

# **PROYECTO INTEGRADOR**

## **3° Cuatrimestre**

Sazón Patrimonial

DevBox

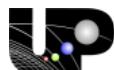
243691 Escobar Nuricumbo Heber Alexander

243683 Zárate Vázquez Diego Alberto

243679 Montoya Noble Eduardo

243738 Ruiz Pérez Brian Luis

27/07/2025



## Índice

<b>1.- Problemática.....</b>	<b>3</b>
<b>2.- Descripción del Proyecto.....</b>	<b>4</b>
<b>3.- Requerimientos.....</b>	<b>6</b>
3.1 Funcionalidades.....	6
3.2 Diagramas de Caso de Uso.....	7
3.3 Diagrama de Actividades.....	9
3.4 Mapa de Navegación.....	10
<b>4.- Diseño.....</b>	<b>11</b>
4.1 Diseño de la base de datos.....	11
4.2 Diseño de las interfaces (figma).....	17
<b>5.- Desarrollo.....</b>	<b>27</b>
5.1 Código API's.....	27
5.1.1 Repositorios.....	34
5.2 Código Fronted.....	128
5.2.1 HTML.....	128
5.2.2 CSS.....	182
5.2.3 JavaScript.....	341
<b>6.- Vistas Fronted.....</b>	<b>520</b>
6.1 Usuario.....	520
6.2 Restaurantero.....	522
6.3 Administrador.....	528

## 1.- Problemática

La ausencia de un sistema completo y eficiente para catalogar los restaurantes en San Cristóbal representa un reto importante para los turistas que visitan la localidad, como también para el propio negocio local. Debido a que, a día de hoy existe una clara falta de una plataforma centralizada que ofrezca información real y actualizada sobre los lugares de comida y bebida ubicados en el municipio de San Cristóbal, resulta difícil para los turistas localizar opciones reales y adecuadas, opción que indudablemente hace una mala contribución a la experiencia del turismo que realicen en la ciudad. De la misma manera, esta falta de un sistema eficiente hace que los propios establecimientos tengan una visibilidad muy baja, en especial aquellos de menor tamaño, que no tienen presupuesto para promocionarse en publicidad electrónica o en otros medios, por lo que la implementación de una facilidad como la propuesta de nuestro proyecto, no solo sería efectiva para mejorar la satisfacción del visitante, sino que además podría favorecer el crecimiento del sector, puesto que podría gestionarse para conectar más consumidores de ese tipo de establecimiento.

## 2.- Descripción del Proyecto

Sazón Patrimonial es una plataforma digital diseñada para promover y preservar la riqueza gastronómica de San Cristóbal de Las Casas, Chiapas. Su objetivo es conectar a restaurantes locales (desde pequeños negocios hasta establecimientos consolidados) con comensales locales, nacionales y extranjeros, facilitando la difusión de la cocina tradicional y apoyando la economía de la región.

### 3.- Requerimientos

#### 3.1 Funcionalidades

##### Para restauranteros:

- Gestión de Perfil
- Edición completa de información (horarios, dirección, descripción).
- Carga de menús digitales en PDF.
- Subida de hasta 3 imágenes (1 destacada + 2 complementarias).
- Selección de etiquetas (ej: "comida mexicana", "pet-friendly").
- Estadísticas sobre el local.
- Reportes de descargas de menús (semanal/acumulado).
- Datos demográficos de clientes (locales y extranjeros).
- Validación de Negocio.
- Sistema de verificación con comprobante de domicilio.
- Personalización.
- Actualización en tiempo real de cambios.

##### Para Usuarios/Comensales:

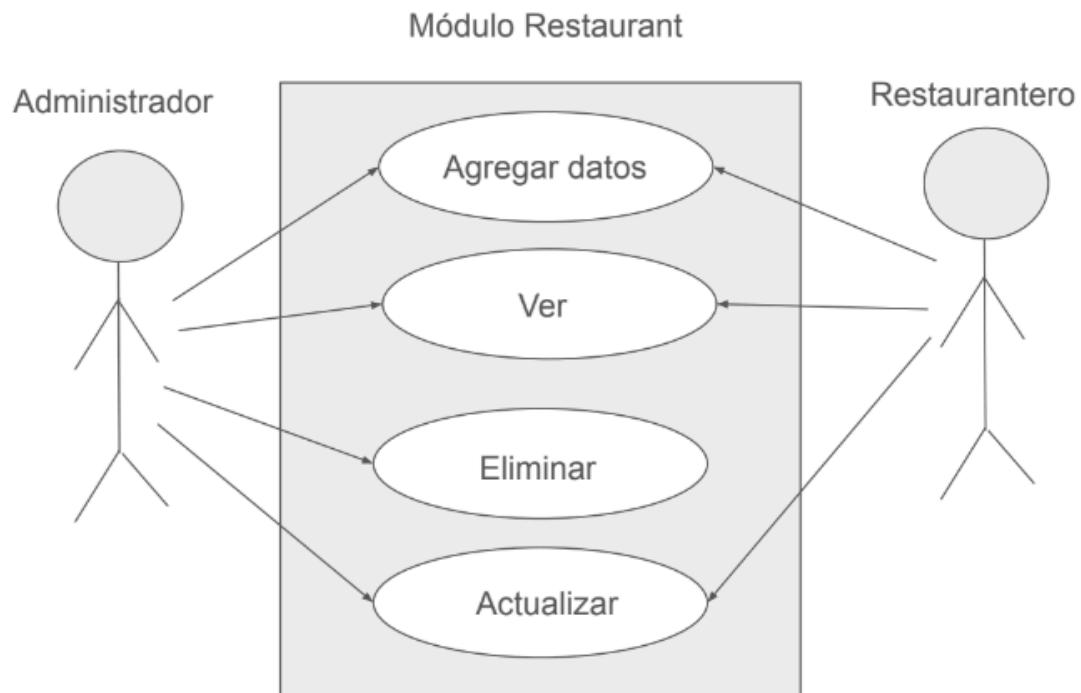
Búsqueda por filtros por ejemplo:

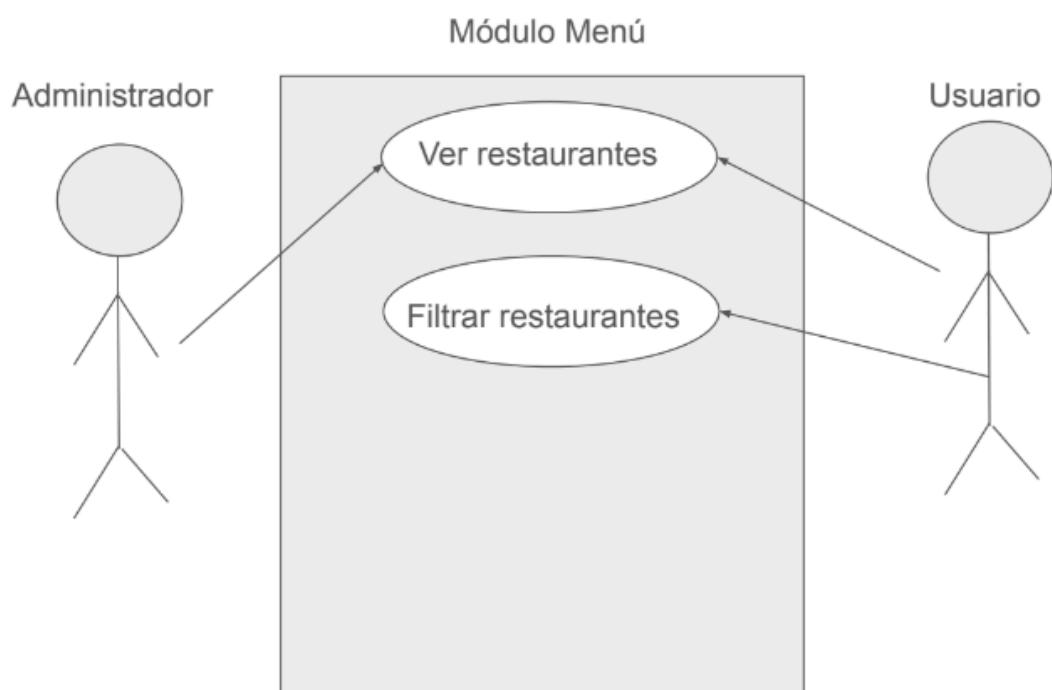
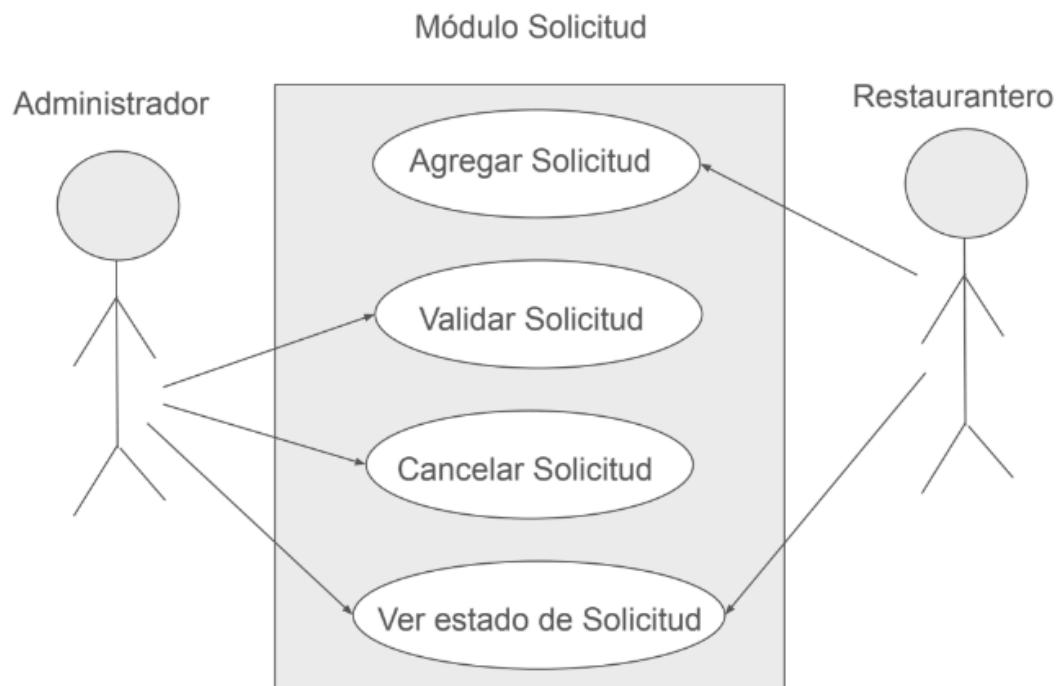
- Tipo de comida (mexicana, internacional, etc.).
  - Ambiente (terraza, familiar, etc.).
  - Servicios (wifi, estacionamiento).
- Visualización de Perfiles de los Restaurantes.
- Información detallada: contacto, redes sociales.
- Galería de imágenes y menú descargable.
- Indicador de horario del local.
- Responder la encuesta antes de descargar el menú.

##### Para Administradores

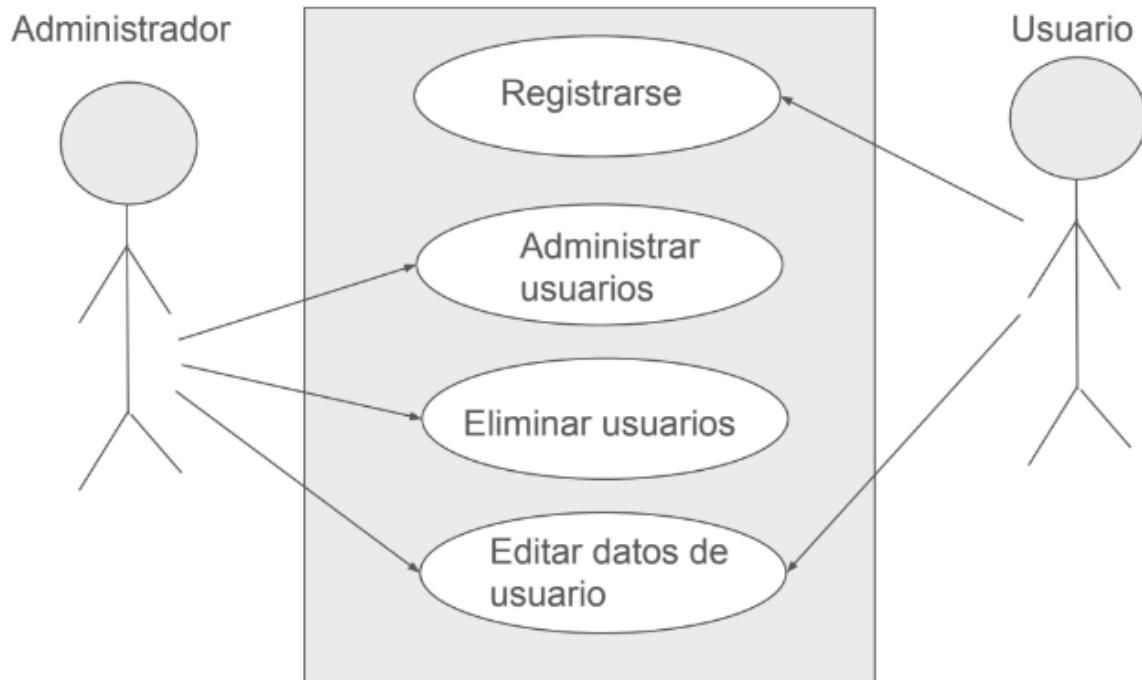
- Moderación
- Validación de perfiles de restaurantes.
- Soporte
- Actualización de etiquetas.

