

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS – UNIMINUTO**

**Evidencias de Laboratorio:**

Desarrollo de una Aplicación CRUD con Flask y MySQL

**Estudiantes:**

Valentina Sánchez Rivera -

Andrés Felipe Molano Montejo.

**Fundamentos de programación web**

**NRC: 80769**

**Nombre del tutor: Alexander Ramírez Camargo**

**Bogotá, 16 de noviembre de 2025**

## **FLASK-VALENTINA-Y-ANDRES-PROGRAMACIONWEB**

**GITHUB:** <https://github.com/DresUni/FLASK-VALENTINA-Y-ANDRES-PROGRAMACIONWEB/tree/main>

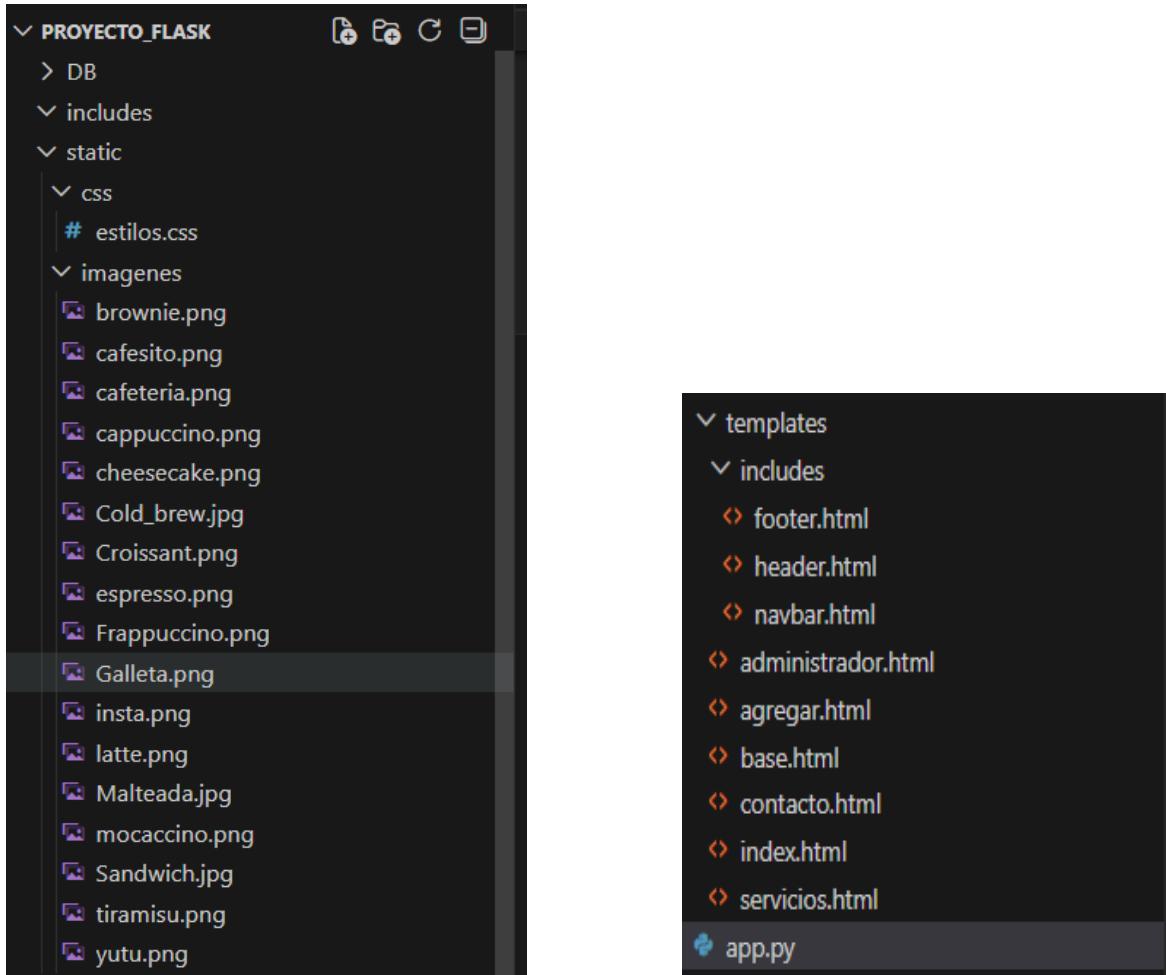
Es una aplicación web de cafetería virtual, para revisar catalogo actual y dar opiniones/aportes al servicio se desarrollada con Python y Flask. El proyecto incluye un sistema de plantillas (Jinja) y gestión básica de conexión a MySQL para ofrecer una experiencia sencilla y accesible.

