

EXPERIENCIA CURRICULAR DE TECNOLOGÍAS WEB

MANUAL DE USUARIO - APPVENTAS

UNIVERSIDAD NACIONAL DE TRUJILLO

Facultad de Ingeniería, Escuela de Ingeniería de Sistemas



ALUMNOS:

- Lucano Nieves, David Adrian,
- Villanueva Rosales, Estefany Lucía

DOCENTE: Ing. Sánchez Ticona, Roberto J.

Trujillo - Perú

2025

Índice

1. Módulo de Unidades.....	4
1.1. Listar Unidades.....	4
1.2. Registrar Unidad.....	4
1.3. Editar Unidad.....	5
1.4. Eliminar/Desactivar Unidad.....	5
2. Módulo de Productos.....	6
2.1. Listar Productos.....	6
2.2. Registrar Producto.....	6
2.3. Editar Producto.....	7
2.4. Eliminar/Desactivar Producto.....	8
3. Módulo de Ventas.....	9
3.1. Listado de Ventas.....	9
3.2. Registrar Venta.....	9
3.3. Ver Detalle de Venta.....	15
3.4. Explicación de la implementación:.....	16
4. Anexos.....	37
4.1. Diagrama de Base de Datos.....	37
4.2. Modelos de los mantenedores adicionales.....	38
4.3. Capturas de Pantalla y explicación de código de los mantenedores relevantes:.....	40

Introducción

- Alcance de la aplicación

La presente aplicación tiene como objetivo la gestión de un sistema de ventas básico utilizando el framework Django. El sistema contempla la administración de entidades clave como clientes, categorías de productos, unidades, tipos de comprobantes, parámetros de numeración, productos, y el registro de ventas mediante cabeceras y detalles.

Dentro del módulo de mantenimiento, la aplicación permite agregar, editar y eliminar registros de clientes, productos, categorías y unidades. Estas funcionalidades están disponibles a través de una interfaz web intuitiva, que incluye formularios validados y notificaciones al usuario.

En el módulo de ventas, se pueden registrar nuevas ventas (cabecera y detalle), así como anularlas lógicamente. Durante el registro, se actualiza automáticamente el stock de los productos vendidos y se genera la numeración del comprobante según el tipo seleccionado. La lógica incluye validaciones por stock disponible y control de integridad.

El sistema está diseñado para ejecutarse en un entorno local, utilizando herramientas como WampServer o XAMPP para levantar los servicios para la gestión visual de la base de datos. La aplicación utiliza MySQL para el desarrollo de la base de datos, completamente integrado con Django.

- Librerías Utilizadas

- **Django** – Framework principal para desarrollo web en Python.
- **Bootstrap** – Para diseño de interfaz responsiva y componentes visuales como botones, formularios y notificaciones.
- **crispy-forms** – Para mejorar la presentación de los formularios HTML en Django.
- **SweetAlert2 (Swal)** – Librería de JavaScript utilizada para mostrar alertas emergentes con diseño atractivo y personalizable.
- **Toast (Bootstrap Toast)** – Notificaciones flotantes que informan acciones realizadas en el sistema, como confirmaciones o errores.

1. Módulo de Unidades

1.1. Listar Unidades

LISTADO DE UNIDADES + NUEVO

Buscar por descripción ...

CODIGO	DESCRIPCION	OPCIONES
1	Unid	Editar Eliminar
2	Kilogramo	Editar Eliminar
3	Litro	Editar Eliminar
4	Caja	Editar Eliminar

Copyright © 2014-2020 AdminLTE.io. All rights reserved. Version 3.1.0-rc

1.2. Registrar Unidad

- Campos requeridos:
 - Descripción
 - Estado

AGREGAR UNIDAD

Descripcion*

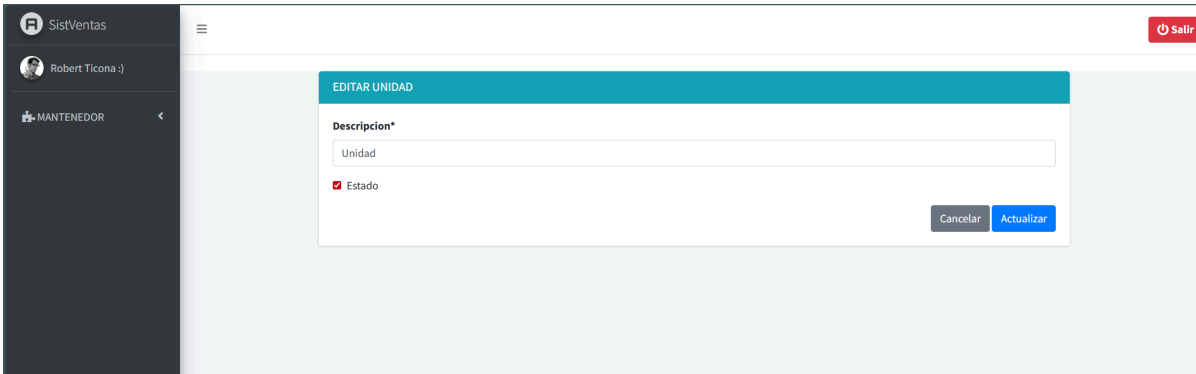
☒ Estado

Cancelar Guardar

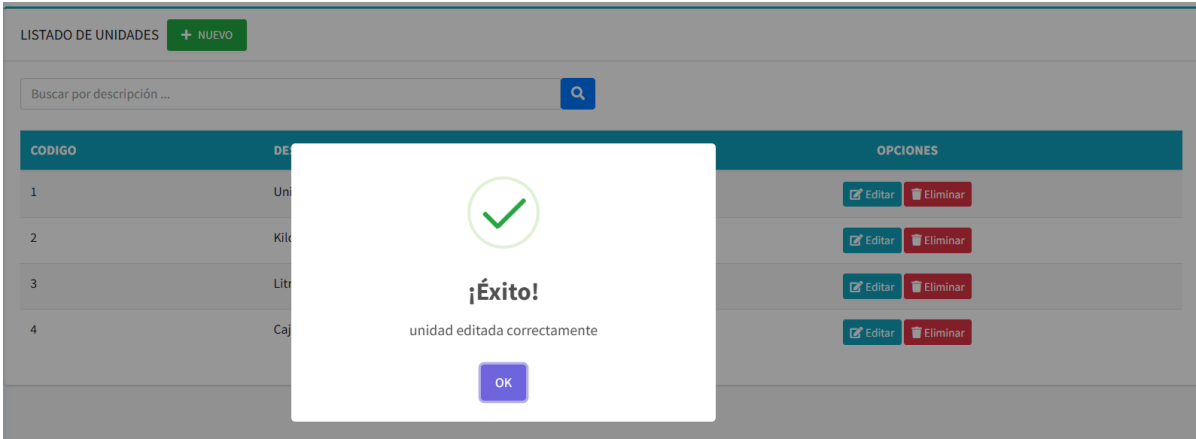
1.3. Editar Unidad

CODIGO	DESCRIPCION	OPCIONES
1	Unidad	 Editar  Eliminar

Ubicándonos en el listado, al hacer click en el botón “Editar”, aparecerá el apartado en el que se puede editar la respectiva unidad



Si clickeamos el botón “Actualizar”, se realizará la actualización pertinente y nos devolverá al listado de unidades, mostrándonos un mensaje de confirmación



1.4. Eliminar/Desactivar Unidad

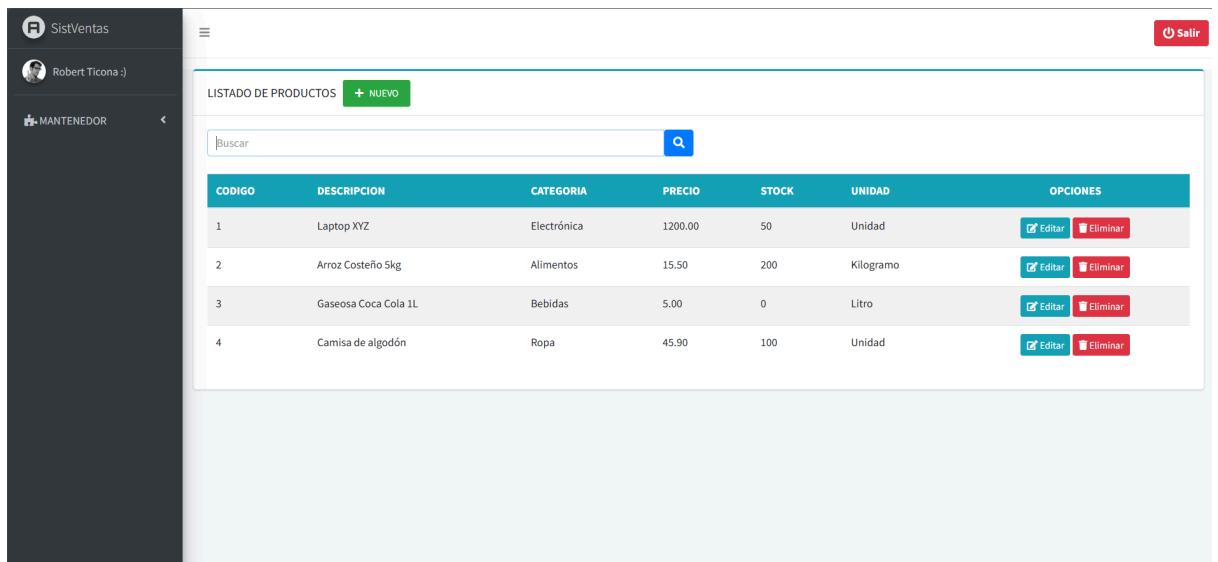
CODIGO	DESCRIPCION	OPCIONES
1	Unidad	 Editar  Eliminar

Al hacer click en el botón “Eliminar”, aparecerá un mensaje para confirmar la decisión de eliminar el registro respectivo



2. Módulo de Productos

2.1. Listar Productos



2.2. Registrar Producto

- Campos requeridos:
 - Descripción
 - Categoría
 - Precio
 - Stock
 - Unidad
 - Estado

The screenshot shows the 'AGREGAR PRODUCTO' (Add Product) form. On the left is a dark sidebar with the logo 'SistVentas', the user 'Robert Ticona :)', and a 'MANTENEDOR' button. The main area has a light blue header with the title 'AGREGAR PRODUCTO'. The form contains several input fields: 'Descripcion*' (text), 'Categoria*' (dropdown), 'Precio*' (text), 'Stock*' (text), and 'Unidad*' (dropdown). There is a checked checkbox for 'Estado'. At the bottom right are 'Cancelar' and 'Guardar' buttons. A red 'Salir' button is in the top right corner of the application.

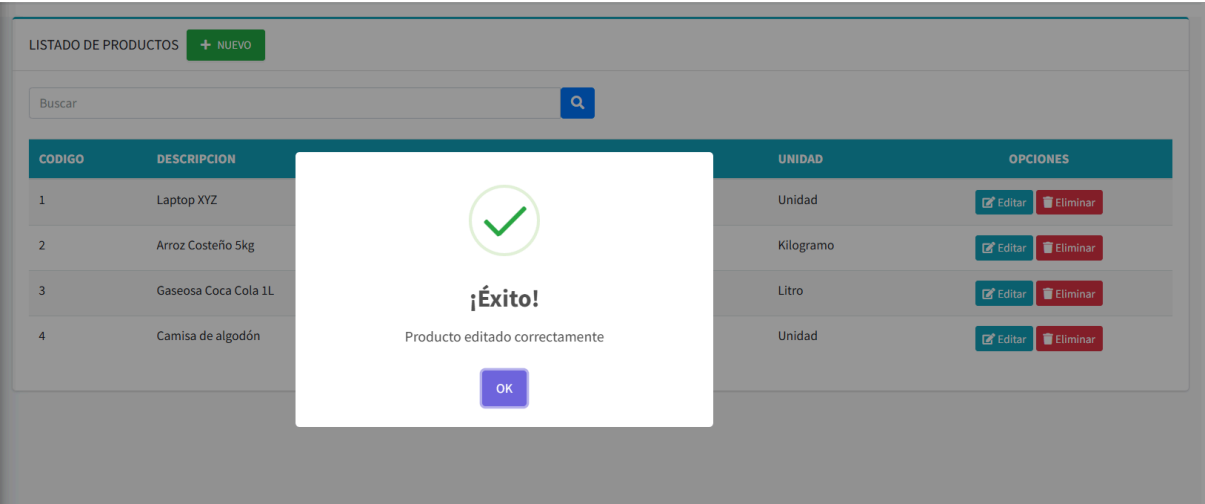
2.3. Editar Producto

CODIGO	DESCRIPCION	CATEGORIA	PRECIO	STOCK	UNIDAD	OPCIONES
1	Laptop XYZ	Electrónica	1200.00	50	Unidad	Editar Eliminar

Ubicándonos en el listado, al hacer click en el botón “Editar”, aparecerá el apartado en el que se puede editar el respectivo producto

The screenshot shows the 'EDITAR PRODUCTO' (Edit Product) form. The sidebar is identical to the previous screenshot. The main area has a light blue header with the title 'EDITAR PRODUCTO'. The form fields are pre-filled with data from the table: 'Descripcion*' is 'Laptop XYZ', 'Categoria*' is 'Electrónica', 'Precio*' is '1200.00', 'Stock*' is '50', and 'Unidad*' is 'Unidad'. The 'Estado' checkbox is checked. At the bottom right are 'Cancelar' and 'Actualizar' buttons. A red 'Salir' button is in the top right corner of the application.

