

**MODALIDAD PRESENCIAL**

**ÁREA TÉCNICA**

**SISTEMAS INFORMÁTICOS Y COMPUTACIÓN**

ARQUITECTURA DE APLICACIONES

**SISTEMA DE REGISTRO Y GESTIÓN DE FICHAS DE MOVIMIENTOS DE MASA “SPRINTS”**

**INTEGRANTES**

* Roosevelt Javier Jaramillo Arias
* Silvana Patricia Vélez Macas

**FECHA:** 03-02-2019

**DOCENTE**: Mgs. Luis Santiago Quiñones Cuenca

AÑO 2018

**1 Introducción:**

El aplicativo web es un software que permite a nuestros Docentes y Alumnos realizar el almacenamiento y la consulta de los datos registrados . El principal objetivo es poder mostrar la información obtenida de manera clara e intuitiva.

**2 Pantalla autenticación:**

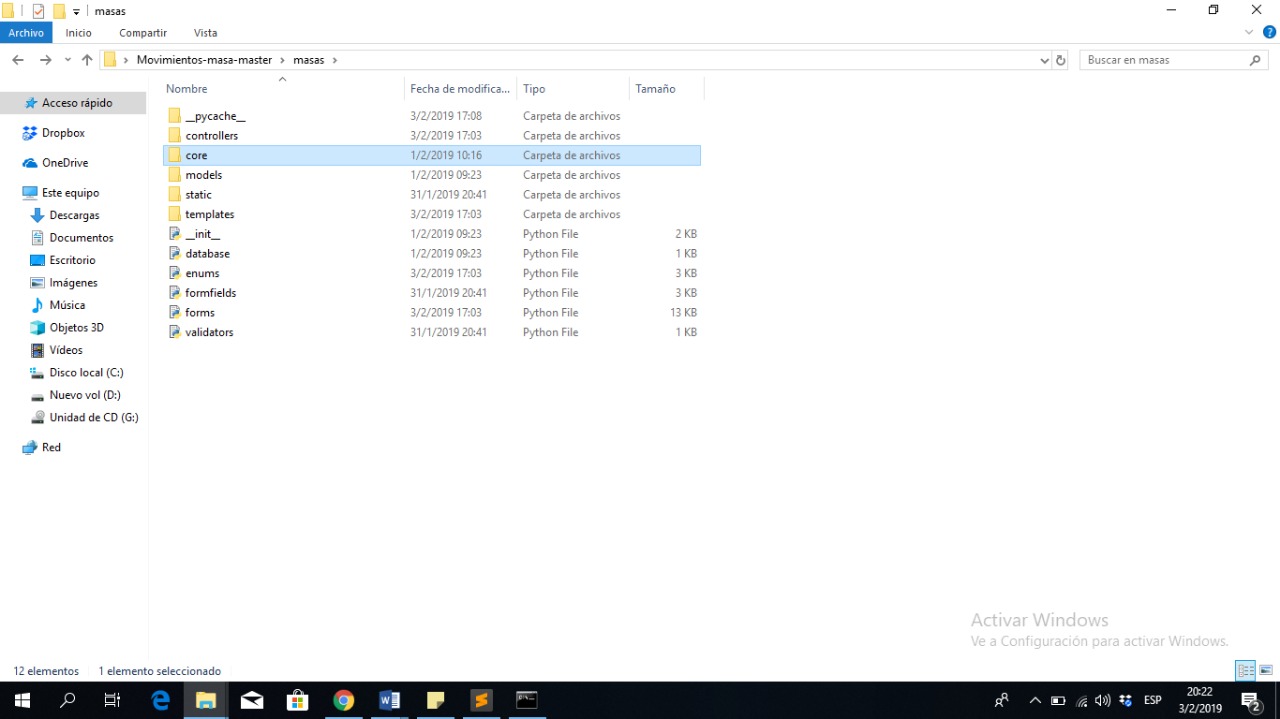
Para poder entrar en la aplicación es obligatorio identificarse, para ello es necesario introducir el usuario y la contraseña.

**Descripción De La Arquitectura Que Se Utilizo En El Aplicativo**

En esta pantalla se observa la arquitectura que se utilizo, en este caso es la arquitectura limpia se  basa en los cuatro conceptos introducidos anteriormente y alinea el proyecto con las mejores prácticas, como el  principio de inversión de dependencia o los  casos de uso . También apunta a una independencia máxima de cualquier marco o herramienta que pueda permanecer en el camino de la capacidad de prueba de la aplicación o su reemplazo.

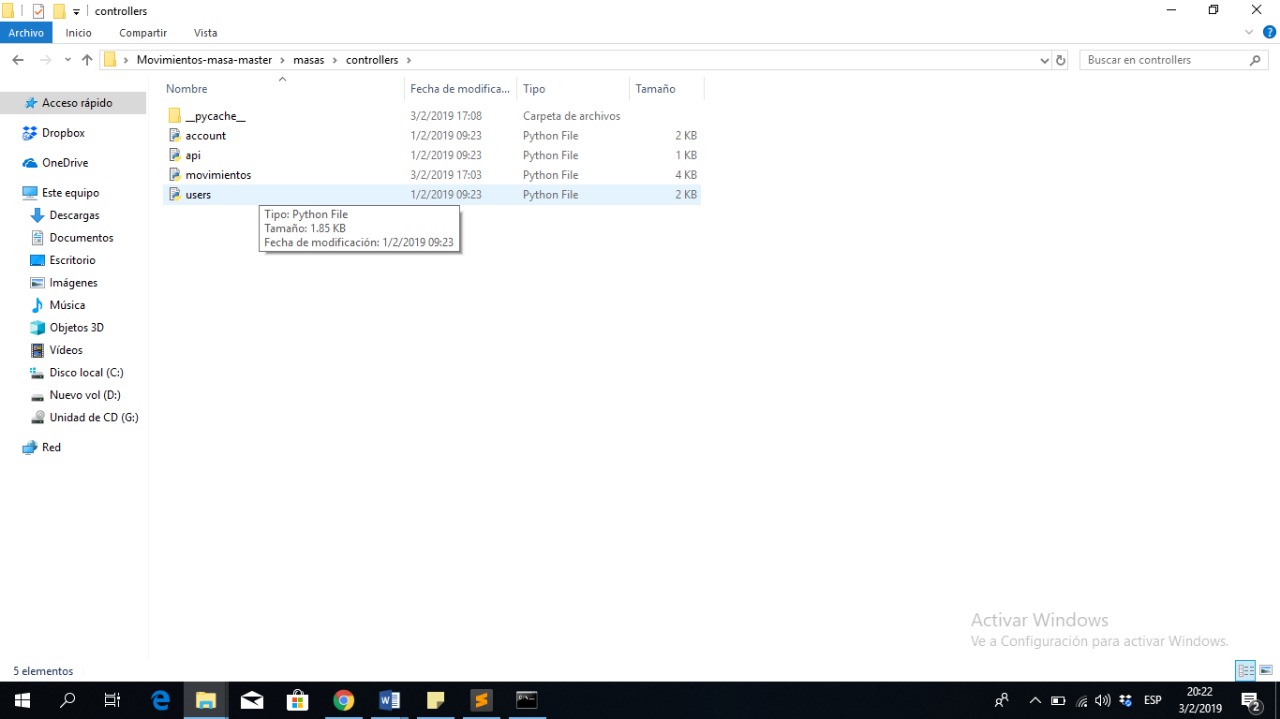
Aquí nos divide los

* controllers
* Core
* Model
* template



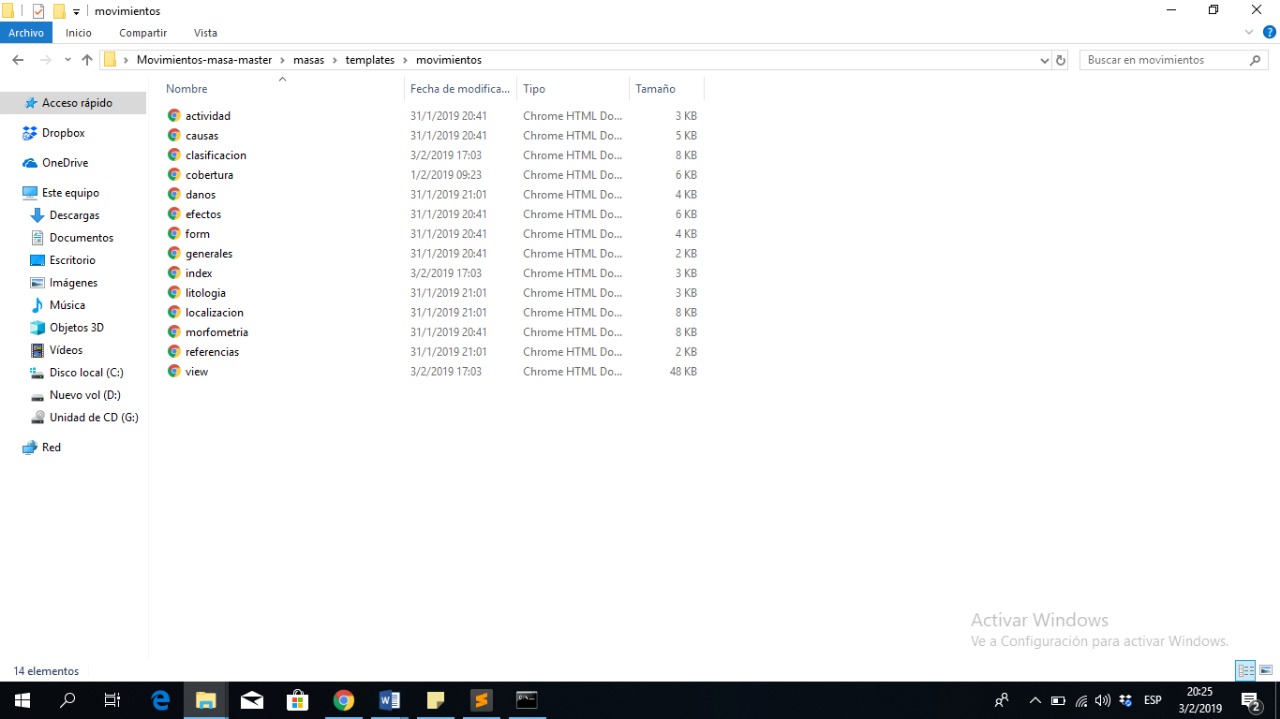
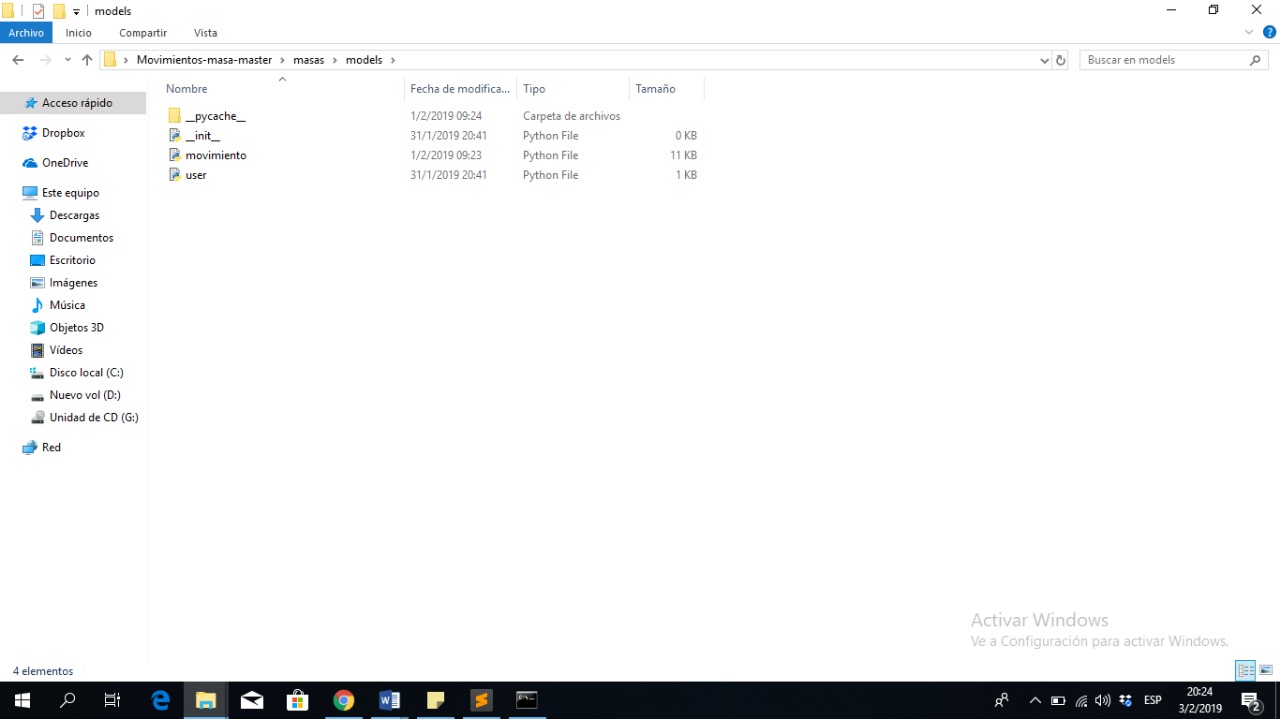
**Casos De Usos**

Aquí encontramos los archivos de los usuarios que se encuentran en la carpeta Controllers.



