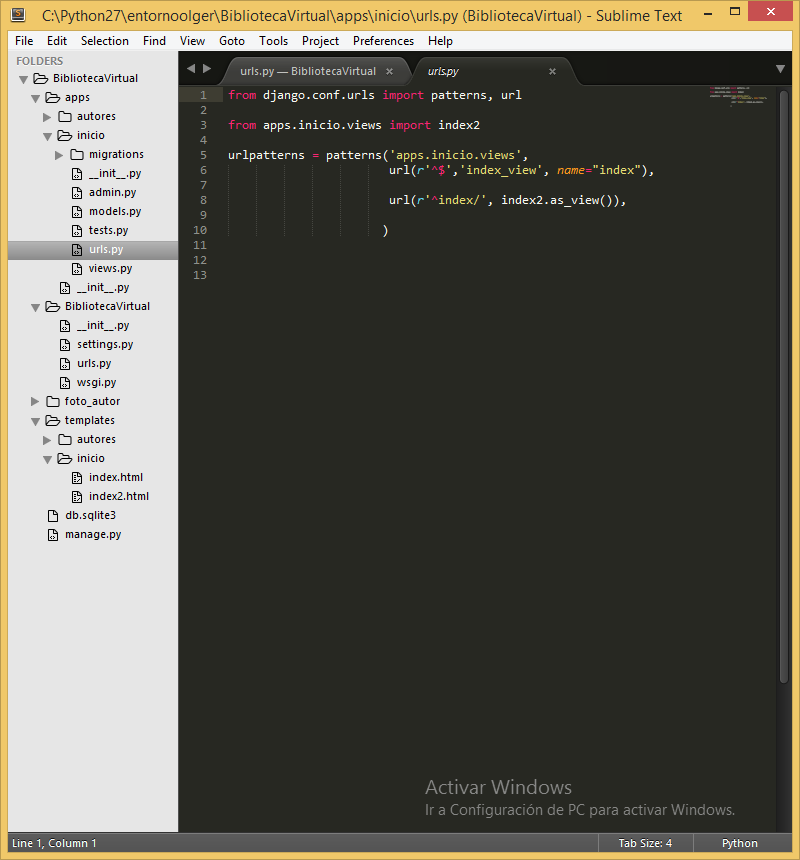


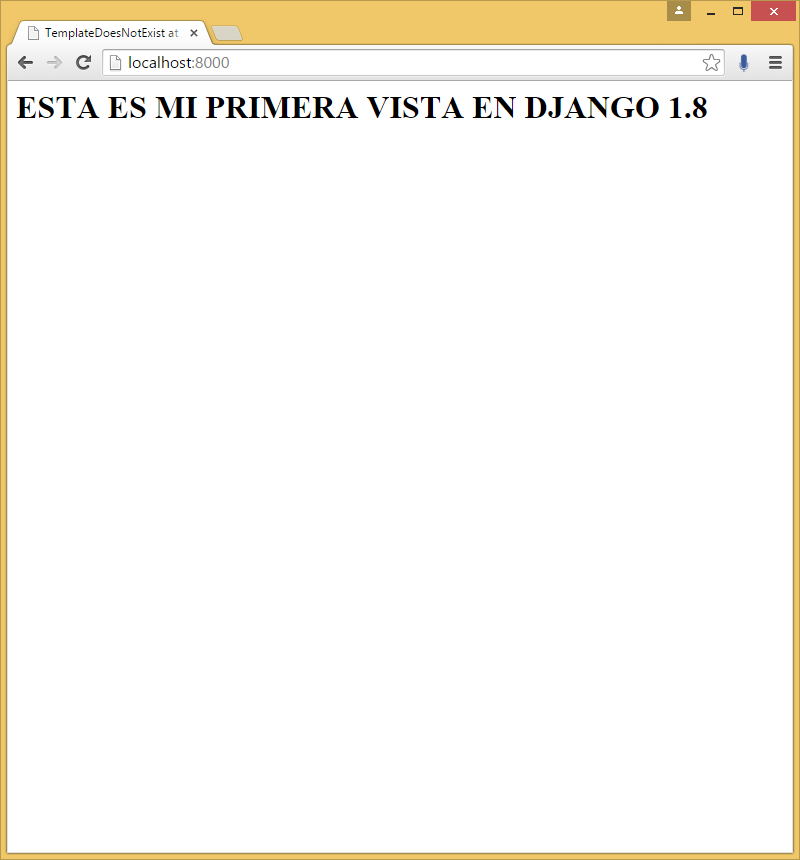
Creamos un archivo html



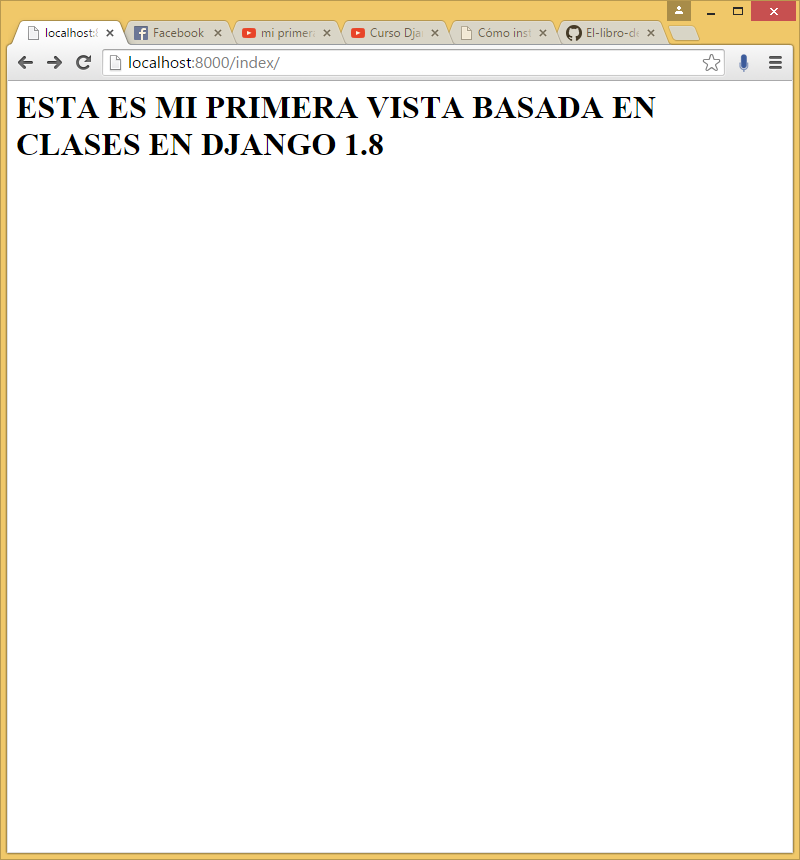


Declaramos nuestro archivo de urls