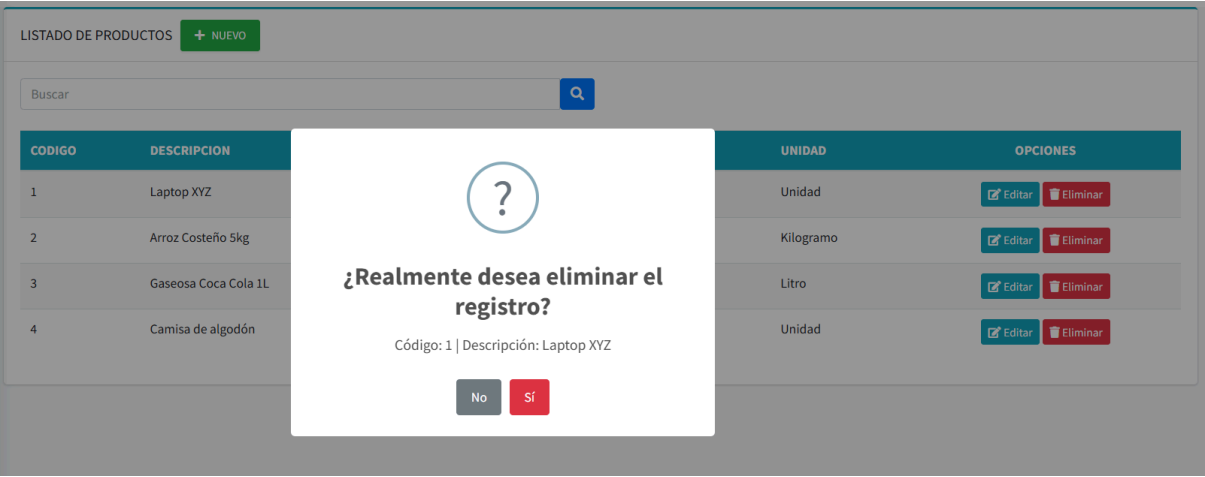
Si clickeamos el botón “Actualizar”, se realizará la actualización pertinente y nos devolverá al listado de productos, mostrándonos un mensaje de confirmación



2.4. Eliminar/Desactivar Producto

CODIGO	DESCRIPCION	CATEGORIA	PRECIO	STOCK	UNIDAD	OPCIONES
1	Laptop XYZ	Electrónica	1200.00	50	Unidad	Editar Eliminar

Al hacer click en el boton “Eliminar”, aparecerá un mensaje para confirmar la decisión de eliminar el registro respectivo



3. Módulo de Ventas

3.1. Listado de Ventas

CÓDIGO	DESCRIPCIÓN	FECHA	RUC/DNI	NOMBRES/RAZÓN	TOTAL	OPCIONES
3	Boleta	July 1, 2025	98765432	Maria Lopez	30.0	Eliminar Ver detalles
4	Factura	July 1, 2025	45678912	Carlos García	40.0	Eliminar Ver detalles
5	Boleta	July 1, 2025	98765432	Maria Lopez	1430.0	Eliminar Ver detalles
8	Boleta	July 2, 2025	45678912	Carlos García	62.0	Eliminar Ver detalles
9	Boleta	July 2, 2025	98765432	Maria Lopez	168.3	Eliminar Ver detalles

3.2. Registrar Venta

Registrar Venta

Fecha: 01/07/2025 Tipo: - Seleccione Tipo - Nro. Documento: []

Cliente: - Seleccione Cliente - RUC/DNI: [] Dirección: []

Producto: - Seleccione Producto - Unidad: [] Precio: [] Cantidad: []

[Agregar al carrito](#)

Opciones	Código	Descripción	Unidad	Cantidad	P. Venta	Importe
----------	--------	-------------	--------	----------	----------	---------

Total S/. []

[Registrar](#) [Cancelar](#)

- Selección del Tipo de Venta (Factura/Boleta)

Tipo	Nro. Documento
Factura	00011
- Seleccione Tipo -	RUC/DNI
Factura	98765432
Boleta	
Unidad	Precio

- Selección de cliente.

Cliente

- Seleccione Cliente -

- Seleccione Cliente -

Juan Pérez

Maria Lopez

Carlos García

- Agregar productos al detalle.

Producto

- Seleccione Producto -

- Seleccione Producto -

Laptop XYZ

Arroz Costeño 5kg

Gaseosa Coca Cola 1L

Camisa de algodón

Registrar

- Listado de Productos en el carrito

Agregar al carrito

Opciones	Código	Descripción	Unidad	Cantidad	P. Venta	Importe
<div><div></div><div></div></div>	<div><div></div><div>3</div></div>	Gaseosa Coca Cola 1L	<div><div></div><div>Litro</div></div>	<div><div></div><div>1</div></div>	<div><div></div><div>5.00</div></div>	5.00
<div><div></div><div></div></div>	<div><div></div><div>4</div></div>	Camisa de algodón	<div><div></div><div>Unidad</div></div>	<div><div></div><div>4</div></div>	<div><div></div><div>45.90</div></div>	183.60
<div><div></div><div></div></div>	<div><div></div><div>1</div></div>	Laptop XYZ	<div><div></div><div>Unidad</div></div>	<div><div></div><div>2</div></div>	<div><div></div><div>1200.00</div></div>	2,400.00

Total S/.

2,588.60

Registrar

Cancelar

- Validaciones del boton “Agregar al carrito”:

Error! Por favor seleccione el Tipo.

Fecha

01/07/2025

Tipo

- Seleccione Tipo -

Nro. Documento

00011

Cliente

Juan Pérez

RUC/DNI

12345678

Dirección

Av. Siempre Viva 123

Producto

Gaseosa Coca Cola 1L

Unidad

Litro

Precio

5.00

Cantidad

3

Agregar al carrito

Error! Por favor seleccione el Cliente.

Fecha

01/07/2025

Tipo

Factura

Nro. Documento

00011

Cliente

- Seleccione Cliente -

RUC/DNI

Dirección

Producto

Gaseosa Coca Cola 1L

Unidad

Litro

Precio

5.00

Cantidad

3

Agregar al carrito

Error! Por favor seleccione el Producto.

Fecha

01/07/2025

Tipo

Factura

Nro. Documento

00011

Cliente

Juan Pérez

RUC/DNI

12345678

Dirección

Av. Siempre Viva 123

Producto

- Seleccione Producto -

Unidad

Litro

Precio

5.00

Cantidad

3

Agregar al carrito

Error! Por favor ingrese cantidad del producto.

Fecha

01/07/2025

Tipo

Factura

Nro. Documento

00011

Cliente

Maria Lopez

RUC/DNI

98765432

Dirección

Calle Falsa 456

Producto

Gaseosa Coca Cola 1L

Unidad

Litro

Precio

5.00

Cantidad

0

Error! No puede volver a vender el mismo producto.

Fecha

01/07/2025

Tipo

Factura

Nro. Documento

00011

Cliente

Maria Lopez

RUC/DNI

98765432

Dirección

Calle Falsa 456

Producto

Gaseosa Coca Cola 1L

Unidad

Litro

Precio

5.00

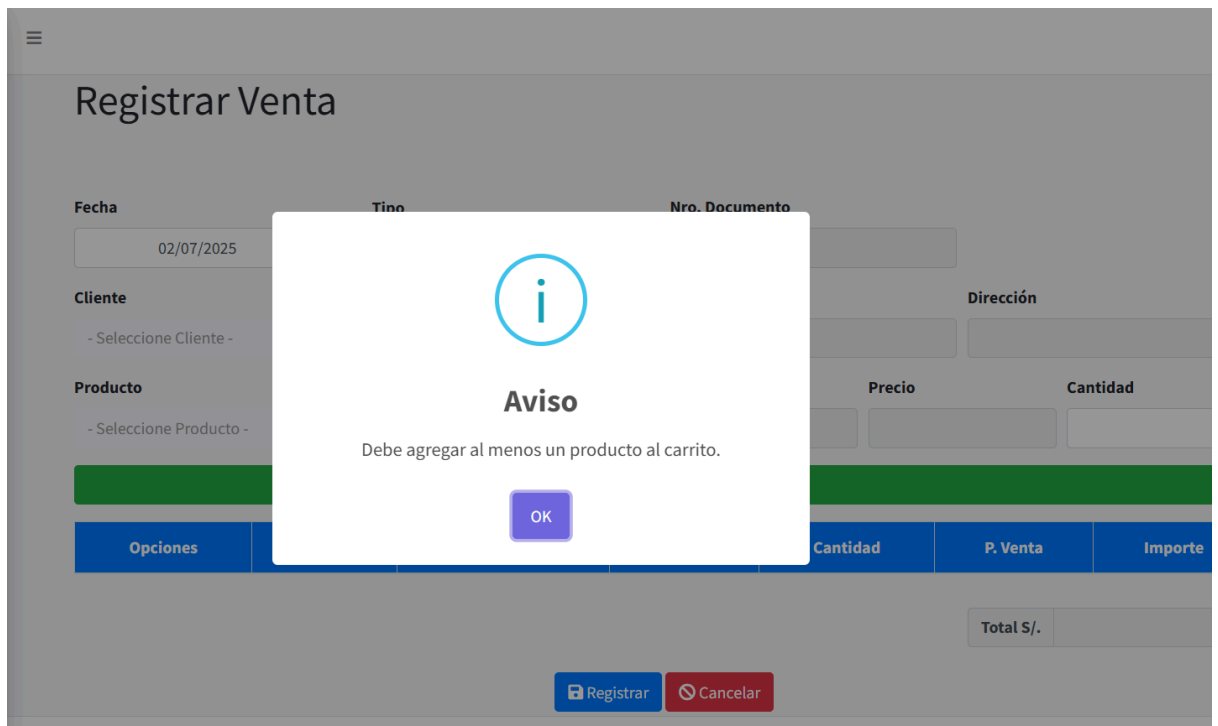
Cantidad

3

Agregar al carrito

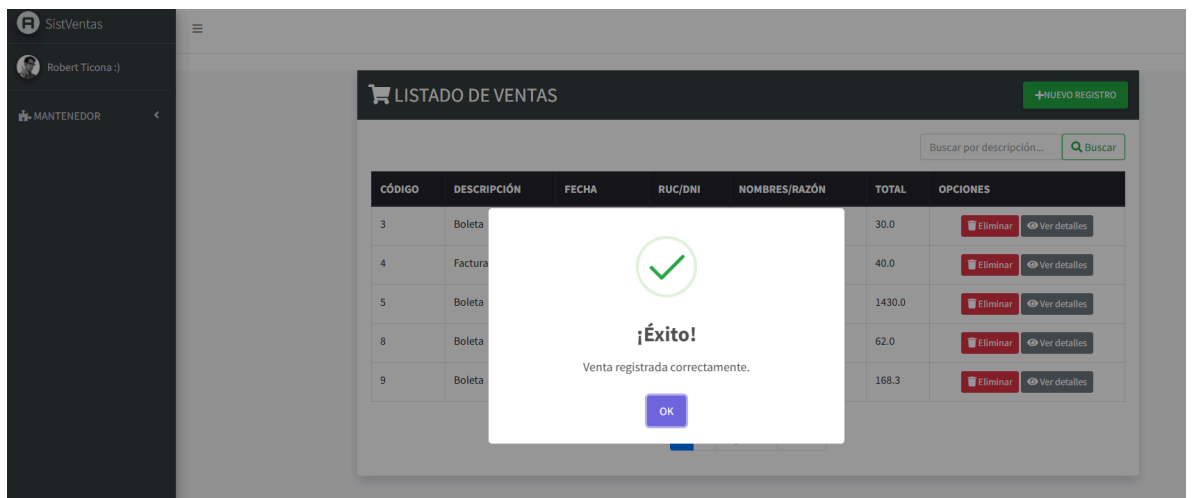
Opciones	Código	Descripción	Unidad	Cantidad	P. Venta	Importe
✖	3	Gaseosa Coca Cola 1L	Litro	1	5.00	5.00

- Validaciones del botón “Registrar”:



Se verifica que haya al menos un producto ingresado al carrito.

- Registro de la Venta:



Actualización del Stock:

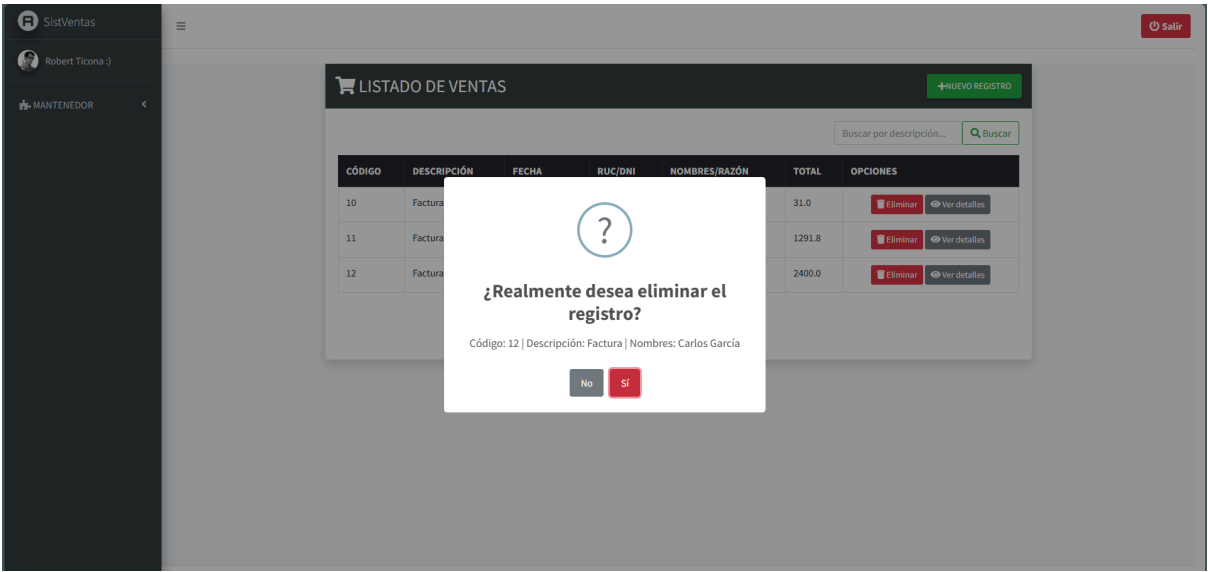
Antes:

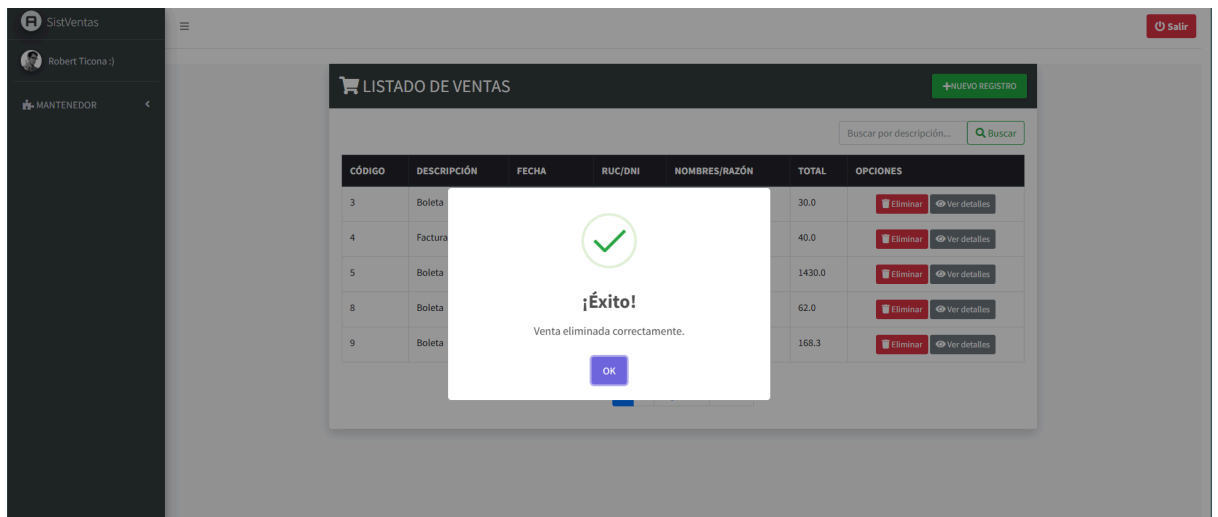
CODIGO	DESCRIPCION	CATEGORIA	PRECIO	STOCK	UNIDAD	OPCIONES
1	Laptop XYZ	Electrónica	1200.00	50	Unidad	Editar Eliminar
2	Arroz Costeño 5kg	Alimentos	15.50	200	Kilogramo	Editar Eliminar
3	Gaseosa Coca Cola 1L	Bebidas	5.00	6	Litro	Editar Eliminar
4	Camisa de algodón	Ropa	45.90	100	Unidad	Editar Eliminar

