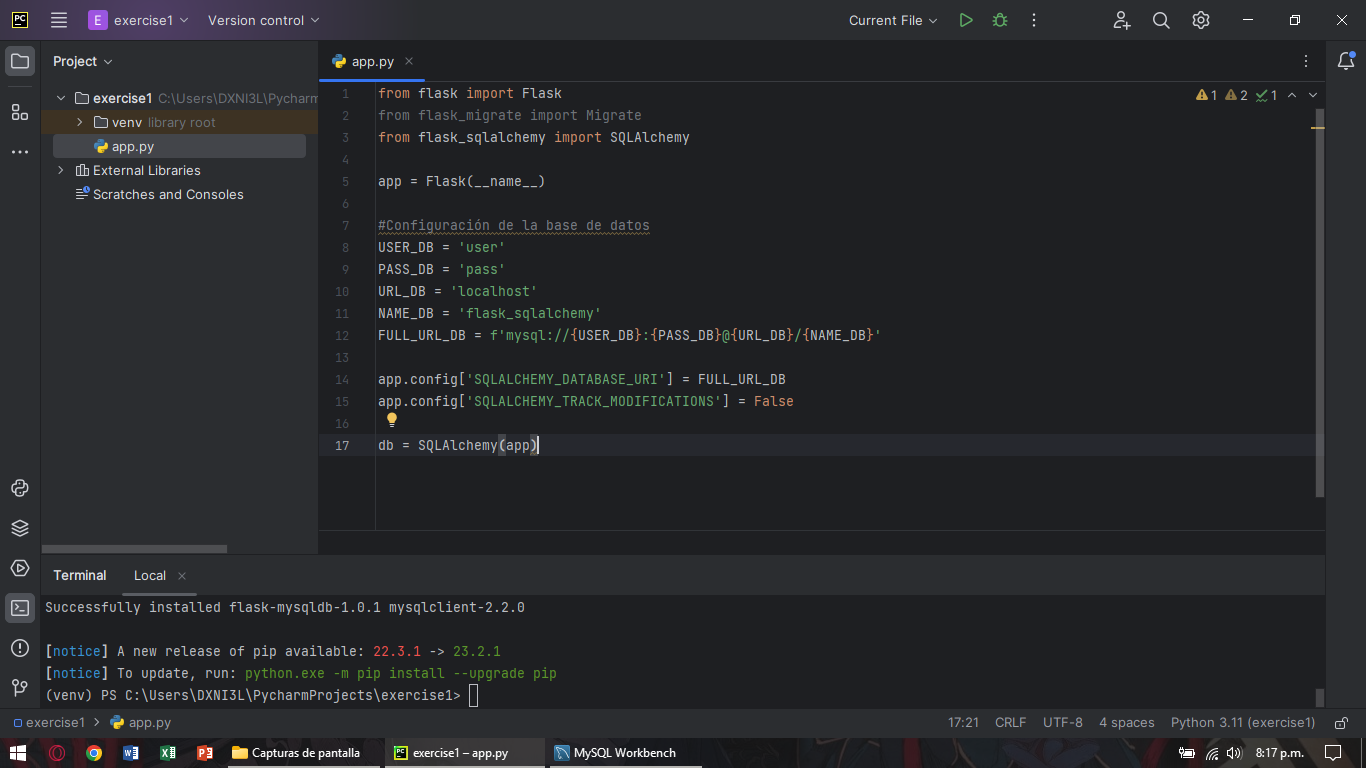
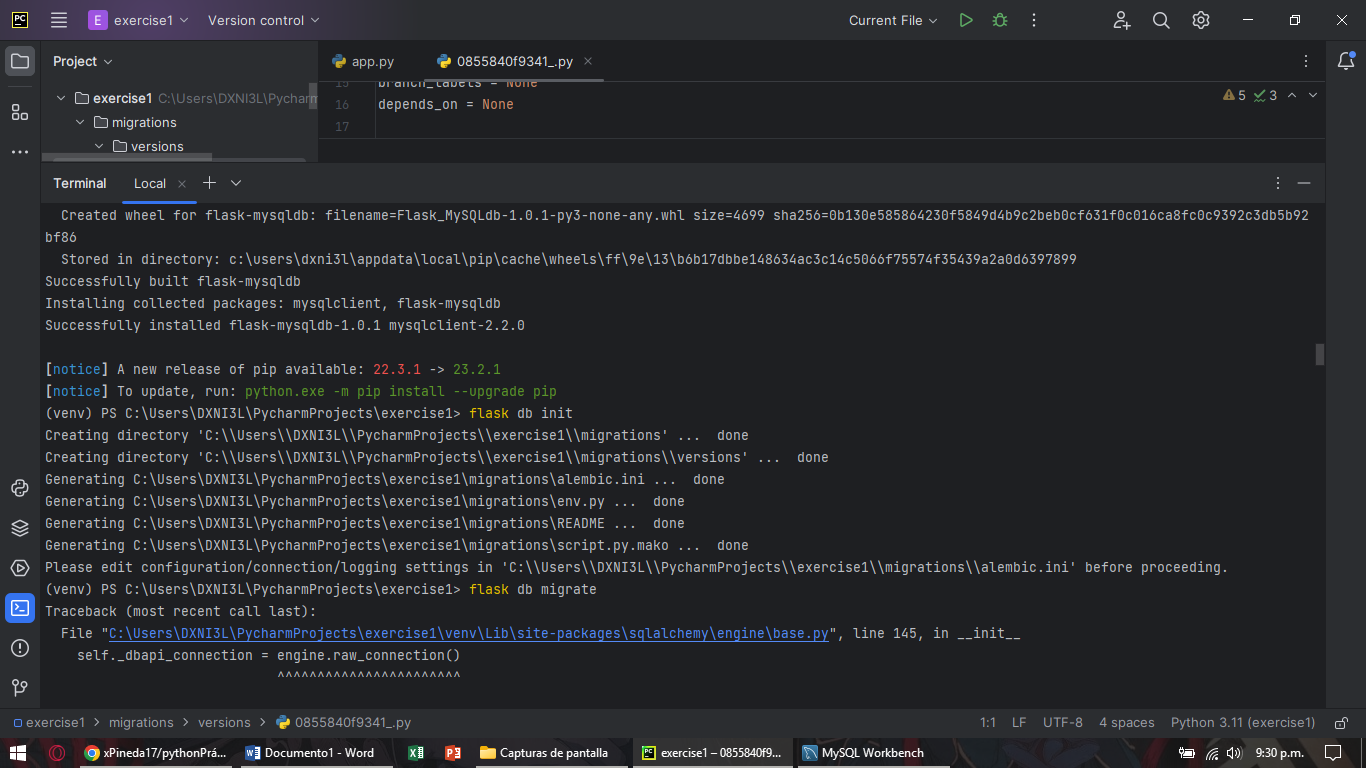
**PRACTICA #2**