la vista simple

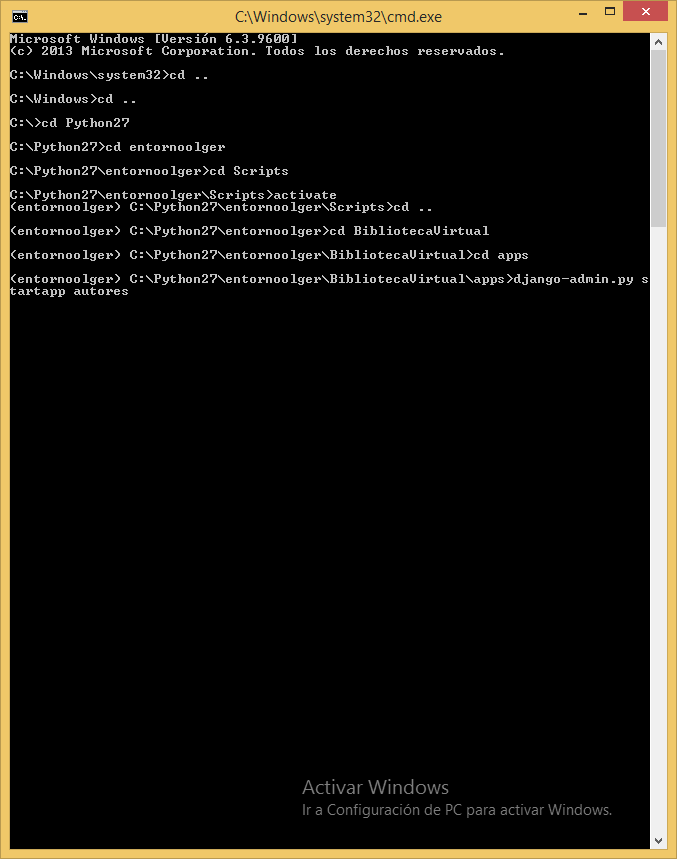


La vista basada en clases con la que trabajaremos

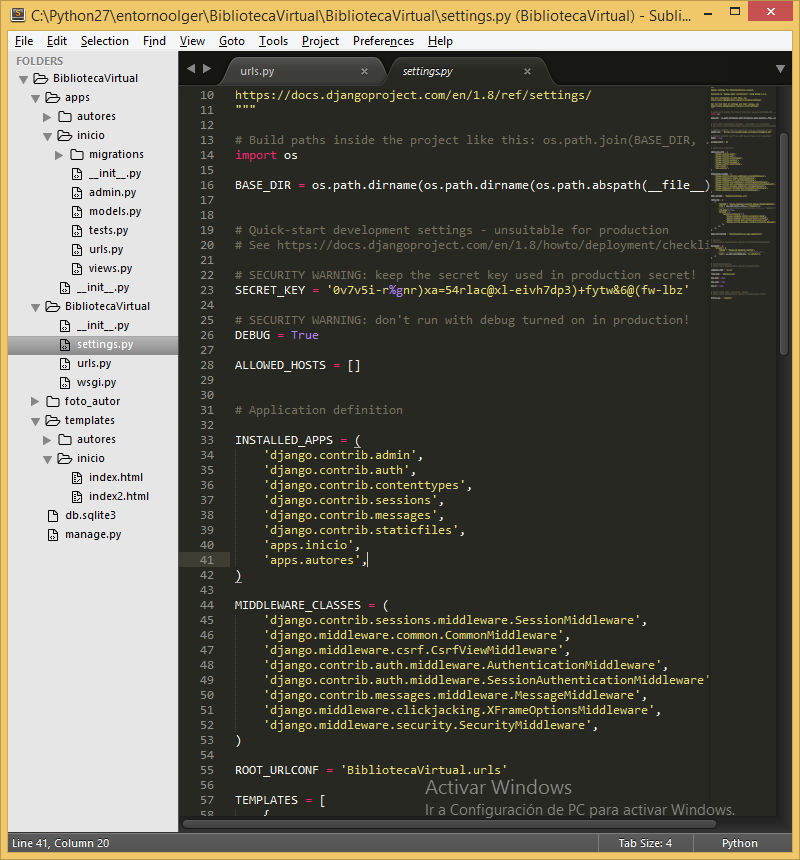


Creando nuestro primer formulario