Después:

CODIGO	DESCRIPCION	CATEGORIA	PRECIO	STOCK	UNIDAD	OPCIONES
1	Laptop XYZ	Electrónica	1200.00	48	Unidad	Editar Eliminar
2	Arroz Costeño 5kg	Alimentos	15.50	200	Kilogramo	Editar Eliminar
3	Gaseosa Coca Cola 1L	Bebidas	5.00	5	Litro	Editar Eliminar
4	Camisa de algodón	Ropa	45.90	96	Unidad	Editar Eliminar

3.3. Eliminación de la Venta:





Actualización del Stock:

Antes:

CODIGO	DESCRIPCION	CATEGORIA	PRECIO	STOCK	UNIDAD	OPCIONES
1	Laptop XYZ	Electrónica	1200.00	48	Unidad	Editar Eliminar
2	Arroz Costeño 5kg	Alimentos	15.50	200	Kilogramo	Editar Eliminar
3	Gaseosa Coca Cola 1L	Bebidas	5.00	5	Litro	Editar Eliminar
4	Camisa de algodón	Ropa	45.90	96	Unidad	Editar Eliminar

Después:

CODIGO	DESCRIPCION	CATEGORIA	PRECIO	STOCK	UNIDAD	OPCIONES
1	Laptop XYZ	Electrónica	1200.00	50	Unidad	Editar Eliminar
2	Arroz Costeño 5kg	Alimentos	15.50	200	Kilogramo	Editar Eliminar
3	Gaseosa Coca Cola 1L	Bebidas	5.00	6	Litro	Editar Eliminar
4	Camisa de algodón	Ropa	45.90	100	Unidad	Editar Eliminar

3.4. Ver Detalle de Venta

- Información completa de la operación.

Al hacer click sobre el boton “Ver detalles”

4	Factura	July 1, 2025	45678912	Carlos García	40.0	Eliminar Ver detalles
---	---------	--------------	----------	---------------	------	---

Nos cargará la siguiente vista de detalles, con los datos de la venta y cada detalleventa:


Detalles de la Venta N° 00010
+ Nuevo Registro

Información del Cliente
 Cliente: Carlos García
 RUC/DNI: 45678912
 Dirección: Jr. Unión 789

Datos de la Venta
 Fecha: 01/07/2025
 Estado: Activa

#	Producto	Unidad	Cantidad	Precio	Importe
1	Gaseosa Coca Cola 1L	Litro	8.0	5.00	8 × 5.00 = 40.00
SUBTOTAL					S/ 33.90
IGV					S/ 6.10
TOTAL					S/ 40.00

← Volver

3.5. Explicación de la implementación:

- Modelo de Tipo:

```

64 class Tipo(models.Model):
65     idtipo = models.AutoField(primary_key=True)
66     descripcion = models.CharField(max_length=20, unique=True)
67     estado = models.BooleanField(default=True)
68     class Meta:
69         db_table = 'tipos'
  
```

- Modelo de Parámetro:

```

71 class Parametro(models.Model):
72     tipo = models.OneToOneField(
73         Tipo,
74         on_delete=models.CASCADE,
75         db_column='idtipo',
76         primary_key=True
77     )
78     numeracion = models.CharField(max_length=15, unique=True)
79     serie = models.CharField(max_length=3)
80     class Meta:
81         db_table = 'parametros'
  
```


- Modelo de CabeceraVenta:

```
84 class CabeceraVenta(models.Model):
85     idventa = models.AutoField(primary_key=True)
86     cliente = models.ForeignKey(
87         Cliente,
88         on_delete=models.CASCADE,
89         db_column='idcliente'
90     )
91     fecha_venta = models.DateField()
92     tipo = models.ForeignKey(
93         Tipo,
94         on_delete=models.CASCADE,
95         db_column='idtipo'
96     )
97     total = models.FloatField()
98     subtotal = models.FloatField()
99     igv = models.FloatField()
100     nrodoc = models.CharField(max_length=12)
101     estado = models.BooleanField(default=True)
102     class Meta:
103         db_table = 'cabeceraventas'
```


- Modelo de DetalleVenta:

```
105 class DetalleVenta(models.Model):
106     pk = models.CompositePrimarykey("producto", "venta")
107     venta = models.ForeignKey(
108         CabeceraVenta,
109         on_delete=models.CASCADE,
110         db_column='idventa'
111     )
112     producto = models.ForeignKey(
113         Producto,
114         on_delete=models.CASCADE,
115         db_column='idproducto'
116     )
117     precio = models.FloatField()
118     cantidad = models.FloatField()
119
120     class Meta:
121         db_table = 'detalleventas'
```

Añadimos todos los modelos de las entidades que participan en el proceso de venta. Luego, realizamos las migraciones (makemigrations) y aplicamos los cambios a la base de datos con migrate para que las tablas nuevas se creen correctamente.

“Listado de Ventas”

Vista:

 LISTADO DE VENTAS

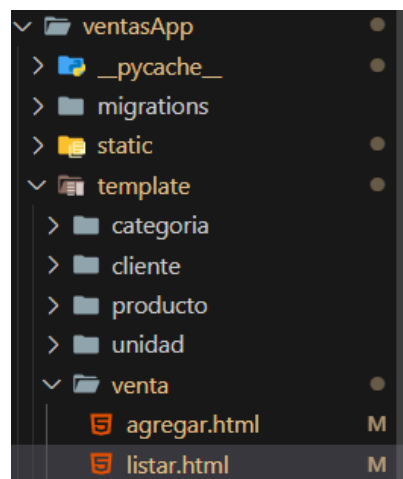
[+NUEVO REGISTRO](#)

[Buscar](#)

CÓDIGO	DESCRIPCIÓN	FECHA	RUC/DNI	NOMBRES/RAZÓN	TOTAL	OPCIONES
3	Boleta	July 1, 2025	98765432	Maria Lopez	30.0	Eliminar
4	Factura	July 1, 2025	45678912	Carlos García	40.0	Eliminar
5	Boleta	July 1, 2025	98765432	Maria Lopez	1430.0	Eliminar

Código:

Nos ubicamos en la carpeta venta, en el archivo listar.html



Colocamos el siguiente código:

```
{% extends "plantilla.html" %}

{% block title %}
Listado de Ventas
{% endblock %}

{% block content %}
<div class="container mt-4">
  <div class="card shadow border-0">
    <!-- Cabecera oscura -->
    <div class="p-3 bg-dark text-white d-flex justify-content-between align-items-center flex-wrap">
      <h3 class="mb-0">
        <i class="fas fa-shopping-cart"></i> LISTADO DE VENTAS
      </h3>
      <a href="{% url 'agregarventa' %}" class="btn btn-success btn-sm d-flex align-items-center fw-bold shadow-sm px-3 py-2">
        <i class="fas fa-plus me-2"></i> NUEVO REGISTRO
      </a>
    </div>
  </div>
</div>
```

```

<div class="card-body">
  <!-- Buscador alineado a la derecha -->
  <div class="d-flex justify-content-end mb-3">
    <form method="GET" class="form-inline">
      <div class="input-group">
        <input type="text" name="buscar" id="buscar" class="form-control" placeholder="Buscar por descripción..."
          value="{{ buscar|default_if_none:'' }}">
        <button class="btn btn-outline-success" type="submit">
          <i class="fas fa-search"></i> Buscar
        </button>
      </div>
    </form>
  </div>
</div>

```

```

<!-- Tabla -->
<div class="table-responsive">
  <table class="table table-hover table-bordered">
    <thead class="thead-dark">
      <tr>
        <th>CÓDIGO</th>
        <th>DESCRIPCIÓN</th>
        <th>FECHA</th>
        <th>RUC/DNI</th>
        <th>NOMBRES/RAZÓN</th>
        <th>TOTAL</th>
        <th>OPCIONES</th>
      </tr>
    </thead>
    <tbody>
      {% if venta %}
      {% for itemventa in venta %}
      <tr>
        <td>{{ itemventa.idventa }}</td>
        <td>{{ itemventa.tipo.descripcion }}</td>
        <td>{{ itemventa.fecha_venta }}</td>
        <td>{{ itemventa.cliente.ruc_dni }}</td>
        <td>{{ itemventa.cliente.nombres }}</td>
        <td>{{ itemventa.total }}</td>
        <td class="text-center">
          <a href="#" onclick="eliminarVenta(event, '{{ itemventa.idventa }}', '{{ itemventa.tipo.descripcion }}', '{{ itemventa.cliente.nombres }}')"
            class="btn btn-danger btn-sm">
            <i class="fas fa-trash"></i> Eliminar
          </a>
        </td>
      </tr>
      {% endfor %}
      {% else %}
      <tr class="text-center">
        <td colspan="7"><p>No hay registros.</p></td>
      </tr>
      {% endif %}
    </tbody>
  </table>
</div>

```

```

<!-- Paginación -->
{% if venta.has_other_pages %}
  <div class="d-flex justify-content-center mt-3">
    <nav>
      <ul class="pagination">
        {% if venta.has_previous %}
        <li class="page-item">
          <a class="page-link" href="?page=1{% if buscar %}&buscar={{ buscar }}{% endif %}">Primero</a>
        </li>
        <li class="page-item">
          <a class="page-link" href="?page={{ venta.previous_page_number }}{% if buscar %}&buscar={{ buscar }}{% endif %}">Anterior</a>
        </li>
        {% endif %}

        {% for num in venta.paginator.page_range %}
        {% if venta.number == num %}
        <li class="page-item active"><span class="page-link">{{ num }}</span></li>
        {% elif num > venta.number|add:-3 and num < venta.number|add:3 %}
        <li class="page-item">
          <a class="page-link" href="?page={{ num }}{% if buscar %}&buscar={{ buscar }}{% endif %}">{{ num }}</a>
        </li>
        {% endif %}
        {% endfor %}

        {% if venta.has_next %}
        <li class="page-item">
          <a class="page-link" href="?page={{ venta.next_page_number }}{% if buscar %}&buscar={{ buscar }}{% endif %}">Siguiente</a>
        </li>
        <li class="page-item">
          <a class="page-link" href="?page={{ venta.paginator.num_pages }}{% if buscar %}&buscar={{ buscar }}{% endif %}">Último</a>
        </li>
        {% endif %}
      </ul>
    </nav>
  </div>
{% endif %}
</div>
</div>
{% endblock %}

```

Agregamos en [views.py](#) la function para listar las ventas:

```
def listarventa(request):
    queryset=request.GET.get("buscar", "")
    venta=CabeceraVenta.objects.filter(estado=True)
    if queryset:
        venta=CabeceraVenta.objects.filter(Q(tipo__descripcion__icontains=queryset),estado=True).distinct()

    paginator = Paginator(venta, 5)
    page_number = request.GET.get('page')
    page_obj = paginator.get_page(page_number)

    context={'venta':page_obj, 'buscar': queryset}
    return render(request,"venta/listar.html",context)
```

Y así mismo, en [urls.py](#), colocamos la ruta respectiva de 'listarventa/':

```
#Venta


path('listarventa/',listarventa,name="listarventa"),
path('agregarventa/',agregarventa ,name="agregarventa"),
path('eliminarventa/<int:id>/',eliminarventa, name="eliminarventa"),


path('EncontrarProducto/<int:producto_id>/', ProductoCodigo, name="EncontrarProducto"),
path('EncontrarTipo/<int:tipo_id>/', PorTipo, name="EncontrarTipo"),
```

“Registrar Ventas”

Vista:

Registrar Venta

Fecha	Tipo	Nro. Documento	
<input type="text" value="02/07/2025"/>	<div> - Seleccione Tipo -</div>	<input type="text"/>	
Cliente	RUC/DNI	Dirección	
<div>- Seleccione Cliente -</div>	<input type="text"/>	<input type="text"/>	
Producto	Unidad	Precio	Cantidad
<div>- Seleccione Producto -</div>	<input type="text"/>	<input type="text"/>	<input type="text"/>

 Agregar al carrito

Opciones	Código	Descripción	Unidad	Cantidad	P. Venta	Importe
----------	--------	-------------	--------	----------	----------	---------

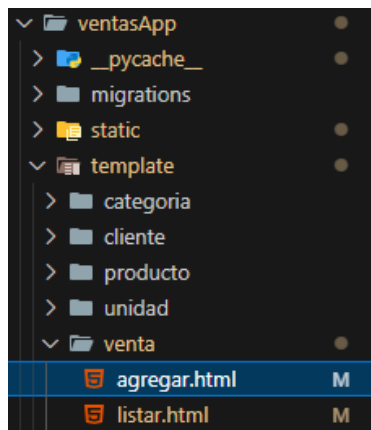
Total S/.

