UNSA
Universidad Nacional de San Agustín

“AÑO DE LA RECUPERACIÓN Y CONSOLIDACIÓN DE LA ECONOMÍA PERUANA”

FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

PRÁCTICA:Análisis de Sensibilidad

ASIGNATURA:

PROGRAMACION WEB 2

DOCENTE:

CARLO JOSE LUIS CORRALES DELGADO

INTEGRANTES:

- SILVA PINO JESUS FRANCISCO

AREQUIPA-PERÚ

2025

Informe de Proyecto: Sistema de Gestión de Reservas Web

Proyecto: Desarrollo de una Aplicación Web para la Gestión de Reservas en el Restaurante "Sazón Peruana".
Repositorio: https://github.com/JesusFSP/Curso-PWeb2.git
pagina Web:https://curso-pweb2-production.up.railway.app/

1. Introducción

El presente documento detalla el plan de desarrollo para una aplicación web destinada a la gestión de reservas de mesas en el restaurante “Sazón Peruana”. El objetivo principal es construir una solución tecnológica robusta y eficiente que modernice el proceso de reserva. Para ello, el proyecto empleará el framework **Django** para el desarrollo del backend, garantizando una base sólida y escalable. En el frontend, se integrará **AngularJS** a través de **Djangular**, lo que permitirá crear una interfaz de usuario interactiva y dinámica. El diseño de la interfaz se apoyará en **Bootstrap 5** para asegurar una experiencia de usuario adaptable y moderna en cualquier dispositivo.

2. Justificación del Proyecto

En el panorama actual de la industria restaurantera en el Perú, se ha identificado una brecha digital significativa: una gran cantidad de establecimientos, especialmente los de tamaño mediano y pequeño, carecen de una página web propia para gestionar sus operaciones y relacionarse con sus clientes. Esta dependencia de intermediarios, como aplicaciones de delivery o directorios, no solo implica el pago de comisiones que afectan su rentabilidad, sino que también limita su capacidad para construir una marca sólida y gestionar directamente la experiencia de sus comensales.
La implementación de un sistema de reservas web propio para el restaurante "Sazón Peruana" aborda esta problemática de frente. Este proyecto no solo representa una solución para optimizar la gestión interna de reservas, sino que también se posiciona como una valiosa oportunidad de negocio. Un sistema de este tipo ofrece los siguientes beneficios:

- **Autonomía y Reducción de Costos:** Elimina la dependencia de plataformas de terceros y sus comisiones asociadas.

- **Canal Directo con el Cliente:** Facilita la comunicación directa, la fidelización y el control total sobre la experiencia del usuario.
- **Ventaja Competitiva:** Proporciona una imagen más profesional y moderna, diferenciando al restaurante de su competencia.
- **Modelo de Negocio Replicable:** El desarrollo de esta plataforma sienta las bases para un modelo de negocio que puede ser adaptado y ofrecido a otros restaurantes que enfrentan la misma necesidad, generando una nueva vía de ingresos.

3. Objetivos del Proyecto

Objetivo General

Desarrollar una aplicación web completa y funcional para la gestión de reservas de mesas en el restaurante "Sazón Peruana".