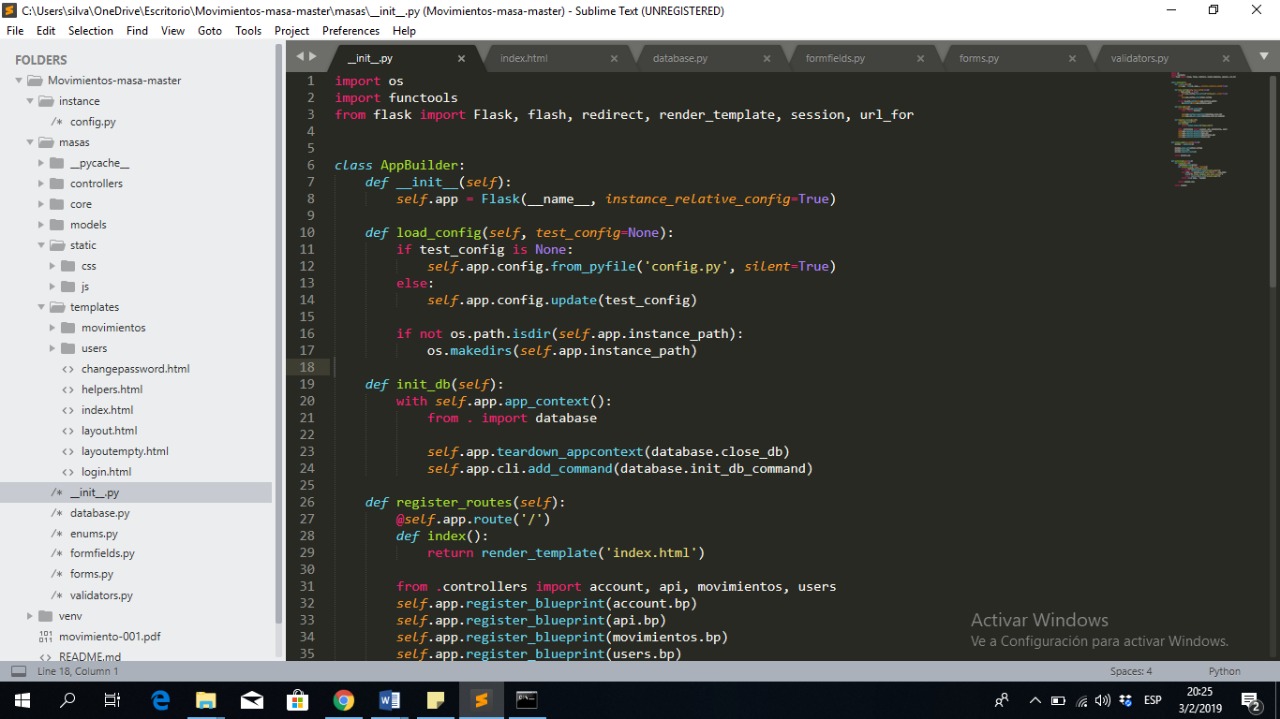
**Models**

Ensta ficha se encuentran las vistas de todas las vistas.



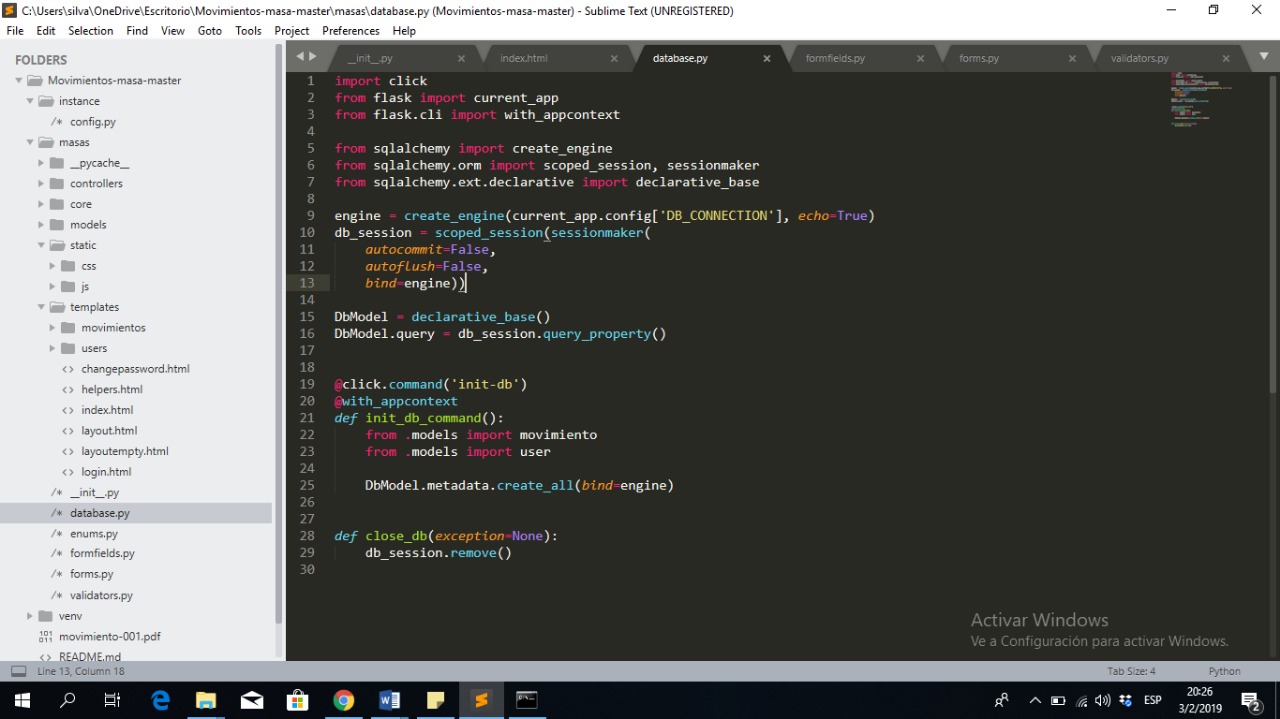
**Patrones de Creacionales que se utilizaron en el proyecto:**

En esta vista se emplea el patron Builder es usado para permitir la creación de una variedad de objetos complejos desde un objeto fuente el objeto fuente se compone de una variedad de partes que contribuyen individualmente a la creación de cada objeto complejo a través de un conjunto de llamadas a interfaces comunes de la clase Abstract Builder.



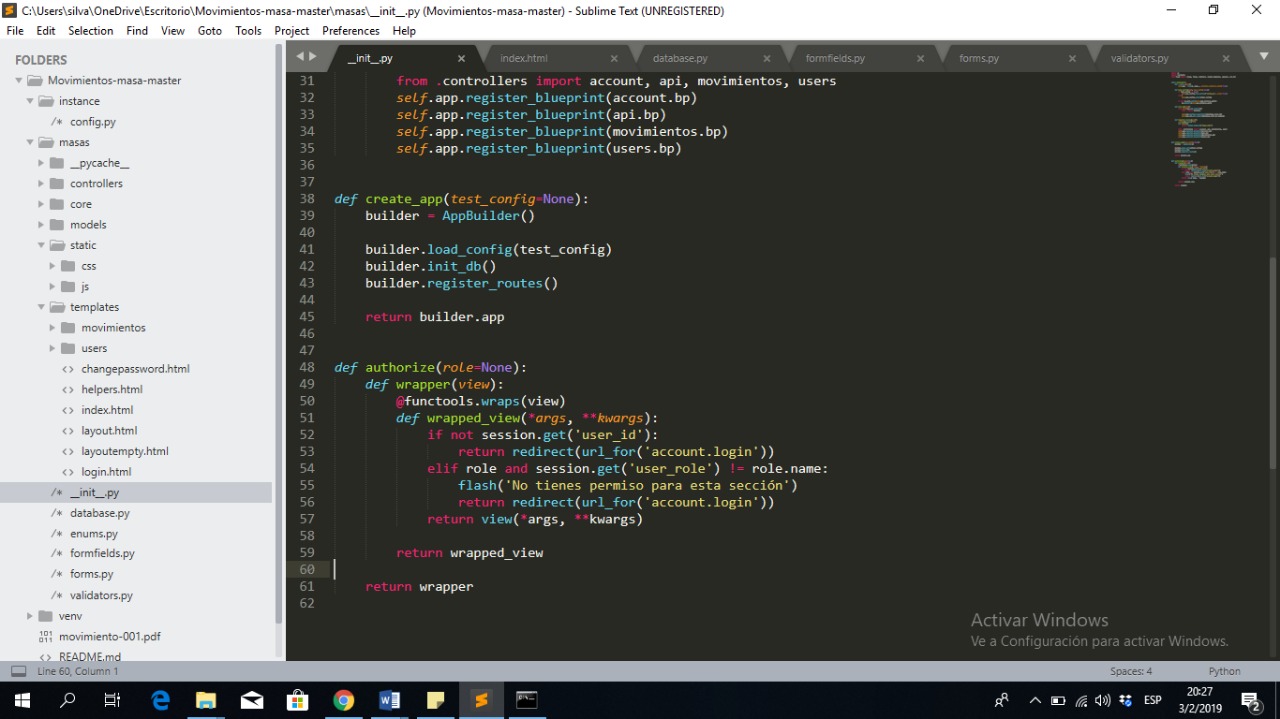
**Singleton**

En esta vista tambien se ocupa *singleton*  es un [patrón de diseño](https://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o) que permite restringir la creación de [objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) pertenecientes a una [clase](https://es.wikipedia.org/wiki/Clase_(programaci%C3%B3n_orientada_a_objetos)) o el valor de un tipo a un único objeto.