### 3.2 Diagramas de Caso de Uso

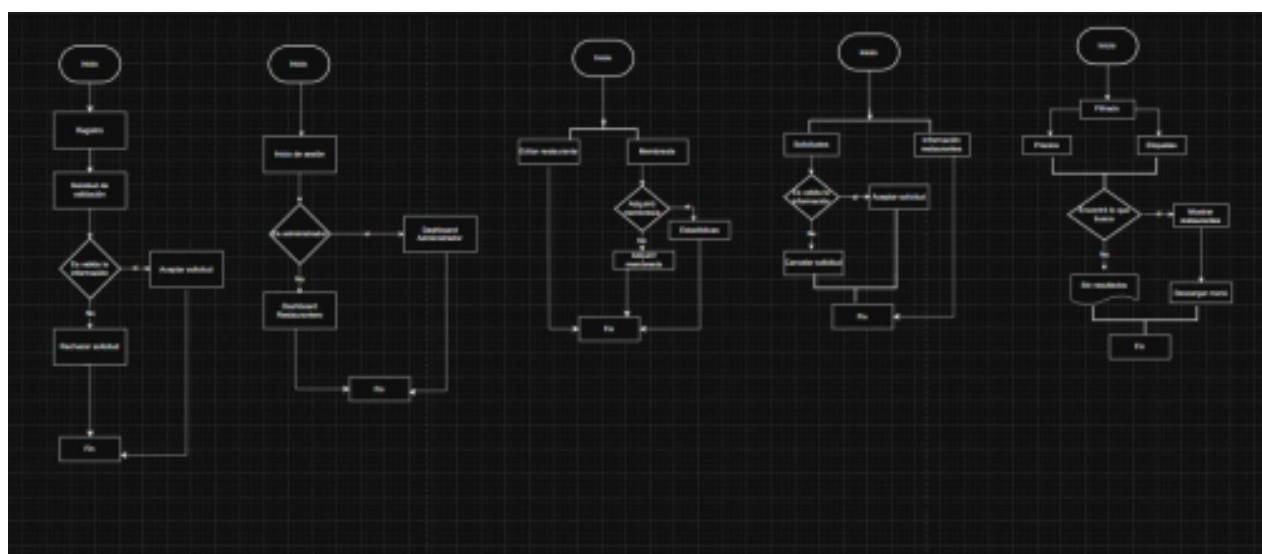




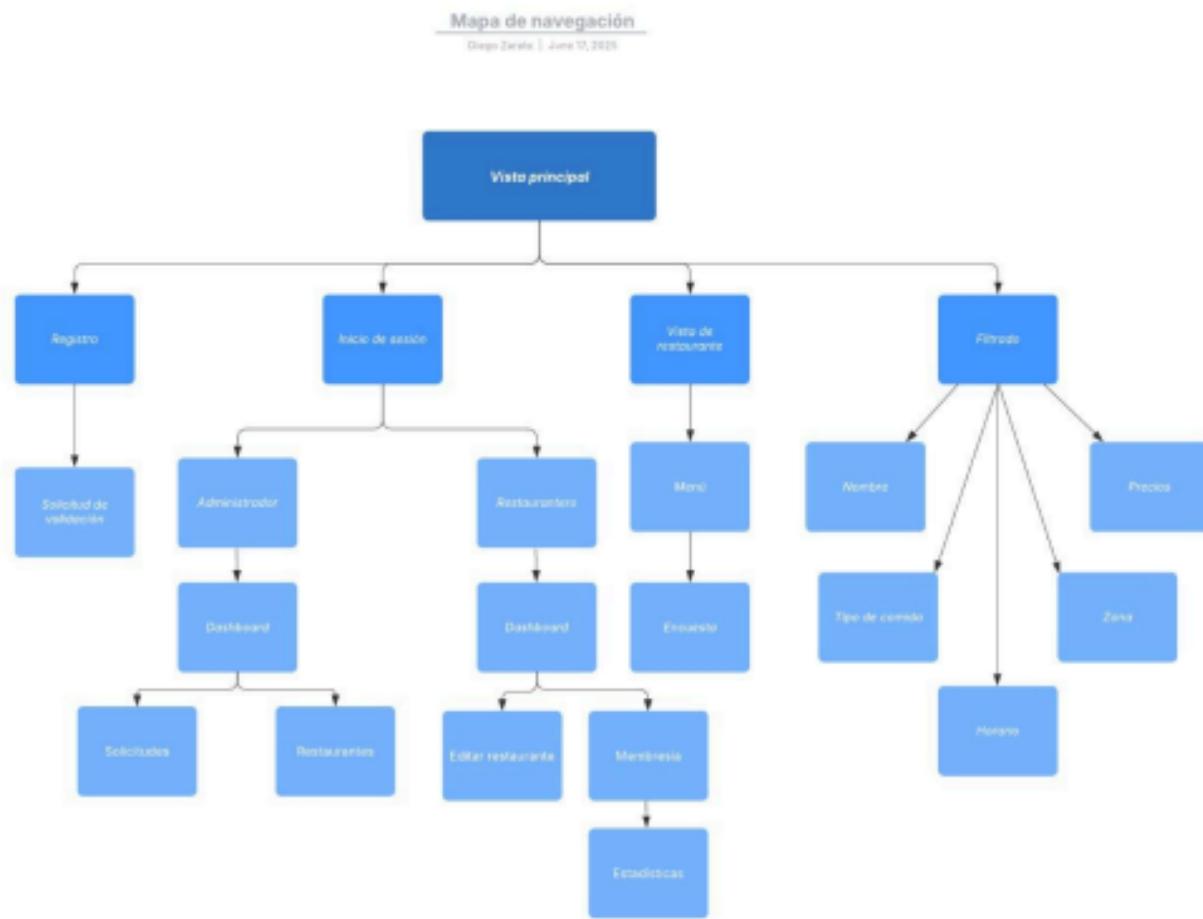
### Módulo Usuarios



### 3.3 Diagrama de Actividades.



### 3.4 Mapa de Navegación.



## 4.- Diseño

### 4.1 Diseño de la base de datos

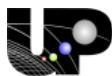
Tabla de Usuario

usuario	
!	id_usuario INT
◆	nombre VARCHAR(45)
◆	correo VARCHAR(45)
◆	contrasena VARCHAR(45)
◆	tipo VARCHAR(45)
Indexes	

Tabla administrador

administrador	
!	codigo_usuario INT
Indexes	

Tabla solicitud de registro



**solicitud\_registro**

▼

- 💡 **id\_solicitud** INT
- 🕒 **fecha** DATETIME(40)
- 🕒 **estado** VARCHAR(45)
- 🕒 **restaurante** VARCHAR(45)
- 🕒 **correo** VARCHAR(45)
- 🕒 **propietario** VARCHAR(45)
- 🕒 **horario** VARCHAR(45)
- 🕒 **numero** VARCHAR(45)
- 🕒 **facebook** VARCHAR(45)
- 🕒 **instagram** VARCHAR(45)
- 🕒 **direccion** VARCHAR(45)
- 🕒 **imagen1** VARCHAR(200)
- 🕒 **imagen2** VARCHAR(200)
- 🕒 **imagen3** VARCHAR(200)
- 🕒 **comprobante** VARCHAR(200)
- 🕒 **menu** VARCHAR(200)

Indexes ►

## Tabla Estadística

estadistica	
id_descarga	INT
nacional	INT
extranjero	INT
correo	INT
descargas	INT
comida	INT
ubicacion	INT
recomendacion	INT
horario	INT
vista	INT

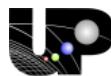
## Tabla Menú

menu	
id_menu	INT
ruta_archivo	VARCHAR(200)
telefono	VARCHAR(45)

## 4.2 Diseño de las interfaces (figma)

### 1.- Vista Principal





**La Mansión**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**Tarumba**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**Taniperla**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**El secreto**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00

**Contáctanos**

M sazonpatrimonial@gmail.com

+52 961 652 2093

@sazonpatrimonial

@sazonpatrimonial



**Universidad Politécnica de Chiapas**

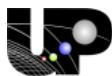
Permite al usuario buscar restaurantes mediante un buscador, aplicar filtros personalizados o seleccionar directamente entre las opciones mostradas en la vista principal; además, ofrece la posibilidad de registrarse (como restaurantero administrador) o iniciar sesión si ya dispone de una cuenta.

## 2.-Registro de Usuario

The screenshot shows a modal window titled "Registro de usuario" (User Registration) overlaid on a background image of a colorful street in San Cristóbal de las Casas. The modal contains fields for "Nombre de usuario:" (Username), "Correo electrónico:" (Email), "Contraseña:" (Password), and "Confirmar contraseña:" (Confirm Password). There are also "Buscar" (Search) and "Registrarse" (Register) buttons. The top right of the modal has links for "Registrarse" and "Iniciar sesión" (Login).

Permite al restaurantero registrarse.

## 3.- Vista de solicitud de validación del restaurante



### Solicitud de validación

Proporciona la siguiente información para verificar tu restaurante

Restaurante

Ingrese el nombre del restaurante

Propietario

Ingrese el nombre del propietario

Correo electrónico

Ingrese su correo

Número de celular

Ingrese su número de celular

Facebook

Ingrese su Facebook

Instagram

Ingrese su Instagram

Dirección

Ingrese la dirección del restaurante

Horarios de cierre y apertura

Ingrese el horario

Imágenes (JPG o PNG)

Imagen principal

Imagen secundaria

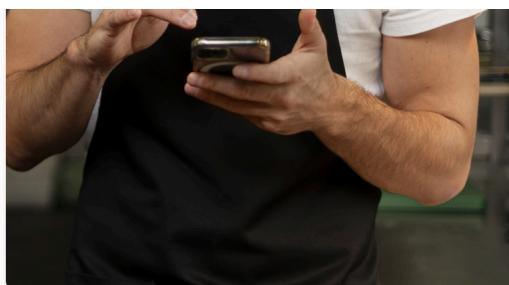
Imagen de platillo

Archivos (PDF)

Comprobante de domicilio

Menú del restaurante

Enviar solicitud



#### Contáctanos

sazonpatrimonial@gmail.com

+52 961 652 2093

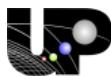
@sazonpatrimonial

@sazonpatrimonial



Permite que el restaurantero valide la existencia de su negocio proporcionando información personal (número de celular y dirección del establecimiento), adjuntando tres imágenes del local en formato PNG o JPG, y presentando un comprobante de domicilio en PDF como parte del proceso de verificación.

#### 4.- Vista de restaurante



## La mansión



Libre de gluten

Libre de gluten

Comida local

Pet-friendly

### Horarios

#### MORARIOS

Lunes a viernes

8:00-20:00

Sábado y domingo

10:00-20:00

### Contactos

#### CONTACTOS

+52 123 467 2543

@LaMansion

@la\_mansion

### Ubicación

#### UBICACION

Av. 4 de junio N°20,  
San Cristóbal de las  
Casas, Chiapas,  
México.

Menú

#### Contáctanos

sazonpatrimonial@gmail.com

+52 961 652 2093

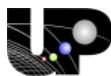
@sazonpatrimonial

@sazonpatrimonial



Permite al usuario visualizar los detalles del restaurante seleccionado, incluyendo fotos, horarios, información de contacto, ubicación, etiquetas descriptivas y la opción de descargar el menú proporcionado por el establecimiento.

## 5.- Encuesta



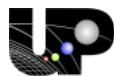
The screenshot shows a restaurant's interior with tables and chairs. A white survey overlay is centered over the image. The overlay has a logo for "SAZÓN PATRIMONIAL" at the top. Below it, the word "Encuesta" is displayed. Two questions are listed: "¿Qué te atrajo del restaurante?" and "¿De dónde nos visitas?", each with a "Selecciona una opción" button. At the bottom of the overlay is a red "Enviar" button.

Pequeña encuesta que aparece al usuario cuando quiere descargar el menú del restaurante, esto nos sirve a los administradores para poder dar las estadísticas al restaurantero.

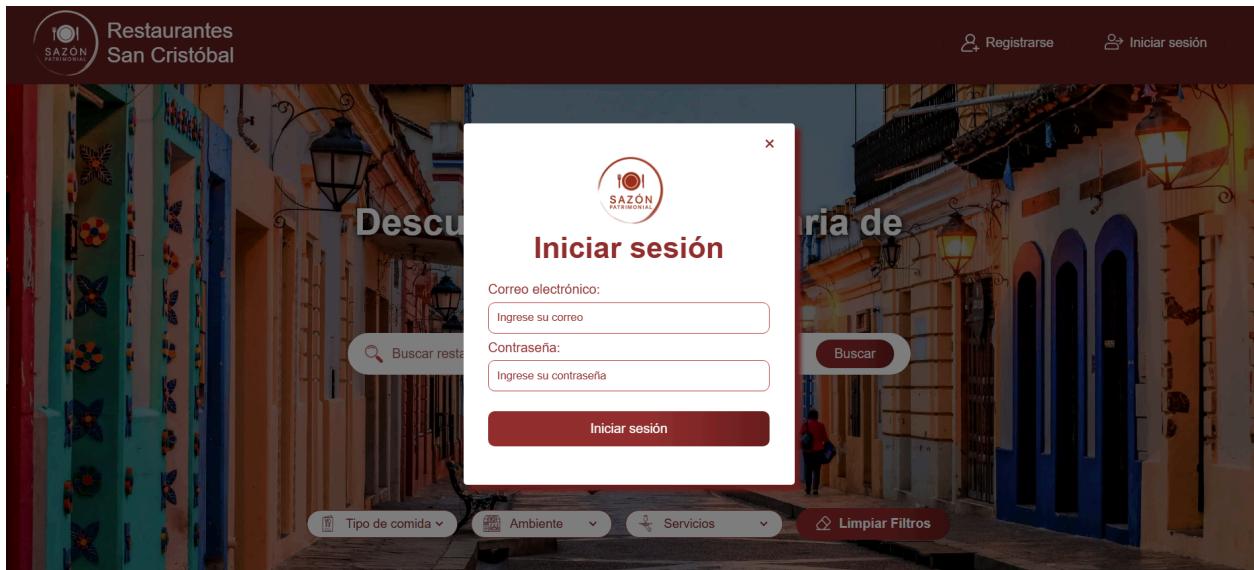
## 6.- Opción de Login

The screenshot shows a street view of colorful buildings in the background. In the foreground, a login modal is open. The modal has a logo for "SAZÓN PATRIMONIAL" at the top. Below it, the text "Ingresar como" is displayed. There are two main buttons: a dark red "Administrador" button with a gear icon and a white "Buscar" button with a magnifying glass icon. At the bottom of the modal is a "Registrarse" link. The top right corner of the modal has a close button (X).

Los usuarios pueden ingresar como Administrador (acceso exclusivo para personal autorizado), Restaurantero , o Registrarse (para crear una nueva cuenta como restaurantero).



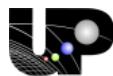
## 7.- Iniciar sesión



Deja iniciar sesión tanto al restaurante como al administrador.

## 8.- Vista principal del restaurantero





**La Mansión**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**Tarumba**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**Taniperla**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**El secreto**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00

**Contáctanos**

M sazonpatrimonial@gmail.com

+52 961 652 2093

@sazonpatrimonial

@sazonpatrimonial



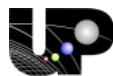
Universidad Politécnica  
de Chiapas



Vista principal cuando el restaurantero inicia sesión, muy similar a la vista principal, pero con el mínimo detalle que ahora tiene las opciones de "Mi Restaurante" o Cerrar sesión.

## 9.- Vista de edición del restaurante

The screenshot shows the main dashboard of the system. At the top, there's a dark header bar with the logo 'SAZÓN PATRIMONIAL' on the left, the text 'Restaurantes San Cristóbal' in the center, and 'Cerrar sesión' (Logout) and a user icon on the right. Below the header, the word 'Bienvenido' (Welcome) is prominently displayed in a large, red, serif font. To the left of 'Bienvenido' is a small red circular arrow icon. The main content area is divided into several sections: a top section with a red button labeled 'Estadísticas' (Statistics), a 'Galería de Imágenes' (Image Gallery) section containing three empty boxes, and a 'Menú en PDF:' (PDF Menu) section with a file input field labeled 'Seleccionar archivo PDF' (Select PDF file). The overall design is clean and modern, using a color palette of red, white, and grey.



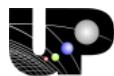
<b>Ubicación:</b> <input type="text" value="Ingrese la dirección"/>	<b>Horarios:</b> <input type="text" value="Ingrese el horario"/>												
<b>Contactos:</b> <table border="1"><tr><td></td><td><input type="text" value="Ingrese su número celular"/></td></tr><tr><td></td><td><input type="text" value="Ingrese su facebook"/></td></tr><tr><td></td><td><input type="text" value="Ingrese su instagram"/></td></tr></table>		<input type="text" value="Ingrese su número celular"/>		<input type="text" value="Ingrese su facebook"/>		<input type="text" value="Ingrese su instagram"/>	<b>Etiquetas:</b> <table border="1"><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>						
	<input type="text" value="Ingrese su número celular"/>												
	<input type="text" value="Ingrese su facebook"/>												
	<input type="text" value="Ingrese su instagram"/>												
<b>Aplicar Cambios</b>													

Esta interfaz facilita la gestión completa del perfil del restaurante, desde horarios y ubicación hasta redes sociales, etiquetas y poder visualizar las estadísticas del local

## 10.- Estadísticas del restaurante



Este panel le ayuda al restaurantero visualizar datos estratégicos como descargas de menú, interés de clientes por origen y aspectos destacados, ayudando al restaurantero a medir su desempeño.