**Código configurado a mysql:**



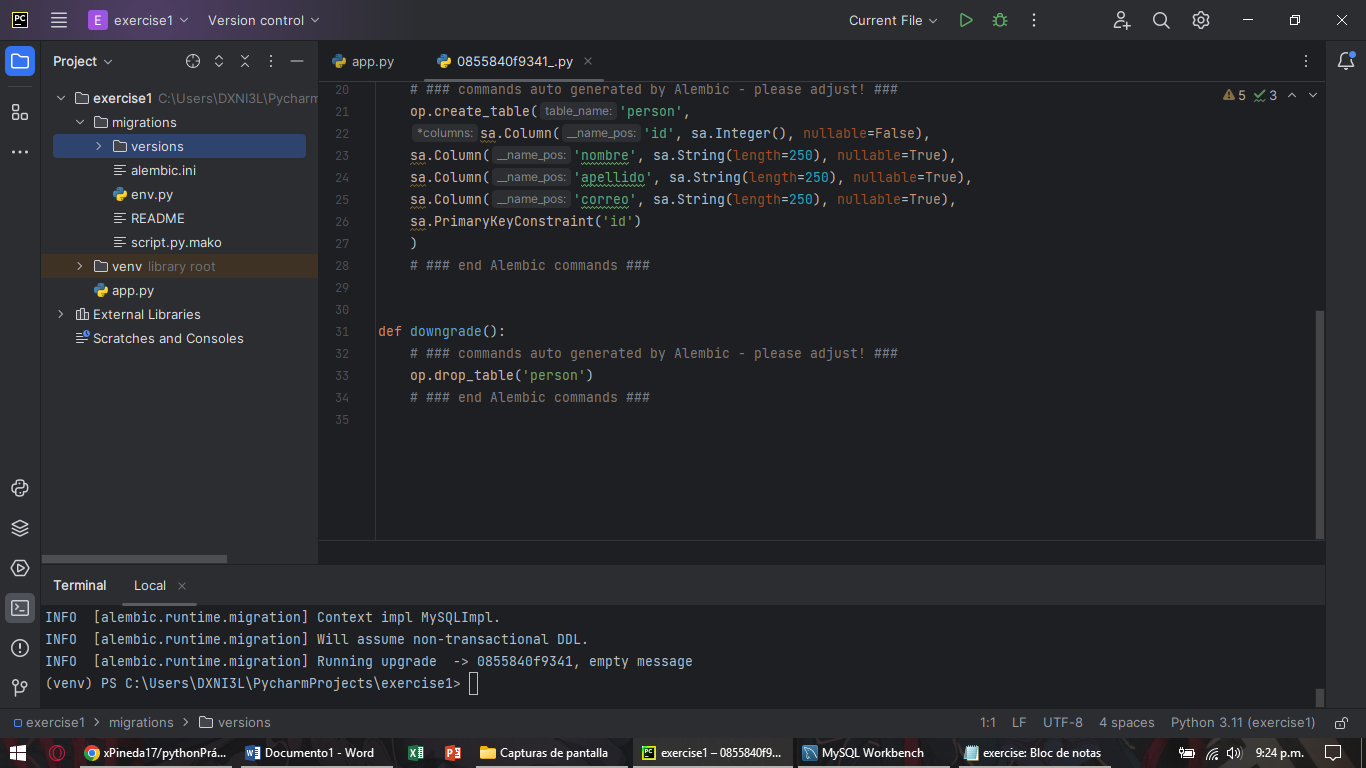
**flask db init**

****

**Revisa tu carpeta:**

* ¿Qué sucedió?

Rta: Creó una carpeta llamada migrations con varios archivos dentro.