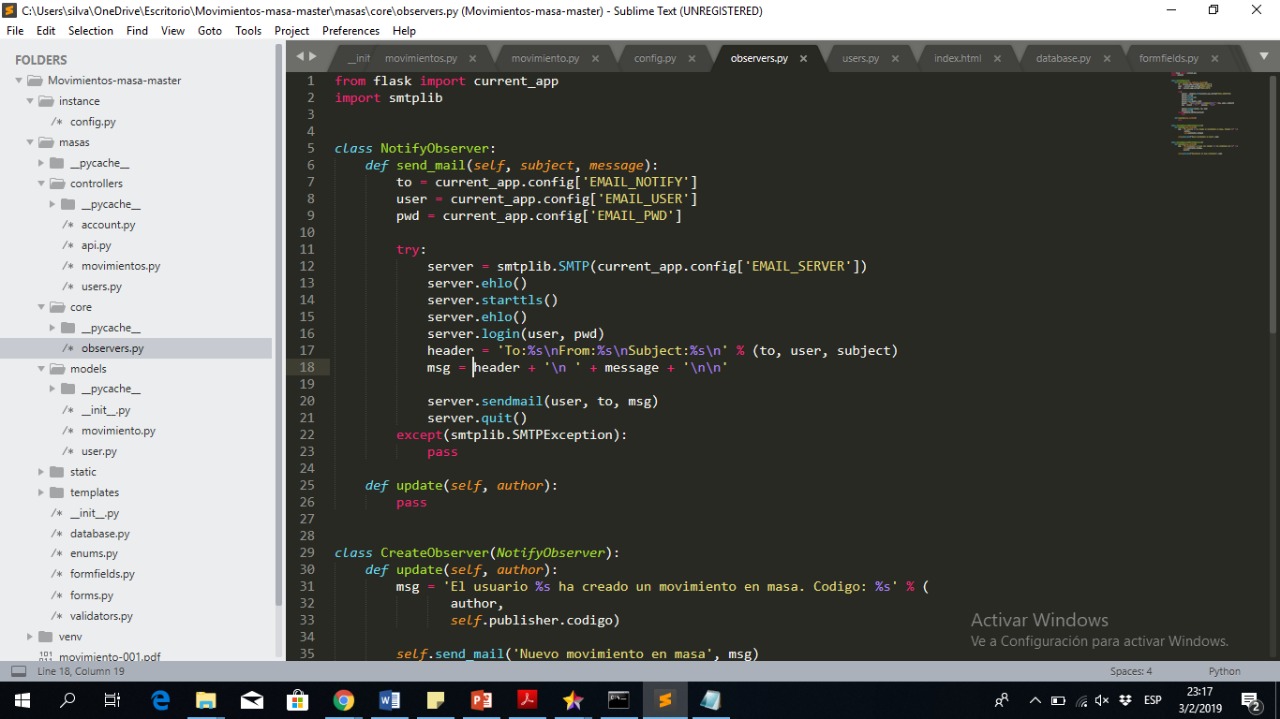
**Patron estructural Decorator**

Asi mismo se emplea el patron decorator para responde a la necesidad de añadir dinámicamente funcionalidad a un Objeto. Esto nos permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera.

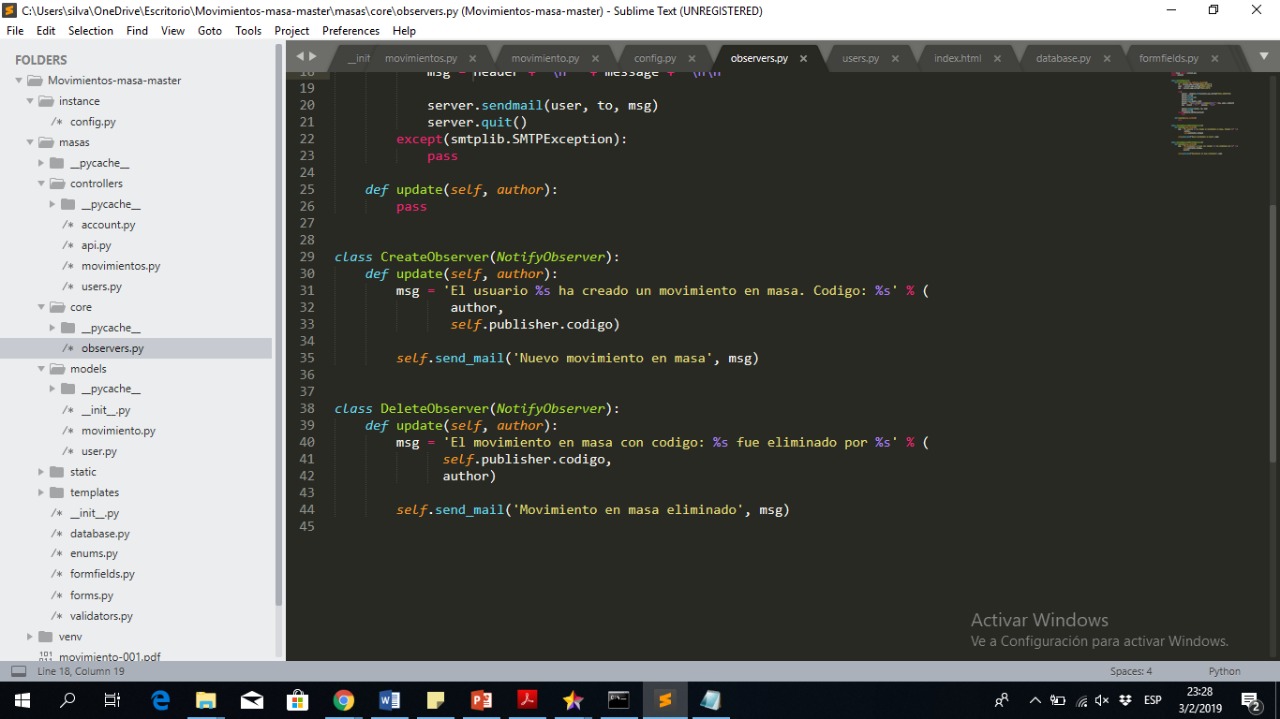


**Patron de comportamiento observer notificdor**

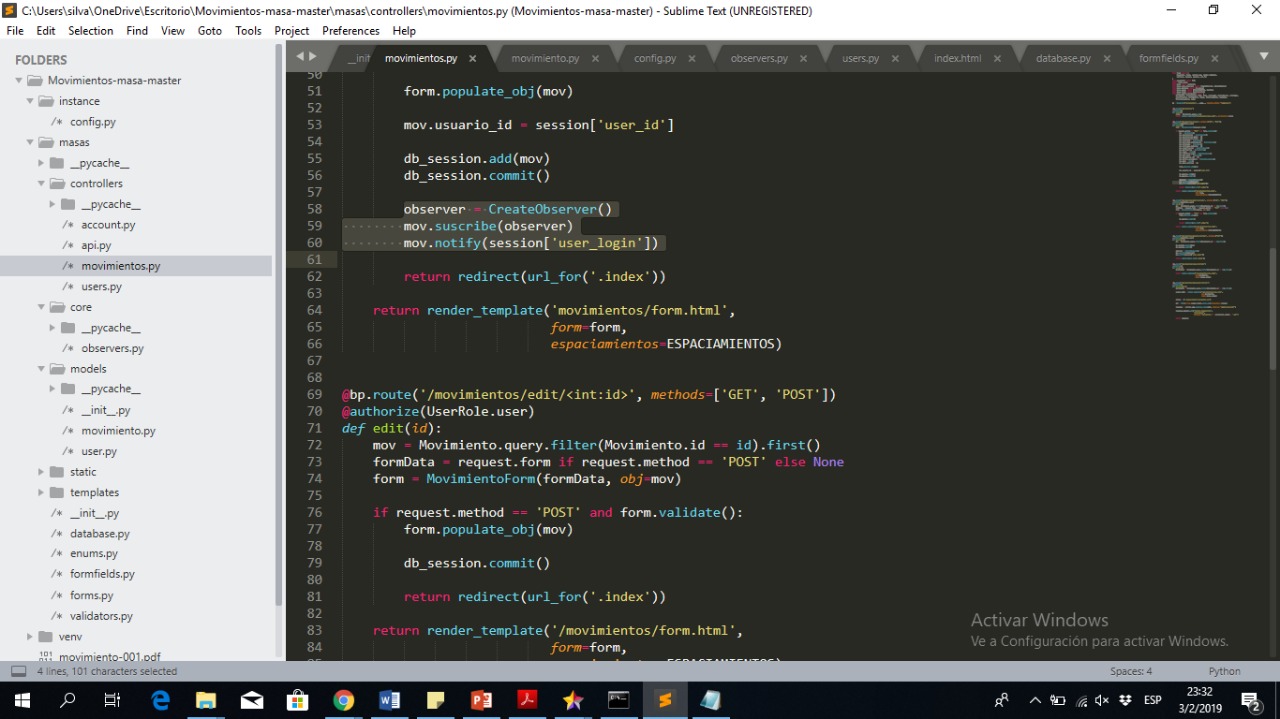
cuando el estado de uno de ellos cambia, todos los objetos dependientes son notificados y actualizados de forma automática”.



