

Tercera Parte

- Proyecto: Parisina.
- Lenguaje: Python.
- Framework: Django.
- Editor: VS code.
- Carpeta del Proyecto: UIII_Parisina_0777
- Proyecto: backend_Parisina
- aplicación: app_Parisina

1 Aquí el modelo models.py que ya está creado

```
# =====  
# MODELO: PRODUCTO  
# =====  
class Producto(models.Model):  
    nombre = models.CharField(max_length=150)  
    descripcion = models.TextField()  
    categoria = models.CharField(max_length=100)  
    precio = models.DecimalField(max_digits=10, decimal_places=2)  
    stock = models.PositiveIntegerField()  
    cliente = models.ForeignKey(Cliente, on_delete=models.CASCADE,  
related_name='productos') # Relación 1 a muchos  
    proveedores = models.ManyToManyField(Proveedor, related_name='productos')  
# Relación muchos a muchos  
  
    def __str__(self):  
        return self.nombre  
=====
```

2 Procedimiento para realizar las migraciones(makemigrations y migrate.

3 Ahora trabajamos con el MODELO: PRODUCTO

4 En view de app_Parisina crear las funciones con sus códigos

correspondientes (agregar_producto, actualizar_producto, realizar_actualizacion_producto, borrar_producto)

5 En la opción de agregar_producto, la tabla Producto está relacionada con Proveedor y Cliente, primero se captura los datos no relacionados de Producto y luego en los campos de Cliente debe aparecer un combobox con los clientes 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” en submenú de Proveedor (Agregar producto, ver producto en forma de tabla, actualizar producto, borrar producto)

7 Crear la subcarpeta producto dentro de app_Parisina\templates.

8 crear los archivos html con su código 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_Parisina\templates\producto

9 No utilizar forms.py.

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

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

11 trabajando con todos los modelos

12 Utilizar colores suaves, atractivos y modernos, el código de las páginas web sencillas para película.

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 producto.

15 proyecto totalmente funcional.

16 finalmente ejecutar servidor en el puerto puerto 0559.