Registrar

Cancelar

Código:

Nos ubicamos en la carpeta venta en el archivo agregar.html



Colocamos el siguiente código:

```
{% extends "plantilla.html" %}

{% block title %}
Agregar Venta
{% endblock %}

{% load static %}

{% block extra_css %}
<link rel="stylesheet" href="{% static '/assets/calendario/css/bootstrap-datepicker.standalone.css' %}">
<link rel="stylesheet" href="{% static '/assets/select2/bootstrap-select.min.css' %}">
{% endblock %}

{% block content %}
<div class="container">
  <h1 class="mb-4">Registrar Venta</h1>
  <div class="alert hidden" role="alert"></div>
  <form method="POST">
    {% csrf_token %}

    <!-- Fecha y Tipo -->
    <div class="form-row align-items-end">
      <div class="form-group col-md-3">
        <label for="fecha">Fecha</label>
        <div class="input-group date form_date" data-date-format="dd/mm/yyyy" data-provide="datepicker">
          <input type="text" class="form-control text-center" name="fecha" id="fecha"
            value="{{ fecha_actual }}">
          <div class="input-group-append">
            <button class="btn btn-primary date-set" type="button">
              <i class="fa fa-calendar"></i>
            </button>
          </div>
        </div>
      </div>
      <div class="form-group col-md-3">
        <label for="seltipo">Tipo</label>
        <select class="form-control selectpicker" id="seltipo" name="seltipo">
          <option value="">- Seleccione Tipo -</option>
          {% for t in tipos %}
            <option value="{{ t.idtipo }}">{{ t.descripcion }}</option>
          {% endfor %}
        </select>
      </div>
    </div>
  </form>
</div>
```

```

    </div>

    <div class="form-group col-md-3">
        <label for="nrodoc">Nro. Documento</label>
        <input type="text" class="form-control" name="nrodoc" id="nrodoc" readonly>
    </div>
</div>

<!-- Cliente -->
<div class="form-row">
    <div class="form-group col-md-6">
        <label for="idcliente">Cliente</label>
        <select class="form-control selectpicker" data-live-search="true" id="idcliente" name="idcliente">
            <option value="">- Seleccione Cliente -</option>
            {% for c in clientes %}
                <option value="{{ c.idcliente }}_{{ c.ruc_dni }}_{{ c.direccion }}">
                    {{ c.apellidos }} {{ c.nombres }}
                </option>
            {% endfor %}
        </select>
    </div>

    <div class="form-group col-md-3">
        <label for="ruc">RUC/DNI</label>
        <input type="text" class="form-control" name="ruc" id="ruc" readonly>
    </div>

    <div class="form-group col-md-3">
        <label for="direccion">Dirección</label>
        <input type="text" class="form-control" name="direccion" id="direccion" readonly>
    </div>
</div>

<!-- Producto -->
<div class="form-row">
    <div class="form-group col-md-6">
        <label for="idproducto">Producto</label>
        <select class="form-control selectpicker" data-live-search="true" id="idproducto" name="idproducto">
            <option value="">- Seleccione Producto -</option>

```

```

            {% for p in productos %}
                <option value="{{ p.idproducto }}">{{ p.descripcion }}</option>
            {% endfor %}
        </select>
    </div>

    <div class="form-group col-md-2">
        <label for="unidad">Unidad</label>
        <input type="text" class="form-control" name="unidad" id="unidad" readonly>
    </div>

    <div class="form-group col-md-2">
        <label for="precio">Precio</label>
        <input type="text" class="form-control" name="precio" id="precio" readonly>
    </div>

    <div class="form-group col-md-2">
        <label for="cantidad">Cantidad</label>
        <input type="text" class="form-control" name="cantidad" id="cantidad">
    </div>
</div>

<!-- Botón agregar -->
<div class="form-row">
    <div class="col-md-12 text-right mb-3">
        <button type="button" id="btnaddet" class="btn btn-success btn-block">
            <i class="fas fa-shopping-cart"></i> Agregar al carrito
        </button>
    </div>
</div>

```

```



<!-- Tabla detalle -->
<div class="table-responsive mb-3">
  <table id="detalles" class="table table-bordered">
    <thead class="bg-primary text-white">
      <tr>
        <th class="text-center">Opciones</th>
        <th class="text-center">Código</th>
        <th>Descripción</th>
        <th>Unidad</th>
        <th class="text-center">Cantidad</th>
        <th class="text-center">P. Venta</th>
        <th class="text-center">Importe</th>
      </tr>
    </thead>
    <tbody>
    </tbody>
  </table>
</div>

<!-- Total -->
<div class="form-row justify-content-end mb-4">
  <div class="col-md-3">
    <div class="input-group">
      <div class="input-group-prepend">
        <span class="input-group-text font-weight-bold">Total S/.</span>
      </div>
      <input type="text" class="form-control text-right" name="total" id="total" readonly>
    </div>
  </div>
</div>

<!-- Botones -->
<div class="form-row">
  <div class="col text-center">
    <button class="btn btn-primary" type="submit">
      <i class="fas fa-save"></i> Registrar
    </button>
    <a href="{% url 'listarventa' %}" class="btn btn-danger">
      <i class="fas fa-ban"></i> Cancelar
    </a>
  </div>
</div>
</form>
</div>
{% endblock %}

```

Muy importante, agregar el siguiente código en la sección ‘extra_css’ de la plantilla

```

{% block extra_css %}
<link rel="stylesheet" href="{% static '/assets/calendario/css/bootstrap-datepicker.standalone.css' %}">
<link rel="stylesheet" href="{% static '/assets/select2/bootstrap-select.min.css' %}">
{% endblock %}

```

Y este otro en la sección ‘js’:

```

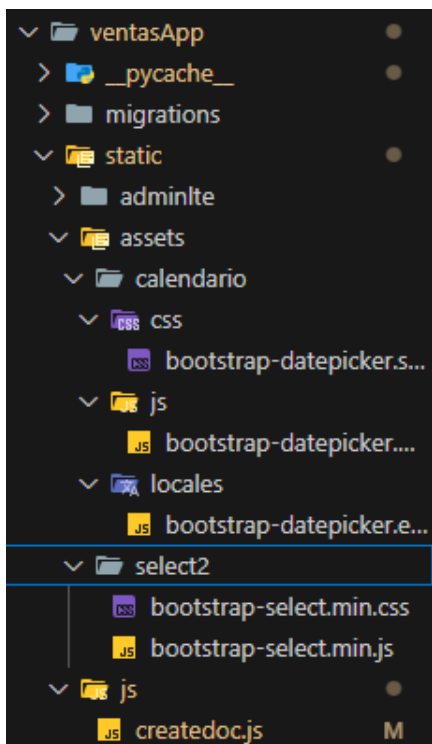
{% block js %}
<script src="{% static '/assets/select2/bootstrap-select.min.js' %}"></script>
<script src="{% static '/assets/calendario/js/bootstrap-datepicker.min.js' %}"></script>
<script src="{% static '/assets/calendario/locales/bootstrap-datepicker.es.min.js' %}"></script>
<script src="{% static '/js/createdoc.js' %}"></script>

<script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

```

```
{% if messages %}
  {% for m in messages %}
    <script>
      Swal.fire({
        "title": "{% if 'success' in m.tags %}¡Éxito!{% elif 'error' in m.tags %}Error!{% else %}Aviso{% endif %}",
        "text": "{{ m }}",
        "icon": "{% if 'success' in m.tags %}success{% elif 'error' in m.tags %}error{% else %}info{% endif %}"
      });
    </script>
  {% endfor %}
{% endif %}
{% endblock %}
```

Colocamos los archivos correspondientes en la carpeta static:



Estos archivos pueden ser descargados desde:

- datapicker: <https://github.com/uxsolutions/bootstrap-datepicker>
- select: <https://developer.snapappointments.com/bootstrap-select/>

Además, el archivo [createdoc.js](#), será el javascript necesario para que el registro de ventas funcione correctamente:


```

var cont=0;
var total=0;
var detalleventa=[];
var subtotal=[];
var controlproducto=[];
$(document).ready(function(){

    $('#idcliente').change(function(){
        mostrarCliente();
    });
    $('#idproducto').change(function(){
        mostrarProducto();
    });
    $('#btnaddde').click(function(){
        agregarDetalle();
    });
    $('#seltipo').change(function(){
        mostrarTipo();
    });

});

```

```

function mostrarCliente(){
    datosCliente=document.getElementById('idcliente').value.split('_');
    $('#ruc').val(datosCliente[1]);
    $('#direccion').val(datosCliente[2]);
}
function mostrarProducto(){
    idproducto=$("#idproducto").val();
    $.get('/EncontrarProducto/'+idproducto, function(data){
        $('#input[name=idproducto]').val(data[0].idproducto);
        $('#input[name=unidad]').val(data[0].unidad);
        $('#input[name=precio]').val(data[0].precio);
        $('#input[name=stock]').val(data[0].stock);
        $('#input[name=cantidad]').val(0);
    });
}
function mostrarTipo(){
    idtipo = $("#seltipo").val();

    $.get('/EncontrarTipo/' + idtipo, function(data){
        $('#input[name=nrodoc]').val(data[0].numeracion);
    })
}

```

```

function mostrarMensajeError(mensaje){
    $(".alert").css('display', 'block');
    $(".alert").removeClass("hidden");
    $(".alert").addClass("alert-danger");
    $(".alert").html("<button type='button' class='close' data-close='alert'></button>" + "<span><b>Error!</b> " + mensaje + "</span>");
    $(".alert").delay(5000).hide(400);
}

```

```
function agregarDetalle()
{
    ruc=$("#ruc").val();
    if (ruc=='')
    {
        mostrarMensajeError("Por favor seleccione el Cliente");
        return false;
    }
    tipo = $('#seltipo option:selected').text();
    if (tipo=='- Seleccione Tipo -')
    {
        mostrarMensajeError("Por favor seleccione el Tipo");
        return false;
    }
    cliente = $('#idcliente option:selected').text();
    if(cliente == '- Seleccione Cliente -')
    {
        mostrarMensajeError("Por favor seleccione el Cliente");
        return false;
    }
    descripcion = $('#idproducto option:selected').text();
    if (descripcion=='- Seleccione Producto -')
    {
        mostrarMensajeError("Por favor seleccione el Producto");
        return false;
    }
    let cantidad=$("#cantidad").val();
    let stock= $("#stock").val();
    if (cantidad==' ' || Number(cantidad)==0 || cantidad==null)
    {
        mostrarMensajeError("Por favor ingrese cantidad del producto");
        return false;
    }
}
```

```

if (cantidad<=0)
{
    mostrarMensajeError("Por favor debe escribir cantidad del producto mayor a 0");
    return false;
}
if (Number(cantidad)>Number(stock))
{
    mostrarMensajeError("No se tiene tal cantidad de producto solo hay " + stock);
    return false;
}
pventa=$("#precio").val();
if (pventa==' ' || pventa==0)
{
    mostrarMensajeError("Por favor ingrese precio de venta del producto");
    return false;
}
cod_producto=$("#idproducto").val();
/* Buscar que codigo de producto no se repita */
var i=0;
var band=false;
while (i<cont)
{
    if (controlproducto[i]==cod_producto)
    {
        band=true;
    }
    i=i+1;
}
if (band==true)
{
    mostrarMensajeError("No puede volver a vender el mismo producto");

    return false;
}

```

```

else
{
    limpiar();
    unidad=$("#unidad").val();
    subtotal[cont]=cantidad*pventa;
    controlproducto[cont]=cod_producto;
    total=total + subtotal[cont];

    var fila='<tr class="selected" id="fila'+cont+'"><td style="text-align:center;"><button
type="button" class="btn btn-danger btn-xs" onclick="eliminardetalle('+cod_producto+', '+cont+'>
"><i class="fa fa-times"></i></button></td><td style="text-align:right;"><input type="text"
name="cod_producto[]" value="'+ cod_producto +'>
"></td><td>'+ descripcion +'</td><td><input type="text" name="unidad[]" value="'+ unidad +'>
readonly style="width:140px; text-align:left;"></td><td style="text-align:right;" ><input
type="number" name="cantidad[]" value="'+ cantidad +'>
style="width:80px; text-align:right;"
readonly></td><td style="text-align:right;"><input type="number" name="pventa[]" value="'+
pventa +'>
style="width:80px; text-align:right;" readonly></td><td style="text-align:right;">'
+number_format(subtotal[cont],2)+'</td></tr>';
    $('#detalles').append(fila);
    detalleventa.push({
        codigo:cod_producto,
        unidad:unidad,
        cantidad:cantidad,
        pventa:pventa,
        subtotal:subtotal
    });
    cont++;
}
$('#total').val(number_format(total,2));
}

```

```

function limpiar(){
    $("#cantidad").val(0);
    //$("#precio").val(0);
}

/* Eliminar productos */
function eliminardetalle(codigo,index){
    total=total-subtotal[index];
    tam=detalleventa.length;
    var i=0;
    var pos;
    while (i<tam)
    {
        if (detalleventa[i].codigo==codigo)
        {
            pos=i;
            break;
        }
        i=i+1;
    }
}

```

```

    detalleventa.splice(pos,1);
    $('#fila'+index).remove();
    controlproducto[index]="";
    $('#total').val(number_format(total,2));
}

function number_format(amount, decimals) {
    amount += ''; // por si pasan un numero en vez de un string
    amount = parseFloat(amount.replace(/[^0-9\.]/g, '')); // elimino cualquier cosa que no sea numero o punto
    decimals = decimals || 0; // por si la variable no fue pasada
    // si no es un numero o es igual a cero retorno el mismo cero
    if (isNaN(amount) || amount === 0)
        return parseFloat(0).toFixed(decimals);
    // si es mayor o menor que cero retorno el valor formateado como numero
    amount = '' + amount.toFixed(decimals);
    var amount_parts = amount.split('.');
    regexp = /(\d+)(\d{3})/;
    while (regexp.test(amount_parts[0]))
        amount_parts[0] = amount_parts[0].replace(regexp, '$1' + ',' + '$2');
    ;
    return amount_parts.join('.');
}

```

En [views.py](#), colocamos la function agregarventa()

```

@transaction.atomic
def agregarventa(request):
    if request.method == "POST":
        try:
            with transaction.atomic():

                # Validar que hay productos
                ids = request.POST.getlist("cod_producto[]")
                if len(ids) == 0:
                    messages.error(request, "Debe agregar al menos un producto")
                    return redirect("agregarventa")

```

Obtenemos el id del cliente con raw split porque el dato pasa como id_ruc_direccion:

```
{% for c in clientes %}
    <option value="{{ c.idcliente }}_{{ c.ruc_dni }}_{{ c.direccion }}">
        {{ c.apellidos }} {{ c.nombres }}
    </option>
{% endfor %}
```

```
# Cliente
idcliente_raw = request.POST["idcliente"]
cliente_id = idcliente_raw.split("_")[0]
cliente = Cliente.objects.get(pk=cliente_id)
```

```
# Cabecera
venta = CabeceraVenta()
venta.cliente = cliente
venta.nrodoc = request.POST["nrodoc"]
idtipo = request.POST["seltipo"]
venta.tipo_id = idtipo
```

```
# Fecha
fecha_str = request.POST["fecha"]
if not fecha_str:
    fecha_str = date.today().strftime("%d/%m/%Y")
venta.fecha_venta = parse_date("-".join(fecha_str.split("/")[::-1]))
venta.estado = True
```

```
total_raw = request.POST["total"]
total = float(total_raw.replace(",", ""))
```

