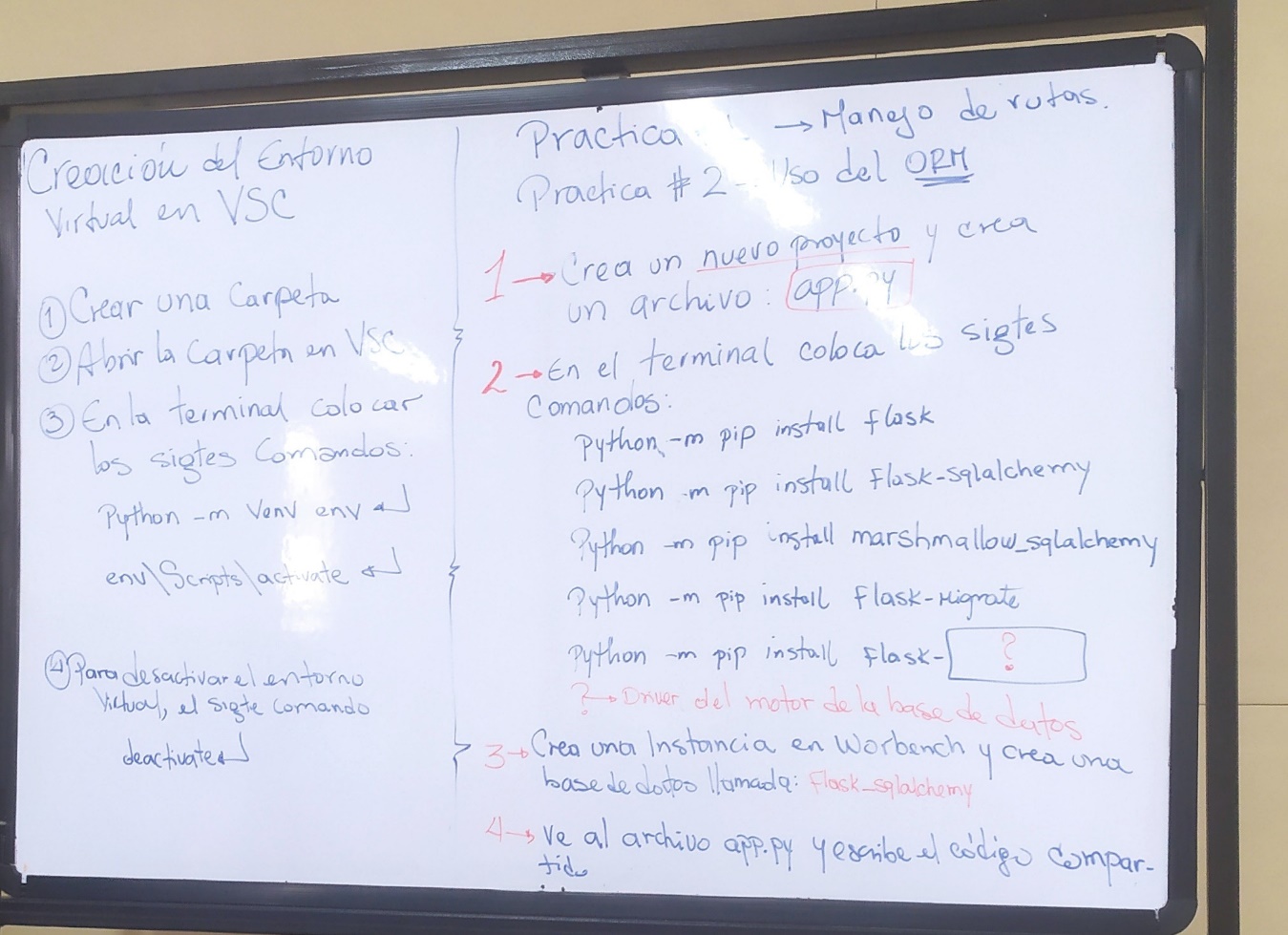
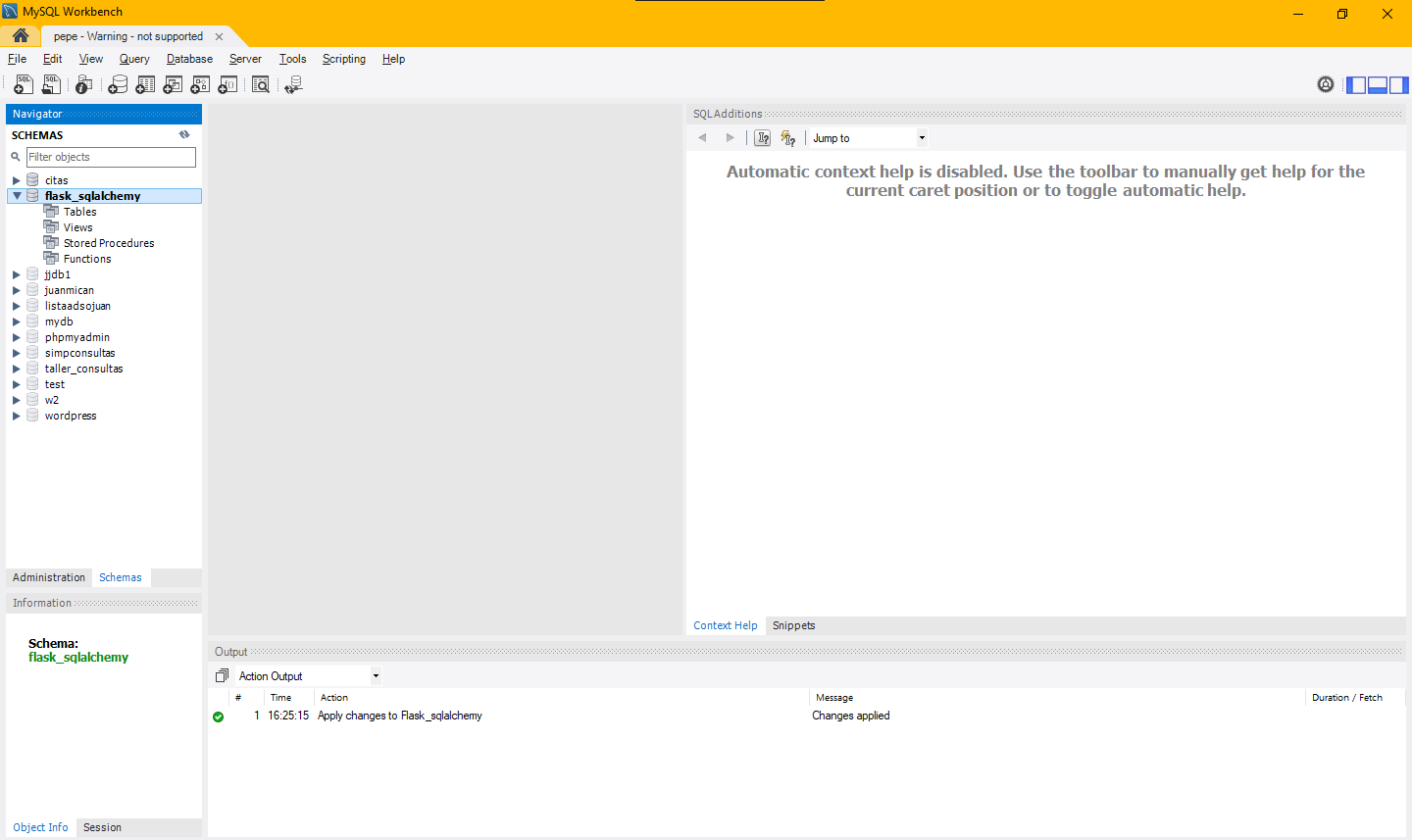
# Explicación ROM