### **INSTRUCCIONES DE FUNCIONAMIENTO:**

- En este repositorio se encuentra el archivo zip con el proyecto de la aplicación web.
- Es necesario descomprimirlo.
- Abrir la carpeta en Visual Studio Code.
- Abrir la terminal / CMD.
- Comprobar que python esté instalado en el equipo.
- Abrir el proyecto en la terminal.
- Crear el entorno virtual con: python -m venv my\_env
- Cuando se haya activado el prompt iniciará con :(my\_env)
- Instalar flask con : pip install flask
- tambien necesario para el funcionamiento de la base de datos instalar en la terminal: pip install flask flask-mysqldb
- tambien importar la base de datos "crud\_flask" con ese mismo nombre, la cual esta en el archivo comprimido; en la carpeta DB
- Verificar que esté ubicado en la carpeta principal: cd '\ proyecto\_flask'
- Ejecutar la aplicación con: python app.py
- Debería aparecer algo así: Running on http://127.0.0.1:5000/
- Si no logra ejecutarlo probar con: py app.py
- La aplicación se ejecutará en el navegador.

## Parte 1: Configuración del Proyecto

### 1. Crear la estructura de carpetas



### 2. Crear base de datos en MySQL e insertar datos

```
1 -- CREAR BASE DE DATOS
2 CREATE DATABASE crud_flask;
3 USE crud_flask;
```

```
1 -- TABLA DE USUARIOS
2 CREATE TABLE Usuarios (
3     idUsuario INT AUTO_INCREMENT PRIMARY KEY,
4     nombre VARCHAR(100) NOT NULL,
5     correo VARCHAR(120) UNIQUE NOT NULL,
6     contraseña VARCHAR(255) NOT NULL,
7     telefono VARCHAR(20),
8     cumpleaños DATE
9 );
```

```
1 -- TABLA CATEGORIAS
2
3 CREATE TABLE Categorias (
4     idCategoria INT AUTO_INCREMENT PRIMARY KEY,
5     nombre_categoria| VARCHAR(50) NOT NULL
6 );
```

```
1 -- TABLA PRODUCTOS (MENÚ)
2
3 CREATE TABLE Productos (
4     idProducto INT AUTO_INCREMENT PRIMARY KEY,
5     nombre_producto VARCHAR(100) NOT NULL,
6     descripcion TEXT,
7     precio DECIMAL(10,2) NOT NULL,
8     idCategoria INT,
9     disponible BOOLEAN DEFAULT TRUE,
10    FOREIGN KEY (idCategoria) REFERENCES Categorias(idCategoria)
11 );
```

```
1 -- TABLA PEDIDOS
2 CREATE TABLE Pedidos (
3     idPedido INT AUTO_INCREMENT PRIMARY KEY,
4     idUsuario INT,
5     fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
6     total DECIMAL(10,2) DEFAULT 0,
7     metodoPago VARCHAR(50),
8     FOREIGN KEY (idUsuario) REFERENCES Usuarios(idUsuario)
9 );
```

```
1 -- DETALLE DE PEDIDO
2
3 CREATE TABLE DetallePedido (
4     idDetalle INT AUTO_INCREMENT PRIMARY KEY,
5     idPedido INT,
6     idProducto INT,
7     cantidad INT NOT NULL,
8     subtotal DECIMAL(10,2),
9     FOREIGN KEY (idPedido) REFERENCES Pedidos(idPedido),
10    FOREIGN KEY (idProducto) REFERENCES Productos(idProducto)
11 );
```

```
1 -- TABLA INVENTARIO
2 CREATE TABLE Inventario (
3     idInventario INT AUTO_INCREMENT PRIMARY KEY,
4     idProducto INT,
5     cantidad_inventario INT NOT NULL,
6     FOREIGN KEY (idProducto) REFERENCES Productos(idProducto)
7 );
```