Verificamos que si el documento es de tipo 2 (“boleta”) no se aplica igv y si es factura hace el cálculo y asigna los valores:

```
# Totales según tipo
if request.POST["seltipo"] == "2":
    venta.total = total
    venta.subtotal = 0
    venta.igv = 0
else:
    venta.total = total
    venta.subtotal = round(total / 1.18, 2)
    venta.igv = round(total - venta.subtotal, 2)
```

```
venta.save()
```

```
# Detalles
#en las validaciones iniciales obtiene los productos
cantidades = request.POST.getlist("cantidad[]")
precios = request.POST.getlist("pventa[]")
```

```
if len(ids) != len(cantidades) or len(ids) != len(precios):
    raise Exception("Error en los datos de productos")
```

```

for i in range(len(ids)):
    try:
        idproducto = int(ids[i])
        cantidad = float(cantidades[i])
        precio = float(precios[i])

        producto = Producto.objects.get(pk=idproducto)
        if producto.stock < cantidad:
            raise Exception(f"Stock insuficiente para {producto.descripcion}")

        detalle = DetalleVenta()
        detalle.venta = venta
        detalle.producto = producto
        detalle.cantidad = cantidad
        detalle.precio = precio
        detalle.save()

        # Actualizar stock
        Producto.objects.filter(pk=idproducto).update(
            stock=F("stock") - cantidad
        )

    except ValueError:
        raise Exception(f"Datos inválidos en producto {i+1}")

```

Obtiene el número de serie actual según el tipo de documento, lo convierte en un int y le suma 1. Luego lo convierte a cadena, le agrega ceros a la izquierda hasta que complete 5 dígitos y lo guarda.

```

# Actualizar número de serie y numeración

parametro = Parametro.objects.get(tipo_id=idtipo)
numeracion_actual = parametro.numeracion

nuevo_numero = int(numeracion_actual) + 1
nueva_numeracion = f"{nuevo_numero:05d}"

parametro.numeracion = nueva_numeracion
parametro.save()

# Mensaje éxito
messages.success(request, "Venta registrada correctamente.")
return redirect("listarventa")

except Exception as e:
    transaction.set_rollback(True)
    messages.error(request, f"Ocurrió un error al registrar la venta: {str(e)}")
    return redirect("listarventa")
else:
    clientes = Cliente.objects.filter(estado=1)
    productos = Producto.objects.filter(estado=1)
    tipos = Tipo.objects.filter(estado=1)
    parametros = Parametro.objects.all() # o el que corresponda
    context = {
        "clientes": clientes,
        "productos": productos,
        "tipos": tipos,
        "parametros": parametros,
        "fecha_actual": date.today().strftime("%d/%m/%Y"),
    }
    return render(request, "venta/agregar.html", context)

```

Además de:

```
def ProductoCodigo(request, producto_id):
    producto = (
        Producto.objects
        .filter(estado=True, idproducto=producto_id)
        .values(
            'idproducto',
            'descripcion',
            'unidad__descripcion',
            'precio',
            'stock'
        )
        .first()
    )
    return JsonResponse([{
        "idproducto": producto["idproducto"],
        "unidad": producto["unidad__descripcion"],
        "descripcion": producto["descripcion"],
        "precio": producto["precio"],
        "stock": producto["stock"]
    }], safe=False)

def PorTipo(request, tipo_id):
    parametro = (
        Parametro.objects.filter(tipo__idtipo=tipo_id).values(
            'tipo__idtipo',
            'tipo__descripcion',
            'serie',
            'numeracion'
        ).first()
    )
    return JsonResponse([{
        "idtipo" : parametro["tipo__idtipo"],
        "descripcion": parametro["tipo__descripcion"],
        "serie": parametro["serie"],
        "numeracion": parametro["numeracion"]
    }], safe=False)
```

Para que en las urls podamos tener:

```
#Venta

path('listarventa/', listarventa, name="listarventa"),
path('agregarventa/', agregarventa, name="agregarventa"),
path('eliminarventa/<int:id>/', eliminarventa, name="eliminarventa"),

path('EncontrarProducto/<int:producto_id>/', ProductoCodigo, name="EncontrarProducto"),
path('EncontrarTipo/<int:tipo_id>/', PorTipo, name="EncontrarTipo"),
```

EncontrarProducto y EncontrarTipo serán usadas al desplegarse los selects, en la primera parte del archivo [createdoc.js](#)

```
$(document).ready(function(){

    $('#idcliente').change(function(){
        mostrarCliente();
    });
    $('#idproducto').change(function(){
        mostrarProducto();
    });
    $('#btnaddde').click(function(){
        agregarDetalle();
    });
    $('#seltipo').change(function(){
        mostrarTipo();
    });

});

function mostrarCliente(){
    datosCliente=document.getElementById('idcliente').value.split('_');
    $('#ruc').val(datosCliente[1]);
    $('#direccion').val(datosCliente[2]);
}

function mostrarProducto(){
    idproducto=$('#idproducto').val();
    $.get('/EncontrarProducto/'+idproducto, function(data){
        $('#input[name=idproducto]').val(data[0].idproducto);
        $('#input[name=unidad]').val(data[0].unidad);
        $('#input[name=precio]').val(data[0].precio);
        $('#input[name=stock]').val(data[0].stock);
        $('#input[name=cantidad]').val(0);
    });
}

function mostrarTipo(){
    idtipo = $('#seltipo').val();

    $.get('/EncontrarTipo/' + idtipo, function(data){
        $('#input[name=nrodoc]').val(data[0].numeracion);
    })
}
```

Mismo archivo donde se realizarán las validaciones correspondientes para evitar que se registren productos al carrito de manera errónea

```
function agregarDetalle()
{
    ruc=$('#ruc').val();
    if (ruc=='')
    {
        mostrarMensajeError("Por favor seleccione el Cliente");
        return false;
    }
    tipo = $('#seltipo option:selected').text();
    if (tipo=='- Seleccione Tipo -')
    {
        mostrarMensajeError("Por favor seleccione el Tipo");
        return false;
    }
    cliente = $('#idcliente option:selected').text();
    if(cliente == '- Seleccione Cliente -')
    {
        mostrarMensajeError("Por favor seleccione el Cliente");
        return false;
    }
}
```



```
descripcion = $('#idproducto option:selected').text();
if (descripcion=='- Seleccione Producto -')
{
    mostrarMensajeError("Por favor seleccione el Producto");
    return false;
}
```

```

}
let cantidad=$("#cantidad").val();
let stock= $("#stock").val();
if (cantidad==' ' || Number(cantidad)==0 || cantidad==null)
{
    mostrarMensajeError("Por favor ingrese cantidad del producto");
    return false;
}
if (cantidad<=0)
{
    mostrarMensajeError("Por favor debe escribir cantidad del producto mayor a 0");
    return false;
}
if (Number(cantidad)>Number(stock))
{
    mostrarMensajeError("No se tiene tal cantidad de producto solo hay " + stock);
    return false;
}
pventa=$("#precio").val();
if (pventa==' ' || pventa==0)
{
    mostrarMensajeError("Por favor ingrese precio de venta del producto");
    return false;
}
cod_producto=$("#idproducto").val();
/* Buscar que codigo de producto no se repita */
var i=0;
var band=false;
while (i<cont)
{
    if (controlproducto[i]==cod_producto)
    {
        band=true;
    }
    i=i+1;
}
if (band==true)
{
    mostrarMensajeError("No puede volver a vender el mismo producto");

    return false;
}
}

```

“Eliminar Ventas”

Vista:

LISTADO DE VENTAS

+NUEVO REGISTRO

Buscar por descripción...

Q Buscar

CÓDIGO	DESCRIPCIÓN	FECHA	RUC/DNI	NOMBRES/RAZÓN	TOTAL	OPCIONES
3	Boleta	2023-01-01	12345678901234	Maria Lopez	30.0	<div>Eliminar</div>
4	Factura	2023-01-02	12345678901234	Maria Lopez	40.0	<div>Eliminar</div>
5	Boleta	2023-01-03	12345678901234	Maria Lopez	1430.0	<div>Eliminar</div>
8	Boleta	2023-01-04	12345678901234	Maria Lopez	62.0	<div>Eliminar</div>

?

¿Realmente desea eliminar el registro?

No

Sí

Código: 3 | Descripción: Boleta | Nombres: Maria Lopez

Código:

En el listar.html, en el apartado de Opciones colocamos la llamada a la función eliminarVenta()

```
<td class="text-center">
  <a href="#" onclick="eliminarVenta(event, '{{ itemventa.idventa }}', '{{ itemventa.tipo.descripcion }}', '{{ itemventa.cliente.nombres }}')"
    class="btn btn-danger btn-sm">
    <i class="fas fa-trash"></i> Eliminar
  </a>
</td>
```

Esta la definimos más abajo en el mismo archivo listar.html, dentro del block js, junto a mensajes de error en caso ocurra algún imprevisto a la hora de registrar la venta y mensajes de éxito si es que la operación realizada se ejecutó con normalidad:

```
{% block js %}
<script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

{% if messages %}
  {% for m in messages %}
    <script>
    Swal.fire({
      title: "{% if 'success' in m.tags %}Éxito!{% elif 'error' in m.tags %}Error!{% else %}Aviso{% endif %}",
      text: "{{ m }}",
      icon: "{% if 'success' in m.tags %}success{% elif 'error' in m.tags %}error{% else %}info{% endif %}"
    });
    </script>
  {% endfor %}
{% endif %}

<script>
function eliminarVenta(event, id, descripcion, nombres) {
  event.preventDefault();
  Swal.fire({
    title: "¿Realmente desea eliminar el registro?",
    text: "Código: " + id + " | Descripción: " + descripcion + " | Nombres: " + nombres,
    icon: "question",
    showCancelButton: true,
    cancelButtonText: "No",
    confirmButtonText: "Sí",
    reverseButtons: true,
    confirmButtonColor: "#dc3545"
  }).then(function(result) {
    if (result.isConfirmed) {
      window.location.href = "/eliminarventa/" + id + "/";
    }
  });
}
</script>
{% endblock %}
```

En las [views.py](#), definimos la funcion para eliminar ventas:

```
def eliminarventa(request, id):
    try:
        venta = CabeceraVenta.objects.get(pk=id)

        detalles = DetalleVenta.objects.filter(venta=venta)
        for detalle in detalles:
            Producto.objects.filter(pk=detalle.producto_id).update(
                stock=F('stock') + detalle.cantidad
            )
        venta.estado = False
        venta.save()
        messages.success(request, 'Venta eliminada correctamente.')
        return redirect("listarventa")
    except Exception as e:
        messages.error(request, f'Ocurrió un error: {str(e)}')
        return redirect("listarventa")
```

Y de igual forma en las urls:

```
#Venta

path('listarventa/', listarventa, name="listarventa"),
path('agregarventa/', agregarventa, name="agregarventa"),
path('eliminarventa/<int:id>/', eliminarventa, name="eliminarventa"),
```

Para detalle Ventas tenemos el siguiente codigo:

```
{% extends "plantilla.html" %}

{% load filtros %}

{% block title %}
Detalles de la Venta
{% endblock %}

{% block content %}
<div class="container mt-4">
    <div class="card shadow border-0">
        <div class="p-3 bg-dark text-white d-flex justify-content-between align-items-center">
            <h3 class="mb-0">
                <i class="fas fa-shopping-cart"></i>
                Detalles de la Venta N° {{ venta.nrodoc }}
            </h3>
            <a href="{% url 'agregarventa' %}" class="btn btn-success btn-sm">
                <i class="fas fa-plus"></i> Nuevo Registro
            </a>
        </div>
    </div>

    <div class="p-4">
        {# Datos Generales #}
        <div class="row mb-3">
            <div class="col-md-6">
                <h5 class="text-primary">Información del Cliente</h5>
                <p class="mb-1">
                    <strong>Cliente:</strong>
                    {{ venta.cliente.nombres|default_if_none:'' }}
                </p>
                <p class="mb-1">
                    <strong>RUC/DNI:</strong> {{ venta.cliente.ruc_dni|default_if_none:'' }}</p>
                <p class="mb-1">
                    <strong>Dirección:</strong> {{ venta.cliente.direccion|default_if_none:'' }}</p>
            </div>
        </div>
    </div>
</div>
```

```

<div class="col-md-6">
  <h5 class="text-primary">Datos de la Venta</h5>
  <p class="mb-1">
    <strong>Fecha:</strong>
    {{ venta.fecha_venta|date:"d/m/Y" }}
  </p>
  <p class="mb-1">
    <strong>Estado:</strong>

```

```

    {% if venta.estado %}Activa{% else %}Anulada{% endif %}
  </p>
</div>
</div>

```

```

{# Detalle de Productos #}
<div class="table-responsive">
  <table class="table table-bordered table-hover align-middle">
    <thead class="thead-dark">
      <tr>
        <th>#</th>
        <th>Producto</th>
        <th class="text-center">Unidad</th>
        <th class="text-center">Cantidad</th>
        <th class="text-center">Precio</th>
        <th class="text-center">Importe</th>
      </tr>
    </thead>
    <tbody>
      {% for detalle in detalles %}
      <tr>
        <td>{{ forloop.counter }}</td>
        <td>{{ detalle.producto.descripcion }}</td>
        <td class="text-center">{{ detalle.producto.unidad.descripcion }}</td>
        <td class="text-center">{{ detalle.cantidad }}</td>
        <td class="text-center">{{ detalle.precio|floatformat:2 }}</td>
        <td class="text-center">
          {{ detalle.cantidad|floatformat:2|floatformat }} × {{ detalle.precio|floatformat:2 }} =
          {{ detalle.cantidad|mul:detalle.precio|floatformat:2 }}
        </td>
      </tr>
      {% empty %}
      <tr>
        <td colspan="6" class="text-center">No hay productos.</td>
      </tr>
    </tbody>
  </table>

```

```

    </tr>
    {% endfor %}
  </tbody>
  <tfoot>
    <tr>
      <th colspan="5" class="text-right">SUBTOTAL</th>
      <th class="text-center">S/ {{ venta.subtotal|floatformat:2 }}</th>
    </tr>
    <tr>
      <th colspan="5" class="text-right">IGV</th>
      <th class="text-center">S/ {{ venta.igv|floatformat:2 }}</th>
    </tr>
    <tr>
      <th colspan="5" class="text-right">TOTAL</th>
      <th class="text-center">S/ {{ venta.total|floatformat:2 }}</th>
    </tr>
  </tfoot>
</table>
</div>

{# Botón volver #}
<div class="text-center mt-3">
  <a href="{% url 'listarventa' %}" class="btn btn-secondary">
    <i class="fas fa-arrow-left"></i> Volver
  </a>
</div>
</div>
</div>
{% endblock %}

```

En [views.py](#) colocamos la declaración de la función para mostrar la vista de detalles de la venta respectiva

