

Tercera parte

Proyecto: **victorysports**.

lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

Carpeta del Proyecto: **UIII_victorysports_0362**

Proyecto: **backend_victorysports**

aplicación: **app_victorysports**

1 Aqui el modelo models.py que ya esta creado

```
# =====
```

```
# --- Modelo para Producto Proveedor (Tabla Intermedia/Many-to-Many con Datos Adicionales)
---
```

```
class ProductoProveedor(models.Model):
    producto = models.ForeignKey(Producto, on_delete=models.CASCADE,
                                related_name='relaciones_proveedor')
    proveedor = models.ForeignKey(Proveedor, on_delete=models.CASCADE,
                                related_name='relaciones_producto')
    precio_compra = models.DecimalField(max_digits=10, decimal_places=2)
    fecha_ultima_compra = models.DateField(null=True, blank=True)
    cantidad_comprada = models.IntegerField(default=1)
    referencia_pedido = models.CharField(max_length=50, blank=True, null=True)
    es_proveedor_activo = models.BooleanField(default=True)

    class Meta:
        unique_together = (('producto', 'proveedor'),)
        verbose_name = "Producto por Proveedor"
        verbose_name_plural = "Productos por Proveedores"

    def __str__(self):
        return f"Relación: {self.producto.nombre} - {self.proveedor.nombre_empresa}"
```

2 Procedimiento para realizar las migraciones(makemigrations y migrate.

3 Ahora trabajamos con el MODELO: Producto Proveedor

4 En view de app_victorysports crear las funciones con sus códigos correspondientes (agregar_conexion, actualizar_conexion, realizar_actualizacion_conexion borrar_conexion)

5 En la opcion de agregar_conexion, la tabla Producto Proveedor está relacionada con producto y proveedor, primero se captura los datos no relacionados de producto y luego en los campos de proveedor debe aparecer un combobox con los proveedores por su nombre y poder seleccionar una de ellas, en el campo asignar sala debe aparecer un combobox con los proveedores disponibles por su nombre y poder seleccionar una de

ellas 6 Modificar el archivo navbar.html que está dentro de templates, para actualizar la opción "Producto Proveedor"

7 Crear la subcarpeta **producto_proveedor** dentro de **app_victorysports/templates**.

8 crear los archivos html con su código correspondientes de (agregar_conexion .html, ver_conexion.html mostrar en tabla con los botones ver, editar y borrar, actualizar_conexion.html,

borrar_conexion.html) dentro de app_victorysports/templates/producto_proveedor

9 No utilizar forms.py.

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

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

11 trabajando con todos los modelos

12 Utilizar el mismo diseño que ya estábamos trabajando

13 No validar entrada de datos.

14 Al inicio crear la estructura completa de carpetas y archivos actualizados, donde se muestra la carpeta películas dentro de templates con los archivos html correspondientes a las operaciones crud películas.

15 proyecto totalmente funcional.

16 finalmente ejecutar servidor en el puerto puerto 8362.