## INSERTAR DATOS

```
1 -- INSERTAR CATEGORIAS
2 INSERT INTO Categorias (nombre_categoria)
3 VALUES
4 ('Bebidas Calientes'),
5 ('Bebidas Frías'),
6 ('Panaderia'),
7 ('Postres');
```

```
-- INSERTAR PRODUCTOS
INSERT INTO Productos (nombre_producto, descripcion, precio, idCategoria)
VALUES
('Latte', 'Café con leche espumosa', 10.000, 1),
('Capuccino', 'Café con espuma y canela', 10.000, 1),
('Espresso', 'Bebida de café concentrada', 7.000, 1),
('Mocaccino', 'Bebida a base de café que combina espresso, leche vaporizada y chocolate', 15.000, 1),

('Frappuccino', 'Bebida fría cremosa', 12.000, 2),
('Cold Brew', 'Café frío infusionado', 10.000, 2),
('Malteada', 'Bebida helada y espesa a base de helado y leche', 20.000, 2),

('Sandwich', 'Pan artesanal con jamón y queso', 10.000, 3),
('Croissant', 'Croissant de mantequilla recién horneado', 7.000, 3),
('Galleta', 'Galleta con chips de chocolate', 5.000, 3),

('Cheesecake', 'Tarta de queso de frutos rojos', 10.000, 4),
('Tiramisú', 'Postre frío de café', 16.000, 4),
('Brownie con helado', 'Brownie de chocolate acompañado de helado de vainilla', 18.000, 4);
```

```
1 -- INSERTAR USUARIOS
2 INSERT INTO Usuarios (nombre, correo, contraseña, telefono, cumpleaños)
3 VALUES
4 ('Laura Perez', 'laura@gmail.com', '123456', '3001234567', '1998-05-21'),
5 ('Carlitos Gutierrez', 'carlos@gmail.com', 'abc123', '3019876543', '1995-11-10'),
6 ('Luisito comunica', 'Luisito@gmail.com', 'abc123', '3019876543', '1995-11-10'),
7 ('Daddy Yankee', 'Daddy@gmail.com', 'abc123', '3019876543', '1995-11-10'),
8 ('Ana Torres', 'ana@gmail.com', 'pass123', '3205558899', '2000-02-14');
```

```
1 -- INSERTAR INVENTARIO
2 INSERT INTO Inventario (idProducto, cantidad_inventario)
3 VALUES
4 (1, 50),
5 (2, 40),
6 (3, 60),
7 (4, 30),
8 (5, 35),
9 (6, 20),
10 (7, 25),
11 (8, 45),
12 (9, 50),
13 (10, 45),
14 (11, 45),
15 (12, 45),
```

```

1 -- INSERTAR PEDIDOS
2
3 INSERT INTO Pedidos (idUsuario, total, metodoPago)
4 VALUES
5 (1, 20.000, 'Tarjeta'),
6 (2, 12.000, 'Efectivo'),
7 (3, 18.000, 'Tarjeta'),
8 (4, 15.000, 'Tarjeta'),
9 (5, 36.000, 'Efectivo');
10