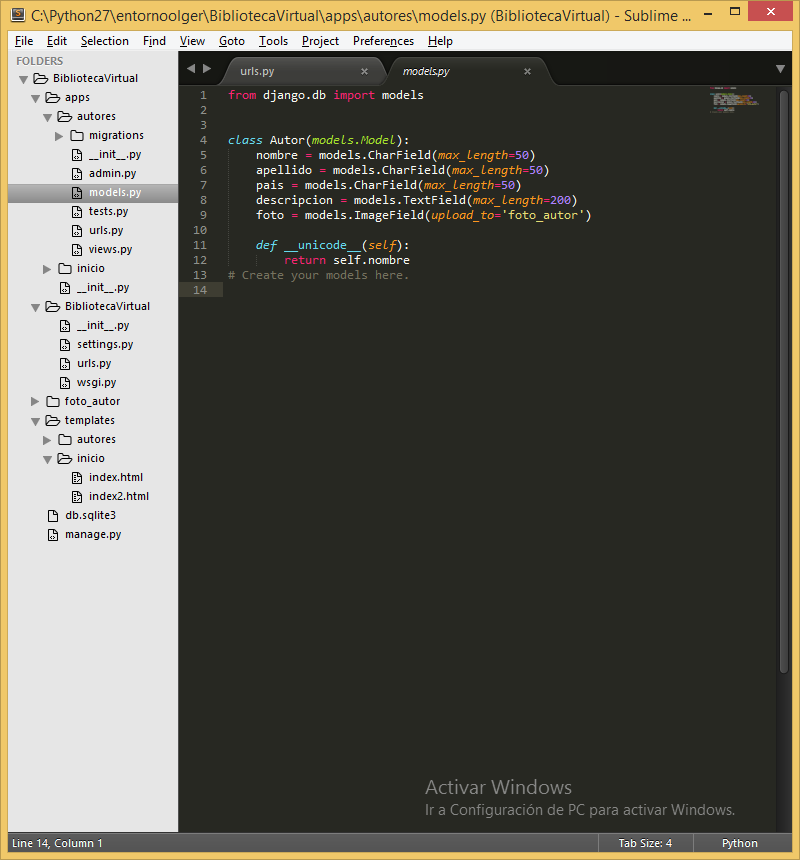
Crearemos una nueva aplicación autores



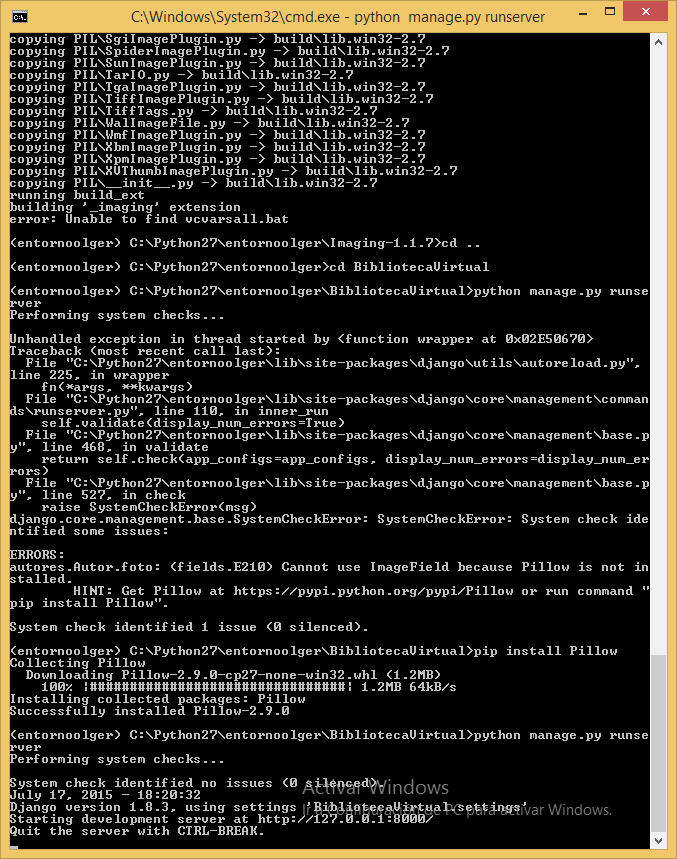
Tengo que darle a conocer que hay un nuevo apps