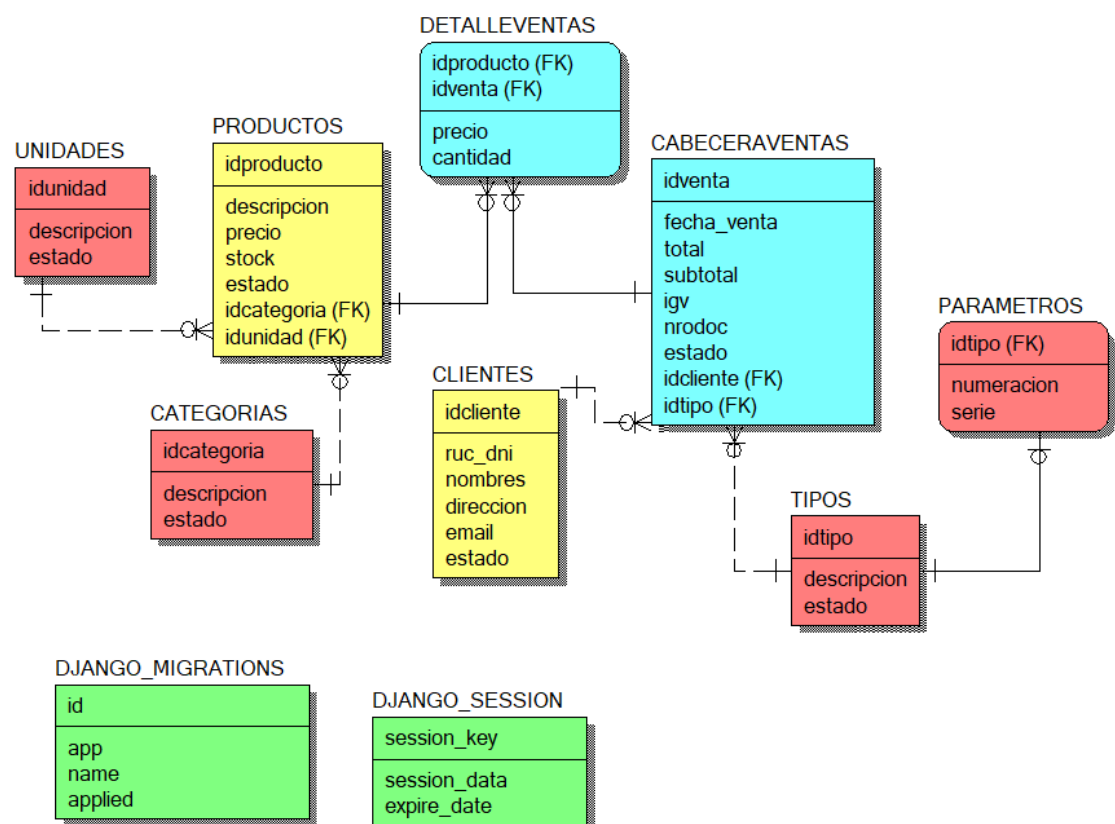
```
def detallesventa(request, id):
    venta = CabeceraVenta.objects.get(pk=id)
    detalles = DetalleVenta.objects.select_related('producto').filter(venta=venta)
    context = {
        'venta': venta,
        'detalles': detalles
    }
    return render(request, "venta/detalles.html", context)
```

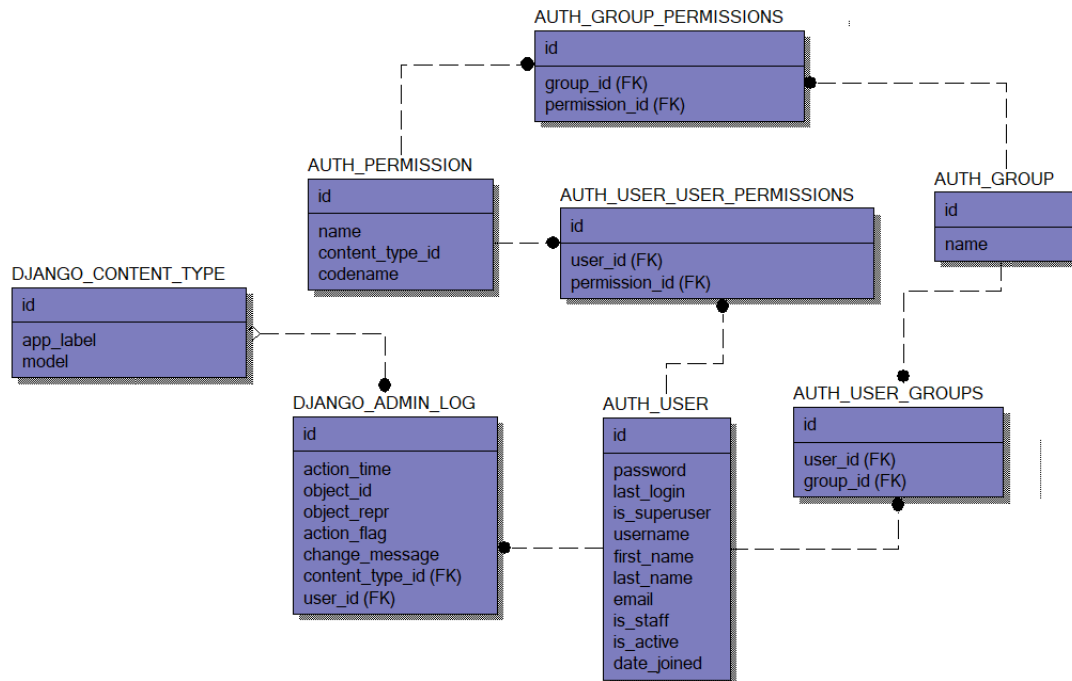
En [urls.py](#) colocamos la ruta para mostrar la vista de detalles de la venta respectiva

```
path('detallesventa/<int:id>/', detallesventa, name="detallesventa"),
```

4. Anexos

4.1. Diagrama de Base de Datos





4.2. Modelos de los mantenedores adicionales

- Modelo de Cliente:

```

2
3 class Cliente(models.Model):
4     idcliente = models.AutoField(primary_key=True)
5     ruc_dni = models.CharField(max_length=11, unique=True)
6     nombres = models.CharField(max_length=80)
7     direccion = models.CharField(max_length=100)
8     email = models.EmailField(max_length=100)
9     estado = models.BooleanField(default=True)
10    def __str__(self):
11        return self.nombres
12    class Meta:
13        db_table = 'clientes'

```

- Modelo de Categoria:

```

15 class Categoria(models.Model):
16     idcategoria = models.AutoField(primary_key=True)
17     descripcion = models.CharField(max_length=30, unique=True)
18     estado = models.BooleanField(default=True)
19
20     def __str__(self):
21         return str(self.descripcion)
22
23     class Meta:
24         verbose_name = "Categoría"
25         verbose_name_plural = "Categorías"
26         ordering = ['descripcion']
27         db_table = 'categorias'
28

```

- Modelo de Unidad:

```

29 class Unidad(models.Model):
30     idunidad = models.AutoField(primary_key=True)
31     descripcion = models.CharField(max_length=30, unique=True)
32     estado = models.BooleanField(default=True)
33
34     class Meta:
35         db_table = 'unidades'
36
37     def __str__(self):
38         return str(self.descripcion)
39

```

- Modelo de Producto:

```

41 class Producto(models.Model):
42     idproducto = models.AutoField(primary_key=True)
43     descripcion = models.CharField(max_length=40, unique=True)
44     categoria = models.ForeignKey(
45         Categoria,
46         on_delete=models.CASCADE,
47         db_column='idcategoria'
48     )
49     precio = models.DecimalField(max_digits=10, decimal_places=2)
50     stock = models.IntegerField()
51     unidad = models.ForeignKey(
52         Unidad,
53         on_delete=models.CASCADE,
54         db_column='idunidad'
55     )
56     estado = models.BooleanField(default=True)
57
58     class Meta:
59         db_table = 'productos'
60
61     def __str__(self):
62         return str(self.descripcion)

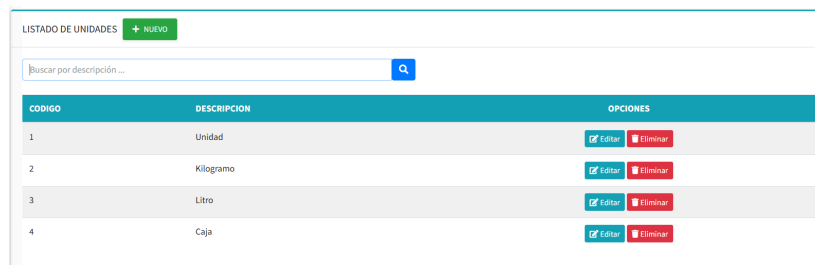
```

4.3. Capturas de Pantalla y explicación de código de los mantenedores adicionales:

CRUD: “Unidades”

- “Listado de Unidades”

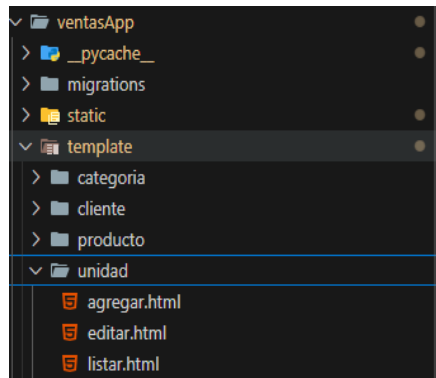
Vista:



CODIGO	DESCRIPCION	OPCIONES
1	Unidad	Editar Eliminar
2	Kilogramo	Editar Eliminar
3	Litro	Editar Eliminar
4	Caja	Editar Eliminar

Código:

En el archivo listar.html de la carpeta unidad



```

1 [% extends "plantilla.html" %]
2
3 [% block title %]
4 Listar Unidades
5 [% endblock %]
6
7 [% block content %]
8 <div class="container-fluid">
9   <div class="card card-outline card-info mt-3">
10     <div class="card-header d-flex align-items-center flex-wrap" style="gap: 10px;">
11       <h1 class="card-title mb-0">LISTADO DE UNIDADES</h1>
12       <a href="{% url 'agregarunidad' %}" class="btn btn-success btn-sm d-flex align-items-center fw-bold shadow-sm px-3 py-2" style="gap: 8px;">
13         <i class="fas fa-plus me-2"></i> NUEVO
14       </a>
15     </div>
16     <div class="card-body">
17       <div class="row">
18         <div class="col-md-6 pull-right">
19           <form action="" method="GET">
20             <div class="input-group">
21               <input type="text" name="buscar" id="buscar" class="form-control" autofocus placeholder="Buscar por descripción ..." value="{% buscar %}">
22               <span class="input-group-btn">
23                 <button class="btn btn-primary">
24                   <i class="fa fa-search"></i>
25                 </button>
26               </span>
27             </div>
28           </form>
29         </div>
30       </div>
31       <div class="table-responsive">
32         <table id="tablasubfamilias" class="table table-striped mt-4" style="width:100%">
33           <thead class="bg-info text-white">
34             <tr>
35               <th>CODIGO</th>
36               <th>DESCRIPCION</th>
37               <th><center>OPCIONES</center></th>
38             </tr>
39           </thead>
40           <tbody>
41             <tr>
42               <td>
43                 <div>
44                   <div>
45                     <div>
46                       <div>
47                         <a href="{% url 'editarunidad' itemunidad.idunidad %}" class="btn btn-info btn-sm">
48                           <i class="fa fa-edit"></i> Editar
49                         </a>
50                         <a href="#" onclick="eliminarUnidad(event, '{{itemunidad.idunidad}}', '{{itemunidad.descripcion}}')" class="btn btn-danger btn-sm">
51                           <i class="fa fa-trash"></i> Eliminar
52                         </a>
53                       </div>
54                     </div>
55                   </div>
56                 </td>
57               <td>
58                 <div>
59                   <div>
60                     <div>
61                       <div>
62                         <a href="{% url 'editarunidad' itemunidad.idunidad %}" class="btn btn-info btn-sm">
63                           <i class="fa fa-edit"></i> Editar
64                         </a>
65                         <a href="#" onclick="eliminarUnidad(event, '{{itemunidad.idunidad}}', '{{itemunidad.descripcion}}')" class="btn btn-danger btn-sm">
66                           <i class="fa fa-trash"></i> Eliminar
67                         </a>
68                       </div>
69                     </div>
70                   </div>
71                 </td>
72             </tr>
73           </tbody>
74         </table>
75       </div>
76     </div>
77   </div>
78   <div class="card card-outline card-info mt-3">
79     <div class="card-header d-flex align-items-center flex-wrap" style="gap: 10px;">
80       <h1 class="card-title mb-0">LISTADO DE UNIDADES</h1>
81       <a href="{% url 'agregarunidad' %}" class="btn btn-success btn-sm d-flex align-items-center fw-bold shadow-sm px-3 py-2" style="gap: 8px;">
82         <i class="fas fa-plus me-2"></i> NUEVO
83       </a>
84     </div>
85     <div class="card-body">
86       <div class="row">
87         <div class="col-md-6 pull-right">
88           <form action="" method="GET">
89             <div class="input-group">
90               <input type="text" name="buscar" id="buscar" class="form-control" autofocus placeholder="Buscar por descripción ..." value="{% buscar %}">
91               <span class="input-group-btn">
92                 <button class="btn btn-primary">
93                   <i class="fa fa-search"></i>
94                 </button>
95               </span>
96             </div>
97           </form>
98         </div>
99       </div>
100       <div class="table-responsive">
101         <table id="tablasubfamilias" class="table table-striped mt-4" style="width:100%">
102           <thead class="bg-info text-white">
103             <tr>
104               <th>CODIGO</th>
105               <th>DESCRIPCION</th>
106               <th><center>OPCIONES</center></th>
107             </tr>
108           </thead>
109           <tbody>
110             <tr>
111               <td>
112                 <div>
113                   <div>
114                     <div>
115                       <div>
116                         <a href="{% url 'editarunidad' itemunidad.idunidad %}" class="btn btn-info btn-sm">
117                           <i class="fa fa-edit"></i> Editar
118                         </a>
119                         <a href="#" onclick="eliminarUnidad(event, '{{itemunidad.idunidad}}', '{{itemunidad.descripcion}}')" class="btn btn-danger btn-sm">
120                           <i class="fa fa-trash"></i> Eliminar
121                         </a>
122                       </div>
123                     </div>
124                   </div>
125                 </td>
126               <td>
127                 <div>
128                   <div>
129                     <div>
130                       <div>
131                         <a href="{% url 'editarunidad' itemunidad.idunidad %}" class="btn btn-info btn-sm">
132                           <i class="fa fa-edit"></i> Editar
133                         </a>
134                         <a href="#" onclick="eliminarUnidad(event, '{{itemunidad.idunidad}}', '{{itemunidad.descripcion}}')" class="btn btn-danger btn-sm">
135                           <i class="fa fa-trash"></i> Eliminar
136                         </a>
137                       </div>
138                     </div>
139                   </div>
140                 </td>
141             </tr>
142           </tbody>
143         </table>
144       </div>
145     </div>
146   </div>
147   <div class="card card-outline card-info mt-3">
148     <div class="card-header d-flex align-items-center flex-wrap" style="gap: 10px;">
149       <h1 class="card-title mb-0">LISTADO DE UNIDADES</h1>
150       <a href="{% url 'agregarunidad' %}" class="btn btn-success btn-sm d-flex align-items-center fw-bold shadow-sm px-3 py-2" style="gap: 8px;">
151         <i class="fas fa-plus me-2"></i> NUEVO
152       </a>
153     </div>
154     <div class="card-body">
155       <div class="row">
156         <div class="col-md-6 pull-right">
157           <form action="" method="GET">
158             <div class="input-group">
159               <input type="text" name="buscar" id="buscar" class="form-control" autofocus placeholder="Buscar por descripción ..." value="{% buscar %}">
160               <span class="input-group-btn">
161                 <button class="btn btn-primary">
162                   <i class="fa fa-search"></i>
163                 </button>
164               </span>
165             </div>
166           </form>
167         </div>
168       </div>
169       <div class="table-responsive">
170         <table id="tablasubfamilias" class="table table-striped mt-4" style="width:100%">
171           <thead class="bg-info text-white">
172             <tr>
173               <th>CODIGO</th>
174               <th>DESCRIPCION</th>
175               <th><center>OPCIONES</center></th>
176             </tr>
177           </thead>
178           <tbody>
179             <tr>
180               <td>
181                 <div>
182                   <div>
183                     <div>
184                       <div>
185                         <a href="{% url 'editarunidad' itemunidad.idunidad %}" class="btn btn-info btn-sm">
186                           <i class="fa fa-edit"></i> Editar
187                         </a>
188                         <a href="#" onclick="eliminarUnidad(event, '{{itemunidad.idunidad}}', '{{itemunidad.descripcion}}')" class="btn btn-danger btn-sm">
189                           <i class="fa fa-trash"></i> Eliminar
190                         </a>
191                       </div>
192                     </div>
193                   </div>
194                 </td>
195               <td>
196                 <div>
197                   <div>
198                     <div>
199                       <div>
200                         <a href="{% url 'editarunidad' itemunidad.idunidad %}" class="btn btn-info btn-sm">
201                           <i class="fa fa-edit"></i> Editar
202                         </a>
203                         <a href="#" onclick="eliminarUnidad(event, '{{itemunidad.idunidad}}', '{{itemunidad.descripcion}}')" class="btn btn-danger btn-sm">
204                           <i class="fa fa-trash"></i> Eliminar
205                         </a>
206                       </div>
207                     </div>
208                   </div>
209                 </td>
210             </tr>
211           </tbody>
212         </table>
213       </div>
214     </div>
215   </div>
216   <div class="card card-outline card-info mt-3">
217     <div class="card-header d-flex align-items-center flex-wrap" style="gap: 10px;">
218       <h1 class="card-title mb-0">LISTADO DE UNIDADES</h1>
219       <a href="{% url 'agregarunidad' %}" class="btn btn-success btn-sm d-flex align-items-center fw-bold shadow-sm px-3 py-2" style="gap: 8px;">
220         <i class="fas fa-plus me-2"></i> NUEVO
221       </a>
222     </div>
223     <div class="card-body">
224       <div class="row">
225         <div class="col-md-6 pull-right">
226           <form action="" method="GET">
227             <div class="input-group">
228               <input type="text" name="buscar" id="buscar" class="form-control" autofocus placeholder="Buscar por descripción ..." value="{% buscar %}">
229               <span class="input-group-btn">
230                 <button class="btn btn-primary">
231                   <i class="fa fa-search"></i>
232                 </button>
233               </span>
234             </div>
235           </form>
236         </div>
237       </div>
238       <div class="table-responsive">
239         <table id="tablasubfamilias" class="table table-striped mt-4" style="width:100%">
240           <thead class="bg-info text-white">
241             <tr>
242               <th>CODIGO</th>
243               <th>DESCRIPCION</th>
244               <th><center>OPCIONES</center></th>
245             </tr>
246           </thead>
247           <tbody>
248             &lt
```

```

71         </li>
72         <li class="page-item">
73             <a class="page-link" href="?page={{ unidad.previous_page_number }}{% if buscar %}&buscar={{ buscar }}{% endif %}">Anterior</a>
74         </li>
75     {% endif %}
76
77     {% for num in unidad.paginator.page_range %}
78         {% if unidad.number == num %}
79             <li class="page-item active">
80                 <a class="page-link">{{ num }}</a>
81             </li>
82         {% elif num > unidad.number|add:-3 and num < unidad.number|add:3 %}
83             <li class="page-item">
84                 <a class="page-link" href="?page={{ num }}{% if buscar %}&buscar={{ buscar }}{% endif %}">{{ num }}</a>
85             </li>
86         {% endif %}
87     {% endfor %}
88
89     {% if unidad.has_next %}
90         <li class="page-item">
91             <a class="page-link" href="?page={{ unidad.next_page_number }}{% if buscar %}&buscar={{ buscar }}{% endif %}">Siguiente</a>
92         </li>
93         <li class="page-item">
94             <a class="page-link" href="?page={{ unidad.paginator.num_pages }}{% if buscar %}&buscar={{ buscar }}{% endif %}">Último</a>
95         </li>
96     {% endif %}
97 </ul>
98 </nav>
99 </div>
100 {% endif %}
101

```

```

102     </div>
103 </div>
104 </div>
105 </div>
106 {% endblock %}
107
108 {% block js %}
109     <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>
110     {% if messages %}
111         {% for m in messages %}
112             <script>
113                 Swal.fire({
114                     "title": "¡Éxito!",
115                     "text": "{{m}}",
116                     "icon": "success"
117                 })
118             </script>
119         {% endfor %}
120     {% endif %}
121
122     <script>
123         function eliminarUnidad(event, id, descripcion) {
124             event.preventDefault();
125
126             Swal.fire({
127                 title: "¿Realmente desea eliminar el registro?",
128                 text: "Código: " + id + " | Descripción: " + descripcion,
129                 icon: "question",
130                 showCancelButton: true,
131                 cancelButtonText: "No",
132                 confirmButtonText: "Sí",
133                 reverseButtons: true,
134                 confirmButtonColor: "#dc3545"
135             }).then(function(result) {
136                 if (result.isConfirmed) {
137                     window.location.href = "/eliminarunidad/" + id + "/";
138                 }
139             });
140         }
141     </script>
142 {% endblock %}

```

Creamos la vista para el listado en [views.py](#) de VentasApp:

```
def listarunidad(request):
    queryset=request.GET.get("buscar", "")
    unidad=Unidad.objects.filter(estado=True)
    if queryset:
        unidad=Unidad.objects.filter(Q(descripcion__icontains=queryset),estado=True).distinct()

    paginator = Paginator(unidad, 5)
    page_number = request.GET.get('page')
    page_obj = paginator.get_page(page_number)

    context={'unidad':page_obj, 'buscar': queryset}
    return render(request,"unidad/listar.html",context)
```

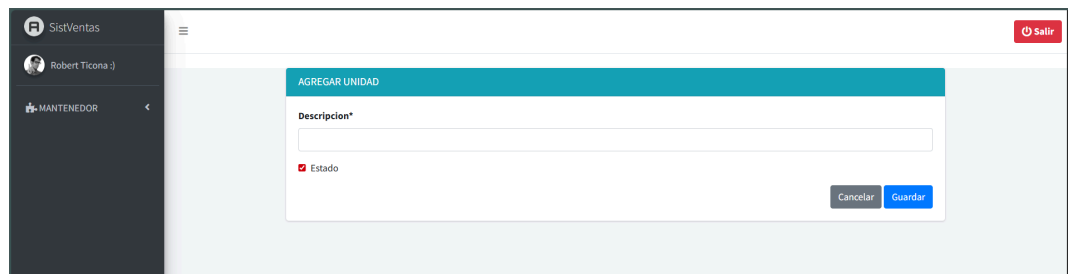
Agregamos la vista a las urls:

```
#Unidad

path('listarunidad/',listarunidad,name="listarunidad"),
path('agregarunidad/',agregarunidad ,name="agregarunidad"),
path('editarunidad/<int:id>/',editarunidad ,name="editarunidad"),
path('eliminarunidad/<int:id>/',eliminarunidad ,name="eliminarunidad"),
```

- “Registrar Unidades”

Vista:



Código:

En el archivo agregar.html de la carpeta unidad

```
agregar.html x
ventasApp > template > unidad > agregar.html > ...
1 {% extends "plantilla.html" %}
2
3 {% block title %}
4 Agregar Unidad
5 {% endblock %}
6
7
8 {% block content %}
9     {% load crispy_forms_tags %}
10     <div class="container">
11         <div class="card card-info mx-4 my-4">
12             <div class="card-header">
13                 <h1 class="card-title">AGREGAR UNIDAD</h1>
14             </div>
15             <div class="card-body">
16                 <form method="post">
17                     {% csrf_token %}
18                     {{ form|crispy }}
19                     <div class="d-flex justify-content-end">
20                         <a href="{% url 'listarunidad' %}" class="btn btn-secondary">Cancelar</a>
21                         &nbsp;
22                         <button type="submit" class="btn btn-primary">Guardar</button>
23                     </div>
24                 </form>
25             </div>
26         </div>
27     </div>
28 {% endblock %}
```

En el archivo [forms.py](#) agregamos UnidadForm:

```
class UnidadForm(forms.ModelForm):
    class Meta:
        model = Unidad
        fields = ['descripcion', 'estado']
```

Creamos la vista para agregar una nueva unidad en [views.py](#) de VentasApp:

```
def agregarunidad(request):
    if request.method=="POST":
        form=UnidadForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'Unidad agregada correctamente')
            return redirect("listarunidad")
        else:
            form=UnidadForm()

    context={'form':form}
    return render(request,"unidad/agregar.html",context)
```

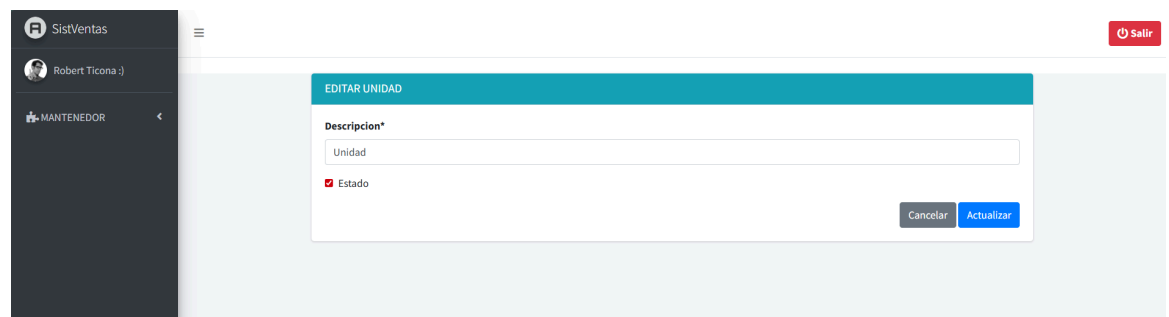
Agregamos la vista a las urls:

```
#Unidad
```

```
path('listarunidad/',listarunidad,name="listarunidad"),  
path('agregarunidad/',agregarunidad ,name="agregarunidad"),  
path('editarunidad/<int:id>/',editarunidad ,name="editarunidad"),  
path('eliminarunidad/<int:id>/',eliminarunidad ,name="eliminarunidad"),
```

- “Editar Unidades”

Vista:



Código:

En el archivo editar.html de la carpeta cliente

```
{% extends "plantilla.html" %}  
  
{% block title %}  
Editar Unidad  
{% endblock %}  
  
{% block content %}  
    {% load crispy_forms_tags %}  
    <div class="container">  
        <div class="card card-info mx-4 my-4">  
            <div class="card-header">  
                <h1 class="card-title">EDITAR UNIDAD</h1>  
            </div>  
            <div class="card-body">  
                <form method="post">  
                    {% csrf_token %}  
                    {{ form|crispy }}  
                    <div class="d-flex justify-content-end">  
                        <a href="{% url 'listarunidad' %}" class="btn btn-secondary">Cancelar</a>  
                        &nbsp;  
                        <button type="submit" class="btn btn-primary">Actualizar</button>  
                    </div>  
                </form>  
            </div>  
        </div>  
    </div>  
{% endblock %}
```

Creamos la vista para editar una unidad en [views.py](#) de VentasApp:

```
def editarunidad(request,id):
    unidad=Unidad.objects.get(idunidad=id)
    if request.method=="POST":
        form=UnidadForm(request.POST,instance=unidad)
        if form.is_valid():
            form.save()
            messages.success(request, 'unidad editada correctamente')
            return redirect("listarunidad")
        else:
            form=UnidadForm(instance=unidad)

    context={"form":form}
    return render(request,"unidad/editar.html",context)
```

Agregamos la vista a las urls:

```
#Unidad

path('listarunidad/',listarunidad,name="listarunidad"),
path('agregarunidad/',agregarunidad ,name="agregarunidad"),
path('editarunidad/<int:id>/',editarunidad ,name="editarunidad"),
path('eliminarunidad/<int:id>/',eliminarunidad ,name="eliminarunidad"),
```

- “Eliminar Unidades”

Vista:



Código:

Creamos la vista para eliminar una unidad en [views.py](#) de VentasApp:

```
def eliminarunidad(request,id):
    unidad=Unidad.objects.get(idunidad=id)
    descripcion = unidad.descripcion
    unidad.estado=False
    unidad.save()
    messages.success(request, 'Unidad eliminada correctamente')
    return redirect("listarunidad")
```

Agregamos la vista a las urls:

```
#Unidad

path('listarunidad/',listarunidad,name="listarunidad"),
path('agregarunidad/',agregarunidad ,name="agregarunidad"),
path('editarunidad/<int:id>/',editarunidad ,name="editarunidad"),
path('eliminarunidad/<int:id>/',eliminarunidad ,name="eliminarunidad"),
```