**Archivo app.py escribimos las líneas de código y responde:**

* ¿Qué crees tú que hace esa parte del código?

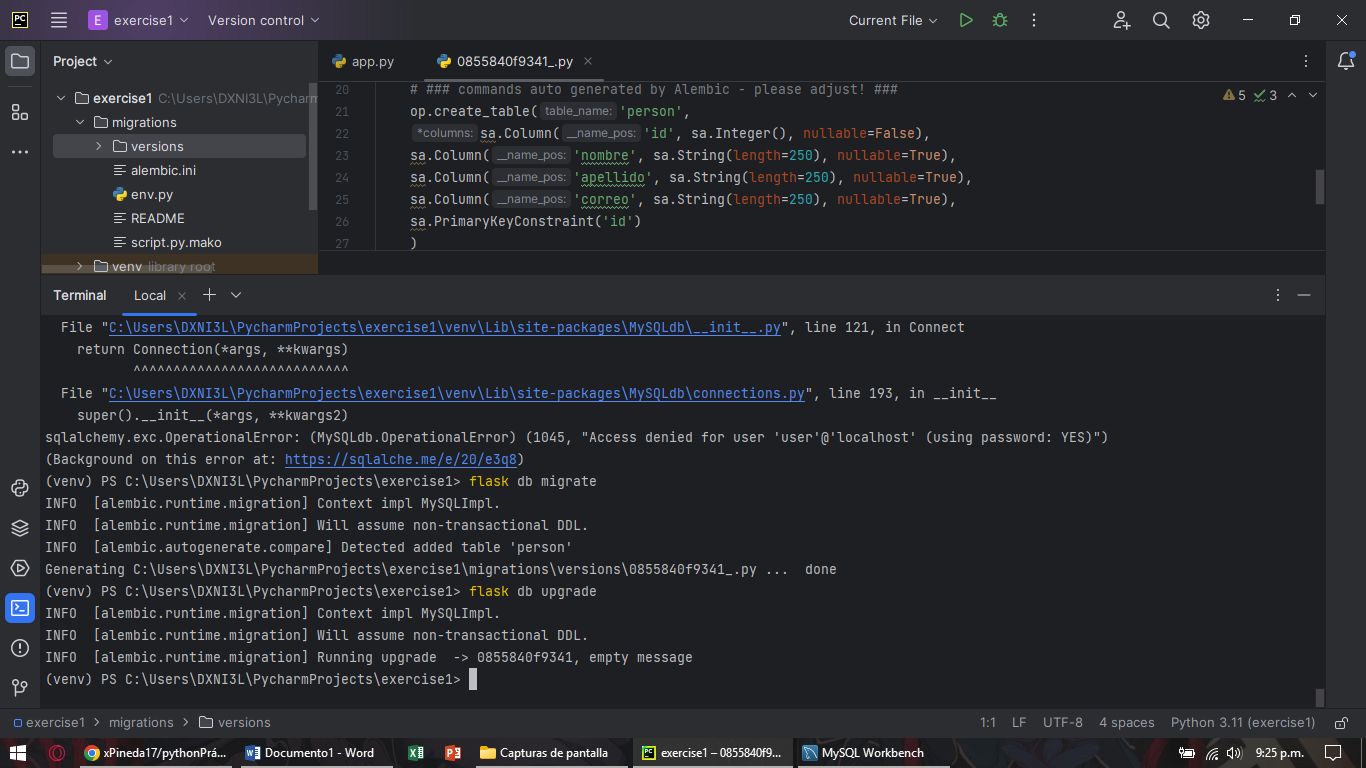
Rta: Se define una clase llamada persona que es como una tabla, que son como los campos de la tabla:

El primero llamado id dice que es de tipo Integer y es llave primaria, el resto de campos como nombre, apellido y correo son de tipo String con un máximo de 250 caracteres.

Luego creo que se accede a los atributos mediante el parámetro self.

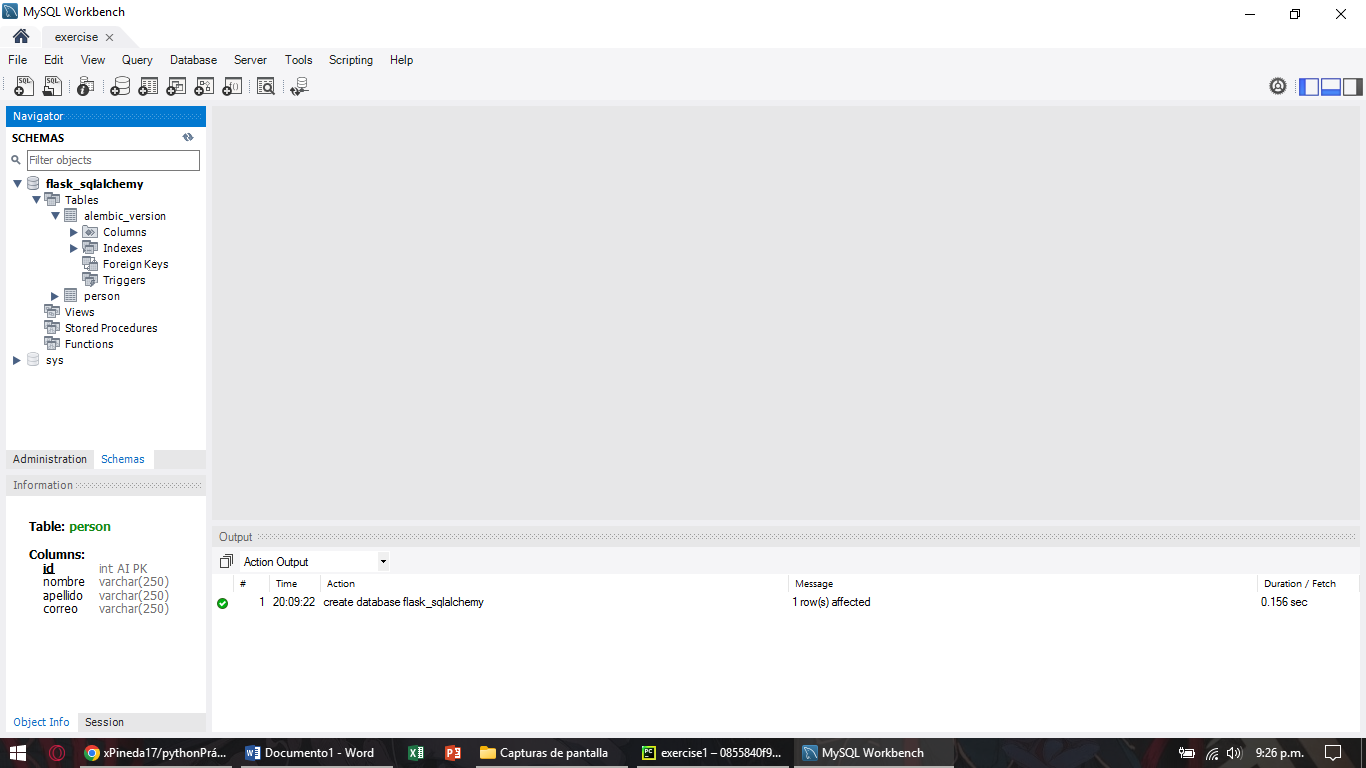
Después parece que almacena los datos en un formato json y lo último no sé qué hace.

