

Modulo 7 Aprendizaje Esperado 2: Ejercicio Individual

Nombre: Cristian Aranda Bórquez

En el siguiente documento se dejará constancia de los comandos utilizados siguiendo las instrucciones de la actividad.

1. Instalación de dependencias

Instala el conector necesario para usar MySQL con Django (mysqlclient).

2. Creación de un proyecto Django

Crea un nuevo proyecto llamado mi_proyecto.

```
PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2_individual> python -m venv myenv
PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2_individual> myenv\scripts\activate
(myenv) PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2_individual> pip install django
Collecting django
  Using cached django-5.2.8-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<=3.8.1 (from django)
  Using cached asgiref-3.10.0-py3-none-any.whl.metadata (9.3 kB)
  Using cached tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached django-5.2.8-py3-none-any.whl (8.3 MB)
Using cached asgiref-3.10.0-py3-none-any.whl (24 kB)
Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)
Using cached tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.10.0 django-5.2.8 sqlparse-0.5.3 tzdata-2025.2

[notice] A new release of pip is available: 24.0 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
(myenv) PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2_individual> pip install mysqlclient
Collecting mysqlclient
  Using cached mysqlclient-2.2.7-cp312-cp312-win_amd64.whl.metadata (4.8 kB)
  Using cached mysqlclient-2.2.7-cp312-cp312-win_amd64.whl (208 kB)
Installing collected packages: mysqlclient
Successfully installed mysqlclient-2.2.7

[notice] A new release of pip is available: 24.0 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
(myenv) PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2_individual> django-admin startproject mi_proyecto
(myenv) PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2_individual> cd mi_proyecto
```

En orden: Se crea entorno virtual; Se activa entorno virtual; Se instala Django en el entorno; Se instala *mysqlclient* para utilizar base de datos MySQL; Se crea proyecto Django; Se ingresa a carpeta del proyecto.

3. Configuración de la conexión a MySQL

Edita el archivo settings.py del proyecto para que la base de datos utilice MySQL.

mi_proyecto/settings.py

```
DATABASES = {
    ...
    'default': {
        ...
        'ENGINE': 'django.db.backends.mysql',
        ...
        'NAME': 'mi_proyecto_db',
        ...
        'USER': 'root',
        ...
        'PASSWORD': 'root',
        ...
        'HOST': 'localhost',
        ...
        'PORT': '3306',
        ...
    }
    ...
}
```

4. Creación de una aplicación Django

Crea una nueva aplicación dentro del proyecto llamada productos. Agrega esta aplicación a la lista INSTALLED_APPS en el archivo settings.py.

```
\Modulo_7\ae2\m7_ae2_individual\mi_proyecto> python manage.py startapp productos
```

mi_proyecto/settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'productos',  
]
```

Primero se crea app en el proyecto, luego se agrega a settings.py

5. Creación del modelo en Django (ORM)

Dentro del archivo models.py de la aplicación productos, define un modelo llamado Producto con los siguientes atributos:

- nombre: Cadena de caracteres única.
- descripcion: Texto largo.
- precio: Número decimal con dos decimales.
- stock: Número entero.
- fecha_creacion: Fecha de creación automática al momento de guardar.

productos/models.py

```
from django.db import models  
  
class Producto(models.Model):  
    nombre = models.CharField(max_length=200, unique=True)  
    descripcion = models.TextField()  
    precio = models.DecimalField(max_digits=10, decimal_places=2)  
    stock = models.IntegerField()  
    fecha_creacion = models.DateTimeField(auto_now_add=True)  
    def __str__(self):  
        return self.nombre
```

6. Aplicación de migraciones

Genera las migraciones correspondientes y aplícalas para que el modelo se cree en la base de datos.

```
(myenv) PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2\m7_ae2_individual\mi_proyecto> python manage.py makemigrations
C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2\m7_ae2_individual\myenv\Lib\site-packages\django\core\management\commands\makemigrations.py performed for database connection 'default': (1049, "Unknown database 'mi_proyecto_db'")
warnings.warn(
Migrations for 'productos':
  productos\migrations\0001_initial.py
    + Create model Producto
(myenv) PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2\m7_ae2_individual\mi_proyecto> python manage.py makemigrations
No changes detected
(myenv) PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2\m7_ae2_individual\mi_proyecto> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, productos, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying productos.0001_initial... OK
  Applying sessions.0001_initial... OK
(myenv) PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2\m7_ae2_individual\mi_proyecto> |
```

En orden: Se crean migraciones (Entrega alerta por no haber creado la DB antes); Se ejecuta migrate para realizar las migraciones a la base de datos (Ya creada).

7. Operaciones CRUD con el ORM de Django

Utiliza la consola interactiva de Django (Django Shell) para realizar las siguientes operaciones:

- a) Crear un producto con valores definidos y guardarlo en la base de datos.

```
(myenv) PS C:\Users\caran\Desktop\Bootcamp_Skillnest\VSCode\Modulo_7\ae2\m7_ae2_individual\mi_proyecto> python manage.py shell
7 objects imported automatically (use -v 2 for details).

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
(InteractiveConsole)
>>> from productos.models import Producto
>>>
>>> producto = Producto(
...     nombre="Laptop HP",
...     descripcion="Laptop de alta gama con procesador Intel i7",
...     precio=999.99,
...     stock=15
... )
>>> producto.save()
>>> print(f"Producto creado con ID: {producto.id}")
Producto creado con ID: 1
>>> |
```

- b) Consultar el producto creado y mostrar sus detalles.

```
>>> # Obtener el producto
>>> producto = Producto.objects.get(id=1)
>>> # O también: producto = Producto.objects.get(nombre="Laptop HP")
>>>
>>> print(f"ID: {producto.id}")
ID: 1
>>> print(f"Nombre: {producto.nombre}")
Nombre: Laptop HP
>>> print(f"Descripción: {producto.descripcion}")
Descripción: Laptop de alta gama con procesador Intel i7
>>> print(f"Precio: ${producto.precio}")
Precio: $999.99
>>> print(f"Stock: {producto.stock}")
Stock: 15
>>> print(f"Fecha de creación: {producto.fecha_creacion}")
Fecha de creación: 2025-11-11 19:23:47.974988+00:00
>>>
```

c) Modificar el precio del producto y guardar los cambios.

```
>>> producto = Producto.objects.get(id=1)
>>> producto.precio = 899.99
>>> producto.save()
>>> print(f"Precio actualizado a: ${producto.precio}")
Precio actualizado a: $899.99
>>>
```

d) Eliminar el producto de la base de datos.

```
>>> producto = Producto.objects.get(id=1)
>>> nombre_eliminado = producto.nombre
>>> producto.delete()
(1, {'productos.Producto': 1})
>>> print(f"Producto '{nombre_eliminado}' eliminado de la base de datos")
Producto 'Laptop HP' eliminado de la base de datos
>>>
>>> print(f"Total de productos: {Producto.objects.count()}")
Total de productos: 0
>>>
```