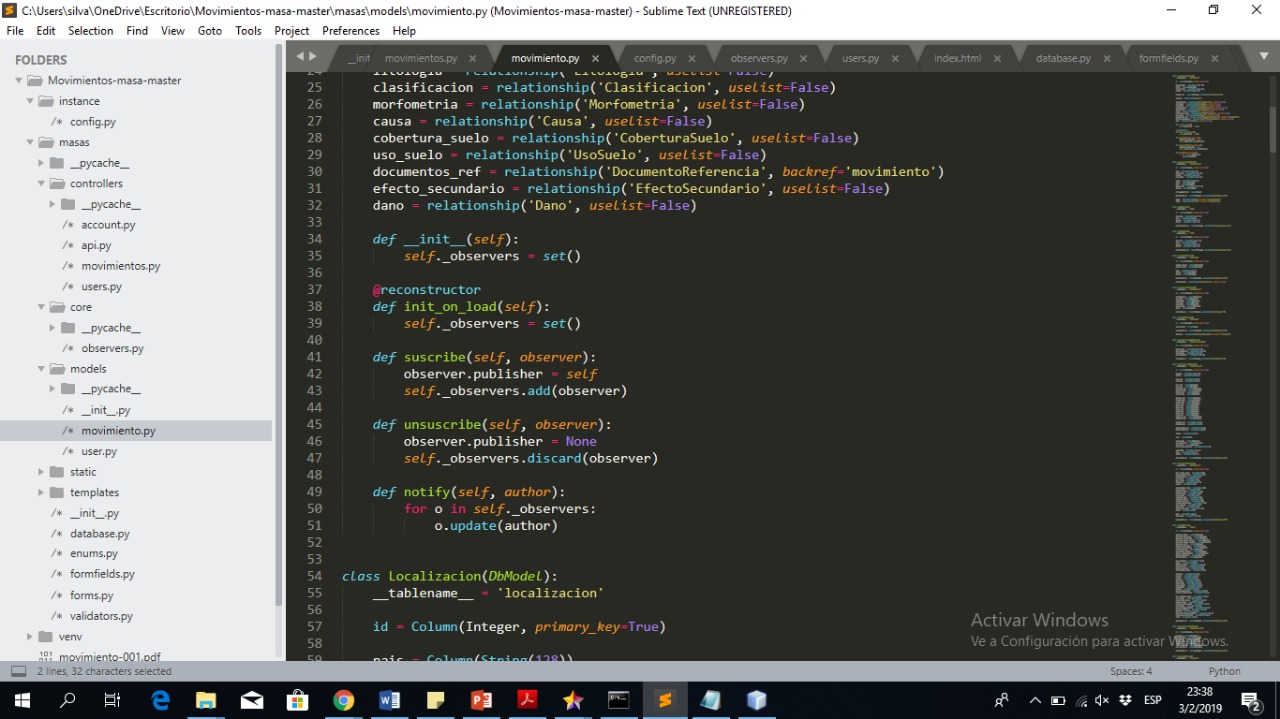
**Observador Al Crear Y Al Eliminar**



**Observador**

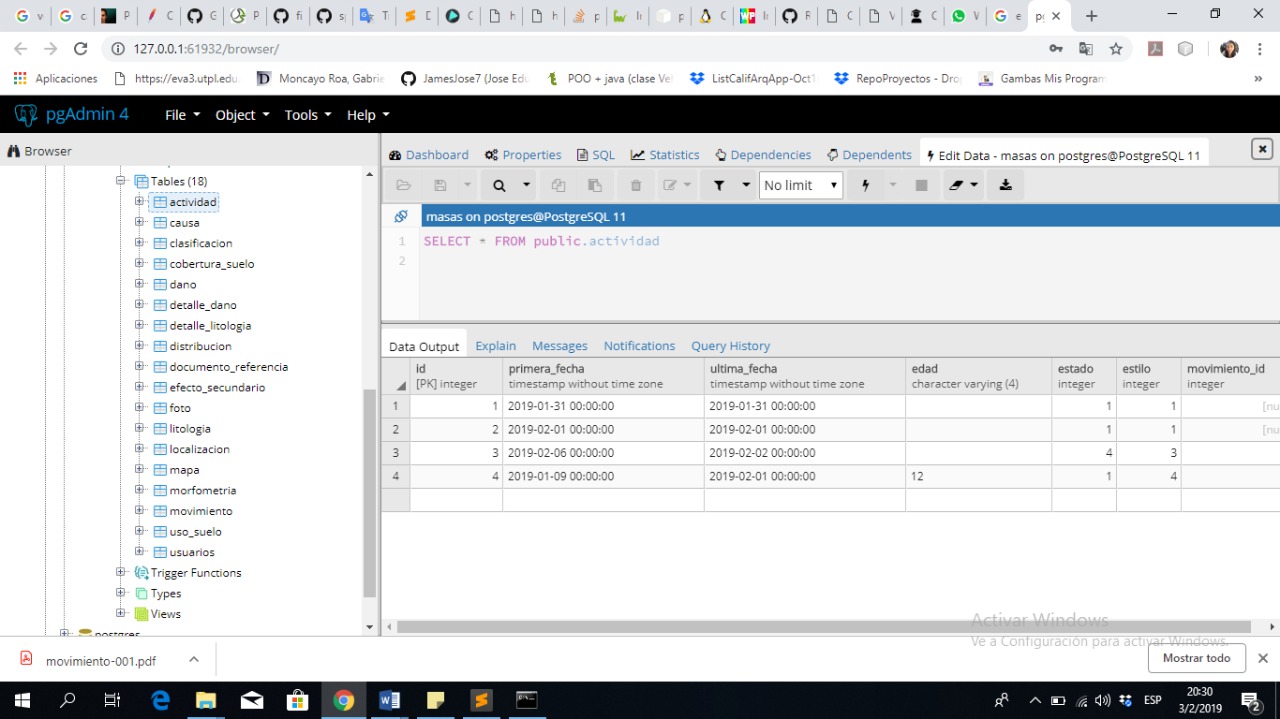


**SUBJECT DEL OBSERVADOR**

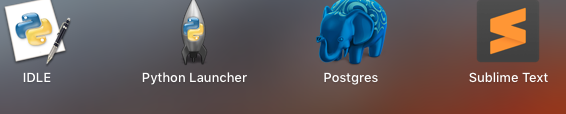


**Finalmente se llega a la base de datos**

En esta pantalla se presenta todos la Data almacenda de todas las fichas.

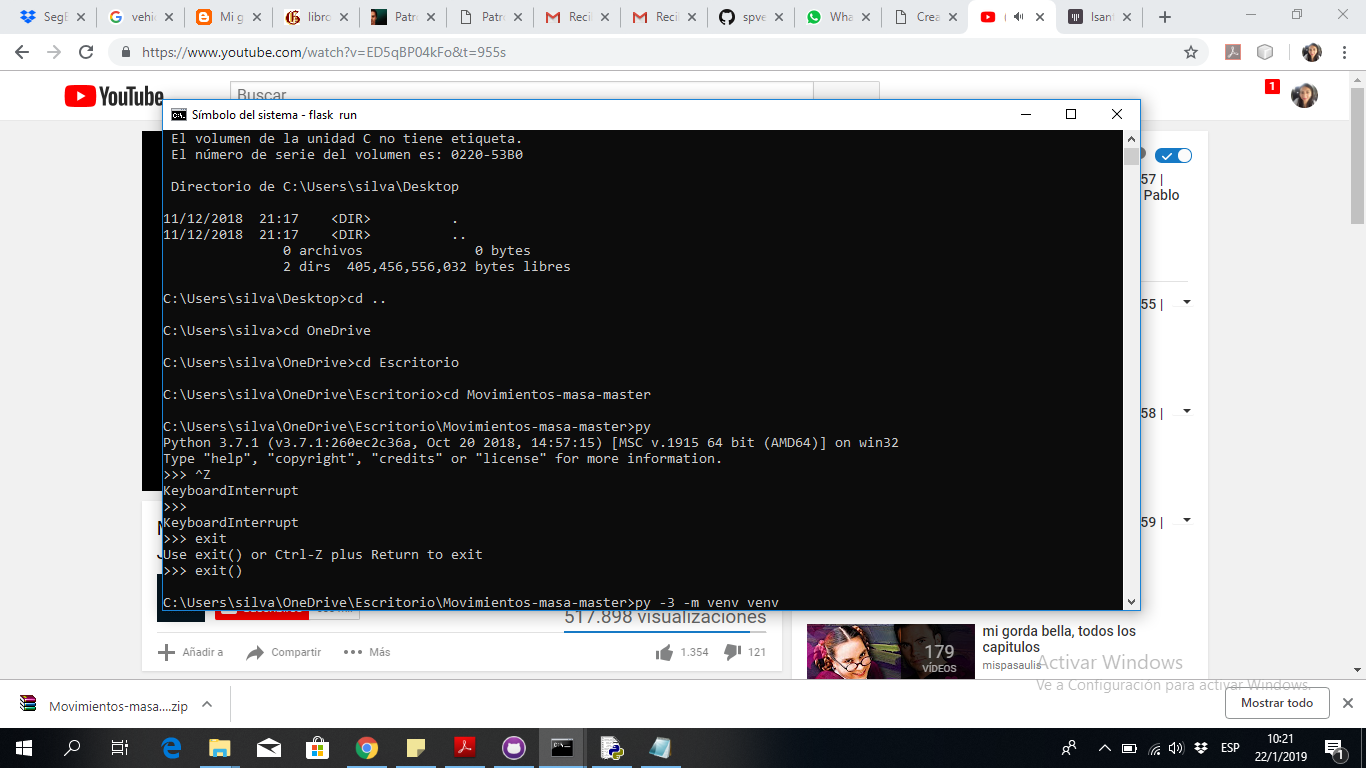


INSTALCION DE HERRAMIENTAS Python,Postgres y Frameworks flask



INSTALACION DEL FRAMEWORK FLAKS

### **1.Crear un entorno**

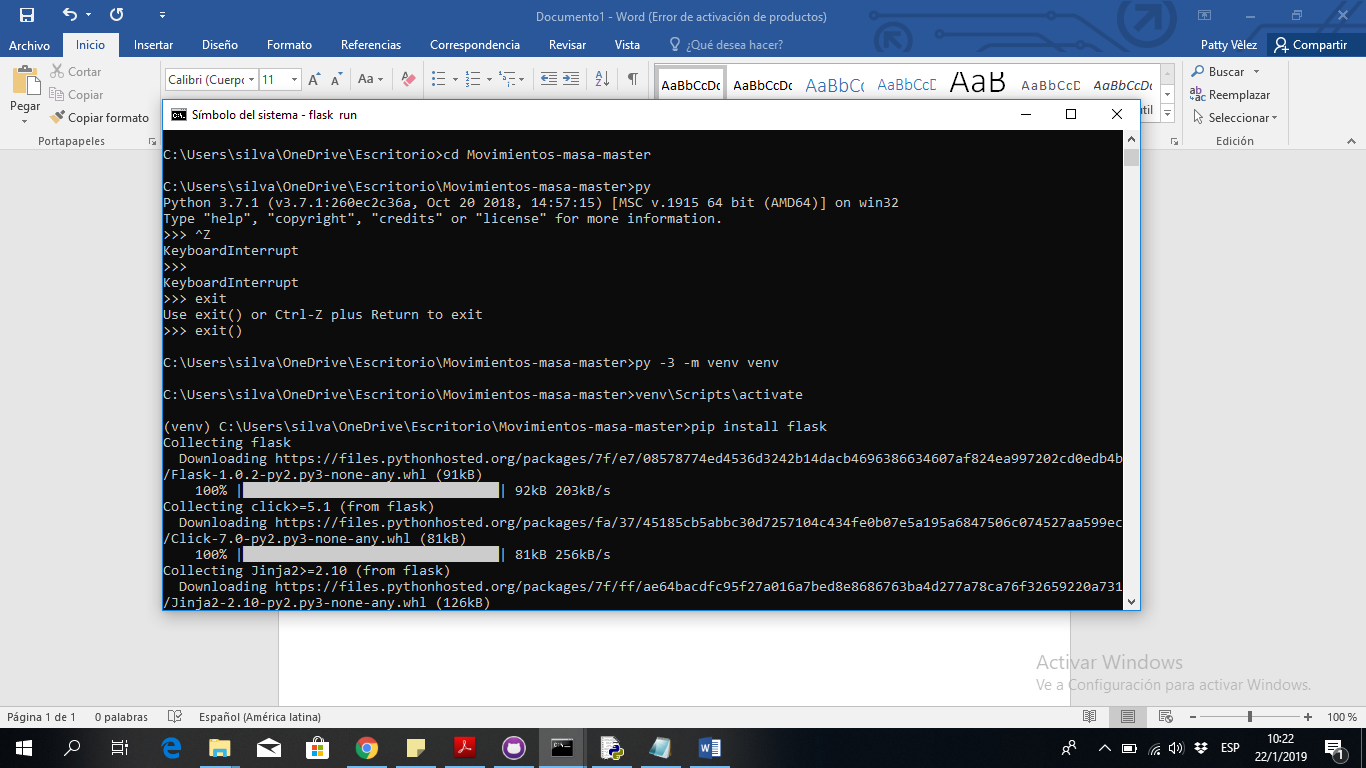


Cree una carpeta de proyecto y una venvcarpeta dentro de:

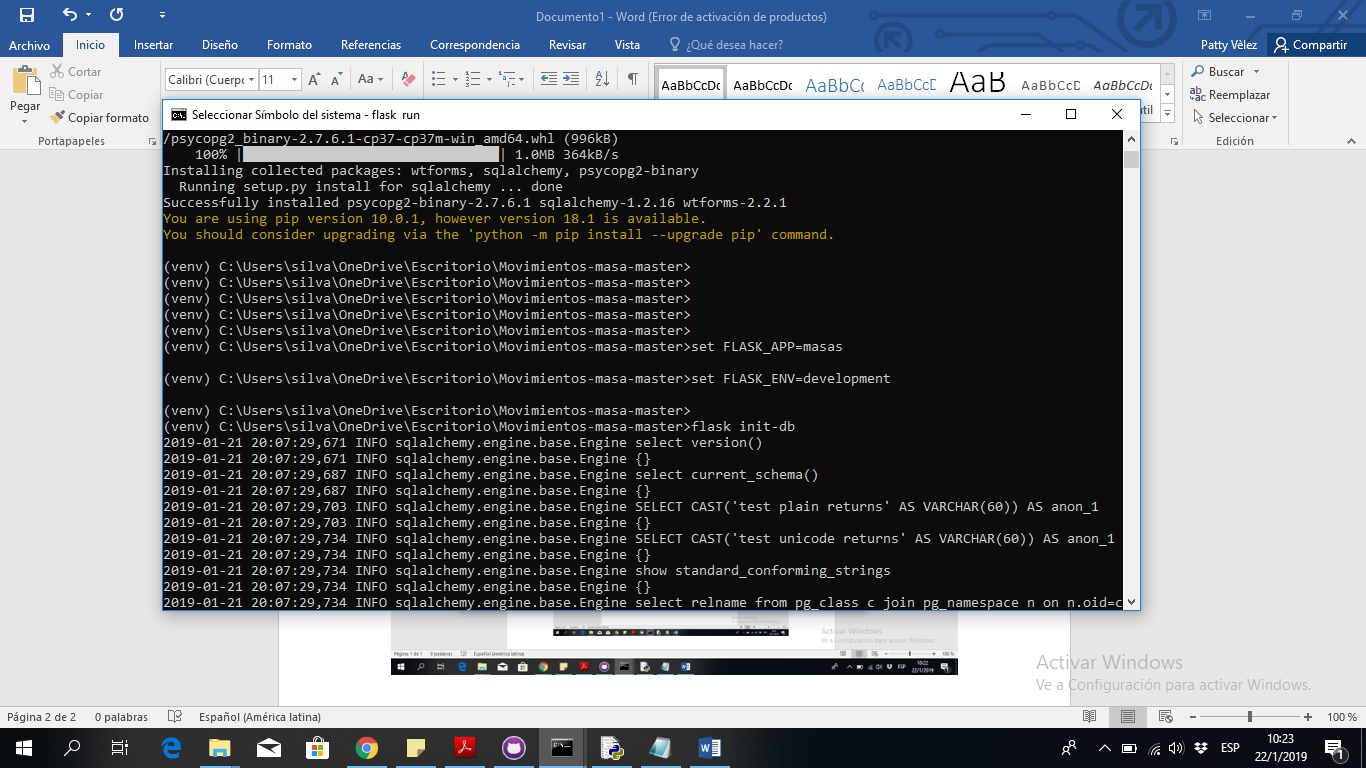
* mkdir myproject
* cd masas
* python3 -m venv venv