**flask db migrate**

****

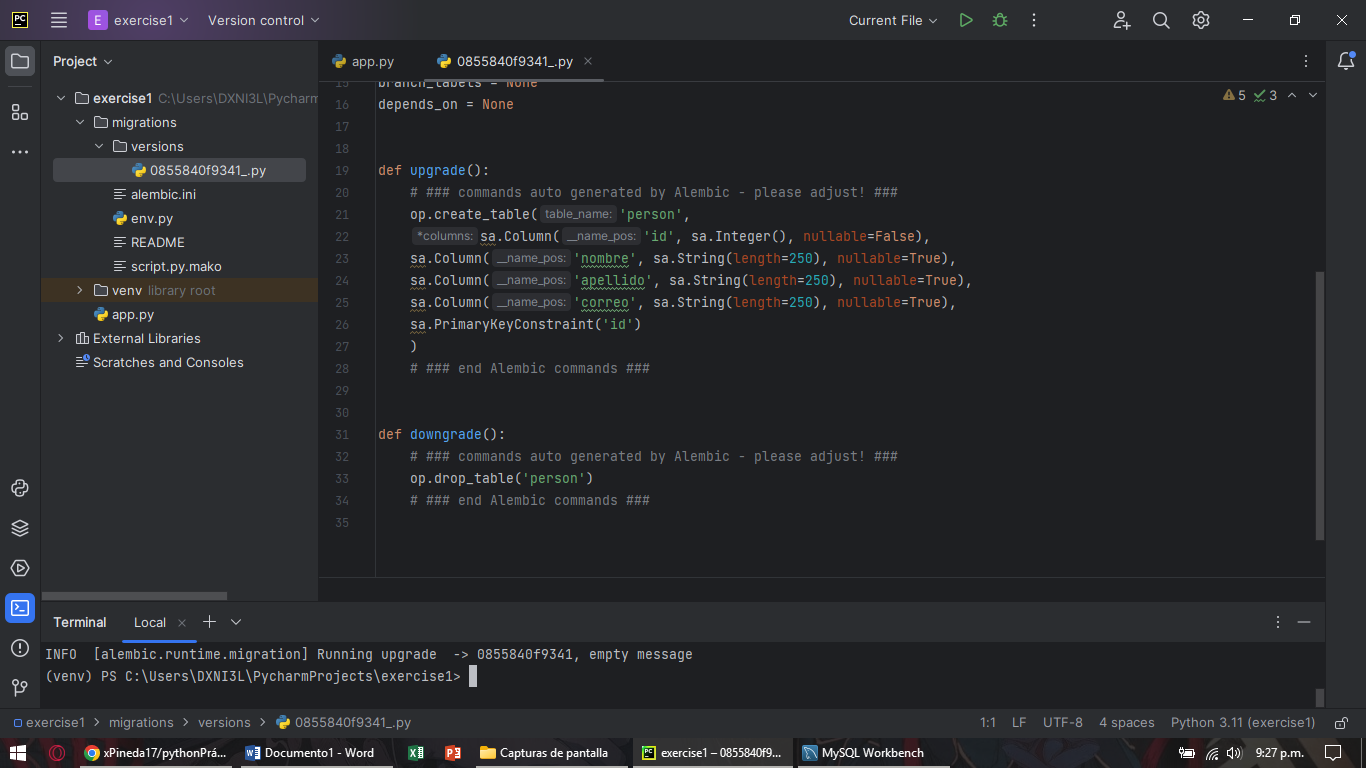
**Revisamos la base de datos en workbench:**

Rta: Me creo una tabla llamada alembic\_version y dentro de ella encontramos un apartados que dicen Columns, Indexes, Foreign Keys y Triggers.



