

Primera parte

Proyecto: **Computadoras y accesorios**

lenguaje: Python.

Framework: **Django**.

Editor: **VS code.**

- 1 Procedimiento para crear carpeta del Proyecto: UIII_Computadoras_1291
 - 2 Procedimiento para abrir vs code sobre la carpeta UIII_Computadoras_1291
 - 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_Computadoras sin duplicar carpeta.
 - 9 Procedimiento para ejecutar servidor en el puerto 8018
 - 10 Procedimiento para copiar y pegar el link en el navegador.
 - 11 Procedimiento para crear aplicación app_Computadoras
 - 12 Aquí el modelo models.py

— — — — —

code

Python

```

proveedor = models.ForeignKey(Proveedor, on_delete=models.CASCADE,
related_name='productos')

def __str__(self):
    return self.nombre

# =====
# MODELO: FACTURA
# =====
class Factura(models.Model):
    id = models.IntegerField(primary_key=True)
    fecha = models.DateField()
    total = models.DecimalField(max_digits=10, decimal_places=2)
    metodo_pago = models.CharField(max_length=100)
    cliente_id = models.IntegerField() # Asumiendo que cliente_id es un ID, si fuera una tabla,
necesitaría un ForeignKey
    empleado = models.CharField(max_length=255, blank=True, null=True)
    observaciones = models.TextField(blank=True, null=True)

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

- 12.5 Procedimiento para realizar las migraciones (makemigrations y migrate).
- 13 Primero trabajamos con el MODELO: PRODUCTO
- 14 En views de app_Computadoras crear las funciones con sus códigos correspondientes (inicio_computadoras, agregar_producto, actualizar_producto, realizar_actualizacion_producto, borrar_producto)
- 15 Crear la carpeta “templates” dentro de “app_Computadoras”.
- 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 de Computadoras y Accesorios”, “Inicio”, “Productos”, en submenu de productos (Agregar Producto, Ver Productos, Actualizar Producto, Borrar Producto), “Proveedores” en submenu de Proveedores (Agregar Proveedor, Ver Proveedores, Actualizar Proveedor, Borrar Proveedor), “Facturas” en submenu de Facturas (Agregar Factura, Ver Facturas, Actualizar Factura, Borrar Factura), incluir iconos a las opciones principales, no en los submenús.
- 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 computadoras y accesorios.
- 21 Crear la subcarpeta 'producto' dentro de app_Computadoras\templates.
- 22 Crear los archivos html con su código correspondientes de (agregar_producto.html, ver_productos.html mostrar en tabla con los botones ver, editar y borrar, actualizar_producto.html, borrar_producto.html) dentro de app_Computadoras\templates\producto.
- 23 No utilizar forms.py.

- 24 Procedimiento para crear el archivo urls.py en app_Computadoras con el código correspondiente para acceder a las funciones de views.py para operaciones de CRUD en productos.
- 25 Procedimiento para agregar app_Computadoras en settings.py de backend_Computadoras
- 26 Realizar las configuraciones correspondiente a urls.py de backend_Computadoras para enlazar con app_Computadoras
- 27 Procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.
- 27 Por lo pronto solo trabajar con “Producto”, dejar pendiente # MODELO: PROVEEDOR y # MODELO: FACTURA
- 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 8018.