### **2.Ejecuta después**

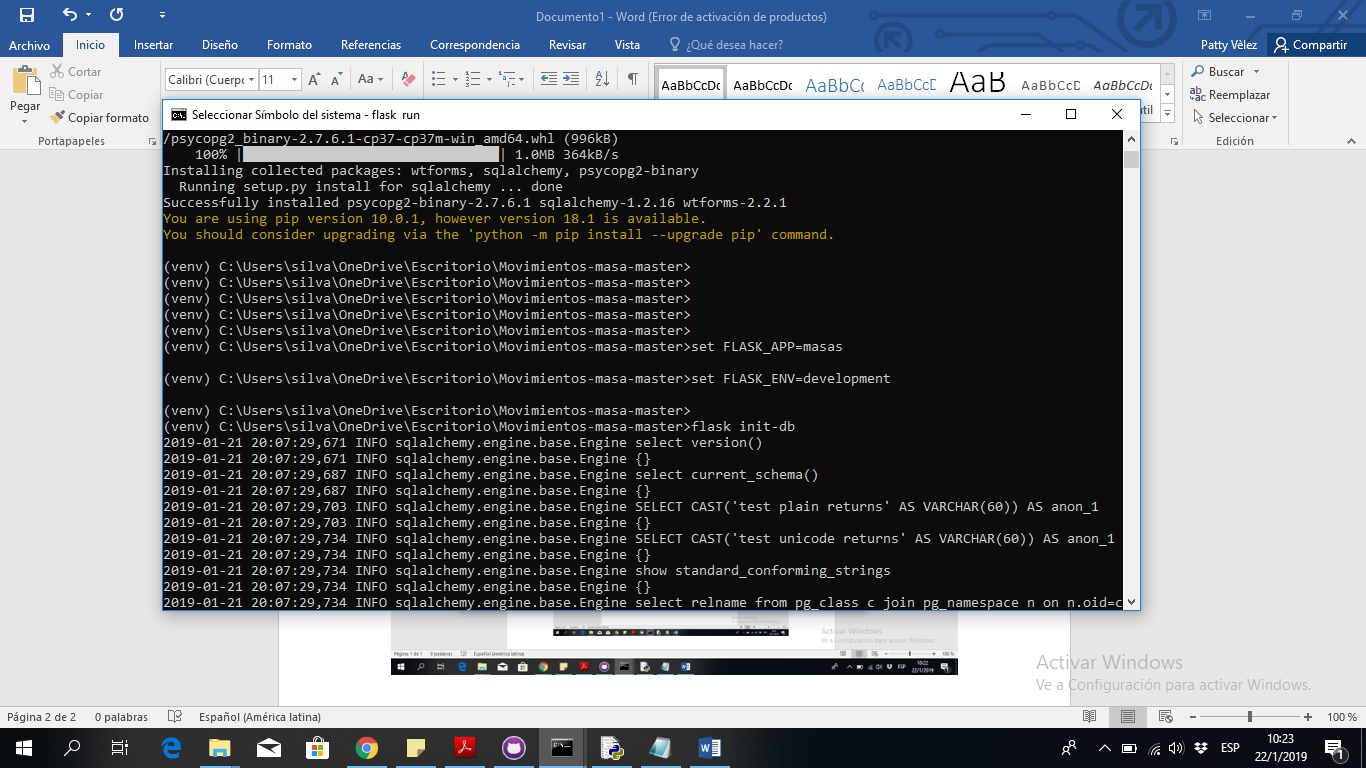
* pip install flask



* pip install wtforms



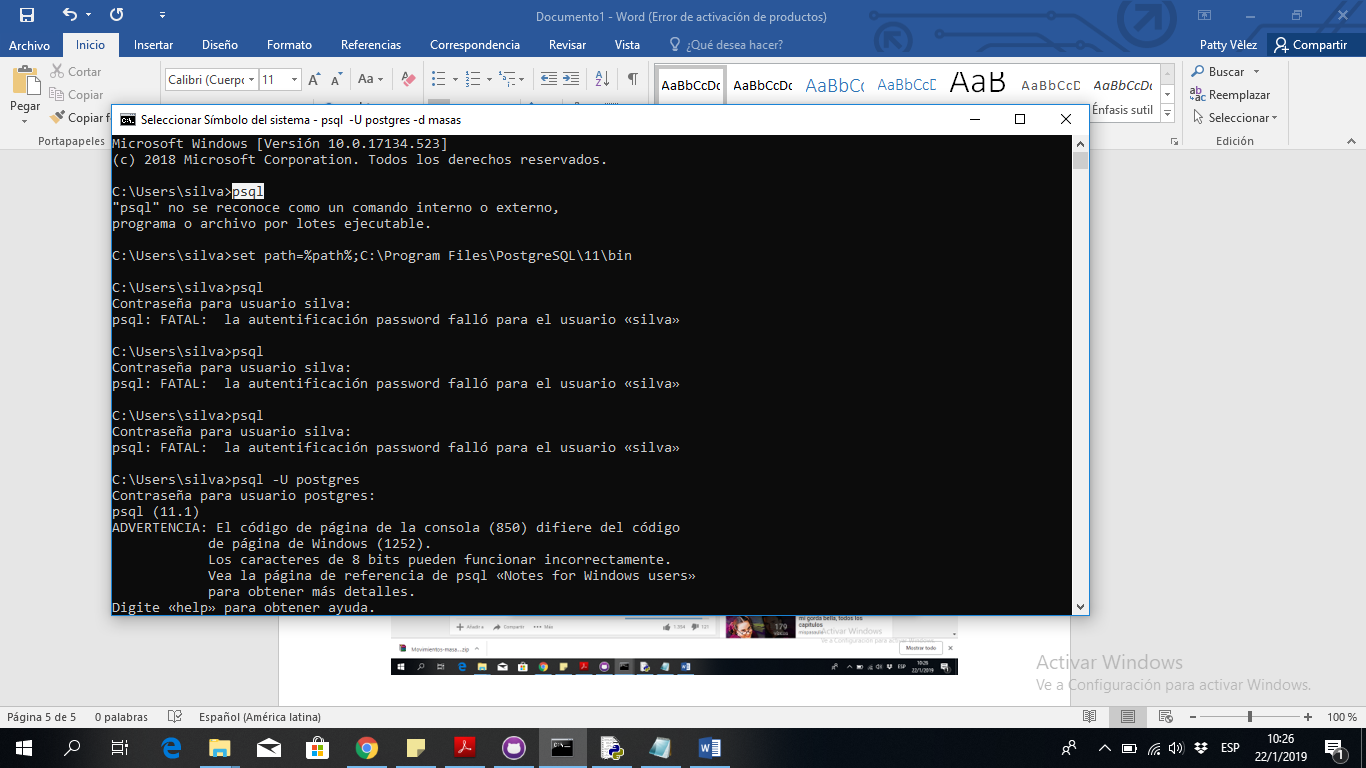
* pip install sqlachemy

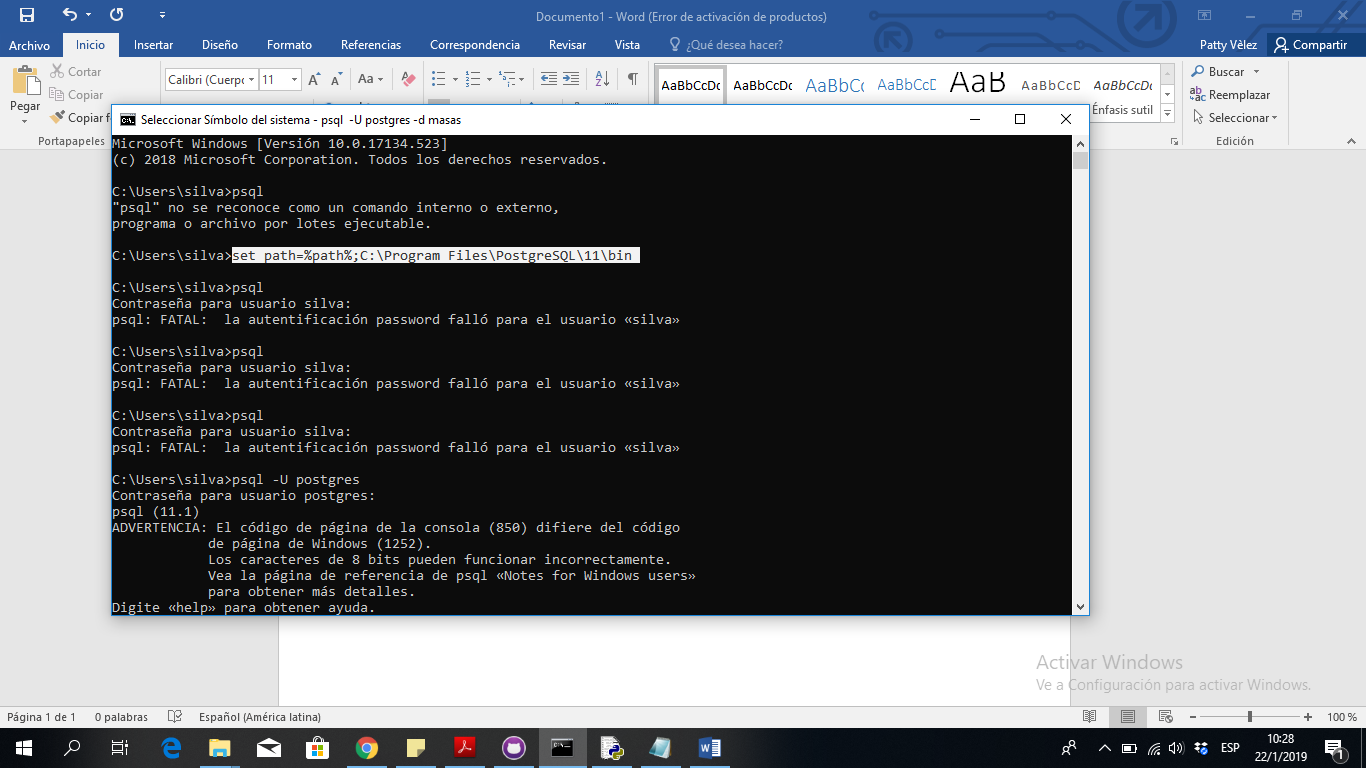


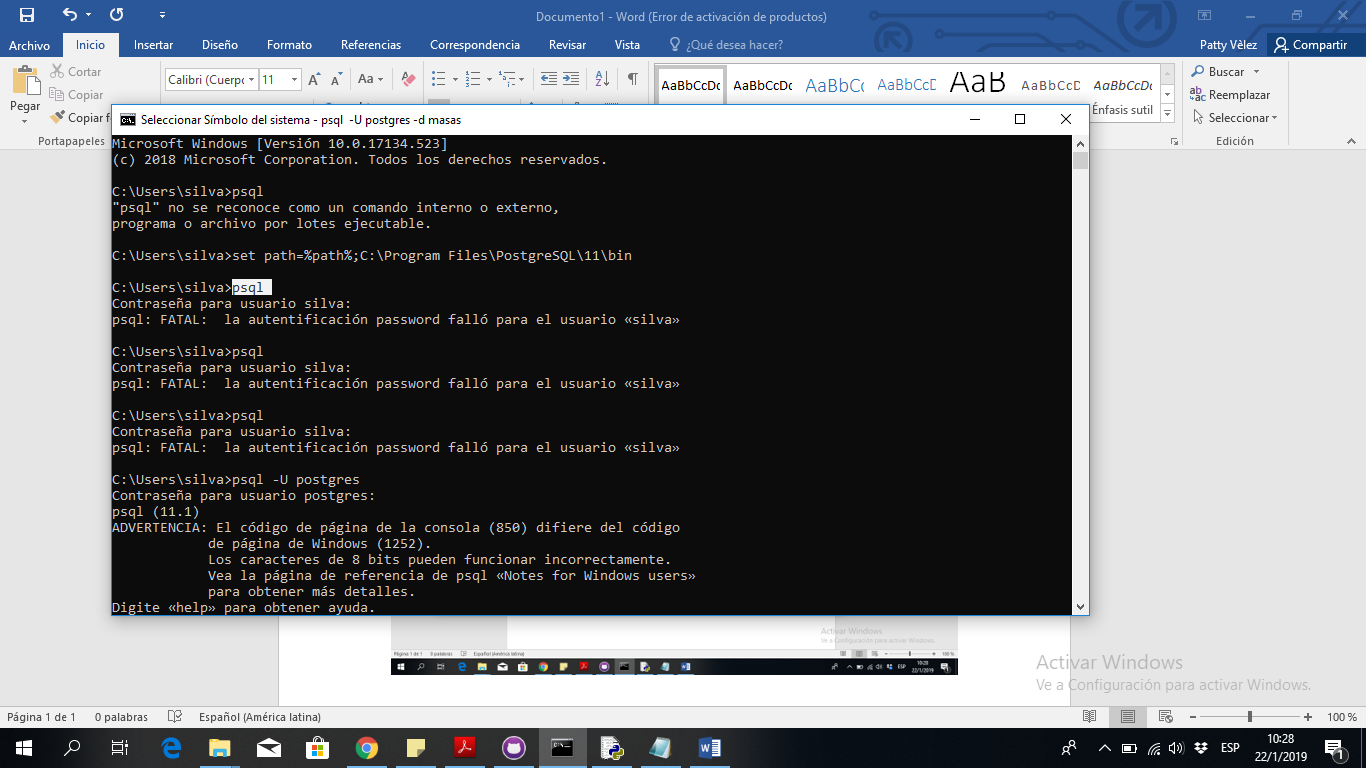
**3. Si necesitabas instalar virtualenv porque estás en una versión anterior de Python, usa el siguiente comando:**

