

Proyecto: TiendaAbarrotes

Adaptado por: Angel Salinas Pérez 5J

Lenguaje: Python

Framework: Django

Editor: VS Code

Procedimientos de Configuración

1. Procedimiento para crear carpeta del Proyecto: UIII_Tienda_0777
2. Procedimiento para abrir VS Code sobre la carpeta UIII_Tienda_0777
3. Procedimiento para abrir terminal en VS Code
4. Procedimiento para crear carpeta entorno virtual '.venv' desde terminal de VS Code
5. Procedimiento para activar el entorno virtual
6. Procedimiento para activar intérprete de Python
7. Procedimiento para instalar Django
8. Procedimiento para crear proyecto backend_Tienda sin duplicar carpeta
9. Procedimiento para ejecutar servidor en el puerto 8033
10. Procedimiento para copiar y pegar el link en el navegador
11. Procedimiento para crear aplicación app_Tienda

Modelos (models.py)

Aquí están los modelos ajustados para tu tienda de abarrotes, basados en tu diagrama:

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

```
# MODELO: CATEGORÍA
```

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

```
    nombre = models.CharField(max_length=100, unique=True)
```

```
    descripcion = models.TextField(blank=True, null=True)
```

```
    pasillo = models.IntegerField(blank=True, null=True)
```

```
    responsable_area = models.CharField(max_length=100, blank=True, null=True)
```

```
    fecha_creacion = models.DateField(auto_now_add=True)
```

```
    activa = models.BooleanField(default=True)
```

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

```
        return self.nombre
```

```
# MODELO: PROVEEDOR
```

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

```

nombre_empresa = models.CharField(max_length=150, unique=True)
nombre_contacto = models.CharField(max_length=100, blank=True, null=True)
telefono = models.CharField(max_length=20, blank=True, null=True)
email = models.EmailField(max_length=100, blank=True, null=True)
direccion = models.TextField(blank=True, null=True)
rfc = models.CharField(max_length=13, blank=True, null=True)

def __str__(self):
    return self.nombre_empresa

# MODELO: PRODUCTO
class Producto(models.Model):
    nombre = models.CharField(max_length=150)
    descripcion = models.TextField(blank=True, null=True)
    precio_venta = models.DecimalField(max_digits=10, decimal_places=2)
    stock = models.IntegerField(default=0)
    codigo_barras = models.CharField(max_length=100, unique=True, blank=True, null=True)
    categoria = models.ForeignKey(Categoria, on_delete=models.SET_NULL, null=True,
related_name="productos")
    proveedores = models.ManyToManyField(Proveedor, related_name="productos")

    def __str__(self):
        return self.nombre

```

Siguientes Pasos del Desarrollo

- 12.5. Procedimiento para realizar las migraciones (makemigrations y migrate)
13. Primero trabajamos con el MODELO: CATEGORÍA
14. En views.py de app_Tienda crear las funciones con sus códigos correspondientes (inicio_tienda, agregar_categoria, actualizar_categoria, realizar_actualizacion_categoria, borrar_categoria)
15. Crear la carpeta "templates" dentro de "app_Tienda"
16. En la carpeta templates crear los archivos html (base.html, header.html, navbar.html, footer.html, inicio.html)
17. En el archivo base.html agregar bootstrap para css y js
18. En el archivo navbar.html incluir las opciones ("Sistema de Administración Abarrotes", "Inicio", "Categorías", en submenú de Categorías (Agregar Categoría, Ver Categorías, etc.), "Productos" (Agregar Producto, Ver Productos, etc.), "Proveedores" (Agregar Proveedor, Ver Proveedores, etc.), incluir iconos a las opciones principales, no en los submenú)
19. En el archivo footer.html incluir derechos de autor, fecha del sistema y "Creado por Angel Salinas Pérez 5J" y mantenerla fija al final de la página

20. En el archivo inicio.html se usa para colocar información del sistema más una imagen tomada desde la red sobre la tienda de abarrotes
21. Crear la subcarpeta categoria dentro de app_Tienda\templates
22. Crear los archivos html con su código correspondiente de (agregar_categoria.html, ver_categorias.html, actualizar_categoria.html, borrar_categoria.html) dentro de app_Tienda\templates\categoria
23. No utilizar forms.py
24. Procedimiento para crear el archivo urls.py en app_Tienda con el código correspondiente para acceder a las funciones de views.py
25. Procedimiento para agregar app_Tienda en settings.py de backend_Tienda
26. Realizar las configuraciones correspondientes a urls.py de backend_Tienda para enlazar con app_Tienda
27. Procedimiento para registrar los modelos (Categoria, Producto, Proveedor) en admin.py y volver a realizar las migraciones
28. Por lo pronto solo trabajar con "categoría". Dejar pendiente # MODELO: PRODUCTO y # MODELO: PROVEEDOR
29. Utilizar colores suaves, atractivos y modernos
30. No validar entrada de datos
31. Al inicio crear la estructura completa de carpetas y archivos
32. Proyecto totalmente funcional
33. Finalmente ejecutar servidor en el puerto 8033