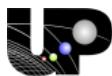
## 11.- Vistas de administrador

The screenshot shows a user interface for managing restaurants. At the top, there's a header bar with the logo 'SAZÓN INTERNACIONAL', the text 'Restaurantes San Cristóbal', and a 'Cerrar sesión' button. On the left, a sidebar has three items: 'Restaurantes' (selected), 'Etiquetas', and 'Usuarios'. The main content area is titled 'Administrar Restaurantes' and contains a table with one row. The table columns are 'Nombre', 'Ubicación', 'Etiquetas', and 'Acciones'. The data row shows 'Nombre 1', 'Ubicación 1', 'Etiqueta 1', and edit (pencil) and delete (trash bin) icons.

Nombre	Ubicación	Etiquetas	Acciones
Nombre 1	Ubicación 1	Etiqueta 1	

The screenshot shows a user interface for managing tags. The layout is similar to the previous one, with a header bar, a sidebar with 'Etiquetas' selected, and a main content area titled 'Administrar Etiquetas'. The main content table has one row with columns 'Nombre' and 'Acciones'. The data row shows 'Etiqueta 1' and edit (pencil) and delete (trash bin) icons. There's also a red button labeled '+ Agregar etiquetas'.

Nombre	Acciones
Etiqueta 1	



Restaurantes  
San Cristóbal

Cerrar sesión

Restaurantes

Etiquetas

Usuarios

### Administrar Solicitudes de Usuarios

Restaurante	Propietario	Correo	Celular	Dirección	Horario	Imágenes	Comprobante	Acciones
Restaurante 1	Propietario 1	correo1@ejemplo.com	1234567890	Dirección 1	Horario 1			

Este panel proporciona información a los administradores sobre la administración de restaurantes, etiquetas y usuarios.

## 5.- Desarrollo

### 5.1 Código API's

```
const API_CONFIG = {  
  
  BASE_URL: 'http://52.23.26.163:7070',  
  
  ENDPOINTS: {  
  
    RESTAURANTEROS: '/api/restauranteros',  
  
    USUARIOS: '/api/usuarios',  
  
    ADMIN: '/api/admin'  
  
  }  
  
};  
  
  
  
class SazonAPI {  
  
  static async request(endpoint, options = {}) {  
  
    const url = `${API_CONFIG.BASE_URL}${endpoint}`;  
  
    const defaultOptions = {  
  
      headers: {  
  
        'Content-Type': 'application/json',  
  
      },  
  
    };  
  
  
  
    try {  
  
      const response = await fetch(url, { ...defaultOptions, ...options });  
  
    } catch (error) {  
  
      console.error(`Error en la solicitud a ${url}: ${error.message}`);  
  
    }  
  
  }  
  
};
```

```
if (!response.ok) {  
  
    throw new Error(`HTTP ${response.status}: ${response.statusText}`);  
  
}  
  
  
  
const contentType = response.headers.get('content-type');  
  
if (contentType && contentType.includes('application/json')) {  
  
    return await response.json();  
  
} else {  
  
    return await response.text();  
  
}  
  
} catch (error) {  
  
    console.error('API Error:', error);  
  
    throw error;  
  
}  
  
}  
  
  
  
static async getAllRestauranteros() {  
  
    return this.request(API_CONFIG.ENDPOINTS.RESTAURANTEROS);  
  
}  
  
  
  
static async getRestauranteroById(id) {  
  
    return this.request(`${API_CONFIG.ENDPOINTS.RESTAURANTEROS}/${id}`);  
  
}
```

```
static async createRestaurantero(data) {  
  
    return this.request(API_CONFIG.ENDPOINTS.RESTAURANTEROS, {  
  
        method: 'POST',  
  
        body: JSON.stringify(data)  
  
    });  
  
}  
  
  
  
static async updateRestaurantero(id, data) {  
  
    return this.request(` ${API_CONFIG.ENDPOINTS.RESTAURANTEROS}/${id}` , {  
  
        method: 'PUT',  
  
        body: JSON.stringify(data)  
  
    });  
  
}  
  
  
  
static async deleteRestaurantero(id) {  
  
    return this.request(` ${API_CONFIG.ENDPOINTS.RESTAURANTEROS}/${id}` , {  
  
        method: 'DELETE'  
  
    });  
  
}  
  
  
  
static async getAllUsuarios() {  
  
    return this.request(API_CONFIG.ENDPOINTS.USUARIOS);
```

}

```
static async createUsuario(data) {  
  
    return this.request(API_CONFIG.ENDPOINTS.USUARIOS, {  
  
        method: 'POST',  
  
        body: JSON.stringify(data)  
  
    });  
  
}
```

```
static async loginUsuario(credentials) {  
  
    return this.request(` ${API_CONFIG.ENDPOINTS.USUARIOS}/login` , {  
  
        method: 'POST',  
  
        body: JSON.stringify(credentials)  
  
    });  
  
}
```

```
static async getAllSolicitudes() {  
  
    return this.request('/solicitudes');  
  
}  
  
}
```

```
class UIUtils {  
  
    static showMessage(message, type = 'info') {
```

```
const toast = document.createElement('div');

toast.className = `toast toast-${type}`;

toast.textContent = message;
```

```
Object.assign(toast.style, {
```

```
    position: 'fixed',
    top: '20px',
    right: '20px',
    padding: '15px 20px',
    borderRadius: '5px',
    color: 'white',
    zIndex: '9999',
    fontWeight: 'bold',
```

```
    backgroundColor: type === 'error' ? '#e74c3c' : type === 'success' ? '#27ae60' :
    '#3498db'
```

```
});
```

```
document.body.appendChild(toast);
```

```
setTimeout(() => {
    if (toast.parentNode) {
        toast.parentNode.removeChild(toast);
    }
}
```

}, 3000);

}

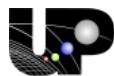
```
static showError(message) {  
    this.showMessage(message, 'error');  
}  
}
```

```
static showSuccess(message) {  
    this.showMessage(message, 'success');  
}  
}
```

```
static showLoading(show = true) {  
    let loader = document.getElementById('global-loader');
```

```
    if (show && !loader) {  
        loader = document.createElement('div');  
        loader.id = 'global-loader';  
        loader.innerHTML = '<div class="spinner"></div>';  
    }
```

```
    Object.assign(loader.style, {  
        position: 'fixed',  
        top: '0',  
        left: '0',  
        width: '100%',  
        height: '100%',  
        overflow: 'hidden',  
        display: 'flex',  
        alignItems: 'center',  
        justifyContent: 'center'  
    });  
}  
}
```



```
width: '100%',  
  
height: '100%',  
  
backgroundColor: 'rgba(0,0,0,0.5)',  
  
display: 'flex',  
  
justifyContent: 'center',  
  
alignItems: 'center',  
  
zIndex: '9998'  
  
});
```

```
const spinner = loader.querySelector('.spinner');  
  
Object.assign(spinner.style, {  
  
width: '50px',  
  
height: '50px',  
  
border: '5px solid #f3f3f3',  
  
borderTop: '5px solid #3498db',  
  
borderRadius: '50%',  
  
animation: 'spin 1s linear infinite'  
  
});
```

```
document.body.appendChild(loader);  
  
} else if (!show && loader) {  
  
loader.remove();  
  
}
```

}

}

```
const style = document.createElement('style');
style.textContent = `
@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}
`;
document.head.appendChild(style);
```

### 5.1.1 Repositorios

#### Admin Repositorio

```
package org.sazonpt.repository;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
```

```
import org.sazonpt.config.DBConfig;  
  
import org.sazonpt.model.Administrador;  
  
  
public class AdminRepository {  
  
    public void createAdmin(Administrador admin) throws SQLException {  
  
        String query = "INSERT INTO administrador(id_usuario) VALUES(?);  
  
        try(Connection conn = DBConfig.getDataSource().getConnection();  
  
            PreparedStatement stmt = conn.prepareStatement(query)){  
  
            stmt.setInt(1, admin.getId_usuario());  
  
            stmt.executeUpdate();  
  
        } catch(SQLException e){  
  
            throw new SQLException("Error al crear el administrador: "+e.getMessage());  
  
        }  
  
    }  
  
    public Administrador findAdminById(int idAdmin) throws SQLException {  
  
        String query = "SELECT u.*, a.id_usuario as admin_id_usuario FROM usuario u " +  
                    "INNER JOIN administrador a ON u.id_usuario = a.id_usuario " +  
                    "WHERE u.id_usuario = ? AND u.status = 'activo"';  
  
        Administrador admin = null;  
  
  
        try (Connection conn = DBConfig.getDataSource().getConnection();  
             PreparedStatement stmt = conn.prepareStatement(query)) {  
            ...  
        }  
    }  
}
```

```
stmt.setInt(1, idAdmin);

try (ResultSet rs = stmt.executeQuery()) {

    if (rs.next()) {

        admin = mapResultSetToAdmin(rs);

    }

}

} catch (SQLException e) {

    throw new SQLException("Error al buscar el administrador: " + e.getMessage());

}

return admin;

}

public List<Administrador> getAllAdmins() throws SQLException {

    List<Administrador> admins = new ArrayList<>();

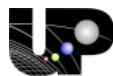
    String query = "SELECT u.*, a.id_usuario as admin_id_usuario FROM usuario u " +

        "INNER JOIN administrador a ON u.id_usuario = a.id_usuario " +

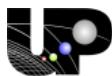
        "WHERE u.status = 'activo'";

    try (Connection conn = DBConfig.getDataSource().getConnection();

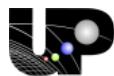
        PreparedStatement stmt = conn.prepareStatement(query);
```



```
ResultSet rs = stmt.executeQuery() {  
  
    while (rs.next()) {  
  
        admins.add(mapResultSetToAdmin(rs));  
  
    }  
  
} catch (SQLException e) {  
  
    throw new SQLException("Error al obtener los administradores: " +  
e.getMessage());  
  
}  
  
  
return admins;  
}  
  
  
public void updateAdmin(Administrador admin) throws SQLException {  
  
    String query = "UPDATE usuario SET nombre = ?, correo = ?, contrasena = ?,  
status = ? WHERE id_usuario = ?";  
  
  
  
  
try (Connection conn = DBConfig.getDataSource().getConnection();  
  
PreparedStatement stmt = conn.prepareStatement(query)) {  
  
    stmt.setString(1, admin.getNombre());  
  
    stmt.setString(2, admin.getCorreo());  
  
    stmt.setString(3, admin.getContrasena());  
}
```



```
stmt.setString(4, admin.getStatus());  
  
stmt.setInt(5, admin.getId_usuario());  
  
  
stmt.executeUpdate();  
  
} catch (SQLException e) {  
  
    throw new SQLException("Error al actualizar el administrador: " +  
e.getMessage());  
  
}  
  
}  
  
  
  
public void deleteAdmin(int idAdmin) throws SQLException {  
  
Connection conn = null;  
  
try {  
  
conn = DBConfig.getDataSource().getConnection();  
  
conn.setAutoCommit(false);  
  
  
  
String queryAdmin = "DELETE FROM administrador WHERE id_usuario = ?";  
  
PreparedStatement stmtAdmin = conn.prepareStatement(queryAdmin);  
  
stmtAdmin.setInt(1, idAdmin);  
  
int rowsAffectedAdmin = stmtAdmin.executeUpdate();  
  
  
  
System.out.println("Query executed: " + queryAdmin);  
  
System.out.println("Admin rows affected: " + rowsAffectedAdmin);
```



```
String queryUsuario = "DELETE FROM usuario WHERE id_usuario = ?";  
PreparedStatement stmtUsuario = conn.prepareStatement(queryUsuario);  
stmtUsuario.setInt(1, idAdmin);  
int rowsAffectedUsuario = stmtUsuario.executeUpdate();  
  
System.out.println("Query executed: " + queryUsuario);  
System.out.println("Admin ID: " + idAdmin);  
System.out.println("Usuario rows affected: " + rowsAffectedUsuario);  
  
if (rowsAffectedUsuario == 0) {  
    throw new SQLException("No se encontró el administrador con ID: " +  
idAdmin);  
}  
  
conn.commit();  
  
} catch (SQLException e) {  
    if (conn != null) {  
        conn.rollback();  
    }  
    throw new SQLException("Error al eliminar el administrador: " +  
e.getMessage());  
}
```

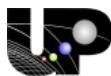
```
    } finally {  
  
        if (conn != null) {  
  
            conn.setAutoCommit(true);  
  
            conn.close();  
  
        }  
  
    }  
  
}
```

```
private Administrador mapResultSetToAdmin(ResultSet rs) throws SQLException {  
  
    return new Administrador(  
  
        rs.getInt("id_usuario"),  
  
        rs.getString("nombre"),  
  
        rs.getString("correo"),  
  
        rs.getString("contrasena"),  
  
        rs.getString("tipo"),  
  
        rs.getInt("admin_id_usuario"),  
  
        rs.getString("status")  
  
    );  
  
}  
  
}
```

## Estadística Repositorio

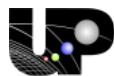
```
package org.sazonpt.repository;
```

```
import java.sql.Connection;  
  
import java.sql.PreparedStatement;  
  
import java.sql.ResultSet;  
  
import java.sql.SQLException;  
  
import java.util.ArrayList;  
  
import java.util.List;  
  
  
import org.sazonpt.config.DBConfig;  
  
import org.sazonpt.model.Estadistica;  
  
  
public class EstadisticaRepository {  
  
  
  
  
  
  
    public void addEstadistica(Estadistica e) throws SQLException {  
  
        String query = "INSERT INTO estadistica(nacional, extranjero, correo, descargas,  
comida, ubicacion, recomendacion, horario, vista) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";  
  
        try (Connection conn = DBConfig.getDataSource().getConnection());  
  
            PreparedStatement stmt = conn.prepareStatement(query)) {  
  
                stmt.setObject(1, e.getNacional());  
  
                stmt.setObject(2, e.getExtranjero());  
  
                stmt.setString(3, e.getCorreo());  
  
                stmt.setInt(4, e.getDescargas());  
  
                stmt.setInt(5, e.getComida());  
            }  
        }  
    }  
}
```



```
stmt.setInt(6, e.getUbicacion());  
  
stmt.setInt(7, e.getRecomendacion());  
  
stmt.setInt(8, e.getHorario());  
  
stmt.setInt(9, e.getVista());  
  
stmt.executeUpdate();  
  
} catch (SQLException ex) {  
  
    throw new SQLException("Error al crear la estadística: " + ex.getMessage());  
  
}  
  
}
```

```
public Estadistica findEstadistica(int id) throws SQLException {  
  
    String query = "SELECT * FROM estadistica WHERE id_estadistica = ?";  
  
    Estadistica estadistica = null;  
  
    try (Connection conn = DBConfig.getDataSource().getConnection();  
  
         PreparedStatement stmt = conn.prepareStatement(query)) {  
  
        stmt.setInt(1, id);  
  
        try (ResultSet rs = stmt.executeQuery()) {  
  
            if (rs.next()) {  
  
                estadistica = mapResultSetToEstadistica(rs);  
  
            }  
  
        }  
  
    } catch (SQLException ex) {
```



```
        throw new SQLException("Error al buscar la estadística: " + ex.getMessage());  
    }  
  
    return estadistica;  
}
```

