

## Tercera parte

Proyecto: **victorysports**.

Lenguaje: **Python**.

Framework: **Django**.

Editor: **VS code**.

Carpeta del Proyecto: **UIII\_victoriesports\_0362**

Proyecto: **backend\_victoriesports**

aplicación: **app\_victoriesports**

1 Aquí el modelo models.py que ya está 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\_victoriesports crear las funciones con sus códigos correspondientes ( agregarConexion, actualizarConexion, realizarActualizacionConexion borrarConexion)

**5 En la opción de agregarConexion, 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 asignarSala 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 opcion "Producto Proveedor"  
"

7 Crear la subcarpeta **producto\_proveedor** dentro de **app\_victorystops\templates**.

8 crear los archivos html con su codigo correspondientes de  
(agregarConexion .html, verConexion.html mostrar en tabla con  
los botones ver, editar y borrar, actualizarConexion.html,

borrarConexion.html) dentro de app\_victorystops\templates\producto\_proveedor

9 No utilizar forms.py.

10 procedimiento para agregar en urls.py en app\_victorystops 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 estabamos 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.