

Segunda Parte

Proyecto: **victorysports**

lenguaje: **Python**

Framework: **Django**

Editor: **VS code**

Carpeta del Proyecto: **U11_victorysports_0362**

Proyecto: **backend_victorysports**.

App: **app_victorysports**

1 Aqui el modelo [models.py](#) ya esta creado

=====

```
from django.db import models
```

```
# --- Modelo para Producto ---
```

```
class Producto(models.Model):
```

```
    # id_producto int pk - Igual que antes, se define explícitamente si se desea ese nombre
```

```
    id_producto = models.AutoField(primary_key=True)
```

```
    nombre = models.CharField(max_length=255)
```

```
    precio_unitario = models.DecimalField(max_digits=10, decimal_places=2)
```

```
    stock = models.IntegerField(default=0)
```

```
    marca = models.CharField(max_length=100)
```

```
    img_url = models.URLField(max_length=255, blank=True, null=True) # URLField para URLs,
permite vacío
```

```
    categoria = models.CharField(max_length=50)
```

```
    color = models.CharField(max_length=50)
```

```
class Meta:
```

```
verbose_name = "Producto"
```

```
verbose_name_plural = "Productos"
```

```
def __str__(self):
```

```
    return f"{self.nombre} ({self.marca})"
```

=====

2 Procedimiento para realizar las migraciones(makemigrations y migrate.

3 primero trabajamos con el **MODELO: Productos**

4 En view de app_victorysports crear las funciones con sus códigos correspondientes (inicio_victorysports, agregar_producto, actualizar_producto, realizar_actualizacion_producto, borrar_producto)

5 Modificar el archivo navbar.html que esta dentro de templates para actualizar el menu productos con submenu de productos(Agregar Producto, ver producto, actualizar producto, borrar producto).

6 Crear la subcarpeta carpeta producto dentro de **app_victorysports\templates**.

7 Crear los archivos html con su codigo correspondientes de (agregar_producto.html, ver_producto.html mostrar en tabla con los botones ver, editar y borrar, actualizar_producto.html, borrar_producto.html) dentro de **app_victorysports\templates\producto**.

8 No utilizar forms.py.

9 procedimiento para agregar a el archivo urls.py en app_victorysports con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en productos.

10 procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.

11 por lo pronto solo trabajar con "productos"

12 Sigue el estilo que ya me habias dado.

13 No validar entrada de datos.

14 Al inicio crear la estructura completa de carpetas y archivos actualizado, donde se muestra la carpeta productos dentro de templates con los archivos html correspondientes a la operación crud productos.

15 proyecto totalmente funcional.

16 finalmente ejecutar servidor en el puerto puerto 8390.