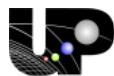
```
public boolean deleteEstadistica(int id) throws SQLException {  
  
    String query = "DELETE FROM estadistica WHERE id_estadistica = ?";  
  
    try (Connection conn = DBConfig.getDataSource().getConnection();  
  
         PreparedStatement stmt = conn.prepareStatement(query)) {  
  
        stmt.setInt(1, id);  
  
        int rowsAffected = stmt.executeUpdate();  
  
        return rowsAffected > 0;  
    } catch (SQLException ex) {  
  
        throw new SQLException("Error al eliminar la estadística: " + ex.getMessage());  
    }  
}
```

```
public Estadistica updateEstadistica(Estadistica e) throws SQLException {  
  
    String query = "UPDATE estadistica SET nacional = ?, extranjero = ?, correo = ?,  
descargas = ?, comida = ?, ubicacion = ?, recomendacion = ?, horario = ?, vista = ?  
WHERE id_estadistica = ?";
```

```
try (Connection conn = DBConfig.getConnection().getDataSource().getConnection()) {  
    PreparedStatement stmt = conn.prepareStatement(query) {  
        stmt.setObject(1, e.getNacional());  
        stmt.setObject(2, e.getExtranjero());  
        stmt.setString(3, e.getCorreo());  
        stmt.setInt(4, e.getDescargas());  
        stmt.setInt(5, e.getComida());  
        stmt.setInt(6, e.getUbicacion());  
        stmt.setInt(7, e.getRecomendacion());  
        stmt.setInt(8, e.getHorario());  
        stmt.setInt(9, e.getVista());  
        stmt.setInt(10, e.getId_estadistica());  
        int rowsAffected = stmt.executeUpdate();  
        if (rowsAffected > 0) {  
            return findEstadistica(e.getId_estadistica());  
        } else {  
            throw new SQLException("No se encontró la estadística con ID: " +  
e.getId_estadistica());  
        }  
    } catch (SQLException ex) {  
        throw new SQLException("Error al actualizar la estadística: " +  
ex.getMessage());  
    }  
}
```

}

```
public List<Estadistica> listAllEstadisticas() throws SQLException {  
  
    List<Estadistica> estadisticas = new ArrayList<>();  
  
    String query = "SELECT * FROM estadistica";  
  
    try (Connection conn = DBConfig.getDataSource().getConnection();  
  
        PreparedStatement stmt = conn.prepareStatement(query);  
  
        ResultSet rs = stmt.executeQuery()) {  
  
        while (rs.next()) {  
  
            estadisticas.add(mapResultSetToEstadistica(rs));  
  
        }  
  
    } catch (SQLException ex) {  
  
        throw new SQLException("Error al listar las estadísticas: " + ex.getMessage());  
  
    }  
  
    return estadisticas;  
  
}  
  
  
private Estadistica mapResultSetToEstadistica(ResultSet rs) throws SQLException {  
  
    return new Estadistica(  
  
        rs.getInt("id_estadistica"),  
  
        (Integer) rs.getObject("nacional"),  
  
        (Integer) rs.getObject("extranjero"),  
  
        ...  
    );  
}
```

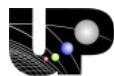


```
        rs.getString("correo"),  
        rs.getInt("descargas"),  
        rs.getInt("comida"),  
        rs.getInt("ubicacion"),  
        rs.getInt("recomendacion"),  
        rs.getInt("horario"),  
        rs.getInt("vista")  
    );  
}  
}
```

## Menú Repositorio

```
package org.sazonpt.repository;  
  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.List;  
  
  
import org.sazonpt.config.DBConfig;  
import org.sazonpt.model.Menu;
```

```
public class MenuRepository {  
  
    public void save(Menu menu) throws SQLException {  
  
        String query = "INSERT INTO menu(telefono, ruta_archivo) VALUES(?, ?)";  
  
        try (Connection conn = DBConfig.getDataSource().getConnection();  
             PreparedStatement stmt = conn.prepareStatement(query)) {  
  
            stmt.setString(1, menu.getTelefono());  
            stmt.setString(2, menu.getRuta_archivo());  
            stmt.executeUpdate();  
  
        } catch (SQLException e) {  
            throw new SQLException("Error al crear el menú: " + e.getMessage());  
        }  
    }  
  
    public Menu findById(int id) throws SQLException {  
  
        String query = "SELECT * FROM menu WHERE id_menu = ?";  
        Menu menu = null;  
  
        try (Connection conn = DBConfig.getDataSource().getConnection();  
             PreparedStatement stmt = conn.prepareStatement(query)) {  
    }
```



```
stmt.setInt(1, id);

try (ResultSet rs = stmt.executeQuery()) {

    if (rs.next()) {

        menu = mapResultSetToMenu(rs);

    }

}

} catch (SQLException e) {

    throw new SQLException("Error al buscar el menú: " + e.getMessage());

}

return menu;

}

public List<Menu> findAll() throws SQLException {

    List<Menu> menus = new ArrayList<>();

    String query = "SELECT * FROM menu";

    try (Connection conn = DBConfig.getDataSource().getConnection();

        PreparedStatement stmt = conn.prepareStatement(query);

        ResultSet rs = stmt.executeQuery()) {
```

```
        while (rs.next()) {  
  
            menus.add(mapResultSetToMenu(rs));  
  
        }  
  
    } catch (SQLException e) {  
  
        throw new SQLException("Error al listar los menus: " + e.getMessage());  
  
    }  
  
    return menus;  
}  
  
  
public List<Menu> findByRestaurant(int codigoRestaurante) throws SQLException {  
  
    List<Menu> menus = new ArrayList<>();  
  
    String query = "SELECT * FROM menu WHERE id_restaurante = ?";  
  
  
    try (Connection conn = DBConfig.getDataSource().getConnection();  
  
         PreparedStatement stmt = conn.prepareStatement(query)) {  
  
        stmt.setInt(1, codigoRestaurante);  
  
  
        try (ResultSet rs = stmt.executeQuery()) {  
  
            while (rs.next()) {  
  
                menus.add(mapResultSetToMenu(rs));  
  
            }  
        }  
    }  
}
```

```
    }

} catch (SQLException e) {

    throw new SQLException("Error al listar los menús del restaurante: " +
e.getMessage());

}

return menus;

}

public void update(Menu menu) throws SQLException {

    String query = "UPDATE menu SET telefono = ?, ruta_archivo = ? WHERE
id_menu = ?";

try (Connection conn = DBConfig.getDataSource().getConnection()) {

    PreparedStatement stmt = conn.prepareStatement(query) {

        stmt.setString(1, menu.getTelefono());
        stmt.setString(2, menu.getRuta_archivo());
        stmt.setInt(3, menu.getId_menu());

        stmt.executeUpdate();

    } catch (SQLException e) {

        throw new SQLException("Error al actualizar el menú: " + e.getMessage());
    }
}
```

}

}

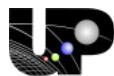
```
public void delete(int id) throws SQLException {  
  
    String query = "DELETE FROM menu WHERE id_menu = ?";  
  
    try (Connection conn = DBConfig.getDataSource().getConnection();  
         PreparedStatement stmt = conn.prepareStatement(query)) {  
  
        stmt.setInt(1, id);  
  
        int rowsAffected = stmt.executeUpdate();  
  
        if (rowsAffected == 0) {  
            throw new SQLException("No se encontró el menú con ID: " + id);  
        }  
  
    } catch (SQLException e) {  
        throw new SQLException("Error al eliminar el menú: " + e.getMessage());  
    }  
  
}  
  
private Menu mapResultSetToMenu(ResultSet rs) throws SQLException {  
    return new Menu(  
}
```

```
        rs.getInt("id_menu"),  
  
        rs.getString("telefono"),  
  
        rs.getString("ruta_archivo")  
  
    );  
  
}  
  
}
```

### **Solicitud\_registro Repositorio**

```
package org.sazonpt.repository;  
  
  
import java.sql.Connection;  
  
import java.sql.PreparedStatement;  
  
import java.sql.ResultSet;  
  
import java.sql.SQLException;  
  
import java.util.ArrayList;  
  
import java.util.List;  
  
  
import org.sazonpt.config.DBConfig;  
  
import org.sazonpt.model.Menu;  
  
  
  
  
public class MenuRepository {  
  
  
  
  
    public void save(Menu menu) throws SQLException {  
  
        String query = "INSERT INTO menu(telefono, ruta_archivo) VALUES(?, ?)";
```

```
try (Connection conn = DBConfig.getDataSource().getConnection();  
     PreparedStatement stmt = conn.prepareStatement(query)) {  
  
    stmt.setString(1, menu.getTelefono());  
    stmt.setString(2, menu.getRuta_archivo());  
    stmt.executeUpdate();  
  
} catch (SQLException e) {  
    throw new SQLException("Error al crear el menú: " + e.getMessage());  
}  
}  
  
public Menu findById(int id) throws SQLException {  
  
    String query = "SELECT * FROM menu WHERE id_menu = ?";  
    Menu menu = null;  
  
    try (Connection conn = DBConfig.getDataSource().getConnection();  
         PreparedStatement stmt = conn.prepareStatement(query)) {  
  
        stmt.setInt(1, id);  
  
        try (ResultSet rs = stmt.executeQuery()) {  
            ...  
        }  
    }  
}
```



```
if (rs.next()) {  
  
    menu = mapResultSetToMenu(rs);  
  
}  
  
}  
  
} catch (SQLException e) {  
  
    throw new SQLException("Error al buscar el menú: " + e.getMessage());  
  
}  
  
  
return menu;  
}  
  
  
public List<Menu> findAll() throws SQLException {  
  
    List<Menu> menus = new ArrayList<>();  
  
    String query = "SELECT * FROM menu";  
  
  
    try (Connection conn = DBConfig.getDataSource().getConnection();  
  
        PreparedStatement stmt = conn.prepareStatement(query);  
  
        ResultSet rs = stmt.executeQuery()) {  
  
  
        while (rs.next()) {  
  
            menus.add(mapResultSetToMenu(rs));  
  
        }  
  
    } catch (SQLException e) {  
  
    }
```

```
        throw new SQLException("Error al listar los menus: " + e.getMessage());  
    }  
  
    return menus;  
}  
  
public List<Menu> findByRestaurant(int codigoRestaurante) throws SQLException {  
    List<Menu> menus = new ArrayList<>();  
  
    String query = "SELECT * FROM menu WHERE id_restaurante = ?";  
  
    try (Connection conn = DBConfig.getDataSource().getConnection()) {  
        PreparedStatement stmt = conn.prepareStatement(query)) {  
            stmt.setInt(1, codigoRestaurante);  
  
            try (ResultSet rs = stmt.executeQuery()) {  
                while (rs.next()) {  
                    menus.add(mapResultSetToMenu(rs));  
                }  
            }  
        } catch (SQLException e) {  
            throw new SQLException("Error al listar los menús del restaurante: " +  
e.getMessage());  
        }  
    }  
}
```

}

return menus;

}

public void update(Menu menu) throws SQLException {

    String query = "UPDATE menu SET telefono = ?, ruta\_archivo = ? WHERE  
    id\_menu = ?";

try (Connection conn = DBConfig.getDataSource().getConnection());

    PreparedStatement stmt = conn.prepareStatement(query)) {

        stmt.setString(1, menu.getTelefono());

        stmt.setString(2, menu.getRuta\_archivo());

        stmt.setInt(3, menu.getId\_menu());

    stmt.executeUpdate();

} catch (SQLException e) {

    throw new SQLException("Error al actualizar el menú: " + e.getMessage());

}

}

public void delete(int id) throws SQLException {

```
String query = "DELETE FROM menu WHERE id_menu = ?";  
  
try (Connection conn = DBConfig.getDataSource().getConnection();  
     PreparedStatement stmt = conn.prepareStatement(query)) {  
  
    stmt.setInt(1, id);  
  
    int rowsAffected = stmt.executeUpdate();  
  
    if (rowsAffected == 0) {  
  
        throw new SQLException("No se encontró el menú con ID: " + id);  
    }  
  
} catch (SQLException e) {  
  
    throw new SQLException("Error al eliminar el menú: " + e.getMessage());  
}  
  
}  
  
private Menu mapResultSetToMenu(ResultSet rs) throws SQLException {  
  
    return new Menu(  
  
        rs.getInt("id_menu"),  
  
        rs.getString("telefono"),  
  
        rs.getString("ruta_archivo")  
    );
```

}

}

## Usuario Repositorio

```
package org.sazonpt.repository;

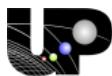
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import org.sazonpt.config.DBConfig;
import org.sazonpt.model.Usuario;

public class UsuarioRepository {

    public Usuario findByCorreo(String correo) throws SQLException {
        String query = "SELECT * FROM usuario WHERE correo = ?;";
        Usuario user = null;
        try(Connection conn = DBConfig.getDataSource().getConnection();
            PreparedStatement stmt = conn.prepareStatement(query)){
            stmt.setString(1, correo);
            try(ResultSet rs = stmt.executeQuery()){

```



```
if(rs.next()){

    user = mapResulsetToUser(rs);

}

}

} catch(SQLException e){

    throw new SQLException("Error al buscar usuario por correo:
"+e.getMessage());

}

return user;

}

public int save(Usuario user) throws SQLException {

if (findByCorreo(user.getCorreo()) != null) {

    throw new SQLException("Ya existe un usuario con ese correo");

}

String query = "INSERT INTO usuario(nombre, correo, contrasena, tipo, status)
VALUES (?, ?, ?, ?, ?);";

int id_usuario = -1;

System.out.println("[UsuarioRepository.save] Recibido:");

System.out.println("nombre: " + user.getNombre());

System.out.println("correo: " + user.getCorreo());

System.out.println("contrasena: " + user.getContrasena());

System.out.println("tipo: " + user.getTipo());

System.out.println("status: " + user.getStatus());
```

```
try(Connection conn = DBConfig.getDataSource().getConnection();

        PreparedStatement stmt = conn.prepareStatement(query,
java.sql.Statement.RETURN_GENERATED_KEYS){

    stmt.setString(1, user.getNombre());
    stmt.setString(2, user.getCorreo());
    stmt.setString(3, user.getContrasena());
    stmt.setString(4, user.getTipo());
    stmt.setString(5, user.getStatus());
    stmt.executeUpdate();

try(ResultSet generatedKeys = stmt.getGeneratedKeys(){

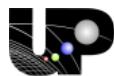
    if(generatedKeys.next()){
        id_usuario = generatedKeys.getInt(1);
    }
} catch(SQLException e){

    System.out.println("[UsuarioRepository.save] Error al obtener el ID generado:
" + e.getMessage());

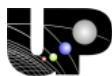
    throw new SQLException("Error al crear el ID del usuario creado: " +
e.getMessage());
}

} catch(SQLException e){

    System.out.println("[UsuarioRepository.save] Error SQL: " + e.getMessage());
```



```
        throw new SQLException("Error al crear el usuario: "+e.getMessage());  
    }  
  
    System.out.println("[UsuarioRepository.save] id_usuario generado: " + id_usuario);  
  
    return id_usuario;  
}  
  
  
public Usuario findByIdUser(int idUser) throws SQLException {  
  
    String query = "SELECT * FROM usuario WHERE id_usuario= ?;";  
  
    Usuario user = null;  
  
  
    try(Connection conn = DBConfig.getDataSource().getConnection();  
  
        PreparedStatement stmt = conn.prepareStatement(query)){  
  
        stmt.setInt(1, idUser);  
  
  
        try(ResultSet rs = stmt.executeQuery()){  
  
            if(rs.next()){  
  
                user = mapResulsetToUser(rs);  
  
            }  
  
        }  
  
    } catch(SQLException e){  
  
        throw new SQLException("Error al buscar el usuario: "+e.getMessage());  
    }  
  
    return user;
```