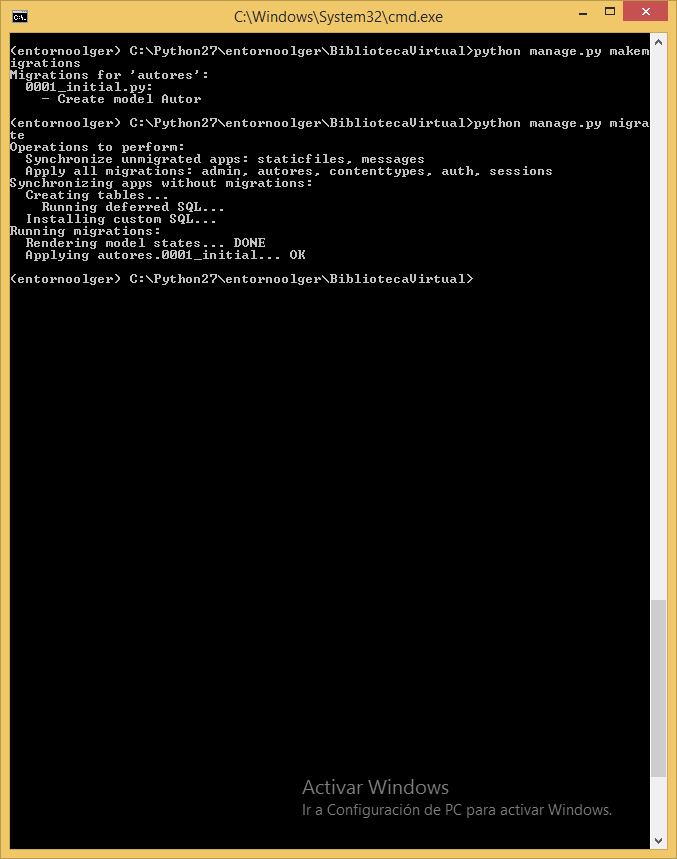
Creamos nuestro primer modelo y nos vamos al archivo models.py



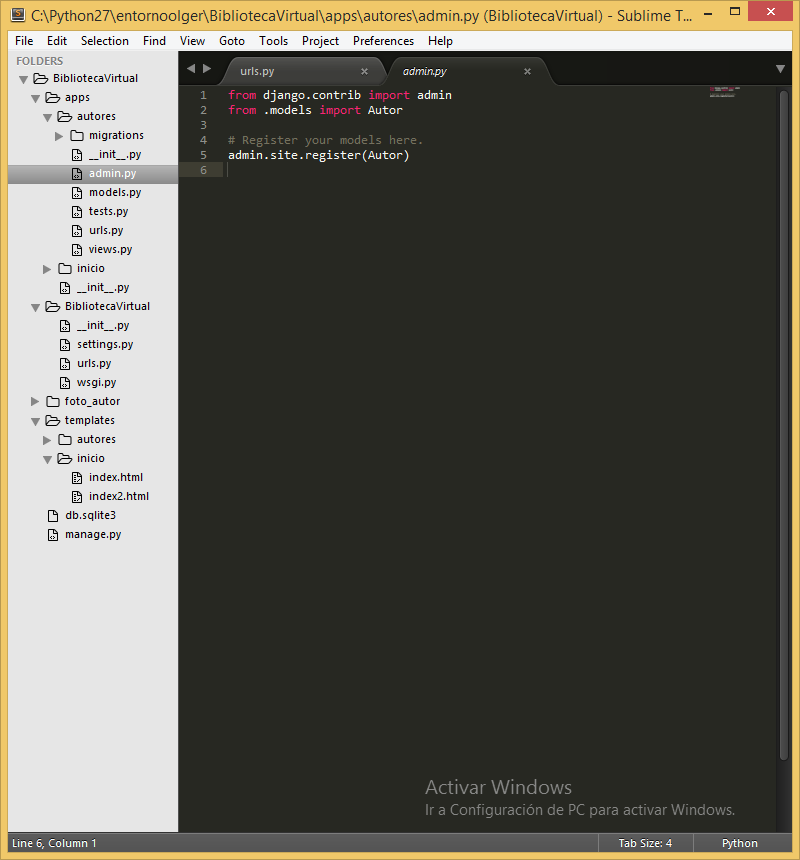
Para poder utilizar el ImageField tenemos que instalar el Pillow que es una librería de imágenes para Python



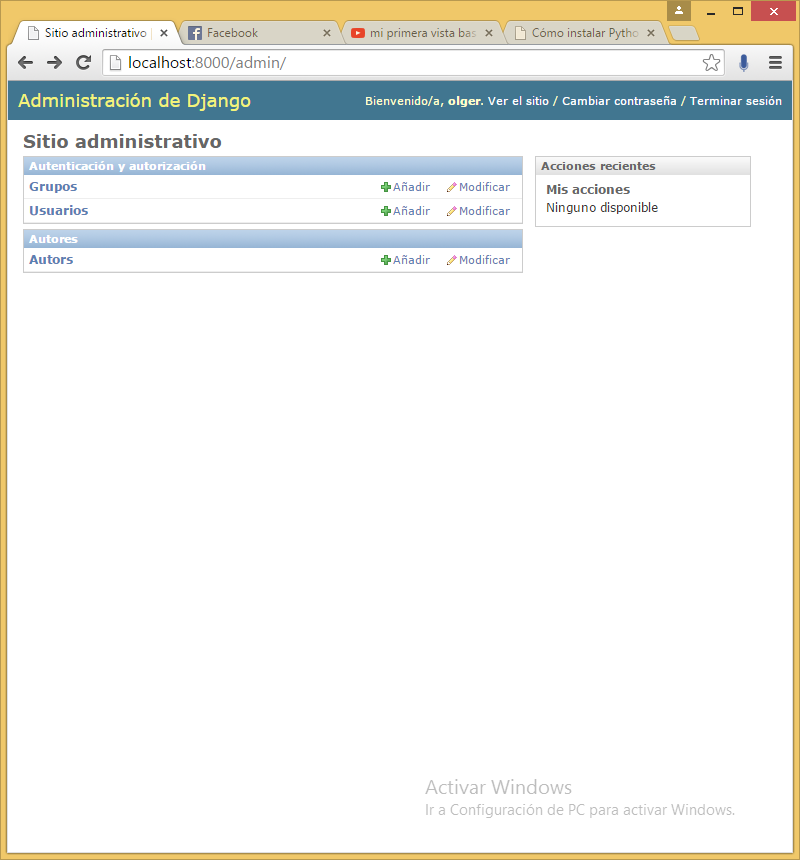
Ahora sincronizaremos la base de datos donde se creara nuestra tabla autores y autor

