

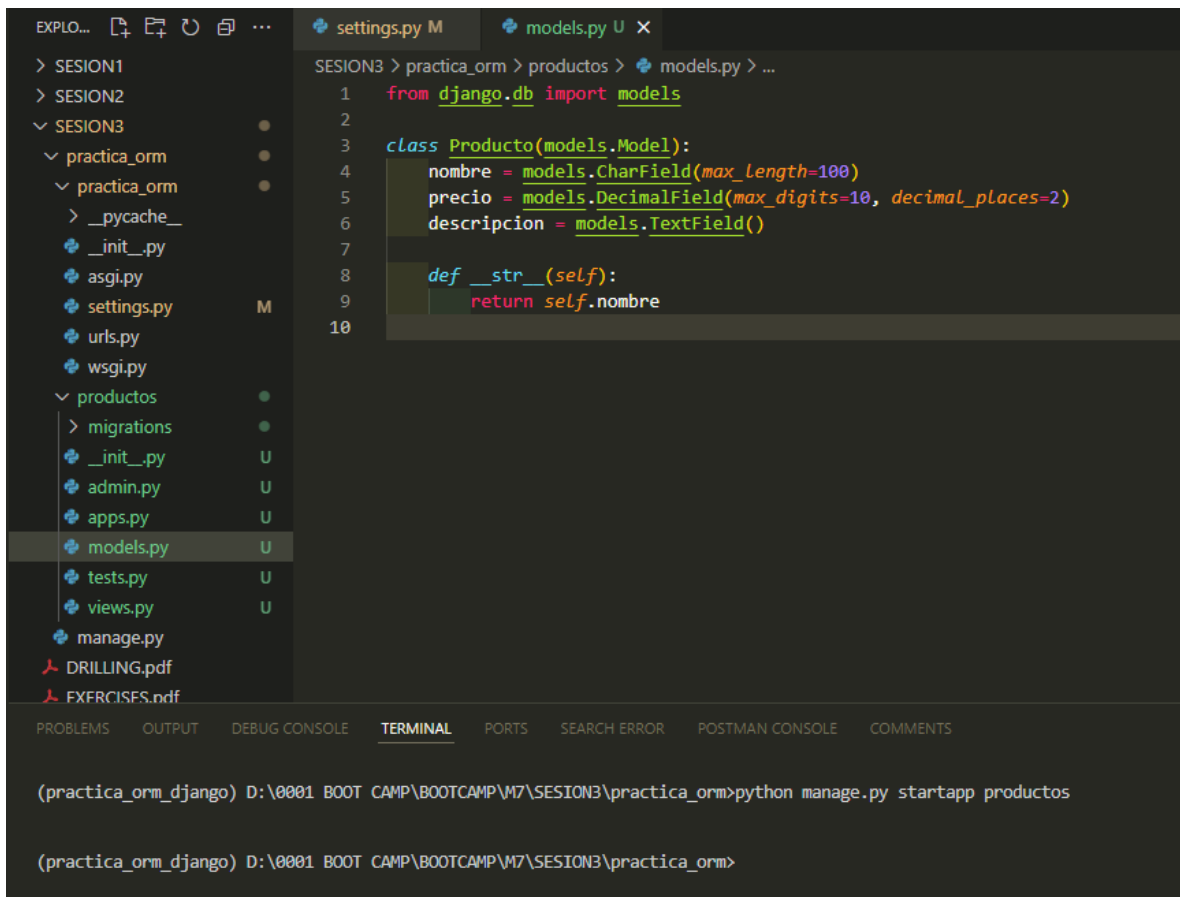
DRILLING M7 S3 SR YURI URZUA LEBUY

DRILLING: CONTINUANDO – CONFIGURAR UNA BASE DE DATOS EN DJANGO

1. Diseñar una aplicación que se va a llamar “productos del proyecto”.

- `python manage.py startapp productos`

2. Los campos de los productos son:



```
EXPLO...  [Icons] ...  settings.py M  models.py U X
> SESION1
> SESION2
v SESION3
  v practica_orm
    v practica_orm
      > __pycache__
      [Icons] __init__.py
      [Icons] asgi.py
      [Icons] settings.py M
      [Icons] urls.py
      [Icons] wsgi.py
      v productos
        > migrations
        [Icons] __init__.py U
        [Icons] admin.py U
        [Icons] apps.py U
        [Icons] models.py U
        [Icons] tests.py U
        [Icons] views.py U
      [Icons] manage.py
      DRILLING.pdf
      EXERCISES.pdf

SESION3 > practica_orm > productos > [Icons] models.py > ...
1  from django.db import models
2
3  class Producto(models.Model):
4      nombre = models.CharField(max_length=100)
5      precio = models.DecimalField(max_digits=10, decimal_places=2)
6      descripcion = models.TextField()
7
8      def __str__(self):
9          return self.nombre
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR POSTMAN CONSOLE COMMENTS

```
(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESION3\practica_orm>python manage.py startapp productos

(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESION3\practica_orm>
```

### 3. Generar las migraciones respectivas.

```
(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESSION3\practica_orm>python manage.py makemigrations
Migrations for 'productos':
  productos\migrations\0001_initial.py
    + Create model Producto

(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESSION3\practica_orm>
```

```
(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESSION3\practica_orm>python manage.py makemigrations
Migrations for 'productos':
  productos\migrations\0001_initial.py
    + Create model Producto

(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESSION3\practica_orm>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, productos, sessions
Running migrations:
  Applying productos.0001_initial... OK

(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESSION3\practica_orm>
```

### 4. Usar el comando en Django que muestra el SQL que se ejecuta al realizar las migraciones y para crear el modelo.

```
(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESSION3\practica_orm>python manage.py sqlmigrate productos 0001
BEGIN;
--
-- Create model Producto
--