}

```
public List<Usuario> findAll() throws SQLException {  
  
    List<Usuario> users = new ArrayList<>();  
  
    String query = "SELECT * FROM usuario";  
  
  
    try (Connection conn = DBConfig.getDataSource().getConnection());  
  
        PreparedStatement stmt = conn.prepareStatement(query);  
  
        ResultSet rs = stmt.executeQuery() {  
  
            while (rs.next()) {  
  
                users.add(mapResulsetToUser(rs));  
  
            }  
  
        } catch (SQLException e) {  
  
            throw new SQLException("Error al listar los usuarios: " + e.getMessage());  
  
        }  
  
  
    return users;  
}  
  
  
  
  
public void UpdateUser(Usuario user) throws SQLException{  
  
    String query = "UPDATE usuario SET nombre= ?, correo= ?, contrasena= ?, tipo= ? , status= ? WHERE id_usuario= ?;";
```

```
try(Connection conn = DBConfig.getDataSource().getConnection();

PreparedStatement stmt = conn.prepareStatement(query)){

    stmt.setString(1, user.getNombre());
    stmt.setString(2, user.getCorreo());
    stmt.setString(3, user.getContrasena());
    stmt.setString(4, user.getTipo());
    stmt.setString(5, user.getStatus());
    stmt.setInt(6, user.getId_usuario());

    stmt.executeUpdate();

} catch(SQLException e){

    throw new SQLException("Error al actualizar el usuario: "+e.getMessage());
}

}

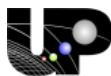
public void deleteUser(int idUser) throws SQLException{

    Connection conn = null;

    try {

        conn = DBConfig.getDataSource().getConnection();
        conn.setAutoCommit(false);

        String queryTipo = "SELECT tipo FROM usuario WHERE id_usuario = ?";
```

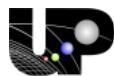


```
PreparedStatement stmtTipo = conn.prepareStatement(queryTipo);
stmtTipo.setInt(1, idUser);
ResultSet rs = stmtTipo.executeQuery();
```

```
String tipoUsuario = null;
if (rs.next()) {
    tipoUsuario = rs.getString("tipo");
} else {
    throw new SQLException("No se encontró el usuario con ID: " + idUser);
}
```

```
System.out.println("Eliminando usuario ID: " + idUser + ", Tipo: " + tipoUsuario);
```

```
if ("administrador".equals(tipoUsuario)) {
    String queryAdmin = "DELETE FROM administrador WHERE id_usuario = ?";
    PreparedStatement stmtAdmin = conn.prepareStatement(queryAdmin);
    stmtAdmin.setInt(1, idUser);
    int adminRows = stmtAdmin.executeUpdate();
    System.out.println("Filas eliminadas de administrador: " + adminRows);
} else if ("restaurantero".equals(tipoUsuario)) {
    String queryRest = "DELETE FROM restaurantero WHERE id_usuario = ?";
    PreparedStatement stmtRest = conn.prepareStatement(queryRest);
    stmtRest.setInt(1, idUser);
```



```
        int restRows = stmtRest.executeUpdate();

        System.out.println("Filas eliminadas de restaurantero: " + restRows);

    }

String queryUsuario = "DELETE FROM usuario WHERE id_usuario = ?";

PreparedStatement stmtUsuario = conn.prepareStatement(queryUsuario);

stmtUsuario.setInt(1, idUser);

int rowsAffected = stmtUsuario.executeUpdate();

System.out.println("Query executed: " + queryUsuario);

System.out.println("User ID: " + idUser);

System.out.println("Rows affected by delete: " + rowsAffected);

if (rowsAffected == 0) {

    throw new SQLException("No se pudo eliminar el usuario con ID: " + idUser);

}

conn.commit();

} catch(SQLException e){

    if (conn != null) {

        conn.rollback();

    }

}
```

```
        throw new SQLException("Error al eliminar el usuario: " + e.getMessage());  
  
    } finally {  
  
        if (conn != null) {  
  
            conn.setAutoCommit(true);  
  
            conn.close();  
  
        }  
  
    }  
  
}
```

```
public Usuario findByCorreoAndContrasenaAndRol(String correo, String contrasena,  
String rol) throws SQLException {  
  
    String query = "SELECT * FROM usuario WHERE correo = ? AND contrasena = ?  
AND tipo = ?;";  
  
    Usuario user = null;  
  
    try(Connection conn = DBConfig.getDataSource().getConnection();  
  
        PreparedStatement stmt = conn.prepareStatement(query)){  
  
        stmt.setString(1, correo);  
  
        stmt.setString(2, contrasena);  
  
        stmt.setString(3, rol);  
  
        try(ResultSet rs = stmt.executeQuery()){  
  
            if(rs.next()){  
  
                user = mapResulsetToUser(rs);  
  
            }  
  
        }
```

```
    }

} catch(SQLException e){

    throw new SQLException("Error al buscar el usuario: "+e.getMessage());

}

return user;

}

private Usuario mapResulsetToUser(ResultSet rs) throws SQLException {

    return new Usuario(

        rs.getInt("id_usuario"),

        rs.getString("nombre"),

        rs.getString("correo"),

        rs.getString("contrasena"),

        rs.getString("tipo"),

        rs.getString("status")

    );

}

}
```

### 5.1.2 Conexiones

#### Admin Controller

```
package org.sazonpt.controller;
```

```
import io.javalin.http.Context;
import io.javalin.http.HttpStatus;
import org.sazonpt.model.Administrador;
import org.sazonpt.service.AdminService;

import java.sql.SQLException;
import java.util.List;

public class AdminController {
    private final AdminService adminService;

    public AdminController(AdminService adminService) {
        this.adminService = adminService;
    }

    public void getAll(Context ctx) {
        try {
            List<Administrador> admins = adminService.getAllAdmins();
            ctx.json(admins);
        } catch (SQLException e) {
            ctx.status(500).result("Error al obtener administradores");
        }
    }
}
```

```
public void getAdminById(Context ctx) {  
  
    try {  
  
        int id = Integer.parseInt(ctx.pathParam("id"));  
  
        Administrador admin = adminService.getByIdAdmin(id);  
  
        if (admin != null) {  
  
            ctx.json(admin);  
  
        } else {  
  
            ctx.status(HttpStatus.NOT_FOUND).result("Administrador no encontrado");  
  
        }  
  
    } catch (Exception e) {  
  
        ctx.status(404).result("Error al obtener administrador");  
  
    }  
  
}
```

```
public void create(Context ctx) {  
  
    try {  
  
        Administrador admin = ctx.bodyAsClass(Administrador.class);  
  
        adminService.createAdmin(admin);  
  
        ctx.status(201).result("Administrador creado");  
  
    } catch (Exception e) {  
  
        ctx.status(400).result("Error al crear administrador");  
  
    }  
  
}
```

}

```
public void update(Context ctx) {  
  
    try {  
  
        int id = Integer.parseInt(ctx.pathParam("id"));  
  
        Administrador admin = ctx.bodyAsClass(Administrador.class);  
  
        admin.setIdUsuario(id);  
  
        adminService.updateAdmin(admin);  
  
        ctx.status(200).result("Administrador actualizado");  
  
    } catch (Exception e) {  
  
        ctx.status(400).result("Error al actualizar administrador");  
  
    }  
  
}
```

```
public void delete(Context ctx) {  
  
    try {  
  
        int id = Integer.parseInt(ctx.pathParam("id"));  
  
        System.out.println("Intentando eliminar el administrador con Id: " + id);  
  
        adminService.deleteAdmin(id);  
  
        ctx.status(200).result("Administrador eliminado exitosamente");  
  
    } catch (NumberFormatException e) {  
  
        System.out.println("formato de Id invalido: " + ctx.pathParam("id"));  
  
        ctx.status(400).result("ID de administrador inválido");  
  
    }  
  
}
```

```
        } catch (SQLException e) {  
  
            System.out.println("Error de SQL al eliminar el Id: " + e.getMessage());  
  
            ctx.status(500).result("Error en base de datos: " + e.getMessage());  
  
        } catch (Exception e) {  
  
            System.out.println("Error inesperado al borrar el administrador: " +  
e.getMessage());  
  
            ctx.status(500).result("Error inesperado: " + e.getMessage());  
  
        }  
  
    }  
  
}
```

### DeBug Controller

```
package org.sazonpt.controller;  
  
  
import io.javalin.http.Context;  
  
import org.sazonpt.service.DebugService;  
  
  
  
public class DebugController {  
  
    private final DebugService debugService;  
  
  
  
    public DebugController(DebugService debugService) {  
  
        this.debugService = debugService;  
  
    }  
}
```

```
public void checkDatabase(Context ctx {  
  
    try {  
  
        String result = debugService.checkDatabaseIntegrity();  
  
        ctx.json(result);  
  
    } catch (Exception e) {  
  
        ctx.status(500).json("Error: " + e.getMessage());  
  
    }  
  
}
```

```
public void fixForeignKeys(Context ctx {  
  
    try {  
  
        String result = debugService.fixForeignKeyIssues();  
  
        ctx.json(result);  
  
    } catch (Exception e) {  
  
        ctx.status(500).json("Error: " + e.getMessage());  
  
    }  
  
}
```

## Estadística Controller

```
package org.sazonpt.controller;
```

```
import java.sql.SQLException;
```

```
import java.util.List;

import org.sazonpt.model.Estadistica;
import org.sazonpt.service.EstadisticaService;

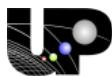
import io.javalin.http.Context;
import io.javalin.http.HttpStatus;

public class EstadisticaController {
    private final EstadisticaService estadisticaService;

    public EstadisticaController(EstadisticaService estadisticaService) {
        this.estadisticaService = estadisticaService;
    }

    public void getAll(Context ctx) {
        try {
            List<Estadistica> estadisticas = estadisticaService.getAllEstadisticas();
            ctx.json(estadisticas);
        } catch (SQLException e) {
            ctx.status(500).result("Error al obtener estadísticas");
        }
    }
}
```

```
public void getById(Context ctx) {  
  
    try {  
  
        int idEstadistica = Integer.parseInt(ctx.pathParam("id"));  
  
        Estadistica estadistica = estadisticaService.getById(idEstadistica);  
  
        if (estadistica != null) {  
  
            ctx.json(estadistica);  
  
        } else {  
  
            ctx.status(HttpStatus.NOT_FOUND).result("Estadística no encontrada");  
  
        }  
  
    } catch (SQLException e) {  
  
        ctx.status(404).result("Error al obtener estadística");  
  
    }  
  
}  
  
  
public void create(Context ctx) {  
  
    try {  
  
        Estadistica estadistica = ctx.bodyAsClass(Estadistica.class);  
  
        if (estadistica.getDescargas() == 0) estadistica.setDescargas(0);  
  
        if (estadistica.getComida() == 0) estadistica.setComida(0);  
  
        if (estadistica.getUbicacion() == 0) estadistica.setUbicacion(0);  
  
        if (estadistica.getRecomendacion() == 0) estadistica.setRecomendacion(0);  
  
        if (estadistica.getHorario() == 0) estadistica.setHorario(0);  
  
    }  
}
```



```
if (estadistica.getVista() == 0) estadistica.setVista(0);

estadisticaService.createEstadistica(estadistica);

ctx.status(201).json(java.util.Map.of(

    "success", true,
    "message", "Estadística creada correctamente"
));

} catch (Exception e) {

    ctx.status(400).json(java.util.Map.of(
        "success", false,
        "message", "Error al crear estadística: " + e.getMessage()
    ));
}

}

public void update(Context ctx) {

try {

    int idEstadistica = Integer.parseInt(ctx.pathParam("id"));

    Estadistica estadistica = ctx.bodyAsClass(Estadistica.class);

    estadistica.setId_estadistica(idEstadistica);

    if (estadistica.getDescargas() == 0) estadistica.setDescargas(0);

    if (estadistica.getComida() == 0) estadistica.setComida(0);

    if (estadistica.getUbicacion() == 0) estadistica.setUbicacion(0);

    if (estadistica.getRecomendacion() == 0) estadistica.setRecomendacion(0);
}
```

```
if (estadistica.getHorario() == 0) estadistica.setHorario(0);

if (estadistica.getVista() == 0) estadistica.setVista(0);

estadisticaService.updateEstadistica(estadistica);

ctx.status(200).result("Estadística actualizada");

} catch (Exception e) {

    ctx.status(400).result("Error al actualizar estadística: " + e.getMessage());

}

}

public void delete(Context ctx) {

try {

    int idEstadistica = Integer.parseInt(ctx.pathParam("id"));

    System.out.println("Intentando eliminar la estadística con id: " + idEstadistica);

    estadisticaService.deleteEstadistica(idEstadistica);

    ctx.status(200).result("Estadística eliminada exitosamente");

} catch (NumberFormatException e) {

    System.out.println("Formato invalido de id: " + ctx.pathParam("id"));

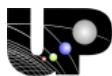
    ctx.status(400).result("ID de estadística inválido");

} catch (SQLException e) {

    System.out.println("Error sql al borrar la estadística: " + e.getMessage());

    ctx.status(500).result("Error en base de datos: " + e.getMessage());

} catch (Exception e) {
```



```
        System.out.println("Error inesperado al intentar borrar la estadística: " +  
e.getMessage());  
  
        ctx.status(500).result("Error inesperado: " + e.getMessage());  
  
    }  
  
}
```

## Menu Controller

```
package org.sazonpt.controller;  
  
  
import java.sql.SQLException;  
import java.util.List;  
  
  
import org.sazonpt.model.Menu;  
import org.sazonpt.service.MenuService;  
  
  
import io.javalin.http.Context;  
import io.javalin.http.HttpStatus;  
  
  
public class MenuController {  
    private final MenuService menuService;  
  
  
    public MenuController(MenuService menuService) {  
        this.menuService = menuService;
```

}

```
public void getAll(Context ctx) {  
  
    try {  
  
        List<Menu> menus = menuService.getAllMenus();  
  
        ctx.json(menus);  
  
    } catch (SQLException e) {  
  
        ctx.status(500).result("Error al obtener menús");  
  
    }  
  
}
```

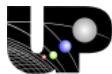
```
public void getById(Context ctx) {  
  
    try {  
  
        int idMenu = Integer.parseInt(ctx.pathParam("id"));  
  
        Menu menu = menuService.getById(idMenu);  
  
        if (menu != null) {  
  
            ctx.json(menu);  
  
        } else {  
  
            ctx.status(HttpStatus.NOT_FOUND).result("Menú no encontrado");  
  
        }  
  
    } catch (SQLException e) {  
  
        ctx.status(404).result("Error al obtener menú");  
  
    }  
  
}
```

}

}

```
public void getByRestaurant(Context ctx) {  
  
    try {  
  
        int codigoRestaurante = Integer.parseInt(ctx.pathParam("restaurante"));  
  
        List<Menu> menus = menuService.getMenusByRestaurant(codigoRestaurante);  
  
        ctx.json(menus);  
  
    } catch (SQLException e) {  
  
        ctx.status(500).result("Error al obtener menús del restaurante");  
  
    }  
  
}
```

```
public void create(Context ctx) {  
  
    try {  
  
        Menu menu = ctx.bodyAsClass(Menu.class);  
  
        menuService.createMenu(menu);  
  
        ctx.status(201).json(java.util.Map.of(  
  
            "success", true,  
  
            "message", "Menú creado correctamente"  
  
        ));  
  
    } catch (Exception e) {  
  
        ctx.status(400).json(java.util.Map.of(  
  
            "error", e.getMessage()  
        ));  
  
    }  
  
}
```



```
"success", false,  
  
        "message", "Error al crear menú: " + e.getMessage()  
));  
  
}  
  
}  
  
  
public void update(Context ctx) {  
  
    try {  
  
        int idMenu = Integer.parseInt(ctx.pathParam("id"));  
  
        Menu menu = ctx.bodyAsClass(Menu.class);  
  
        // Asegurarse de que el ID del menú coincida con el parámetro de la URL  
  
        Menu updatedMenu = new Menu(idMenu, menu.getTelefono(),  
menu.getRuta_archivo());  
  
        menuService.updateMenu(updatedMenu);  
  
        ctx.status(200).result("Menú actualizado");  
  
    } catch (Exception e) {  
  
        ctx.status(400).result("Error al actualizar menú: " + e.getMessage());  
  
    }  
  
}  
  
  
public void delete(Context ctx) {  
  
    try {  
  
        int idMenu = Integer.parseInt(ctx.pathParam("id"));  
  
    }  
  
}
```

```
System.out.println("Intentando eliminar el menu con id: " + idMenu);

menuService.deleteMenu(idMenu);

ctx.status(200).result("Menú eliminado exitosamente");

} catch (NumberFormatException e) {

    System.out.println("formato invalido de id: " + ctx.pathParam("id"));

    ctx.status(400).result("ID de menú inválido");

} catch (SQLException e) {

    System.out.println("Error sql al borrar el menu: " + e.getMessage());

    ctx.status(500).result("Error en base de datos: " + e.getMessage());

} catch (Exception e) {

    System.out.println("Error inesperado al intentar borrar el menu: " +
e.getMessage());

    ctx.status(500).result("Error inesperado: " + e.getMessage());

}

}

}
```