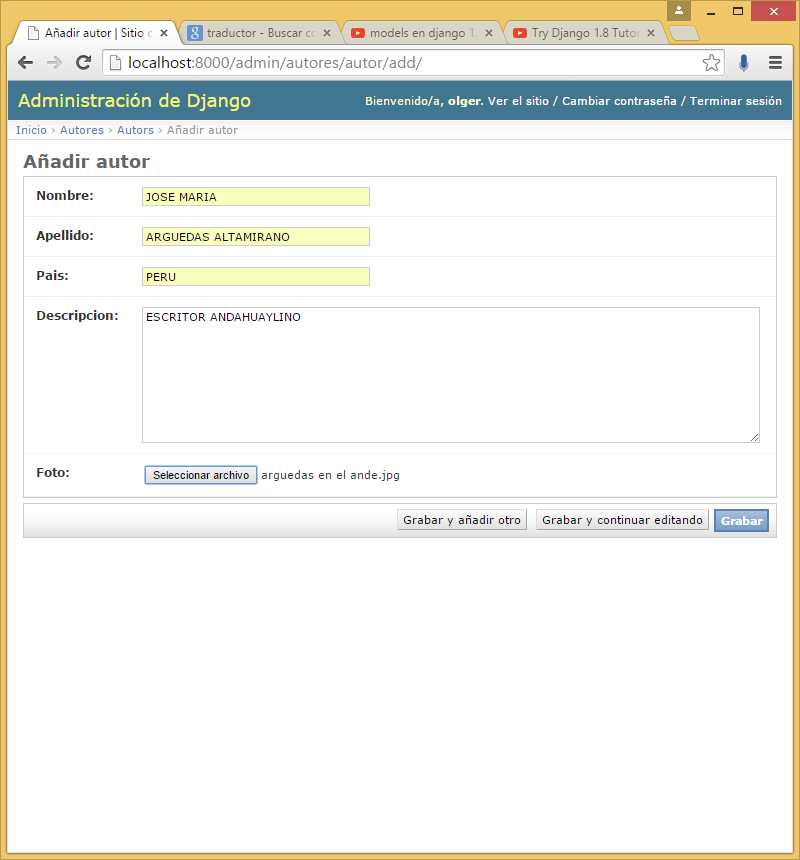


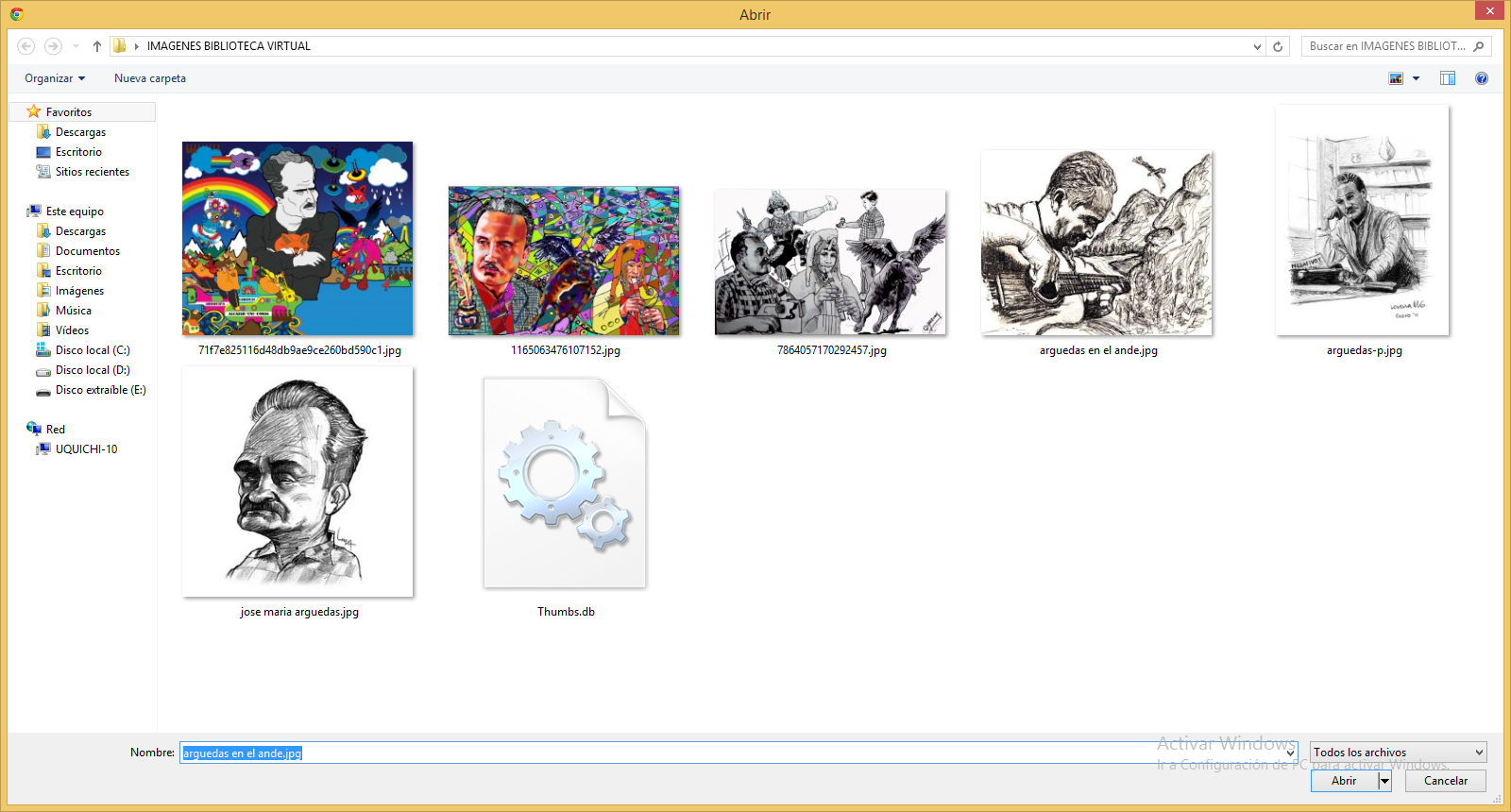
Creamos un archivo admin .py y que nos muestre el nuevo modelo que acabamos de crear



