

## REBOUND M7 S4 YURI URZUA LEBUY

1. Agregar campos desde la consola o shell de django al modelo de Producto.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  POSTMAN CONSOLE  COMMENTS

(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESSION3\practica_orm>python manage.py shell
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug  6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> █
```

```
(practica_orm_django) D:\0001 BOOT CAMP\BOOTCAMP\M7\SESSION3\practica_orm>python manage.py shell
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug  6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from productos.models import Producto
>>> Producto.objects.create(nombre="Laptop", precio=999.99, descripcion="Potente laptop para trabajo y juegos")
<Producto: Laptop>
>>> Producto.objects.create(nombre="Smartphone", precio=599.99, descripcion="Teléfono inteligente de última generación")
<Producto: Smartphone>
>>> Producto.objects.create(nombre="Auriculares", precio=89.99, descripcion="Auriculares inalámbricos con cancelación de ruido")
<Producto: Auriculares>
>>> print(Producto.objects.all())
<QuerySet [<Producto: Laptop>, <Producto: Smartphone>, <Producto: Auriculares>]>
>>> █
```

2. Listar los productos desde la consola de psql.

```
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=# SELECT * FROM productos_producto;
 id | nombre | precio | descripción
-----+-----+-----+-----
  1 | Laptop | 999.99 | Potente laptop para trabajo y juegos
  2 | Smartphone | 599.99 | Teléfono inteligente de última generación
  3 | Auriculares | 89.99 | Auriculares inalámbricos con cancelación de ruido
(3 filas)

db_practica_orm=#
```

### 3. Consultar todos los productos desde la consola de Python.

```
from productos.models import Producto

# Obtener todos los productos
todos_los_productos = Producto.objects.all()

# Imprimir cada producto
for producto in todos_los_productos:
    print(f"ID: {producto.id}, Nombre: {producto.nombre}, Precio: {producto.precio}, Descripción: {producto.descripcion}")

# Filtrar productos por precio
productos_caros = Producto.objects.filter(precio__gt=500)
print("\nProductos con precio mayor a 500:")
for producto in productos_caros:
    print(f"{producto.nombre}: {producto.precio}")
```

```
# Buscar un producto específico
try:
    laptop = Producto.objects.get(nombre="Laptop")
    print(f"\nDetalles de la Laptop: {laptop.nombre}, Precio: {laptop.precio}")
except Producto.DoesNotExist:
    print("No se encontró el producto 'Laptop'")

# Actualizar un producto
smartphone = Producto.objects.get(nombre="Smartphone")
smartphone.precio = 649.99
smartphone.save()
print(f"\nPrecio actualizado del Smartphone: {smartphone.precio}")

# Eliminar un producto
auriculares = Producto.objects.get(nombre="Auriculares")
auriculares.delete()
print("\nAuriculares eliminados")

# Verificar la eliminación
print("\nProductos restantes:")
for producto in Producto.objects.all():
    print(producto.nombre)
```