```

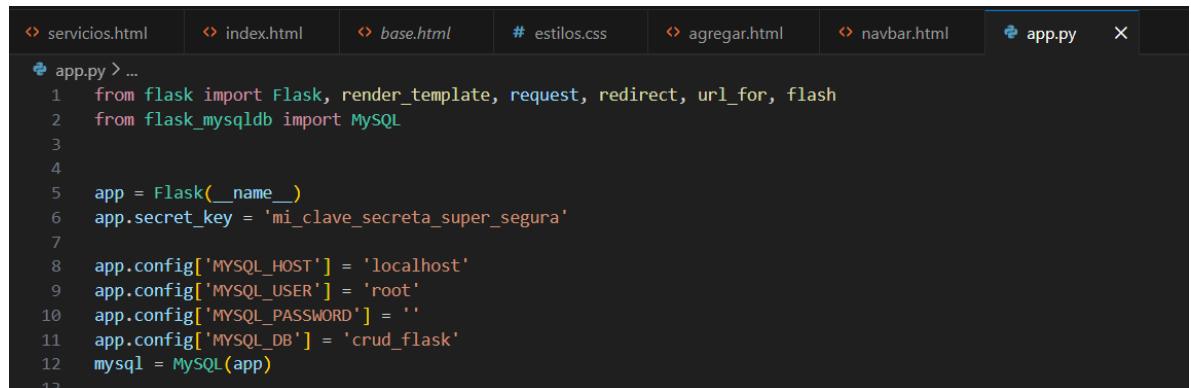
---

```

1 -- INSERTAR DETALLE PEDIDO
2 INSERT INTO DetallePedido (idPedido, idProducto, cantidad, subtotal)
3 VALUES
4 -- Pedido 1 (Laura)
5 (1, 1, 1, 10.000),
6 (1, 2, 1, 10.000),
7
8 -- Pedido 2 (Carlos)
9 (2, 5, 1, 12.000),
10
11 -- Pedido 3 (Luisito)
12 (3, 13, 1, 18.000),
13
14 -- Pedido 4 (Daddy)
15 (4, 10, 1, 5.000),
16 (4, 11, 1, 10.000),
17
18 -- Pedido 5 (Ana)
19 (5, 12, 1, 16.000),
20 (5, 10, 2, 5.000);

```

### 3. Conectar Flask a la base de datos



```

servicios.html index.html base.html estilos.css agregar.html navbar.html app.py
❶ app.py > ...
1  from flask import Flask, render_template, request, redirect, url_for, flash
2  from flask_mysqldb import MySQL
3
4
5  app = Flask(__name__)
6  app.secret_key = 'mi_clave_secreta_super_segura'
7
8  app.config['MYSQL_HOST'] = 'localhost'
9  app.config['MYSQL_USER'] = 'root'
10 app.config['MYSQL_PASSWORD'] = ''
11 app.config['MYSQL_DB'] = 'crud_flask'
12 mysql = MySQL(app)
13