Nos genera un apartado donde hay autores donde está mi tabla autor







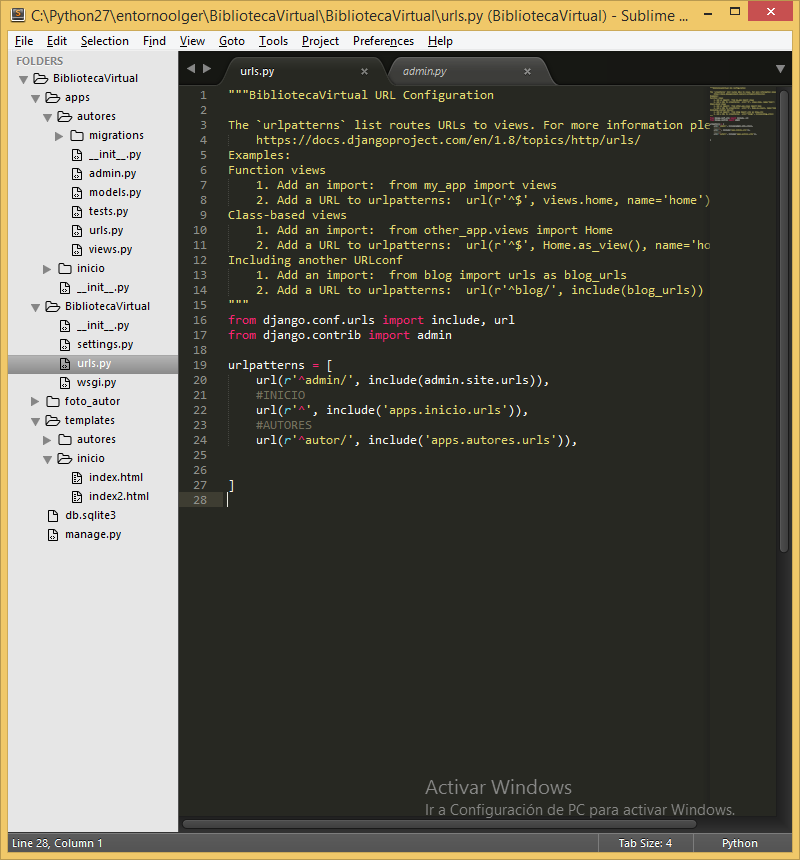
Para mostrar el nombre:





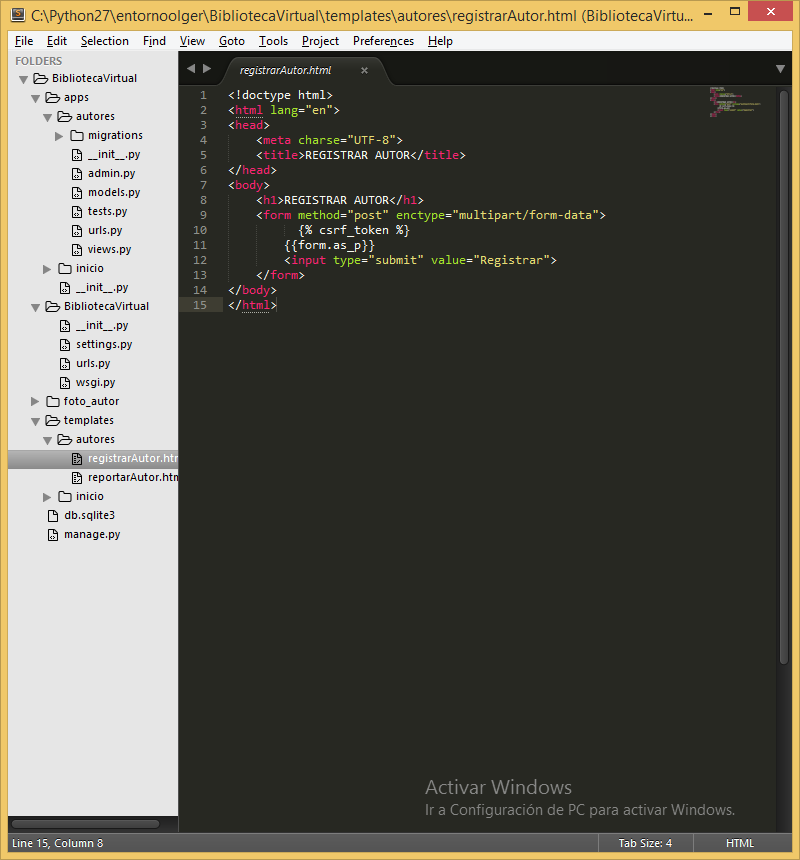
Ahora crearemos nuestro formulario y una vista

Creamos su propio urls pero en nuestro archivo de urls principal le vamos a decir que vamos utilizar esta url.

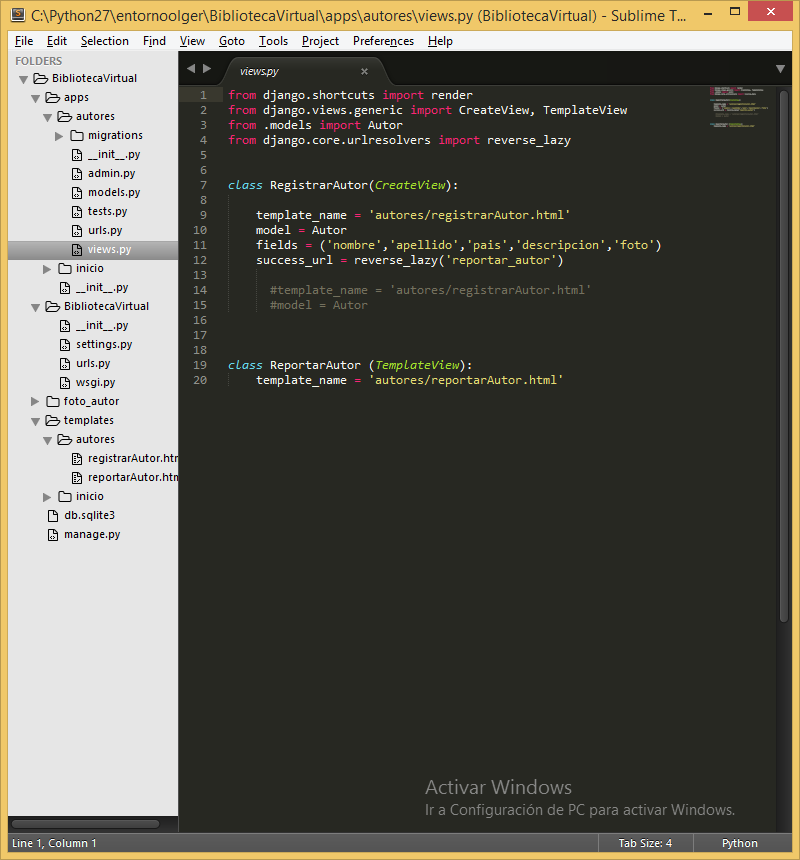


Creamos un templates donde estará el formulario para esto creamos una carpeta autores y dentro de ella un archivo html que se llamara registrarAutor.html