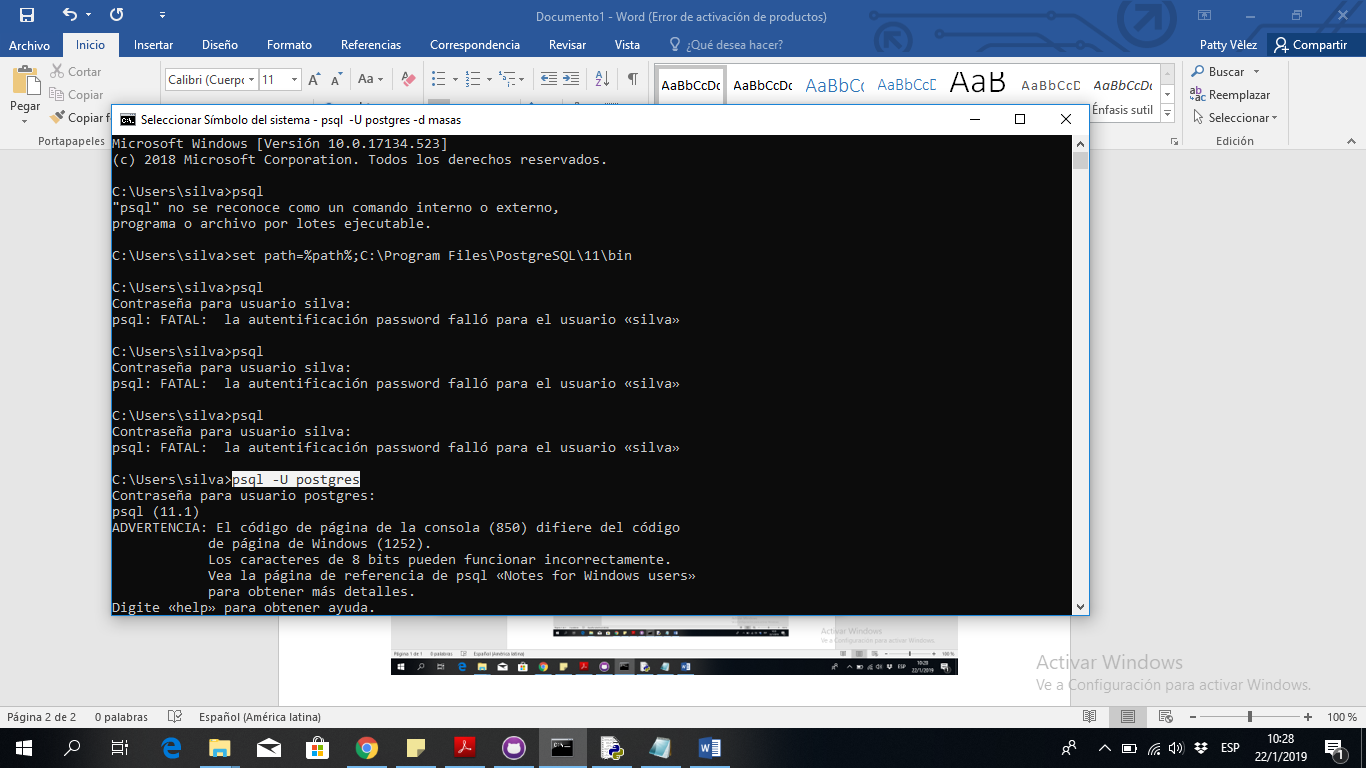
* virtualenv venv

**PostgreSQL le muestra cómo trabajar con la base de datos PostgreSQL usando el lenguaje de programación Python,**







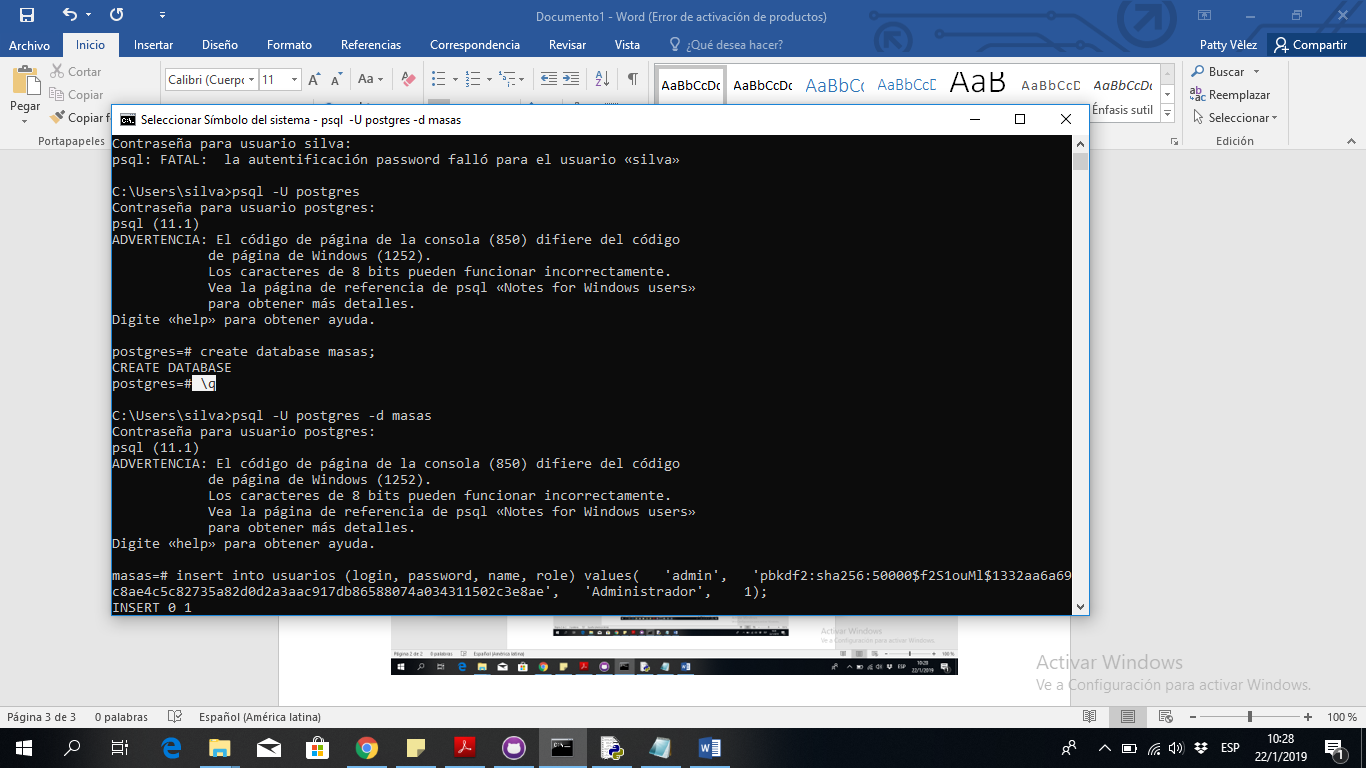


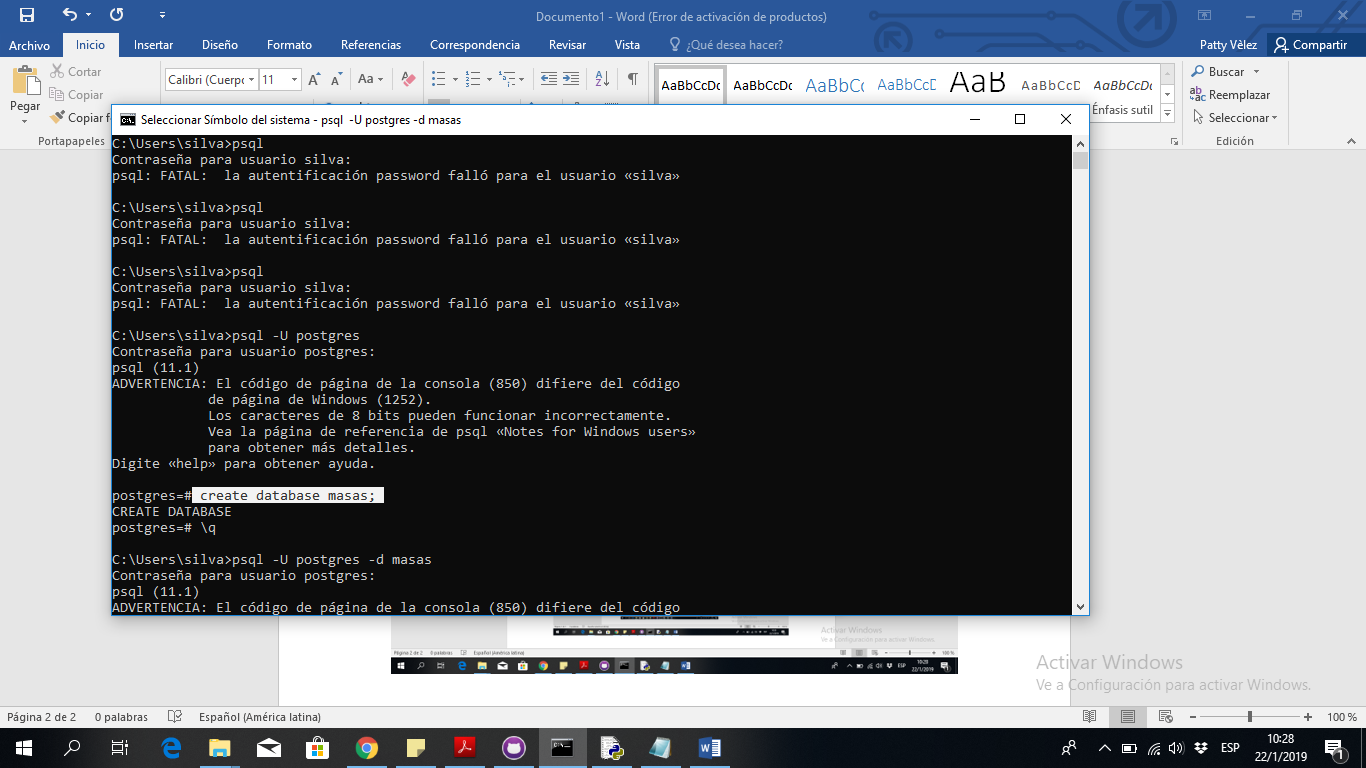
Crea una db en postgres, pon los datos de conexión en el archivo instance/config.py