```

## Parte 2: Desarrollo del CRUD

### 1. Crear (Create)

En nuestra web de cafetería permitimos que el usuario tenga un apartado de agregar usuario/opinión sobre lo que le gusto en nuestra cafetería y podemos evidenciar “CREATE” en **agregar.html**

```
@app.route('/agregar')
def agregar_usuario_form():
    """Muestra la plantilla con el formulario para agregar un nuevo usuario."""
    return render_template('agregar.html')

@app.route('/add_usuario', methods=['POST'])
def add_usuario():
    """Ruta para procesar el formulario (Create) hacia MySQL."""
    if request.method == 'POST':
        nombre = request.form.get('nombre')
        correo = request.form.get('correo')
        telefono = request.form.get('telefono')
        cumpleanos = request.form.get('cumpleanos')

        comentario = request.form.get('comentario', '')

        if not all([nombre, correo, telefono, cumpleanos]):
            flash('Error: El formulario no envió todos los campos esperados.', 'danger')
            return redirect(url_for('agregar_usuario_form'))

        try:
            cur = mysql.connection.cursor()

            cur.execute("""
                INSERT INTO usuarios (nombre, correo, telefono, cumpleaños, comentario)
                VALUES (%s, %s, %s, %s, %s)
            """, (nombre, correo, telefono, cumpleanos, comentario))

            mysql.connection.commit()
            cur.close()
            flash('¡Usuario agregado correctamente a la base de datos!', 'success')

        except Exception as e:
            flash(f'Error al insertar el usuario: {e}', 'danger')

        return redirect(url_for('index'))

    return redirect(url_for('agregar_usuario_form'))
```

```
8 <form action="{{ url_for('add_usuario') }}" method="POST">
9   <div class="mb-4">
10     <label for="nombre" class="block text-gray-700 font-bold mb-2">Nombre Completo:</label>
11     <input type="text" id="nombre" name="nombre" required
12       style="width: 700px; padding: 10px; border: 1px solid #ccc; border-radius: 5px; margin-left: 10px;">
13   </div><br>
14
15   <div class="mb-4">
16     <label for="correo" class="block text-gray-700 font-bold mb-2">Correo Electrónico:</label>
17     <input type="email" id="correo" name="correo" required
18       style="width: 700px; padding: 10px; border: 1px solid #ccc; border-radius: 5px; margin-left: 10px;">
19   </div><br>
20
21   <div class="mb-4">
22     <label for="telefono" class="block text-gray-700 font-bold mb-2">Teléfono: &ampnbsp&ampnbsp&ampnbsp&ampnbsp&ampnbsp&nbsp;</label>
23     <input type="tel" id="telefono" name="telefono" required
24       style="width: 700px; padding: 10px; border: 1px solid #ccc; border-radius: 5px; margin-left: 10px;">
25   </div><br>
26
27   <div class="mb-6">
28     <label for="cumpleanos" class="block text-gray-700 font-bold mb-2">Cumpleaños:</label>
29     <input type="date" id="cumpleanos" name="cumpleanos" required
30       style="width: 700px; padding: 10px; border: 1px solid #ccc; border-radius: 5px; margin-left: 10px;">
31   </div><br>
32
33   <div class="mb-6">
34     <label for="comentario" class="block text-gray-700 font-bold mb-2">Comentario: </label>
35     <textarea id="comentario" name="comentario" rows="4" placeholder="Damos tu opinión del servicio!">
36       style="width: 700px; padding: 10px; border: 1px solid #ccc; border-radius: 5px; resize: vertical; margin-left: 10px;"</textarea>
37   </div>
38
39   <button type="submit"
40         style="background-color: #5d4037; color: white; padding: 12px 20px; border: none; border-radius: 5px; cursor: pointer; width: 250px; font-size: 1.1em; transition: background-color 0.3s ease;">
41     Registrar!
42   </button>
43 </form><br>
44
45 
46
47 <% endblock %>
```



**Regístrate**

---

Nombre Completo:

Correo Electrónico:

Teléfono:

Cumpleaños:  [calendario]

Comentario:

**Regístral**



#### 2-3-4. “(READ) (UPDATE) (DELETE)”

En nuestra web de cafetería las acciones de leer/actualizar y borrar usuario “**(READ) (UPDATE) (DELETE)**” tienen un apartado que maneja el administrador, el cual lo dejamos abierto en esta versión del documento en **administrador.html**

```

@app.route('/administrador')
def administrador():
    """
    Ruta para el panel de administración.
    Permite listar todos los usuarios y, opcionalmente, entrar en modo de edición
    para un usuario específico usando el parámetro 'editar' en la URL.
    """
    try:
        id_para_editar = request.args.get('editar', None, type=int)

        cur = mysql.connection.cursor()

        cur.execute('SELECT idUsuario, nombre, correo, telefono, cumpleaños, comentario FROM usuarios')
        usuarios_data = cur.fetchall()
        cur.close()

        return render_template('administrador.html', usuarios=usuarios_data, id_para_editar=id_para_editar)

    except Exception as e:
        flash(f'Error al cargar el panel de administrador: {e}', 'danger')
        return render_template('administrador.html', usuarios=[], id_para_editar=None)
    
```

```

@app.route('/update/<int:id_usuario>', methods=['POST'])
def update_usuario(id_usuario):
    """Procesa la actualización de un usuario desde la tabla de administrador."""
    if request.method == 'POST':
        nombre = request.form.get('nombre')
        correo = request.form.get('correo')
        telefono = request.form.get('telefono')
        cumpleanos = request.form.get('cumpleanos')
        comentario = request.form.get('comentario', '')

    try:
        cur = mysql.connection.cursor()
        cur.execute("""
            UPDATE usuarios
            SET nombre = %s, correo = %s, telefono = %s, cumpleaños = %s, comentario = %s
            WHERE idUsuario = %s
        """, (nombre, correo, telefono, cumpleanos, comentario, id_usuario))

        mysql.connection.commit()
        cur.close()
        flash(f'Usuario ID {id_usuario} actualizado correctamente!', 'success')

    except Exception as e:
        flash(f'Error al actualizar el usuario: {e}', 'danger')

    return redirect(url_for('administrador'))

@app.route('/delete/<int:id_usuario>')
def delete_usuario(id_usuario):
    """Elimina un usuario por su ID."""
    try:
        cur = mysql.connection.cursor()
        cur.execute('DELETE FROM usuarios WHERE idUsuario = %s', (id_usuario,))
        mysql.connection.commit()
        cur.close()
        flash(f'Usuario ID {id_usuario} eliminado correctamente.', 'success')
    except Exception as e:
        flash(f'Error al eliminar el usuario: {e}', 'danger')

    return redirect(url_for('administrador'))

```

```

@app.route('/')
def index():
    """Ruta para Leer/Mostrar todos los usuarios (Read) desde MySQL."""
    try:
        cur = mysql.connection.cursor()
        cur.execute('SELECT idUsuario, nombre, correo, telefono, cumpleanos FROM usuarios')
        data = cur.fetchall()
        cur.close()

        column_names = ['id', 'nombre', 'correo', 'telefono', 'cumpleanos']
        usuarios = [dict(zip(column_names, row)) for row in data]

    return render_template('index.html', usuarios=usuarios)
    except Exception as e:
        flash(f'Error al conectar o leer la base de datos: {e}', 'danger')
        return render_template('index.html', usuarios[])

```

```

servicios.html index.html # estilos.css agregar.html navbar.html app.py administrador.html ...

1  (% extends "base.html" %)
2  (% block title %)Administrador(% endblock %)
3
4  (% block content %)
5
6
7  <div class="admin-container">
8      <h2 style="text-align: center; color: #333;">Panel de Administración de Usuarios</h2>
9
10     (% with messages = get_flashed_messages(with_categories=true) %)
11         (% if messages %)
12             (% for category, message in messages %)
13                 <div class="flash-message flash{{ category }}">{{ message }}</div>
14             (% endfor %)
15         (% endif %)
16     (% endwith %)
17
18     <table class="data-table">
19         <thead>
20             <tr>
21                 <th>ID</th>
22                 <th>Nombre</th>
23                 <th>Correo</th>
24                 <th>Teléfono</th>
25                 <th>Cumpleaños</th>
26                 <th>Comentario</th>
27                 <th style="width: 180px;">Acciones</th>
28             </tr>
29         </thead>
30         <tbody>
31             (% if not usuarios %)
32                 <tr>
33                     <td colspan="7" style="text-align: center;">No hay usuarios registrados.</td>
34                 </tr>
35             (% endif %)
36
37             (% for usuario in usuarios %)
38                 (% if id_para_editar == usuario[0] %)
39
40                     <tr>
41                         <td><form action="{{ url_for('update_usuario', id_usuario=usuario[0]) }}" method="POST">
42                             <td>{{ usuario[0] }}</td>
43                             <td><input type="text" name="nombre" value="{{ usuario[1] }}" class="form-edit-input" required></td>
44                             <td><input type="email" name="correo" value="{{ usuario[2] }}" class="form-edit-input" required></td>
45                             <td><input type="tel" name="telefono" value="{{ usuario[3] }}" class="form-edit-input" required></td>
46                             <td><input type="date" name="cumpleanos" value="{{ usuario[4] | string }}" class="form-edit-input" required></td>
47                             <td><textarea name="comentario" class="form-edit-textarea">{{ usuario[5] if usuario[5] is not none else '' }}</td>
48                         </td>
49                         <td><button type="submit" class="btn-save">Guardar</button>
50                         <a href="{{ url_for('administrador') }}" class="btn-cancel">Cancelar</a>
51                     </td>
52                 </form>
53             </tr>
54
55             (% else %)
56
57                 <tr>
58                     <td>{{ usuario[0] }}</td>
59                     <td>{{ usuario[1] }}</td>
60                     <td>{{ usuario[2] }}</td>
61                     <td>{{ usuario[3] }}</td>
62                     <td>{{ usuario[4] }}</td>
63                     <td>{{ usuario[5] }}</td>
64                     <td>
65                         <a href="{{ url_for('administrador', editar=usuario[0]) }}" class="admin-link">Editar</a>
66
67                         <a href="{{ url_for('delete_usuario', id_usuario=usuario[0]) }}" class="btn-delete"
68                             onclick="return confirm('¿Estás seguro de ELIMINAR a {{ usuario[1] }}?')">
69                             Eliminar
70                         </a>
71                     </td>
72                 </tr>
73
74             (% endif %)
75             (% endfor %)
76         </tbody>
77     </table>
78 </div>
79 (% endblock %)