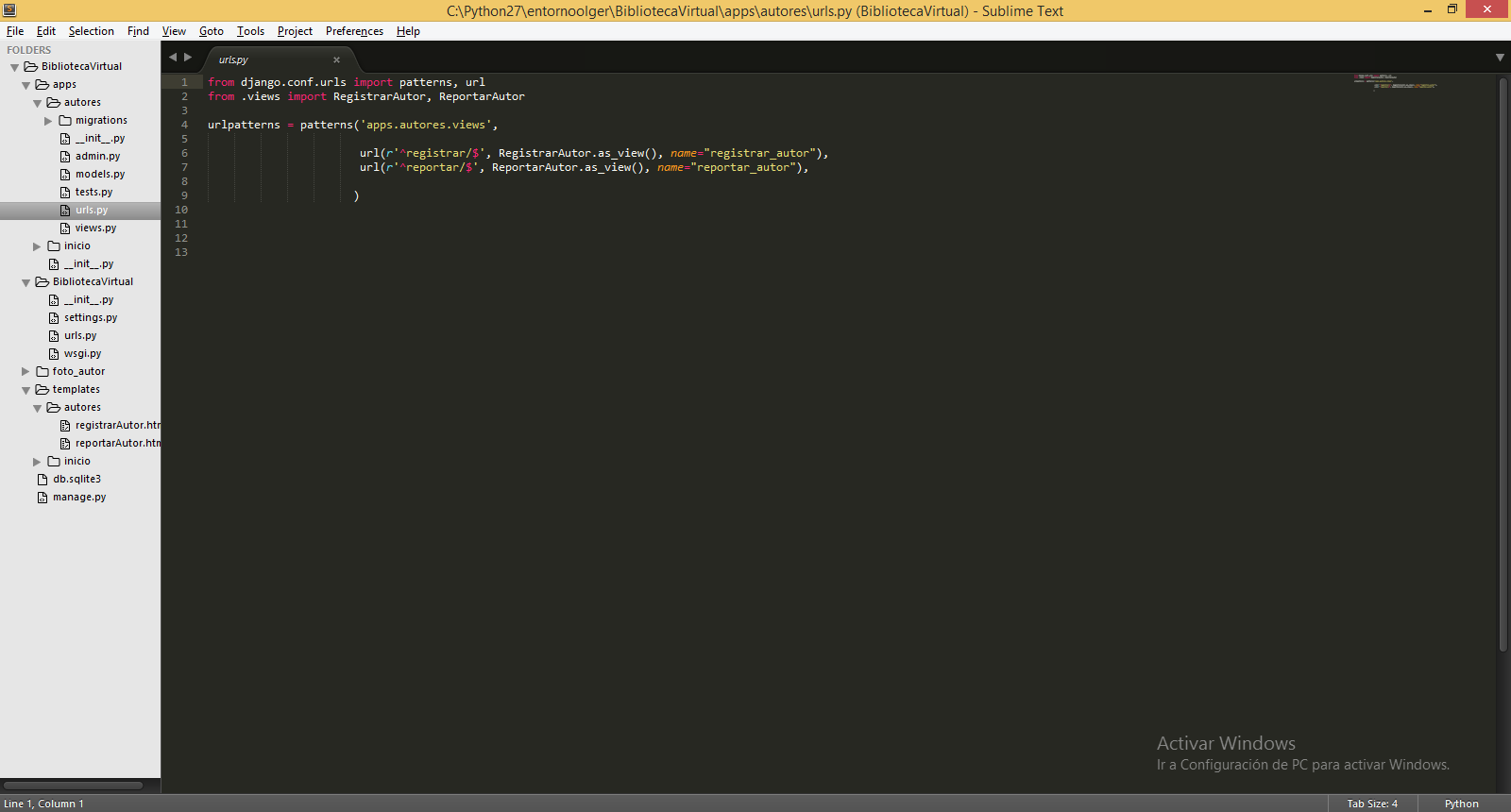
C:\Python27\entornoolger\BibliotecaVirtual\templates\autores



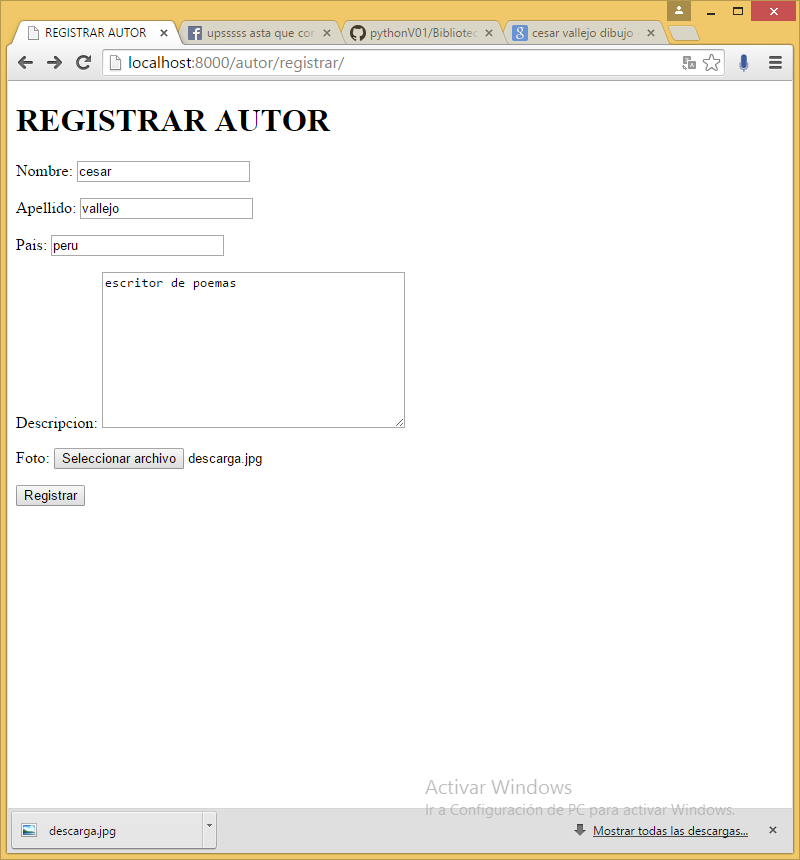
Creamos una vista basada en clases que la vamos a llamar RegistrarAutor C:\Python27\entornoolger\BibliotecaVirtual\apps\autores



Ahora pasamos a crear una urls para la vista



Entonces entramos a la url que pusimos



Asemos click en registrar y nos mostrara otra página que significa que se agrego satisfactoriamente