### Solicitud Register Controller

```
package org.sazonpt.controller;
```

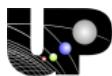
```
import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

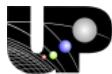
import java.io.InputStream;
```

```
import java.sql.SQLException;  
  
import java.time.LocalDate;  
  
import java.util.List;  
  
  
import org.sazonpt.model.Solicitud_registro;  
  
import org.sazonpt.service.Solicitud_registroService;  
  
  
import io.javalin.http.Context;  
  
import io.javalin.http.HttpStatus;  
  
import io.javalin.http UploadedFile;  
  
  
  
public class Solicitud_registroController {  
  
    public void updateEstado(Context ctx) {  
  
        try {  
  
            int idSolicitud = Integer.parseInt(ctx.pathParam("id"));  
  
            String nuevoEstado = ctx.bodyAsClass(EstadoRequest.class).getEstado();  
  
            if (nuevoEstado == null || nuevoEstado.trim().isEmpty()) {  
  
                ctx.status(400).json(java.util.Map.of(  
  
                    "success", false,  
  
                    "message", "El campo 'estado' es obligatorio"  
  
                ));  
  
                return;  
            }  
        }  
    }  
}
```



```
solicitudService.updateEstado(idSolicitud, nuevoEstado.trim());  
  
ctx.status(200).json(java.util.Map.of(  
  
    "success", true,  
  
    "message", "Estado actualizado correctamente"  
  
));  
  
} catch (NumberFormatException e) {  
  
    ctx.status(400).json(java.util.Map.of(  
  
        "success", false,  
  
        "message", "ID de solicitud inválido"  
  
));  
  
} catch (SQLException e) {  
  
    ctx.status(500).json(java.util.Map.of(  
  
        "success", false,  
  
        "message", "Error al actualizar estado: " + e.getMessage()  
  
));  
  
} catch (Exception e) {  
  
    ctx.status(500).json(java.util.Map.of(  
  
        "success", false,  
  
        "message", "Error interno del servidor"  
  
));  
  
}  
}
```

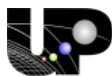
```
public static class EstadoRequest {  
    private String estado;  
  
    public String getEstado() { return estado; }  
  
    public void setEstado(String estado) { this.estado = estado; }  
  
}  
  
private final Solicitud_registroService solicitudService;  
  
  
public Solicitud_registroController(Solicitud_registroService solicitudService) {  
    this.solicitudService = solicitudService;  
}  
  
  
public void getAll(Context ctx) {  
    try {  
        List<Solicitud_registro> solicitudes = solicitudService.getAllSolicitudes();  
        ctx.json(solicitudes);  
    } catch (SQLException e) {  
        ctx.status(500).result("Error al obtener solicitudes");  
    }  
}  
  
  
public void getById(Context ctx) {  
    try {  
        int idSolicitud = Integer.parseInt(ctx.pathParam("id"));  
    }  
}
```



```
Solicitud_registro solicitud = solicitudService.getById(idSolicitud);
```

```
if (solicitud != null) {  
  
    ctx.json(solicitud);  
  
} else {  
  
    ctx.status(HttpStatus.NOT_FOUND).result("Solicitud no encontrada");  
  
}  
  
} catch (SQLException e) {  
  
    ctx.status(404).result("Error al obtener solicitud");  
  
}  
  
}
```

```
public void create(Context ctx) {  
  
    try {  
  
        Solicitud_registro solicitud = ctx.bodyAsClass(Solicitud_registro.class);  
  
        if (solicitud.getEtiqueta1() == null) solicitud.setEtiqueta1("");  
  
        if (solicitud.getEtiqueta2() == null) solicitud.setEtiqueta2("");  
  
        if (solicitud.getEtiqueta3() == null) solicitud.setEtiqueta3("");  
  
        solicitudService.createSolicitud(solicitud);  
  
        ctx.status(201).result("Solicitud creada correctamente");  
  
    } catch (Exception e) {  
  
        ctx.status(400).json(java.util.Map.of(  
  
            "success", false,
```



```
"message", "Error al crear solicitud: " + e.getMessage()

));

}

}

public void createWithFiles(Context ctx {

try {

var restaurante = ctx.formParam("restaurante");

var propietario = ctx.formParam("propietario");

var correo = ctx.formParam("correo");

var numero = ctx.formParam("numero");

var direccion = ctx.formParam("direccion");

var horario = ctx.formParam("horario");

var estado = ctx.formParam("estado");

var facebook = ctx.formParam("facebook");

var instagram = ctx.formParam("instagram");

var menu = ctx.formParam("menu");

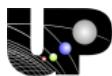
var etiqueta1 = ctx.formParam("etiqueta1");

var etiqueta2 = ctx.formParam("etiqueta2");

var etiqueta3 = ctx.formParam("etiqueta3");

if (restaurante == null || restaurante.trim().isEmpty() ||

correo == null || correo.trim().isEmpty() ||
```

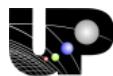


```
direccion == null || direccion.trim().isEmpty()) {  
  
    ctx.status(400).json(java.util.Map.of(  
  
        "success", false,  
  
        "message", "Los campos restaurante, correo y direccion son obligatorios"  
  
    ));  
  
    return;  
  
}
```

```
var imagen1 = ctx.uploadedFile("imagen1");  
  
var imagen2 = ctx.uploadedFile("imagen2");  
  
var imagen3 = ctx.uploadedFile("imagen3");  
  
var comprobante = ctx.uploadedFile("comprobante");  
  
var menuFile = ctx.uploadedFile("menu");
```

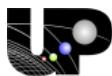
```
String urlImagen1 = imagen1 != null ? saveImageFile(imagen1) : null;  
  
String urlImagen2 = imagen2 != null ? saveImageFile(imagen2) : null;  
  
String urlImagen3 = imagen3 != null ? saveImageFile(imagen3) : null;  
  
String urlComprobante = comprobante != null ?  
    saveDocumentFile(comprobante) : null;  
  
String urlMenu = menuFile != null ? saveDocumentFile(menuFile) : "";
```

```
Solicitud_registro solicitud = new Solicitud_registro();
```

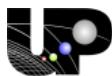


```
solicitud.setRestaurante(restaurante.trim());  
  
solicitud.setPropietario(propietario != null ? propietario.trim() : "");  
  
solicitud.setCorreo(correo.trim());  
  
solicitud.setNumero(numero != null ? numero.trim() : "");  
  
solicitud.setDireccion(direccion.trim());  
  
solicitud.setHorario(horario != null ? horario.trim() : "");  
  
solicitud.setFacebook(facebook != null ? facebook.trim() : "");  
  
solicitud.setInstagram(instagram != null ? instagram.trim() : "");  
  
solicitud.setMenu(urlMenu);  
  
solicitud.setEtiqueta1(etiqueta1 != null ? etiqueta1.trim() : "");  
  
solicitud.setEtiqueta2(etiqueta2 != null ? etiqueta2.trim() : "");  
  
solicitud.setEtiqueta3(etiqueta3 != null ? etiqueta3.trim() : "");  
  
solicitud.setEstado("pendiente");  
  
solicitud.setFecha(LocalDate.now());  
  
solicitud.setImagen1(urllImagen1);  
  
solicitud.setImagen2(urllImagen2);  
  
solicitud.setImagen3(urllImagen3);  
  
solicitud.setComprobante(urlComprobante);  
  
solicitudService.createSolicitud(solicitud);
```

```
ctx.status(201).json(java.util.Map.of(  
    "success", true,  
    "message", "Solicitud creada correctamente con archivos",
```

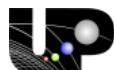


```
"data", java.util.Map.of(  
  
    "restaurante", solicitud.getRestaurante(),  
  
    "correo", solicitud.getCorreo(),  
  
    "facebook", solicitud.getFacebook(),  
  
    "instagram", solicitud.getInstagram(),  
  
    "menu", urlMenu != null ? urlMenu : "No se subió menú",  
  
    "imagen1", urlImagen1 != null ? urlImagen1 : "No se subió imagen",  
  
    "imagen2", urlImagen2 != null ? urlImagen2 : "No se subió imagen",  
  
    "imagen3", urlImagen3 != null ? urlImagen3 : "No se subió imagen",  
  
    "comprobante", urlComprobante != null ? urlComprobante : "No se subió  
comprobante"  
  
)  
));  
  
}  
} catch (SQLException sqlEx) {  
  
    System.err.println("Error de base de datos: " + sqlEx.getMessage());  
  
    ctx.status(500).json(java.util.Map.of(  
  
        "success", false,  
  
        "message", "Error de base de datos: " + sqlEx.getMessage()  
  
));  
  
}  
} catch (IllegalArgumentException argEx) {  
  
    System.err.println("Error de validación: " + argEx.getMessage());
```



```
ctx.status(400).json(java.util.Map.of(  
    "success", false,  
    "message", argEx.getMessage()  
  
} catch (Exception e) {  
  
    System.err.println("Error interno: " + e.getMessage());  
    e.printStackTrace();  
  
    ctx.status(500).json(java.util.Map.of(  
        "success", false,  
        "message", "Error interno del servidor"  
}  
}  
  
private String saveImageFile(UploadedFile file) throws IOException {  
  
    String contentType = file.contentType();  
  
    if (contentType == null || !contentType.startsWith("image/")) {  
  
        throw new IllegalArgumentException("El archivo debe ser una imagen");  
    }  
  
    String staticPath = "uploads/images";  
  
    File uploadDir = new File(staticPath);
```

```
if (!uploadDir.exists()) {  
  
    boolean created = uploadDir.mkdirs();  
  
    if (!created) {  
  
        throw new IOException("No se pudo crear el directorio de imágenes");  
  
    }  
  
}  
  
  
String fileName = System.currentTimeMillis() + "_" + file.filename();  
  
File destino = new File(uploadDir, fileName);  
  
  
try (InputStream is = file.content();  
  
        FileOutputStream os = new FileOutputStream(destino)) {  
  
    is.transferTo(os);  
  
}  
  
  
String host = "52.23.26.163";  
  
return String.format("http://%s:7070/uploads/images/%s", host, fileName);  
  
}  
  
  
private String saveDocumentFile(UploadedFile file) throws IOException {  
  
    String contentType = file.contentType();  
  
    if (contentType == null || !contentType.equals("application/pdf")) {
```



```
        throw new IllegalArgumentException("El comprobante debe ser un archivo  
PDF");
```

```
}
```

```
String staticPath = "uploads/documents";
```

```
File uploadDir = new File(staticPath);
```

```
if (!uploadDir.exists()) {
```

```
    boolean created = uploadDir.mkdirs();
```

```
    if (!created) {
```

```
        throw new IOException("No se pudo crear el directorio de documentos");
```

```
}
```

```
}
```

```
String fileName = System.currentTimeMillis() + "_" + file.filename();
```

```
File destino = new File(uploadDir, fileName);
```

```
try (InputStream is = file.content());
```

```
    FileOutputStream os = new FileOutputStream(destino)) {
```

```
    is.transferTo(os);
```

```
}
```

```
String host = "52.23.26.163";
```

```
return String.format("http://%s:7070/uploads/documents/%s", host, fileName);
```

}

```
public void updateWithFiles(Context ctx) {  
  
    try {  
  
        int idSolicitud = Integer.parseInt(ctx.pathParam("id"));  
  
        Solicitud_registro solicitudExistente = solicitudService.getById(idSolicitud);  
  
        if (solicitudExistente == null) {  
  
            ctx.status(404).json(java.util.Map.of(  
  
                "success", false,  
  
                "message", "No existe una solicitud con este ID"  
  
            ));  
  
            return;  
        }  
  
        var restaurante = ctx.formParam("restaurante");  
  
        var propietario = ctx.formParam("propietario");  
  
        var correo = ctx.formParam("correo");  
  
        var numero = ctx.formParam("numero");  
  
        var direccion = ctx.formParam("direccion");  
  
        var horario = ctx.formParam("horario");  
  
        var estado = ctx.formParam("estado");  
  
        var facebook = ctx.formParam("facebook");  
    }  
}
```

```
var instagram = ctx.formParam("instagram");
```

```
var menu = ctx.formParam("menu");
```

```
var etiqueta1 = ctx.formParam("etiqueta1");
```

```
var etiqueta2 = ctx.formParam("etiqueta2");
```

```
var etiqueta3 = ctx.formParam("etiqueta3");
```

```
var imagen1 = ctx.uploadedFile("imagen1");
```

```
var imagen2 = ctx.uploadedFile("imagen2");
```

```
var imagen3 = ctx.uploadedFile("imagen3");
```

```
var comprobante = ctx.uploadedFile("comprobante");
```

```
var menuFile = ctx.uploadedFile("menu");
```

```
String urlImagen1 = imagen1 != null ? saveImageFile(imagen1) :  
solicitudExistente.getImagen1();
```

```
String urlImagen2 = imagen2 != null ? saveImageFile(imagen2) :  
solicitudExistente.getImagen2();
```

```
String urlImagen3 = imagen3 != null ? saveImageFile(imagen3) :  
solicitudExistente.getImagen3();
```

```
String urlComprobante = comprobante != null ?  
saveDocumentFile(comprobante) : solicitudExistente.getComprobante();
```

```
String urlMenu = menuFile != null ? saveDocumentFile(menuFile) :  
solicitudExistente getMenu();
```

```
Solicitud_registro solicitud = new Solicitud_registro();
```

```
solicitud.setId_solicitud(idSolicitud);

        solicitud.setRestaurante(restaurante != null ? restaurante.trim() : solicitudExistente.getRestaurante());

        solicitud.setPropietario(propietario != null ? propietario.trim() : solicitudExistente.getPropietario());

        solicitud.setCorreo(correo != null ? correo.trim() : solicitudExistente.getCorreo());

        solicitud.setNumero(numero != null ? numero.trim() : solicitudExistente.getNumero());

        solicitud.setDireccion(direccion != null ? direccion.trim() : solicitudExistente.getDireccion());

        solicitud.setHorario(horario != null ? horario.trim() : solicitudExistente.getHorario());

        solicitud.setFacebook(facebook != null ? facebook.trim() : solicitudExistente.getFacebook());

        solicitud.setInstagram(instagram != null ? instagram.trim() : solicitudExistente.getInstagram());

        solicitud.setMenu(urlMenu);

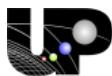
        solicitud.setEtiqueta1(etiqueta1 != null ? etiqueta1.trim() : solicitudExistente.getEtiqueta1());

        solicitud.setEtiqueta2(etiqueta2 != null ? etiqueta2.trim() : solicitudExistente.getEtiqueta2());

        solicitud.setEtiqueta3(etiqueta3 != null ? etiqueta3.trim() : solicitudExistente.getEtiqueta3());

        solicitud.setEstado(estado != null ? estado.trim() : solicitudExistente.getEstado());

        solicitud.setFecha(solicitudExistente.getFecha());
```



```
solicitud.setImagen1(urlImagen1);

solicitud.setImagen2(urlImagen2);

solicitud.setImagen3(urlImagen3);

solicitud.setComprobante(urlComprobante);

solicitudService.updateSolicitud(solicitud);

ctx.status(200).json(java.util.Map.of(
    "success", true,
    "message", "Solicitud actualizada correctamente con archivos",
    "data", java.util.Map.of(
        "restaurante", solicitud.getRestaurante(),
        "correo", solicitud.getCorreo(),
        "facebook", solicitud.getFacebook(),
        "instagram", solicitud.getInstagram(),
        "menu", urlMenu != null ? urlMenu : "No se subió menú",
        "imagen1", urlImagen1 != null ? urlImagen1 : "No se subió imagen",
        "imagen2", urlImagen2 != null ? urlImagen2 : "No se subió imagen",
        "imagen3", urlImagen3 != null ? urlImagen3 : "No se subió imagen",
        "comprobante", urlComprobante != null ? urlComprobante : "No se subió
comprobante"
    )
));
```

```
        } catch (Exception e) {

            ctx.status(400).json(java.util.Map.of(
                "success", false,
                "message", "Error al actualizar solicitud: " + e.getMessage()
            ));

        }

    }

public void delete(Context ctx) {
    try {
        int idSolicitud = Integer.parseInt(ctx.pathParam("id"));

        System.out.println("Intentando borrar la solicitud con id: " + idSolicitud);
        solicitudService.deleteSolicitud(idSolicitud);

        ctx.status(200).result("Solicitud eliminada exitosamente");
    } catch (NumberFormatException e) {
        System.out.println("Formato invalido de id: " + ctx.pathParam("id"));

        ctx.status(400).result("ID de solicitud inválido");
    } catch (SQLException e) {
        System.out.println("Error sql al borrar el registro: " + e.getMessage());

        ctx.status(500).result("Error en base de datos: " + e.getMessage());
    } catch (Exception e) {
        System.out.println("Error inesperado al borrar la solicitud: " + e.getMessage());

        ctx.status(500).result("Error inesperado: " + e.getMessage());
    }
}
```