Por Juan Carlos Mican Muñoz

1: Iniciamos el proyecto con los siguientes pasos y descargas de dependencias



2: creamos nuestra base de datos por medio de Workbench:



3: copiamos el siguiente código, pero lo modificamos a nuestras necesidades con MySQL



Modificando los usuarios y el nombre de la base de datos además de su ruta por medio de la siguiente configuración

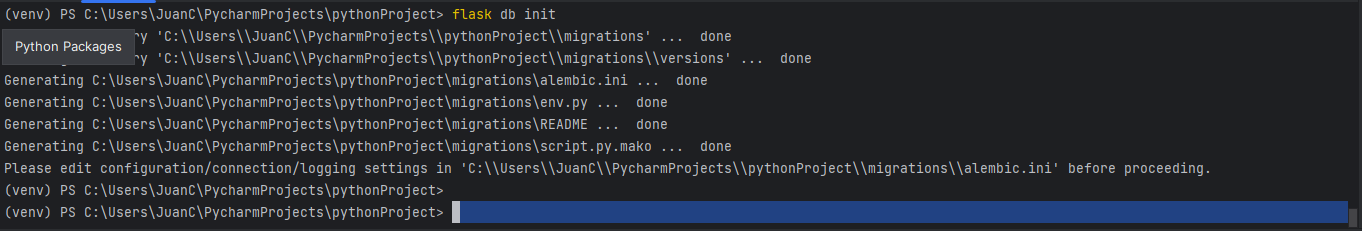
Instalamos el por medio del comando pip install pymysql