```

**El Cafesito ☕**

Inicio | Servicios | Registrar | Contacto **ADMINISTRADOR**

**Panel de Administración de Usuarios**

¡Usuario agregado correctamente a la base de datos!

ID	Nombre	Correo	Teléfono	Cumpleaños	Comentario	Acciones
1	Laura Perez	laura@gmail.com	3001234567	1998-05-21	None	<a href="#">Editar</a> <a href="#">Eliminar</a>
2	Carlitos Gutierrez	carlos@gmail.com	3019876543	1995-11-10	None	<a href="#">Editar</a> <a href="#">Eliminar</a>
3	Luisito comunica	Luisito@gmail.com	3019876543	1995-11-10	None	<a href="#">Editar</a> <a href="#">Eliminar</a>
4	Daddy Yankee	Daddy@gmail.com	3019876543	1995-11-10	None	<a href="#">Editar</a> <a href="#">Eliminar</a>
5	Ana Torres	ana@gmail.com	3205558899	2000-02-14	None	<a href="#">Editar</a> <a href="#">Eliminar</a>
12	Griselda Llana	grilla@japo.com	3124983166	2024-09-16	que rico todo, supEr	<a href="#">Editar</a> <a href="#">Eliminar</a>

Mostrar todo | Número de filas: 25 ▾

Filtrar filas: Buscar

Opciones extra

← →	idUsuario	nombre	correo	contraseña	telefono	cumpleaños	comentario
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	1	Laura Perez	laura@gmail.com	123456	3001234567	1998-05-21	NULL
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	2	Carlitos Gutierrez	carlos@gmail.com	abc123	3019876543	1995-11-10	NULL
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	3	Luisito comunica	Luisito@gmail.com	abc123	3019876543	1995-11-10	NULL
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	4	Daddy Yankee	Daddy@gmail.com	abc123	3019876543	1995-11-10	NULL
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	5	Ana Torres	ana@gmail.com	pass123	3205558899	2000-02-14	NULL
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	12	Griselda Llana	grilla@japo.com		3124983166	2024-09-16	que rico todo, supEr

↑  Seleccionar todo Para los elementos que están marcados: [Editar](#) [Copiar](#) [Borrar](#) [Exportar](#)