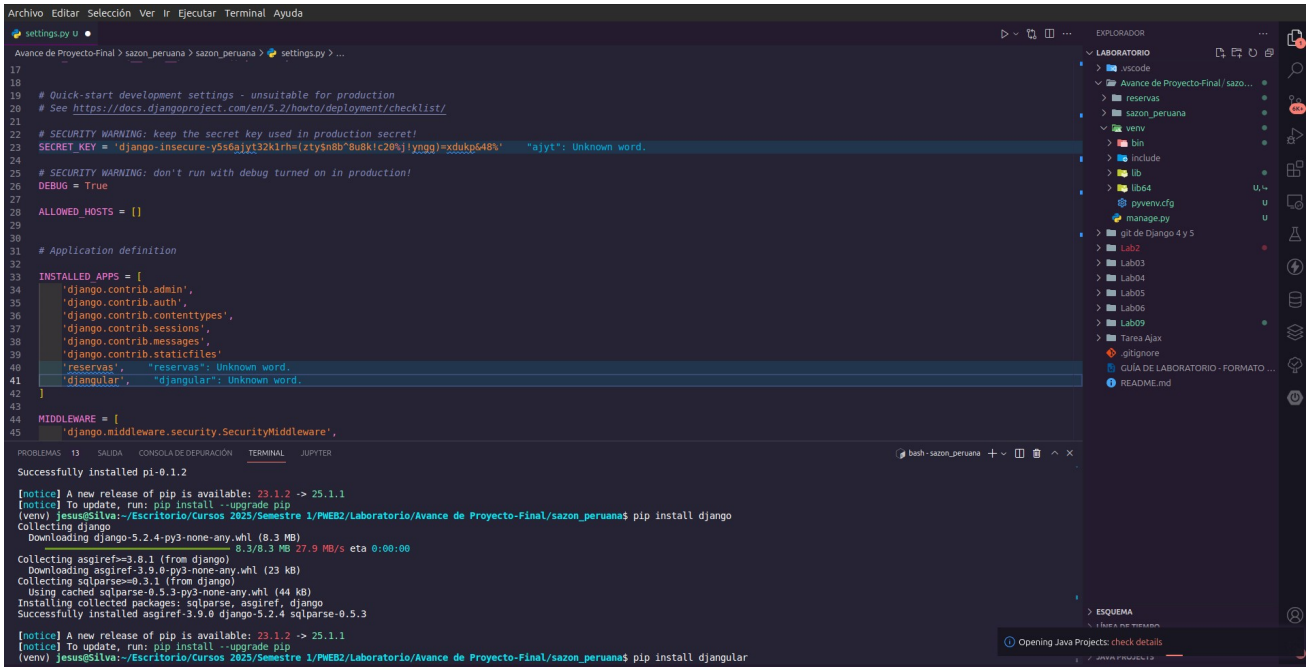
Objetivos Específicos

- Implementar un backend robusto utilizando el framework Django.
- Crear un sistema completo de tipo CRUD (Crear, Leer, Actualizar, Eliminar) para la administración de las reservas.
- Integrar AngularJS mediante Djangular para desarrollar un frontend interactivo.
- Implementar una función de consulta de disponibilidad de mesas en tiempo real mediante tecnología AJAX.
- Configurar una plantilla base utilizando Bootstrap para asegurar un diseño web adaptable (responsive).
- Gestionar el código fuente a través de un repositorio Git y documentar el proyecto en GitHub.

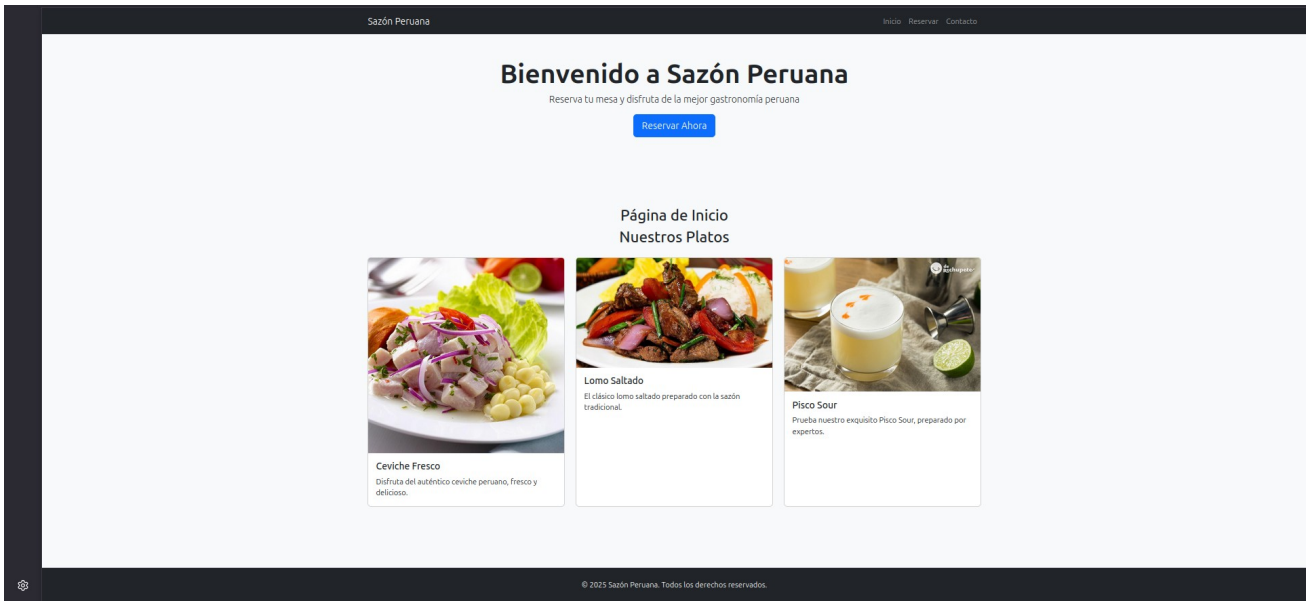
4. Plan de Implementación por Fases

El desarrollo del proyecto se estructurará en las siguientes fases metodológicas:

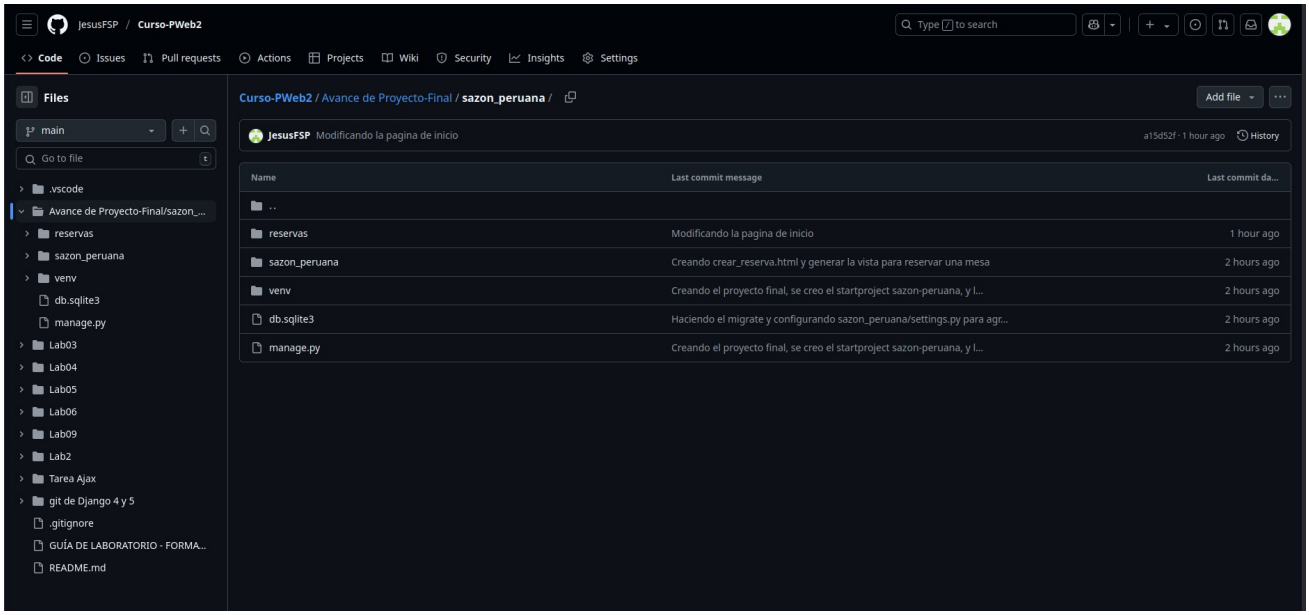
- **Fase 1: Configuración del Entorno e Inicialización del Proyecto:** Preparación del entorno de desarrollo local, instalación de dependencias (Django, Djangular) y creación de la estructura inicial del proyecto.



- **Fase 2: Estructura del Frontend y Plantillas:** Creación de la plantilla base base.html con la integración de Bootstrap 5 y la estructura para la herencia de plantillas de Django.



- **Fase 3: Gestión de Versiones y Repositorio:** Inicialización del repositorio Git y configuración del repositorio remoto en GitHub para el control de versiones.



- **Fase 4: Desarrollo del Backend y Frontend:** Implementación de los modelos, vistas y URLs en Django para el CRUD de reservas y el desarrollo del formulario de cliente con AngularJS.
- **Fase 5: Funcionalidades Adicionales y Despliegue:** Implementación de características opcionales (notificaciones, reportes) y despliegue de la aplicación en un servidor web.

5. Ejemplos de Código y Explicaciones Técnicas

A continuación, se presentan fragmentos de código esenciales que ilustran la arquitectura y funcionamiento del sistema.

5.1. Modelo de Datos (reservas/models.py)

Este código define la estructura de la tabla de Reserva en la base de datos. Django utiliza este modelo para crear, leer y manipular los registros de reservas de forma automática.

```
from django.db import models

class Reserva(models.Model):
    """
    Modelo que representa una reserva en el restaurante.
    """
    nombre_cliente = models.CharField(max_length=100)
    correo_cliente = models.EmailField()
```

```
    telefono_cliente = models.CharField(max_length=15)
    fecha_reserva = models.DateField()
    hora_reserva = models.TimeField()
    cantidad_personas = models.PositiveIntegerField()
    estado = models.CharField(max_length=20,
                              choices=[('pendiente', 'Pendiente'),
                                       ('confirmada', 'Confirmada'),
                                       ('cancelada', 'Cancelada')],
                              default='pendiente')
    creado_en = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"Reserva de {self.nombre_cliente} para el {self.fecha_reserva}"
```

Explicación:

- Cada campo de la clase Reserva (ej. nombre_cliente, fecha_reserva) corresponde a una columna en la base de datos.
- Django provee tipos de campo específicos (CharField, EmailField, DateField) que validan los datos automáticamente.
- El campo estado utiliza choices para limitar los valores posibles, asegurando la integridad de los datos.