Y agregamos alas ruta

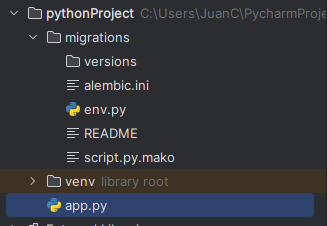
FULL\_URL\_DB = f'mysql+pymysql://{USER\_DB}:{PASS\_DB}@{URL\_DB}/{NAME\_DB}'

Además del cliente de Mysql con el comando “Python -m pip install flask mysql-client”

Y iniciamos la conexión con “flask db init”



4: al ejecutar el anterior nos crea una carpeta llamada migraciones que es para permitirnos la migración de los datos de este script a la base de datos que realizamos en workbench



5: copiamos el código suministrado que es el responsable de inyectar a la base de datos las tablas y sus campos; y ejecutamos el comando “flask db migrate “

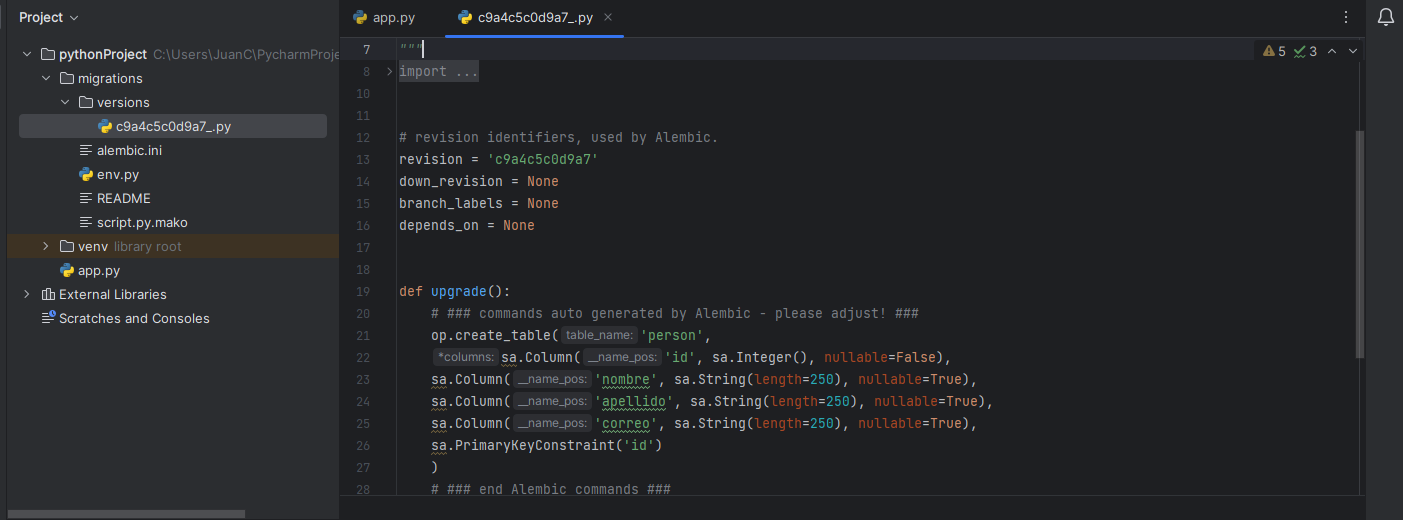
Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

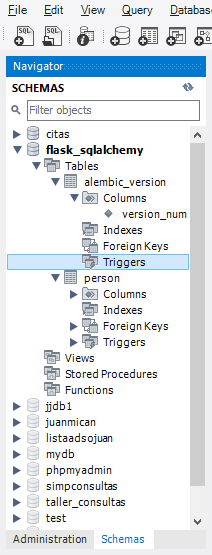
Lo que sucedió es que se actualizo la BD con nueva tablas y nuevos campos además se creó una carpeta llamada versiones y un archivo que guarda las interacciones con la base de datos



6: ahora ejecutamos en la terminal el siguiente comando “flask db upgrade”, que realizara una actualización ala base de datos (¿Otra vez?):

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente



6: un ROM es “Object-Relational Mapping” y es una técnica que mapea una base de datos y nos permite interactuar por medio de la manipulación de objetos con la base de datos supuestamente dando una mayor legibilidad; es similar al DOM que generan los navegadores con la información de páginas web.