## Parte 3: Plantillas y Herencia (Templates)

### 1. Crear layout.html con bloques `{% block content %}`.

```
servicios.html index.html # estilos.css agregar.html navbar.html app.py base.html

templates > base.html > html
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>{% block title %}Título por Defecto{% endblock %} - Mi Aplicación Flask</title>
7      <link rel="stylesheet" href="{{ url_for('static', filename='css/estilos.css') }}>
8  </head>
9  <body>
10     {% include "includes/header.html" %}
11     {% include "includes/navbar.html" %}
12     <main>
13         {% block content %}
14         {% endblock %}
15     </main>
16     {% include "includes/footer.html" %}
17 </body>
18 </html>
```

### 2. Crear index.html, agregar.html y editar.html que hereden del layout.

```
servicios.html index.html # estilos.css agregar.html navbar.html app.py base.html

templates > index.html > ...
1  {% extends "base.html" %}
2  {% block title %}Inicio{% endblock %}
3
4
5  {% block content %}
6  <div class="titulosborde">
7      <h2 class="titulos" style="text-align: center;" id="valor">Sobre Nuestra Cafetería </h2>
8  </div>
9  <div class="formvalida">
10     <p style="text-align: justify;">
11         Nuestra cafetería es un lugar acogedor donde puedes disfrutar del mejor café
12         acompañado de postres deliciosos. Nos esforzamos por brindar calidad en cada
13         taza y un ambiente ideal para compartir con amigos o relajarte.
14     </p>
15
16     <p>
17         ¡Te invitamos a explorar nuestro menú! Revisa el apartado de
18         <strong> Servicios</strong>
19         y, una vez que sepas lo que quieras, dirígete al apartado de
20         <strong>Registrar</strong>
21         para completar tu pedido o registrarte.
22     </p>
23
24     
25
26  {% endblock %}
```

### **3. Agregar hojas de estilo y condicionales en Jinja2.**

```
servicios.html    index.html    # estilos.css X    agregar.html    navbar.html    app.py

static > css > # estilos.css > formvalida2
1  h2.titulos {
2  |   color: #fffffff;
3  }
4  body{
5  background-color: #ffff0e8c7;
6  }
7  .titulosborde{
8  margin: auto; width: 700px; height: 35px;
9  background-color: #360101;
10 border-radius: 8px;
11 }
12 .logoempresa{
13 width:50px;
14 height:50px; }

15 .redes{
16 width:30px;
17 height:20px;
18 }
19 .formvalida {
20 color: #fffffb; text-align: center;
21 margin: 20px auto; width: 700px; border-radius: 5px;
22 background-color: #rgba(157, 22, 22, 0.85);
23 padding: 20px;
24 }
25 .cafeteria {
26 display: block;
27 margin: 15px auto; width: 400px;
28 height: auto; border-radius: 8px; }

29 .formvalida2 []
30 color: #fffffb;
31 text-align: center; margin: 20px auto; width: 900px; border-radius: 5px;
32 background-color: #rgba(157, 22, 22, 0.85);
33 padding: 20px;
34 }
35 }
36 }
```

```
37  /* Aquí todo lo de la parte administrador */
38
39
40 .admin-container {
41     padding: 20px;
42     max-width: 1300px;
43     margin: 0 auto;
44 }
45 .data-table {
46     width: 100%;
47     border-collapse: collapse;
48     margin-top: 15px;
49     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
50 }
51 .data-table th, .data-table td {
52     border: 1px solid #e0e0e0;
53     padding: 12px;
54     text-align: left;
55     vertical-align: middle;
56 }
57 .data-table th {
58     background-color: #007bff;
59     color: white;
60     font-weight: bold;
61 }
62 .data-table tr:nth-child(even) {
63     background-color: #f9f9f9;
64 }
65 .admin-link, .btn-save, .btn-cancel, .btn-delete {
66     padding: 5px 10px;
67     border-radius: 5px;
68     text-decoration: none;
69     margin-right: 5px;
70     display: inline-block;
71     font-size: 14px;
72     cursor: pointer;
73     transition: background-color 0.2s;
74 }
75 .admin-link {
76     background-color: #28a745;
77     color: white;
78 }
79 .admin-link:hover {
80     background-color: #218838;
```