**1.Ejecuta en la terminal**

         set FLASK\_APP=masas  
         set FLASK\_ENV=development

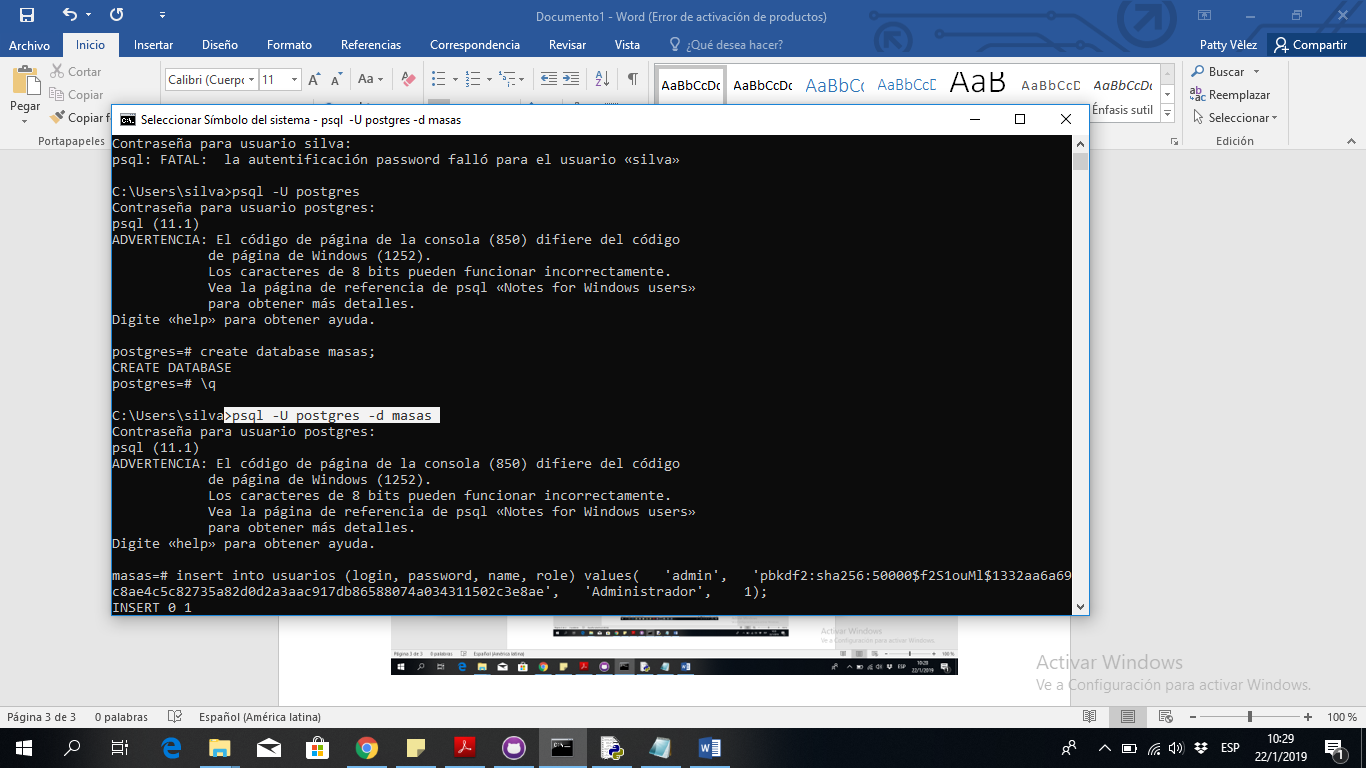
         flask init-db

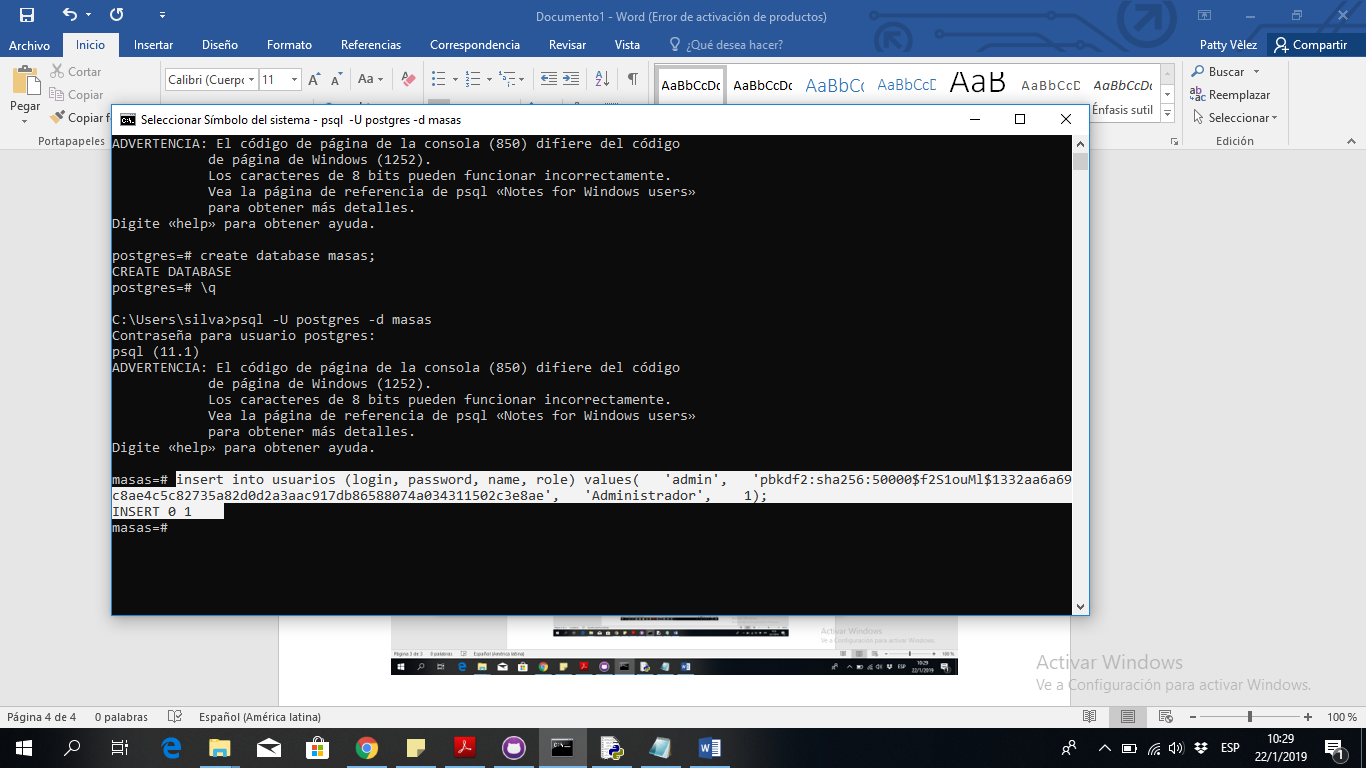


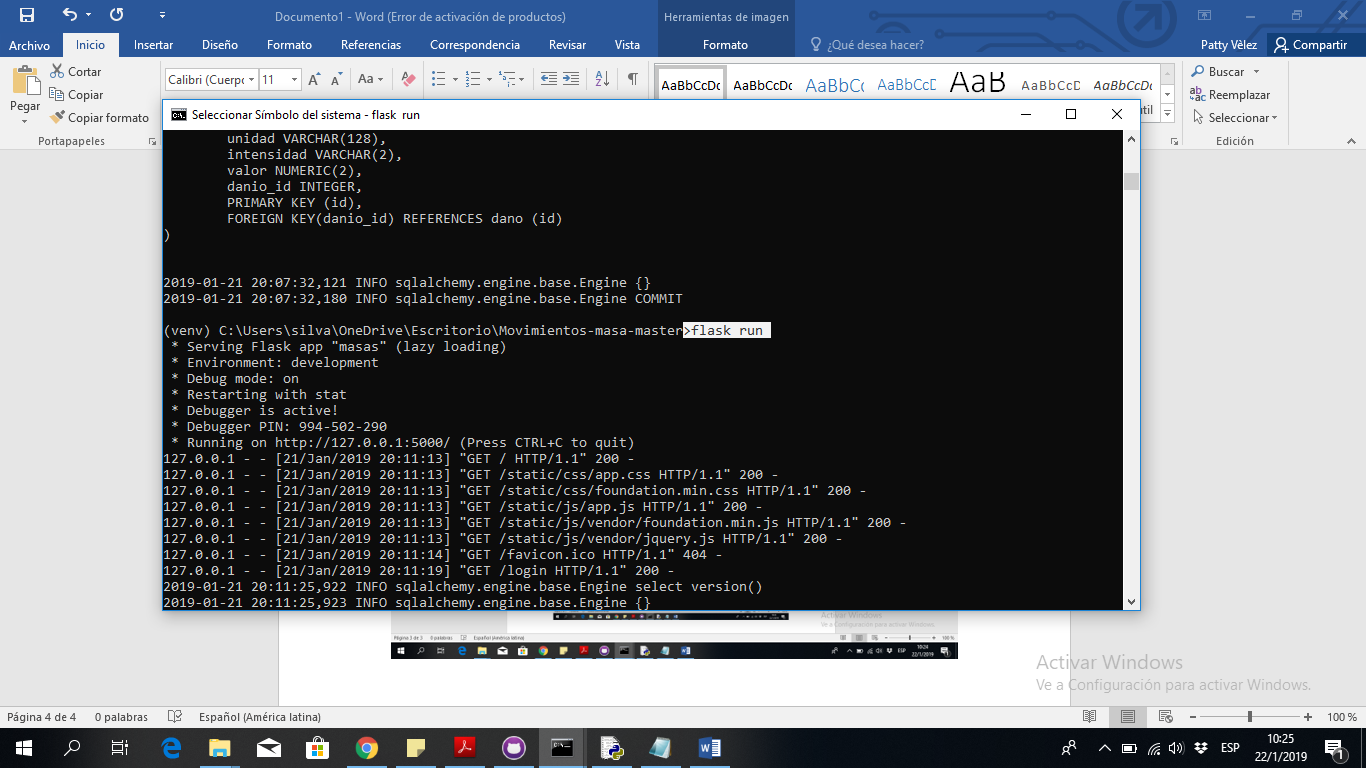


**2.Ejecuta en la db:**

          insert into usuarios (login, password, name, role)  
          values(  
               'admin',  
               'pbkdf2:sha256:50000$f2S1ouMl$1332aa6a69c8ae4c5c82735a82d0d2a3aac917db86588074a034311502c3e8ae',  
               'Administrador',  
              1);

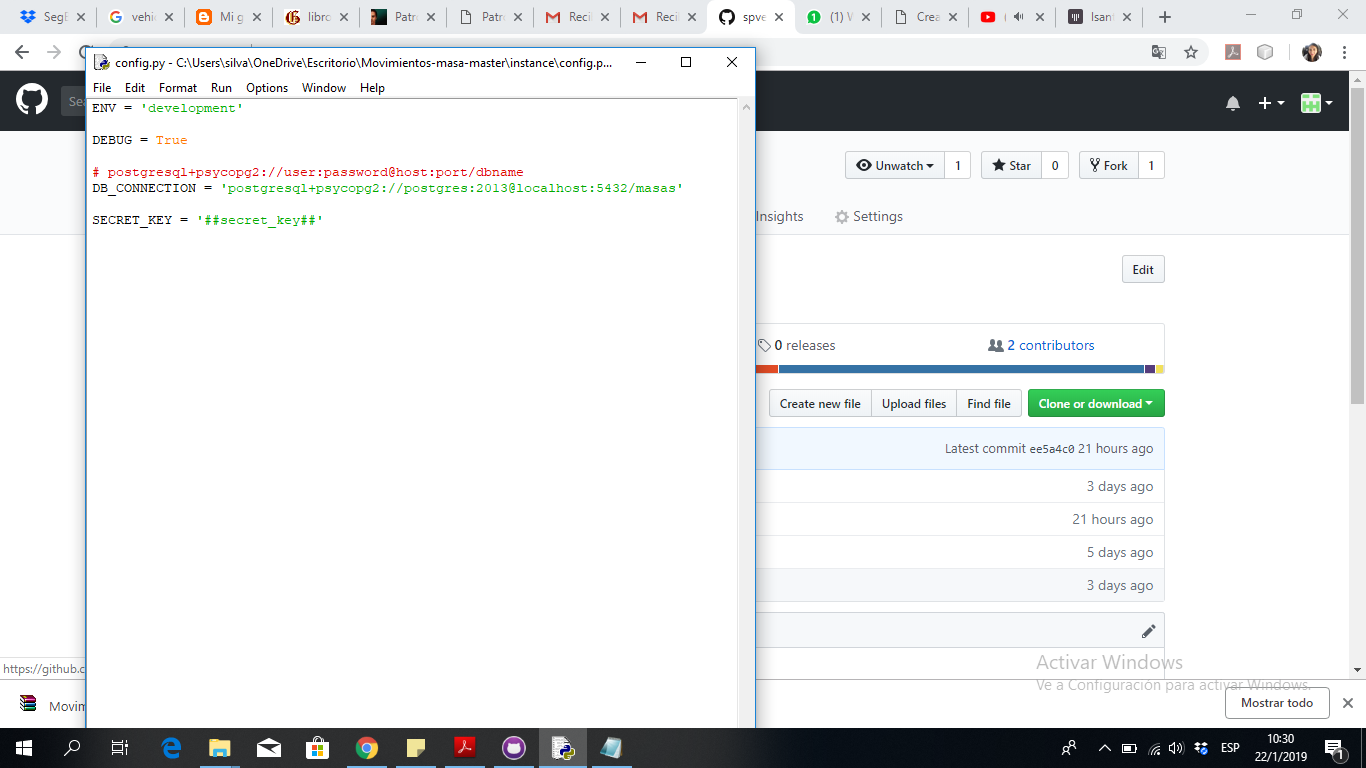






**CONEXIÓN A BASE DE DATOS**

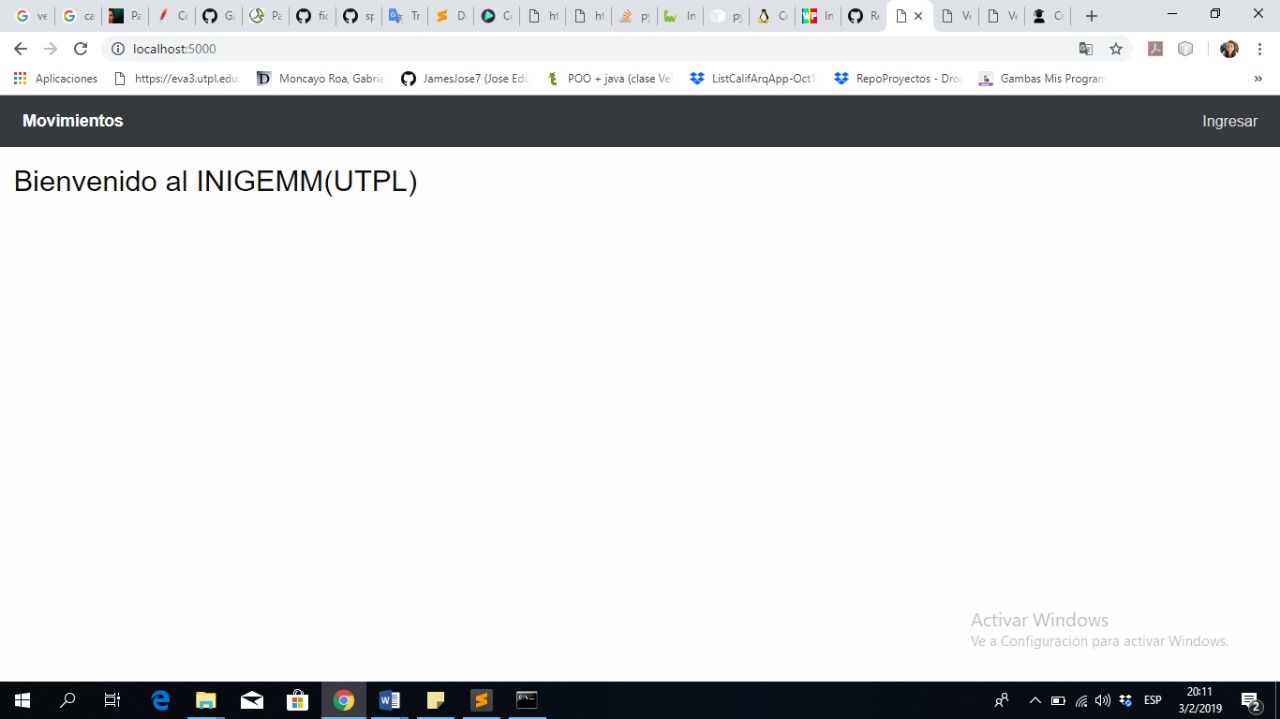
Como Podemos visualizar en esta pantalla aquí podemos hacer la conexión a la base de datos se tiene que poner la ruta de dond esta la base cons su usuario clave y pueto,si no tienes alguno de estos campos no se conectara.



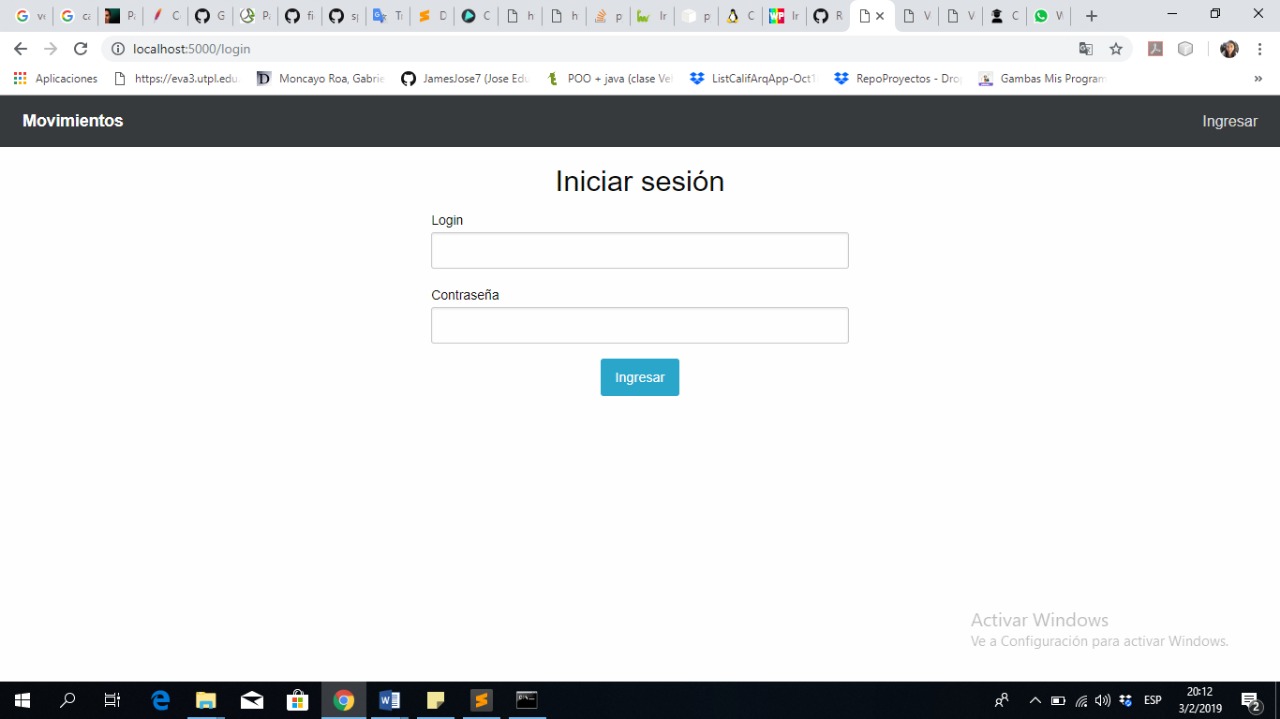
Y luego para ejecutar

python run.py

Abre un navegador e ingresa a [http://localhost:5000](http://192.168.0.1:5000/)



El usuario es admin, la clave es masas



**La aplicación está en:**

<https://mega.nz/#!dNd3DKjL!DIgctaAdpfAxreofCeT7hWrz7E-leQkj2FFNn-0uv-M>