**Revisamos la carpeta migrations y abrimos el archivo py de la carpeta versions. Explica que ves:**

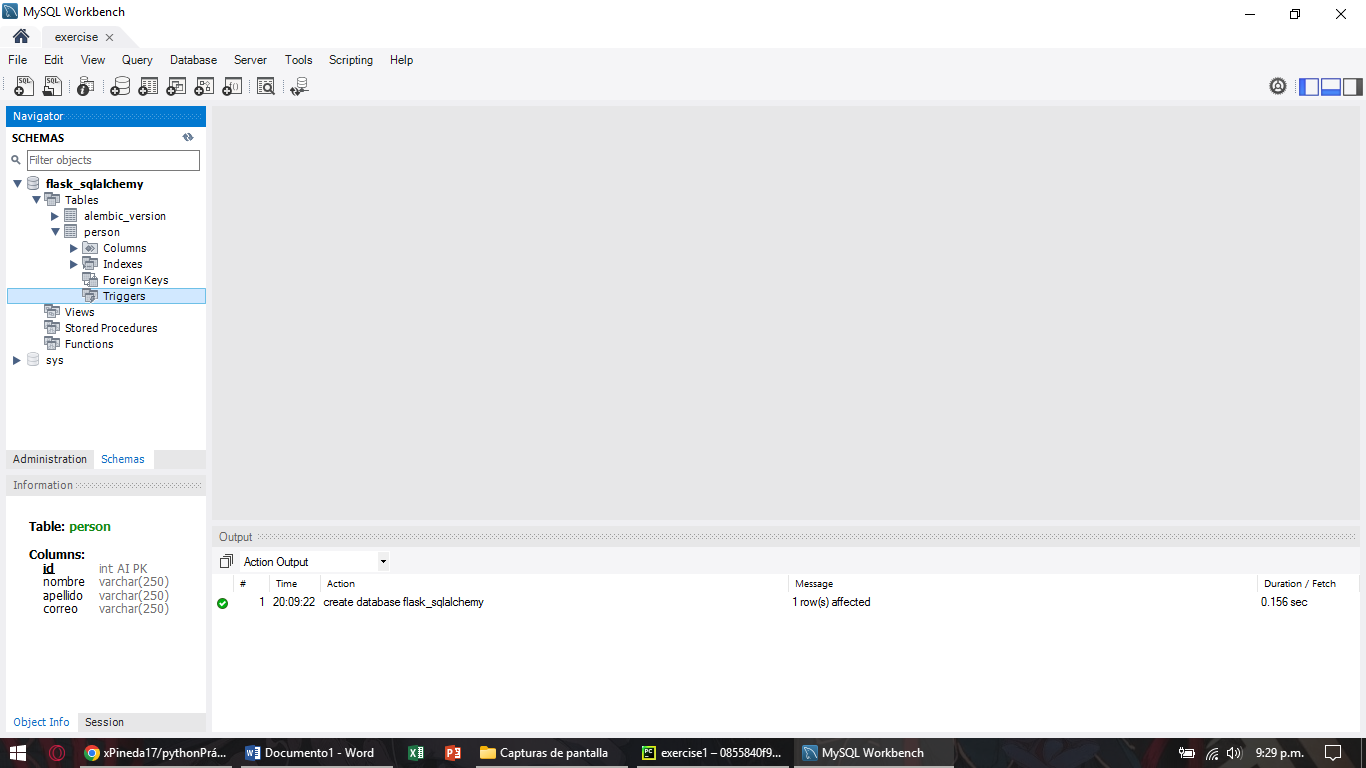
Rta: Muestra un def upgrade() donde vemos un create table person, y con sus respectivos campos id, nombre, apellido, correo y define la llave primaria que es id. Aunque luego hay un def downgrade() que tiene un drop table lo cual sirve para eliminar la tabla.

****

**flask db upgrade**

**Revisamos la base de datos en workbench:**

Rta: Ya creó la tabla person con los campos id, nombre, apellido y correo.



**Explica con tus propias palabras que es un ORM:**

Rta: Para mí es como una forma simplificada para crear y conectarse a bases de datos mediante Python sin necesidad de xampp como hemos visto hasta ahora.