CREATE TABLE "productos_producto" ("id" bigint NOT NULL PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY, "nombre" varchar(100) NOT NULL, "precio" numeric(10, 2) NOT NULL, "descripcion" text NOT NULL);
COMMIT;

(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESSION3\practica_orm>
```

### 5. Consultar la base de datos y el modelo creado en la shell de postgresql.

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
psql (17.0)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

postgres=# \c db_practica_orm
Ahora está conectado a la base de datos «db_practica_orm» con el usuario «postgres».
db_practica_orm=# \dt
Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----+-----+-----+-----
public | auth_group | tabla | user_db
public | auth_group_permissions | tabla | user_db
public | auth_permission | tabla | user_db
public | auth_user | tabla | user_db
public | auth_user_groups | tabla | user_db
public | auth_user_user_permissions | tabla | user_db
public | django_admin_log | tabla | user_db
public | django_content_type | tabla | user_db
public | django_migrations | tabla | user_db
public | django_session | tabla | user_db
public | productos_producto | tabla | user_db
(11 filas)

db_practica_orm=# \d productos_producto
Tabla «public.productos_producto»
Columna | Tipo | Ordenamiento | Nulable | Por omisión
-----+-----+-----+-----+-----
id | bigint | | not null | generated by default as identity
nombre | character varying(100) | | not null |
precio | numeric(10,2) | | not null |
descripcion | text | | not null |
Índices:
"productos_producto_pkey" PRIMARY KEY, btree (id)

db_practica_orm=#
```

## En dbeaver

The screenshot shows the 'Propiedades' (Properties) window in DBeaver for a PostgreSQL table named 'productos\_producto'. The window is divided into several sections:

- Table Information:** Table Name: productos\_producto, ID Objeto: 17734, Tablespace: pg\_default, Propietario: user\_db.
- Security and Partitions:** Has Row-Level Security (unchecked), Partitions (unchecked).
- Partition by:** (empty field).
- Comentario:** (empty text area).
- Opciones extra:** (empty text area).

Below these fields is a table showing the columns of the table:

Column Name	#	Tipo de datos	Identidad	Collation	No Nulo	Por defecto	Comentario
id	1	int8	By Default		[v]		
nombre	2	varchar(100)		default	[v]		
precio	3	numeric(10, 2)			[v]		
descripcion	4	text		default	[v]		

On the left side of the window, there is a sidebar with various database object categories:

- Restricciones
- Claves foráneas
- Indíces
- Dependencias
- Referencias
- Particiones
- Disparadores
- Reglas
- Políticas
- Statistics
- Permisos
- DDL