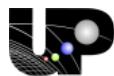
```
    }  
  
}
```

```
public void aprobarSolicitud(Context ctx) {  
  
    try {  
  
        int idSolicitud = Integer.parseInt(ctx.pathParam("id"));  
  
        solicitudService.aprobarSolicitud(idSolicitud);  
  
  
        ctx.status(200).json(java.util.Map.of(  
  
            "success", true,  
  
            "message", "Solicitud aprobada correctamente. El restaurante ha sido  
creado."  
  
        ));  
  
    } catch (NumberFormatException e) {  
  
        ctx.status(400).json(java.util.Map.of(  
  
            "success", false,  
  
            "message", "ID de solicitud inválido"  
  
        ));  
  
    } catch (IllegalArgumentException e) {  
  
        ctx.status(404).json(java.util.Map.of(  
  
            "success", false,  
  
            "message", e.getMessage()  
  
        ));  
  
    }  
}
```

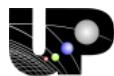
```
        } catch (SQLException e) {  
  
            System.err.println("Error al aprobar solicitud: " + e.getMessage());  
  
            ctx.status(500).json(java.util.Map.of(  
  
                "success", false,  
  
                "message", "Error al aprobar solicitud: " + e.getMessage()  
  
            ));  
  
        } catch (Exception e) {  
  
            System.err.println("Error inesperado: " + e.getMessage());  
  
            e.printStackTrace();  
  
            ctx.status(500).json(java.util.Map.of(  
  
                "success", false,  
  
                "message", "Error interno del servidor"  
  
            ));  
  
        }  
  
    }  
  
}
```

```
public void rechazarSolicitud(Context ctx) {  
  
    try {  
  
        int idSolicitud = Integer.parseInt(ctx.pathParam("id"));  
  
        solicitudService.rechazarSolicitud(idSolicitud);  
  
    }  
  
}
```

```
    ctx.status(200).json(java.util.Map.of(  
  
        "success", true,  
  
    ));  
  
}
```



```
"message", "Solicitud rechazada correctamente"
));
} catch (NumberFormatException e) {
    ctx.status(400).json(java.util.Map.of(
        "success", false,
        "message", "ID de solicitud inválido"
));
} catch (IllegalArgumentException e) {
    ctx.status(404).json(java.util.Map.of(
        "success", false,
        "message", e.getMessage()
));
} catch (SQLException e) {
    System.err.println("Error al rechazar solicitud: " + e.getMessage());
    ctx.status(500).json(java.util.Map.of(
        "success", false,
        "message", "Error al rechazar solicitud: " + e.getMessage()
));
} catch (Exception e) {
    System.err.println("Error inesperado: " + e.getMessage());
    e.printStackTrace();
    ctx.status(500).json(java.util.Map.of(
        "success", false,
```



```
        "message", "Error interno del servidor"  
    ));  
}  
}  
}
```

### **Usuario Controller**

```
package org.sazonpt.controller;
```

```
import java.sql.SQLException;  
import java.util.List;  
  
import org.sazonpt.model.Usuario;  
import org.sazonpt.service.UsuarioService;
```

```
import io.javalin.http.Context;  
import io.javalin.http.HttpStatus;
```

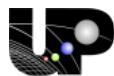
```
public class UsuarioController {  
    private final UsuarioService userService;  
  
    public UsuarioController(UsuarioService userService){  
        this.userService = userService;  
    }
```

```
public void getAll(Context ctx) {  
  
    try {  
  
        List<Usuario> users = userService.getAllUsers();  
  
        ctx.json(users);  
  
    } catch (SQLException e) {  
  
        ctx.status(500).result("Error al obtener usuarios");  
  
    }  
  
}  
  
  
public void getById(Context ctx) {  
  
    try {  
  
        int idUser = Integer.parseInt(ctx.pathParam("id"));  
  
        Usuario user = userService.getByIdUser(idUser);  
  
  
        if (user != null) {  
  
            ctx.json(user);  
  
        } else {  
  
            ctx.status(HttpStatus.NOT_FOUND).result("Usuario no encontrado");  
  
        }  
  
    } catch (SQLException e) {  
  
        ctx.status(404).result("Error al obtener usuario");  
  
    }  
}
```

}

```
public void create(Context ctx) {  
  
    try {  
  
        Usuario user = ctx.bodyAsClass(Usuario.class);  
  
        int id_usuario = userService.createUser(user);  
  
  
        java.util.Map<String, Object> data = new java.util.HashMap<>();  
  
        data.put("id_usuario", id_usuario);  
  
        data.put("nombre", user.getNombre());  
  
        data.put("correo", user.getCorreo());  
  
        data.put("tipo", user.getTipo());  
  
        data.put("status", user.getStatus());  
  
  
        ctx.status(201).json(java.util.Map.of(  
  
            "success", true,  
  
            "message", "Usuario creado correctamente",  
  
            "data", data  
));  
  
    } catch (Exception e) {  
  
        String msg = e.getMessage();  
  
        if (msg != null && (msg.contains("Duplicate entry") || msg.contains("UNIQUE") ||  
msg.contains("duplicate key"))){  

```



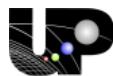
```
ctx.status(409).json(java.util.Map.of(  
    "success", false,  
    "message", "Ya existe un usuario con ese correo"  
));  
} else {  
    ctx.status(400).json(java.util.Map.of(  
        "success", false,  
        "message", "Error al crear usuario: " + msg  
    ));  
}  
}  
}
```

```
public void update(Context ctx) {  
    try {  
        int idUser = Integer.parseInt(ctx.pathParam("id"));  
        Usuario user = ctx.bodyAsClass(Usuario.class);  
        user.setIdUsuario(idUser);  
        userService.updateUser(user);  
        ctx.status(200).result("Usuario actualizado");  
    } catch (Exception e) {  
        ctx.status(400).result("Error al actualizar usuario");  
    }  
}
```

}

```
public void delete(Context ctx) {  
  
    try {  
  
        int idUser = Integer.parseInt(ctx.pathParam("id"));  
  
        System.out.println("Intentando borrar el usuario con id: " + idUser);  
  
        userService.deleteUser(idUser);  
  
        ctx.status(200).result("Usuario eliminado exitosamente");  
  
    } catch (NumberFormatException e) {  
  
        System.out.println("Formato invalido de id: " + ctx.pathParam("id"));  
  
        ctx.status(400).result("ID de usuario inválido");  
  
    } catch (SQLException e) {  
  
        System.out.println("Error sql al borrar el usuario: " + e.getMessage());  
  
        ctx.status(500).result("Error en base de datos: " + e.getMessage());  
  
    } catch (Exception e) {  
  
        System.out.println("Error inesperado al borrar el usuario: " + e.getMessage());  
  
        ctx.status(500).result("Error inesperado: " + e.getMessage());  
  
    }  
  
}
```

```
public void login(Context ctx) {  
  
    try {  
  
        var body = ctx.bodyAsClass(java.util.Map.class);  
  
    }
```



```
String correo = (String) body.get("correo");

String contrasena = (String) body.get("contrasena");

String rol = (String) body.get("rol");



Usuario usuario = userService.login(correo, contrasena, rol);

if (usuario != null) {

    ctx.json(java.util.Map.of(
        "success", true,
        "rol", usuario.getTipo(),
        "id_usuario", usuario.getId_usuario(),
        "message", "Login exitoso"
    ));

} else {

    ctx.json(java.util.Map.of(
        "success", false,
        "message", "Credenciales incorrectas o rol no válido"
    ));

}

} catch (Exception e) {

    ctx.status(500).json(java.util.Map.of(
        "success", false,
        "message", "Error en el servidor"
    )
}
```

```
));  
}  
}  
}
```

## Admin Service

```
package org.sazonpt.service;
```

```
import org.sazonpt.model.Administrador;  
import org.sazonpt.model.Usuario;  
import org.sazonpt.repository.AdminRepository;  
import org.sazonpt.repository.UsuarioRepository;
```

```
import java.sql.SQLException;  
import java.util.List;
```

```
public class AdminService {
```

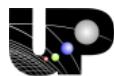
```
    private final AdminRepository adminRepo;
```

```
    public AdminService(AdminRepository adminRepo) {
```

```
        this.adminRepo = adminRepo;  
    }
```

```
    public List<Administrador> getAllAdmins() throws SQLException {
```

```
        return adminRepo.getAllAdmins();  
  
    }  
  
  
    public Administrador getByIdAdmin(int idAdmin) throws SQLException {  
  
        return adminRepo.findAdminById(idAdmin);  
  
    }  
  
  
  
    public void createAdmin(Administrador admin) throws SQLException {  
  
        if(admin.getNombre() == null || admin.getCorreo() == null || admin.getContrasena()  
== null) {  
  
            throw new IllegalArgumentException("Nombre, correo y contraseña son  
obligatorios");  
  
        }  
  
  
  
        if(!admin.getCorreo().contains("@")){  
  
            throw new IllegalArgumentException("El correo debe contener un '@'");  
  
        }  
  
        UsuarioRepository userRepo = new UsuarioRepository();  
  
        if(userRepo.findByCorreo(admin.getCorreo()) != null){  
  
            throw new IllegalArgumentException("Ya existe un usuario con este correo");  
  
        }  
  
  
  
        Usuario user = new Usuario(  
    }
```



```
        admin.getNombre(),  
        admin.getCorreo(),  
        admin.getContrasena(),  
        "administrador"  
    );  
  
    int id_usuario = userRepo.save(user);  
  
    adminRepo.createAdmin(new Administrador(id_usuario));  
  
}
```

```
public void updateAdmin(Administrador admin) throws SQLException {  
  
    adminRepo.updateAdmin(admin);  
  
}
```

```
public void deleteAdmin(int idAdmin) throws SQLException {  
  
    adminRepo.deleteAdmin(idAdmin);  
  
}
```

### Debug Service

```
package org.sazonpt.service;  
  
  
import org.sazonpt.model.Administrador;  
import org.sazonpt.model.Usuario;  
import org.sazonpt.repository.AdminRepository;
```

```
import org.sazonpt.repository.UsuarioRepository;

import java.sql.SQLException;
import java.util.List;

public class AdminService {

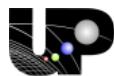
    private final AdminRepository adminRepo;

    public AdminService(AdminRepository adminRepo) {
        this.adminRepo = adminRepo;
    }

    public List<Administrador> getAllAdmins() throws SQLException {
        return adminRepo.getAllAdmins();
    }

    public Administrador getByIdAdmin(int idAdmin) throws SQLException {
        return adminRepo.findAdminById(idAdmin);
    }

    public void createAdmin(Administrador admin) throws SQLException {
        if(admin.getNombre() == null || admin.getCorreo() == null || admin.getContrasena()
== null) {
```



```
        throw new IllegalArgumentException("Nombre, correo y contraseña son  
obligatorios");
```

```
}
```

```
if(!admin.getCorreo().contains("@")){
```

```
    throw new IllegalArgumentException("El correo debe contener un '@'");
```

```
}
```

```
UsuarioRepository userRepo = new UsuarioRepository();
```

```
if(userRepo.findByCorreo(admin.getCorreo()) != null){
```

```
    throw new IllegalArgumentException("Ya existe un usuario con este correo");
```

```
}
```

```
Usuario user = new Usuario(
```

```
    admin.getNombre(),
```

```
    admin.getCorreo(),
```

```
    admin.getContrasena(),
```

```
    "administrador"
```

```
);
```

```
int id_usuario = userRepo.save(user);
```

```
adminRepo.createAdmin(new Administrador(id_usuario));
```

```
}
```

```
public void updateAdmin(Administrador admin) throws SQLException {
```

```
    adminRepo.updateAdmin(admin);

}

public void deleteAdmin(int idAdmin) throws SQLException {
    adminRepo.deleteAdmin(idAdmin);
}
```

### **Estadística Service**

```
package org.sazonpt.service;
```

```
import java.sql.SQLException;
```

```
public class DebugService {
```

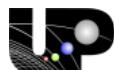
```
    public String checkDatabaseIntegrity() throws SQLException {
```

```
        StringBuilder result = new StringBuilder();
```

```
        String usuariosQuery = "SELECT COUNT(*) as count FROM usuario WHERE tipo  
= 'restaurantero';
```

```
        String restauranterosQuery = "SELECT COUNT(*) as count FROM restaurantero";
```

```
        result.append("Diagnóstico de Base de Datos:\n");
```



```
        result.append("-    Usuarios    tipo    restaurantero:  
").append(getCount(usuariosQuery)).append("\n");
```

```
        result.append("-    Registros    en    tabla    restaurantero:  
").append(getCount(restauranterosQuery)).append("\n");
```

```
String orphanedUsersQuery = """
```

```
    SELECT u.id_usuario, u.nombre, u.correo  
    FROM usuario u  
    LEFT JOIN restaurantero r ON u.id_usuario = r.id_usuario  
    WHERE u.tipo = 'restaurantero' AND r.id_usuario IS NULL
```

```
""";
```

```
        result.append("-    Usuarios    restaurantero    sin    registro:  
").append(getOrphanedUsers(orphanedUsersQuery));
```

```
    return result.toString();
```

```
}
```

```
public String fixForeignKeyIssues() throws SQLException {
```

```
    try {
```

```
        return "Migración completada. Usuarios migrados a tabla restaurantero.";
```

```
    } catch (Exception e) {
```

```
        throw new SQLException("Error durante la migración: " + e.getMessage());
```

```
}
```