Codigo js que llama a sweetalert y eliminar unidad (en listar.html):

```
{% block js %}
<script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>
{% if messages %}
    {% for m in messages %}
        <script>
            Swal.fire({
                "title": "¡Éxito!",
                "text": "{m}",
                "icon": "success"
            })
        </script>
    {% endfor %}
{% endif %}

<script>
    function eliminarUnidad(event, id, descripcion) {
        event.preventDefault();

        Swal.fire({
            title: "¿Realmente desea eliminar el registro?",
            text: "Código: " + id + " | Descripción: " + descripcion,
            icon: "question",
            showCancelButton: true,
            cancelButtonText: "No",
            confirmButtonText: "Sí",
            reverseButtons: true,
            confirmButtonColor: "#dc3545"
        }).then(function(result) {
            if (result.isConfirmed) {
                window.location.href = "/eliminarunidad/" + id + "/";
            }
        });
    }
</script>
{% endblock %}
```

Agregamos la llamada a la función eliminarunidad al botón eliminar en el listar.html

```
<a href="#" onclick="eliminarUnidad(event, '{{itemunidad.idunidad}}', '{{itemunidad.descripcion}}')" class="btn btn-danger btn-sm">
  <i class="fa fa-trash"></i> Eliminar
</a>
```

CRUD: “Productos”

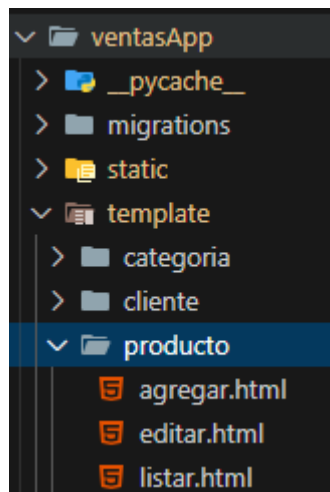
- “Listado de Productos”

Vista:

LISTADO DE PRODUCTOS + NUEVO						
<input type="text" value="Buscar"/> 🔍						
CODIGO	DESCRIPCION	CATEGORIA	PRECIO	STOCK	UNIDAD	OPCIONES
1	Laptop XYZ	Electrónica	1200.00	50	Unidad	✎ Editar 🗑 Eliminar
2	Arroz Costeño 5kg	Alimentos	15.50	200	Kilogramo	✎ Editar 🗑 Eliminar
3	Gaseosa Coca Cola 1L	Bebidas	5.00	6	Litro	✎ Editar 🗑 Eliminar
4	Camisa de algodón	Ropa	45.90	100	Unidad	✎ Editar 🗑 Eliminar

Código:

En el archivo listar.html de la carpeta producto



```
{% extends "plantilla.html" %}

{% block title %}
Listar Productos
{% endblock %}

{% block content %}
<div class="container-fluid">
  <div class="card card-outline card-info mt-3">
    <div class="card-header d-flex align-items-center flex-wrap" style="gap: 10px;">
      <h1 class="card-title mb-0 ">LISTADO DE PRODUCTOS</h1>
      <a href="{% url 'agregarproducto' %}" class="btn btn-success btn-sm d-flex align-items-center fw-bold shadow-sm px-3 py-2" style="gap: 8px;">
        <i class="fas fa-plus me-2"></i> NUEVO
      </a>
    </div>
    <div class="card-body">
      <div class="row">
        <div class="col-md-6 pull-right">
          <form action="" method="GET">
            <div class="input-group">
              <input type="text" name="buscar" id="buscar" class="form-control" autofocus placeholder="Buscar" value="{{buscar}}">
              <span class="input-group-btn">
                <button class="btn btn-primary">
                  <i class="fa fa-search"></i>
                </button>
              </span>
            </div>
          </form>
        </div>
      </div>
      <div class="table-responsive">
        <table id="tablasubfamilias" class="table table-striped mt-4" style="width:100%">
          <thead class="bg-info text-white">
            <tr>
              <th>CODIGO</th>
              <th>DESCRIPCION</th>
              <th>CATEGORIA</th>
              <th>PRECIO</th>
              <th>STOCK</th>
              <th>UNIDAD</th>
              <th><center>OPCIONES</center></th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <td>1</td>
              <td>Producto 1</td>
              <td>Categoria 1</td>
              <td>100</td>
              <td>10</td>
              <td>Unidad 1</td>
              <td><center><a href="#">Editar</a> <a href="#">Eliminar</a></td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>
</div>
{% endblock %}
```

```
{% if producto.has_next %}
    {% for itemproducto in producto %}
        <tr>
            <td>{{ itemproducto.idproducto }}</td>
            <td>{{ itemproducto.descripcion }}</td>
            <td>{{ itemproducto.categoria.descripcion }}</td>
            <td>{{ itemproducto.precio }}</td>
            <td>{{ itemproducto.stock }}</td>
            <td>{{ itemproducto.unidad.descripcion }}</td>
            <td style="text-align: center;">
                <a href="{% url 'editarproducto' itemproducto.idproducto %" class="btn btn-info btn-sm">
                    <i class="fa fa-edit"></i> Editar
                </a>
                <a href="#" onclick="eliminarProducto(event,'{{itemproducto.idproducto}}','{{itemproducto.descripcion}}')" class="btn btn-danger btn-sm">
                    <i class="fa fa-trash"></i> Eliminar
                </a>
            </td>
        </tr>
    {% endfor %}
{% else %}
    <tr style="text-align: center;">
        <td colspan="7">
            <p>No hay Registros</p>
        </td>
    </tr>
{% endif %}
</table>

{% if producto.has_other_pages %}
    <div class="d-flex justify-content-center mt-3">
        <nav>
            <ul class="pagination">
                {% if producto.has_previous %}
                    <li class="page-item">
                        <a class="page-link" href="?page=1{% if buscar %}&buscar={{ buscar }}{% endif %}">Primero</a>
                    </li>
                    <li class="page-item">
                        <a class="page-link" href="?page={{ producto.previous_page_number }}{% if buscar %}&buscar={{ buscar }}{% endif %}">Anterior</a>
                    </li>
                {% endif %}
                {% for num in producto.paginator.page_range %}
                    {% if producto.number == num %}
                        <li class="page-item active">
                            <a class="page-link">{{ num }}</a>
                        </li>
                    {% elif num > producto.number|add:-3 and num < producto.number|add:3 %}
                        <li class="page-item">
                            <a class="page-link" href="?page={{ num }}{% if buscar %}&buscar={{ buscar }}{% endif %}">{{ num }}</a>
                        </li>
                    {% endif %}
                {% endfor %}
                {% if producto.has_next %}
                    <li class="page-item">
                        <a class="page-link" href="?page={{ producto.next_page_number }}{% if buscar %}&buscar={{ buscar }}{% endif %}">Siguiente</a>
                    </li>
                    <li class="page-item">
                        <a class="page-link" href="?page={{ producto.paginator.num_pages }}{% if buscar %}&buscar={{ buscar }}{% endif %}">Último</a>
                    </li>
                {% endif %}
            </ul>
        </nav>
    </div>
{% endif %}
</div>
</div>
</div>
{% endblock %}
```

Creamos la vista para el listado en [views.py](#) de VentasApp:

```
def listarproducto(request):
    queryset=request.GET.get("buscar", "")
    producto=Producto.objects.filter(estado=True)
    if queryset:
        producto=Producto.objects.filter(Q(descripcion__icontains=queryset),estado=True).distinct()

    paginator = Paginator(producto, 5)
    page_number = request.GET.get('page')
    page_obj = paginator.get_page(page_number)

    context={'producto':page_obj, 'buscar': queryset}
    return render(request,"producto/listar.html",context)
```

Agregamos la vista a las urls:

```
#Producto  
path('listarproducto/',listarproducto,name="listarproducto"),
```

- “Registrar Productos”

Vista:

AGREGAR PRODUCTO

Descripcion*

Categoria*

Precio*

Stock*

Unidad*

☒ Estado

Cancelar

Guardar

Código:

En el archivo agregar..html de la carpeta producto

```
{% extends "plantilla.html" %}

{% block title %}
Agregar Producto
{% endblock %}

{% block content %}
    {% load crispy_forms_tags %}
    <div class="container">
        <div class="card card-info mx-4 my-4">
            <div class="card-header">
                <h1 class="card-title">AGREGAR PRODUCTO</h1>
            </div>
            <div class="card-body">
                <form method="POST">
                    {% csrf_token %}
                    {{ form|crispy }}
                    <div class="d-flex justify-content-end">
                        <a href="{% url 'listarproducto' %}" class="btn btn-secondary">Cancelar</a>
                        &nbsp;
                        <button type="submit" class="btn btn-primary">Guardar</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
{% endblock %}
```

En el archivo [forms.py](#) agregamos ProductoForm:

```
class ProductoForm(forms.ModelForm):
    class Meta:
        model = Producto
        fields = ['descripcion', 'categoria', 'precio', 'stock', 'unidad', 'estado']
```

Creamos la vista para agregar un producto en [views.py](#) de VentasApp:

```
def agregarproducto(request):
    if request.method=="POST":
        form=ProductoForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'Producto agregado correctamente')
            return redirect("listarproducto")
        else:
            form=ProductoForm()
            form.fields['categoria'].queryset = Categoria.objects.filter(estado=1)
            form.fields['unidad'].queryset = Unidad.objects.filter(estado=1)
            context={'form':form}
            return render(request,"producto/agregar.html",context)
```

Agregamos la vista a las urls:

```
#Producto

path('listarproducto/',listarproducto,name="listarproducto"),
path('agregarproducto/',agregarproducto ,name="agregarproducto"),
path('editarproducto/<int:id>/',editarproducto ,name="editarproducto"),
path('eliminarproducto/<int:id>/',eliminarproducto, name="eliminarproducto"),
```

- “Editar Productos”

Vista:

EDITAR PRODUCTO

Descripcion*

Laptop XYZ

Categoria*

Electrónica

Precio*

1200.00

Stock*

50

Unidad*

Unidad

☒ Estado

Cancelar

Actualizar

Código:

En el archivo editar.html de la carpeta producto

```
{% extends "plantilla.html" %}

{% block title %}
Editor Producto
{% endblock %}

{% block content %}
    {% load crispy_forms_tags %}
    <div class="container">
        <div class="card card-info mx-4 my-4">
            <div class="card-header">
                <h1 class="card-title">EDITAR PRODUCTO</h1>
            </div>
            <div class="card-body">
                <form method="post">
                    {% csrf_token %}
                    {{ form|crispy }}
                    <div class="d-flex justify-content-end">
                        <a href="{% url 'listarproducto' %}" class="btn btn-secondary">Cancelar</a>
                        &nbsp;
                        <button type="submit" class="btn btn-primary">Actualizar</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
{% endblock %}
```

Creamos la vista para editar un producto en [views.py](#) de VentasApp:

```
def editarproducto(request,id):
    producto=Producto.objects.get(idproducto=id)
    if request.method=="POST":
        form=ProductoForm(request.POST,instance=producto)
        if form.is_valid():
            form.save()
            messages.success(request, 'Producto editado correctamente')
            return redirect("listarproducto")
    else:
        form=ProductoForm(instance=producto)
        form.fields['categoria'].queryset = Categoria.objects.filter(estado=1)
        form.fields['unidad'].queryset = Unidad.objects.filter(estado=1)
    context={"form":form}
    return render(request,"producto/editar.html",context)
```

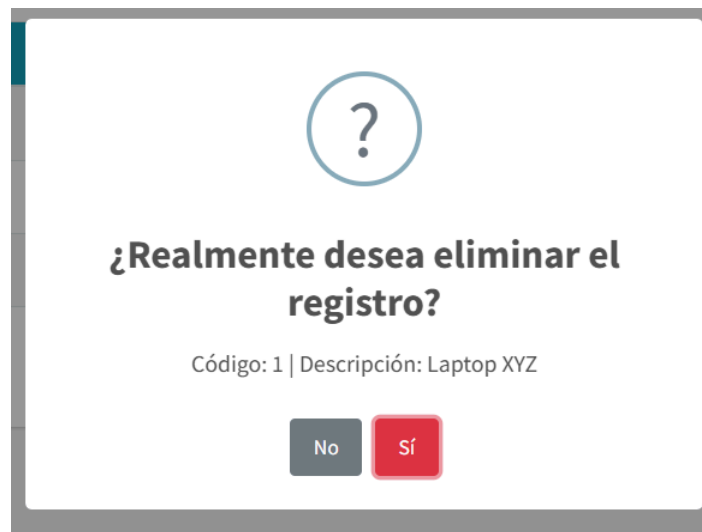
Agregamos la vista a las urls:

```
#Producto

path('listarproducto/',listarproducto,name="listarproducto"),
path('agregarproducto/',agregarproducto ,name="agregarproducto"),
path('editarproducto/<int:id>/',editarproducto ,name="editarproducto"),
path('eliminarproducto/<int:id>/',eliminarproducto, name="eliminarproducto"),
```

- “Eliminar Productos”

Vista:



Código:

Creamos la vista para eliminar un producto en [views.py](#) de VentasApp:

```
def eliminarproducto(request,id):
    producto=Producto.objects.get(idproducto=id)
    descripcion = producto.descripcion
    producto.estado=False
    producto.save()
    messages.success(request, 'Producto eliminado correctamente')
    return redirect("listarproducto")
```

Agregamos la vista a las url:

```
#Producto

path('listarproducto/',listarproducto,name="listarproducto"),
path('agregarproducto/',agregarproducto ,name="agregarproducto"),
path('editarproducto/<int:id>/',editarproducto ,name="editarproducto"),
path('eliminarproducto/<int:id>/',eliminarproducto, name="eliminarproducto"),
```

Codigo js que llama a sweetalert y eliminar producto (en listar.html):

```
{% block js %}
<script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>
{% if messages %}
    {% for m in messages %}
        <script>
            Swal.fire({
                "title": "¡Éxito!",
                "text": "{{m}}",
                "icon": "success"
            })
        </script>
    {% endfor %}
{% endif %}

<script>
    function eliminarProducto(event, id, descripcion) {
        event.preventDefault();

        Swal.fire({
            title: "¿Realmente desea eliminar el registro?",
            text: "Código: " + id + " | Descripción: " + descripcion,
            icon: "question",
            showCancelButton: true,
            cancelButtonText: "No",
            confirmButtonText: "Sí",
            reverseButtons: true,
            confirmButtonColor: "#dc3545"
        }).then(function(result) {
            if (result.isConfirmed) {
                window.location.href = "/eliminarproducto/" + id + "/";
            }
        });
    }
</script>
{% endblock %}
```

Agregamos la llamada a la función `eliminarproducto()` al botón eliminar en el `listar.html`

```
<a href="#" onclick="eliminarProducto(event, '{{itemproducto.idproducto}}', '{{itemproducto.descripcion}}')" class="btn btn-danger btn-sm">
    <i class="fa fa-trash"></i> Eliminar
</a>
```