5.2. Plantilla Base (reservas/templates/reservas/base.html)

Esta es la plantilla principal de la cual heredarán todas las demás vistas. Define la estructura HTML, carga los archivos estáticos (CSS de Bootstrap y JS de AngularJS) e inicializa la aplicación de AngularJS.

```
<!DOCTYPE html>
<html lang="es" ng-app="SazonPeruanaApp">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sazón Peruana - {% block title %}{% endblock %}</title>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
    <!-- AngularJS -->
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
```

```
<body class="d-flex flex-column min-vh-100">
    <header>
        <!-- Barra de navegación -->
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
            <div class="container">
                <a class="navbar-brand" href="/">Sazón Peruana</a>
            </div>
        </nav>
    </header>

    <main class="container mt-4 flex-grow-1">
        {% block content %}
        <!-- El contenido de las vistas hijas se insertará aquí -->
        {% endblock %}
    </main>

    <footer class="bg-light text-center text-lg-start mt-auto">
        <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">
            © 2024 Sazón Peruana
        </div>
    </footer>

    <!-- Bootstrap JS -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></
script>
    <!-- App de AngularJS -->
    <script>
        var app = angular.module('SazonPeruanaApp', []);
    </script>
    {% block scripts %}{% endblock %}
</body>
</html>
```

Explicación:

- ng-app="SazonPeruanaApp": Esta directiva de AngularJS inicializa la aplicación en la etiqueta <html>.
- {% block content %}: Es un marcador de Django. Las plantillas que hereden de base.html pueden rellenar este bloque con su propio contenido.

- Se cargan las librerías de Bootstrap y AngularJS desde un CDN para un rendimiento óptimo.

5.3. Formulario de Reserva con AngularJS (reservas/templates/reservas/crear_reserva.html)

Este fragmento muestra cómo AngularJS se utiliza para crear un formulario interactivo. Los datos del formulario se vinculan a un modelo en el frontend (ng-model), permitiendo validaciones y manipulación de datos en tiempo real.

```
{% extends 'reservas/base.html' %}

{% block content %}
<div ng-controller="ReservaController">
    <h2>Crear Nueva Reserva</h2>
    <form name="reservaForm" ng-submit="submitForm()" novalidate>
        <div class="mb-3">
            <label for="nombre" class="form-label">Nombre Completo</label>
            <input type="text" class="form-control" id="nombre"
                   ng-model="reserva.nombre_cliente" required>
        </div>

        <div class="mb-3">
            <label for="fecha" class="form-label">Fecha de la Reserva</label>
            <input type="date" class="form-control" id="fecha"
                   ng-model="reserva.fecha_reserva" required>
        </div>

        <!-- Otros campos del formulario aquí (correo, teléfono, etc.) -->

        <button type="submit" class="btn btn-primary"
                ng-disabled="reservaForm.$invalid">
            Realizar Reserva
        </button>
    </form>
</div>
{% endblock %}
```

```
{% block scripts %}
<script>
    app.controller('ReservaController', function($scope, $http) {
        $scope.reserva = {}; // Objeto para almacenar los datos del formulario

        $scope.submitForm = function() {
            if ($scope.reservaForm.$valid) {
                // Enviar datos al backend de Django usando $http
                $http.post('/api/reservas/', $scope.reserva)
                    .then(function(response) {
                        alert('¡Reserva creada con éxito!');
                    }, function(error) {
                        alert('Error al crear la reserva.');
                    });
            }
        };
    });
</script>
{% endblock %}
```

Explicación:

- ng-controller="ReservaController": Asocia esta sección del HTML con un controlador de AngularJS.
- ng-model="reserva.nombre_cliente": Vincula el valor del campo de texto a la propiedad nombre_cliente del objeto $scope.reserva en el controlador.
- ng-disabled="reservaForm.$invalid": Deshabilita el botón de envío si el formulario no es válido, proporcionando una retroalimentación instantánea al usuario.
- El controlador (ReservaController) define la lógica para enviar los datos al backend de Django a través de una solicitud $http.post.