

Proyecto: Abarrotes

Lenguaje: Python.

Framework: Django.

Editor: VS code.

- 1 Procedimiento para crear carpeta del Proyecto: UIII_Abarrotes_0633
- 2 procedimiento para abrir vs code sobre la carpeta UIII_Abarrotes_0633
- 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_Abarrotes sin duplicar carpeta.
- 9 procedimiento para ejecutar servidor en el puerto 8036
- 10 procedimiento para copiar y pegar el link en el navegador.
- 11 procedimiento para crear aplicación app_Abarrotes
- 12 Aquí el modelo models.py

=====

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

=====

MODELO: EMPLEADO

=====

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

```
nombre = models.CharField(max_length=255)
puesto = models.CharField(max_length=100)
salario_hora = models.DecimalField(max_digits=8, decimal_places=2)
fecha_contratacion = models.DateField()
horas_semanales = models.PositiveIntegerField()
departamento = models.CharField(max_length=100)
```

text

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

```
    return f'{self.nombre} - {self.puesto}'
```

=====

MODELO: CLIENTE

=====

```
class Cliente(models.Model):
    nombre = models.CharField(max_length=255)
    telefono = models.CharField(max_length=20)
    correo = models.CharField(max_length=255)
    fecha_ultima_compra = models.DateField()
    direccion = models.CharField(max_length=255)
    id_empleado = models.ForeignKey(Empleado,
        on_delete=models.CASCADE, related_name="clientes")
```

text

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

```
    return self.nombre
```

```
=====
```

MODELO: PRODUCTO

```
=====
```

```
class Producto(models.Model):  
    nombre = models.CharField(max_length=150)  
    descripcion = models.TextField(blank=True, null=True)  
    precio = models.DecimalField(max_digits=10, decimal_places=2)  
    stock = models.PositiveIntegerField()  
    categoria = models.CharField(max_length=50)  
    fecha_ingreso = models.DateField()  
    activo = models.BooleanField(default=True)
```

```
text
```

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

```
    return self.nombre
```

```
=====
```

MODELO: VENTA

```
=====
```

```
class Venta(models.Model):  
    fecha = models.DateField()  
    total = models.DecimalField(max_digits=10, decimal_places=2)  
    productos_vendidos = models.TextField()  
    id_empleado = models.ForeignKey(Empleado,  
        on_delete=models.CASCADE, related_name="ventas")  
    id_cliente = models.ForeignKey(Cliente, on_delete=models.CASCADE,  
        related_name="ventas")
```

```
id_producto = models.ForeignKey(Producto,  
on_delete=models.CASCADE, related_name="ventas")
```

```
text
```

```
def __str__(self):  
    return f"Venta {self.id} - {self.fecha}"  
  
=====
```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate.

13 primero trabajamos con el MODELO: EMPLEADO

14 En view de app_Abarrotes crear las funciones con sus códigos correspondientes (inicio_abarrotes, agregar_empleado, actualizar_empleado, realizar_actualizacion_empleado, borrar_empleado)

15 Crear la carpeta "templates" dentro de "app_Abarrotes".

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", "Empleados",en submenu de Empleados(Aregar Empleado,ver Empleados, actualizar Empleado, borrar Empleado), "Clientes" en submenu de Clientes(Aregar clientes,ver

clientes, actualizar clientes, borrar clientes) "Productos" en submenu de Productos(Aregar productos,ver productos, actualizar productos, borrar productos), "Ventas" en submenu de Ventas(Aregar ventas,ver ventas, actualizar ventas, borrar ventas), incluir iconos a las opciones principales, no en los submenu.

19 En el archivo footer.html incluir derechos de autor,fecha del sistema y "Creado por Ing. Eliseo Nava, Cbtis 128" 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 abarros.

21 Crear la subcarpeta carpeta empleado dentro de app_Abarrotes\templates.

22 crear los archivos html con su codigo correspondientes de (agregar_empleado.html, ver_empleados.html mostrar en tabla con los botones ver, editar y borrar, actualizar_empleado.html, borrar_empleado.html) dentro de app_Abarrotes\templates\empleado.

23 No utilizar forms.py.

24 procedimiento para crear el archivo urls.py en app_Abarrotes con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en empleados.

25 procedimiento para agregar app_Abarrotes en settings.py de backend_Abarrotes

26 realizar las configuraciones correspondiente a urls.py de backend_Abarrotes para enlazar con app_Abarrotes

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

27 por lo pronto solo trabajar con "EMPLEADO" dejar pendiente # MODELO: CLIENTE, # MODELO: PRODUCTO y # MODELO: VENTA

28 Utilizar colores suaves, atractivos y modernos, el código de las páginas web sencillas.

28 No validar entrada de datos.

29 Al inicio crear la estructura completa de carpetas y archivos.

30 proyecto totalmente funcional.

31 finalmente ejecutar servidor en el puerto puerto 8036.