}

```
private int getCount(String query) throws SQLException {  
  
    try (var conn = org.sazonpt.config.DBConfig.getDataSource().getConnection();  
  
        var stmt = conn.prepareStatement(query);  
  
        var rs = stmt.executeQuery()) {  
  
        if (rs.next()) {  
  
            return rs.getInt("count");  
  
        }  
  
    }  
  
    return 0;  
  
}
```

```
private String getOrphanedUsers(String query) throws SQLException {  
  
    StringBuilder users = new StringBuilder();  
  
    try (var conn = org.sazonpt.config.DBConfig.getDataSource().getConnection();  
  
        var stmt = conn.prepareStatement(query);  
  
        var rs = stmt.executeQuery()) {  
  
        while (rs.next()) {  
  
            users.append("\n * ID: ").append(rs.getInt("id_usuario"))  
                .append(", Nombre: ").append(rs.getString("nombre"))  
                .append(", Correo: ").append(rs.getString("correo"));  
  
        }  
    }
```

```
}
```

```
    return users.length() > 0 ? users.toString() : "\n (ninguno);
```

```
}
```

```
}
```

## Estadística Service

```
package org.sazonpt.service;
```

```
import java.sql.SQLException;
```

```
import java.util.List;
```

```
import org.sazonpt.model.Estadistica;
```

```
import org.sazonpt.repository.EstadisticaRepository;
```

```
public class EstadisticaService {
```

```
    private final EstadisticaRepository estadisticaRepo;
```

```
    public EstadisticaService(EstadisticaRepository estadisticaRepo) {
```

```
        this.estadisticaRepo = estadisticaRepo;
```

```
}
```

```
    public List<Estadistica> getAllEstadisticas() throws SQLException {
```

```
        return estadisticaRepo.listAllEstadisticas();
```

```
}
```

```
public Estadistica getById(int idEstadistica) throws SQLException {  
    return estadisticaRepo.findEstadistica(idEstadistica);  
}  
  
public void createEstadistica(Estadistica estadistica) throws SQLException {  
    if (estadistica.getCorreo() == null || estadistica.getCorreo().trim().isEmpty()) {  
        throw new IllegalArgumentException("El correo es obligatorio");  
    }  
    if (estadistica.getDescargas() == 0) estadistica.setDescargas(0);  
    if (estadistica.getComida() == 0) estadistica.setComida(0);  
    if (estadistica.getUbicacion() == 0) estadistica.setUbicacion(0);  
    if (estadistica.getRecomendacion() == 0) estadistica.setRecomendacion(0);  
    if (estadistica.getHorario() == 0) estadistica.setHorario(0);  
    if (estadistica.getVista() == 0) estadistica.setVista(0);  
    estadisticaRepo.addEstadistica(estadistica);  
}  
  
public void updateEstadistica(Estadistica estadistica) throws SQLException {  
    if (estadisticaRepo.findEstadistica(estadistica.getId_estadistica()) == null) {  
        throw new IllegalArgumentException("No existe una estadística con este ID");  
    }  
    if (estadistica.getDescargas() == 0) estadistica.setDescargas(0);
```

```
if (estadistica.getComida() == 0) estadistica.setComida(0);

if (estadistica.getUbicacion() == 0) estadistica.setUbicacion(0);

if (estadistica.getRecomendacion() == 0) estadistica.setRecomendacion(0);

if (estadistica.getHorario() == 0) estadistica.setHorario(0);

if (estadistica.getVista() == 0) estadistica.setVista(0);

estadisticaRepo.updateEstadistica(estadistica);

}
```

```
public void deleteEstadistica(int idEstadistica) throws SQLException {

    if (estadisticaRepo.findEstadistica(idEstadistica) == null) {

        throw new IllegalArgumentException("No existe una estadística con este ID");

    }

    boolean deleted = estadisticaRepo.deleteEstadistica(idEstadistica);

    if (!deleted) {

        throw new SQLException("No se pudo eliminar la estadística");

    }

}
```

## Menu Service

```
package org.sazonpt.service;
```

```
import java.sql.SQLException;

import java.util.List;
```

```
import org.sazonpt.model.Menu;

import org.sazonpt.repository.MenuRepository;

public class MenuService {

    private final MenuRepository menuRepo;

    public MenuService(MenuRepository menuRepo) {

        this.menuRepo = menuRepo;
    }

    public List<Menu> getAllMenus() throws SQLException {
        return menuRepo.findAll();
    }

    public Menu getById(int idMenu) throws SQLException {
        return menuRepo.findById(idMenu);
    }

    public List<Menu> getMenusByRestaurant(int codigoRestaurante) throws
SQLException {
        return menuRepo.findByRestaurant(codigoRestaurante);
    }
}
```

```
public void createMenu(Menu menu) throws SQLException {  
  
    if (menu.getRuta_archivo() == null || menu.getRuta_archivo().trim().isEmpty()) {  
  
        throw new IllegalArgumentException("La ruta del archivo es obligatoria");  
  
    }  
  
    if (menu.getTelefono() == null || menu.getTelefono().trim().isEmpty()) {  
  
        throw new IllegalArgumentException("El teléfono es obligatorio");  
  
    }  
  
    menuRepo.save(menu);  
  
}
```

```
public void updateMenu(Menu menu) throws SQLException {  
  
    if (menuRepo.findById(menu.getId_menu()) == null) {  
  
        throw new IllegalArgumentException("No existe un menú con este ID");  
  
    }  
  
    if (menu.getRuta_archivo() == null || menu.getRuta_archivo().trim().isEmpty()) {  
  
        throw new IllegalArgumentException("La ruta del archivo es obligatoria");  
  
    }  
  
    if (menu.getTelefono() == null || menu.getTelefono().trim().isEmpty()) {  
  
        throw new IllegalArgumentException("El teléfono es obligatorio");  
  
    }  
  
    menuRepo.update(menu);  
  
}
```

```
public void deleteMenu(int idMenu) throws SQLException {  
  
    if (menuRepo.findById(idMenu) == null) {  
  
        throw new IllegalArgumentException("No existe un menú con este ID");  
  
    }  
  
    menuRepo.delete(idMenu);  
  
}  
  
}
```

### **Solicitud\_registro servivce**

```
package org.sazonpt.service;  
  
  
import java.sql.SQLException;  
import java.time.LocalDate;  
import java.util.List;  
  
  
import org.sazonpt.model.Solicitud_registro;  
import org.sazonpt.repository.Solicitud_registroRepository;  
  
  
public class Solicitud_registroService {  
  
    public void updateEstado(int idSolicitud, String nuevoEstado) throws SQLException {  
  
        this.solicitudRepo.updateEstado(idSolicitud, nuevoEstado);  
  
    }  
}
```

```
private final Solicitud_registroRepository solicitudRepo;

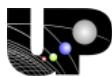
public Solicitud_registroService(Solicitud_registroRepository solicitudRepo) {

    this.solicitudRepo = solicitudRepo;
}

public List<Solicitud_registro> getAllSolicitudes() throws SQLException {
    return solicitudRepo.ListAllSolicitudes();
}

public Solicitud_registro getById(int idSolicitud) throws SQLException {
    return solicitudRepo.FindSolicitudR(idSolicitud);
}

public void createSolicitud(Solicitud_registro solicitud) throws SQLException {
    if (solicitud.getRestaurante() == null || solicitud.getRestaurante().trim().isEmpty()) {
        throw new IllegalArgumentException("El nombre del restaurante es obligatorio");
    }
    if (solicitud.getCorreo() == null || solicitud.getCorreo().trim().isEmpty()) {
        throw new IllegalArgumentException("El correo es obligatorio");
    }
    if (!solicitud.getCorreo().contains("@")) {
        throw new IllegalArgumentException("El correo debe contener un '@'");
    }
}
```



}

```
if (solicitud.getDireccion() == null || solicitud.getDireccion().trim().isEmpty()) {  
    throw new IllegalArgumentException("La dirección es obligatoria");  
}  
  
if (solicitud.getFacebook() == null) solicitud.setFacebook("");  
  
if (solicitud.getInstagram() == null) solicitud.setInstagram("");  
  
if (solicitud getMenu() == null) solicitud.setMenu("");  
  
if (solicitud.getEtiqueta1() == null) solicitud.setEtiqueta1("");  
  
if (solicitud.getEtiqueta2() == null) solicitud.setEtiqueta2("");  
  
if (solicitud.getEtiqueta3() == null) solicitud.setEtiqueta3("");  
  
if (solicitud.getFecha() == null) {  
    solicitud.setFecha(LocalDate.now());  
}  
  
solicitudRepo.AddSolicitudR(solicitud);  
}
```

```
public void updateSolicitud(Solicitud_registro solicitud) throws SQLException {  
    if (solicitudRepo.FindSolicitudR(solicitud.getId_solicitud()) == null) {  
        throw new IllegalArgumentException("No existe una solicitud con este ID");  
    }  
  
    if (solicitud.getRestaurante() == null || solicitud.getRestaurante().trim().isEmpty()) {  
        throw new IllegalArgumentException("El nombre del restaurante es obligatorio");  
    }  
}
```

```
if (solicitud.getCorreo() == null || solicitud.getCorreo().trim().isEmpty()) {  
    throw new IllegalArgumentException("El correo es obligatorio");  
}  
  
if (!solicitud.getCorreo().contains("@")) {  
    throw new IllegalArgumentException("El correo debe contener un '@'");  
}  
  
if (solicitud.getDireccion() == null || solicitud.getDireccion().trim().isEmpty()) {  
    throw new IllegalArgumentException("La dirección es obligatoria");  
}  
  
if (solicitud.getFacebook() == null) solicitud.setFacebook("");  
if (solicitud.getInstagram() == null) solicitud.setInstagram("");  
if (solicitud getMenu() == null) solicitud.setMenu("");  
if (solicitud.getEtiqueta1() == null) solicitud.setEtiqueta1("");  
if (solicitud.getEtiqueta2() == null) solicitud.setEtiqueta2("");  
if (solicitud.getEtiqueta3() == null) solicitud.setEtiqueta3("");  
solicitudRepo.UpdateSolicitud(solicitud);  
}
```

```
public void deleteSolicitud(int idSolicitud) throws SQLException {  
    if (solicitudRepo.FindSolicitudR(idSolicitud) == null) {  
        throw new IllegalArgumentException("No existe una solicitud con este ID");  
    }  
}
```

```
boolean deleted = solicitudRepo.DeleteSolicitudR(idSolicitud);
```

```
if (!deleted) {
```

```
    throw new SQLException("No se pudo eliminar la solicitud");
```

```
}
```

```
}
```

```
public void aprobarSolicitud(int idSolicitud) throws SQLException {
```

```
    if (solicitudRepo.FindSolicitudR(idSolicitud) == null) {
```

```
        throw new IllegalArgumentException("No existe una solicitud con este ID");
```

```
}
```

```
    solicitudRepo.aprobarSolicitud(idSolicitud);
```

```
}
```

```
public void rechazarSolicitud(int idSolicitud) throws SQLException {
```

```
    if (solicitudRepo.FindSolicitudR(idSolicitud) == null) {
```

```
        throw new IllegalArgumentException("No existe una solicitud con este ID");
```

```
}
```

```
    solicitudRepo.rechazarSolicitud(idSolicitud);
```

```
}
```

```
}
```

## Usuario Service

```
package org.sazonpt.service;
```

```
import java.sql.SQLException;  
  
import java.util.List;  
  
  
import org.sazonpt.model.Administrador;  
  
import org.sazonpt.model.Usuario;  
  
import org.sazonpt.repository.AdminRepository;  
  
import org.sazonpt.repository.UsuarioRepository;  
  
  
  
  
public class UsuarioService {  
  
    private final UsuarioRepository userRepo;  
  
  
  
    public UsuarioService(UsuarioRepository userRepo){  
  
        this.userRepo = userRepo;  
  
    }  
  
  
  
    public List<Usuario> getAllUsers() throws SQLException {  
  
        return userRepo.findAll();  
  
    }  
  
  
  
    public Usuario getByIdUser(int idUser) throws SQLException {  
  
        return userRepo.findByIdUser(idUser);  
  
    }  

```

```
public int createUser(Usuario user) throws SQLException {  
  
    if (user.getNombre() == null || user.getCorreo() == null || user.getContrasena() ==  
null) {  
  
        throw new IllegalArgumentException("Nombre, correo y contraseña  
obligatorios");  
  
    }  
  
    if (!user.getCorreo().contains("@")) {  
  
        throw new IllegalArgumentException("El correo debe contener un '@'");  
  
    }  
  
    if (userRepository.findByCorreo(user.getCorreo()) != null) {  
  
        throw new IllegalArgumentException("Ya existe un usuario con este correo");  
  
    }  
  
    int id_usuario = userRepository.save(user);  
  
    if("administrador".equalsIgnoreCase(user.getTipo())){  
  
        AdminRepository adminRepo = new AdminRepository();  
  
        adminRepo.createAdmin(new Administrador(id_usuario));  
  
    }  
}
```

```
        return id_usuario;  
    }  
  
    public void updateUser(Usuario user) throws SQLException {  
        userRepo.UpdateUser(user);  
    }  
  
    public void deleteUser(int idUser) throws SQLException {  
        userRepo.deleteUser(idUser);  
    }  
  
    public Usuario login(String correo, String contrasena, String rol) throws SQLException {  
        return userRepo.findByCorreoAndContrasenaAndRol(correo, contrasena, rol);  
    }  
}
```

## 5.2 Código Fronted

### 5.2.1 HTML

**Encuesta:**

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Encuesta</title>

    <link rel="stylesheet" href="../styles/encuesta.css">

</head>

<body>

    <form id="encuestaForm" class="form">

        <h2>Encuesta</h2>

        <label for="atraccion">¿Qué te atrajo del restaurante?</label>

        <select id="atraccion" name="atraccion">

            <option value="" disabled selected>Selecciona una opción</option>

            <option value="comida">La comida</option>

            <option value="ubicacion">La ubicación</option>

            <option value="recomendacion">Recomendación</option>

            <option value="horario">El horario</option>

            <option value="vista">La vista</option>

        </select>

        <span class="error-message" id="error-atraccion"></span>

    </form>

</body>
```

```
<label for="origen">¿De dónde nos visitas?</label>  
  
<select id="origen" name="origen">  
  
    <option value="" disabled selected>Selecciona una opción</option>  
  
    <option value="nacional">Nacional</option>  
  
    <option value="extranjero">Extranjero</option>  
  
</select>  
  
<span class="error-message" id="error-origen"></span>  
  
<button type="submit" class="button">Enviar</button>  
  
</form>  
  
  
  
<script src="../Scripts/encuesta.js"></script>  
  
</body>  
  
</html>
```

### Estadísticas:

```
<!DOCTYPE html>  
  
<html lang="es">  
  
<head>  
  
    <meta charset="utf-8">  
  
    <meta name="viewport" content="initial-scale=1, width=device-width">  
  
    <link rel="stylesheet" href="../styles/estadisticas.css"/>  
  
    <title>Estadísticas del restaurante</title>  
  
</head>  
  
<body>
```

```
<div class="vista-principal">

    <header class="header-principal">

        <div class="rectangle-parent">

            <div class="logo-blanco-parent">

                <div class="catlogo-de-restaurantes">Restaurantes San Cristóbal</div>

            </div>

            <nav class="acciones-usuario">

                <button class="log-out-parent" type="button">

                    <span class="log-out">Cerrar sesión</span>

                </button>

            </nav>

            <div class="icono-restaurantero">

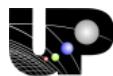
            </div>

        </div>

    </header>

    <main>

        <div class="restaurante-estadisticas-container">
```



```
<button class="btn-back" aria-label="Regresar" onclick="window.history.back()">  
      
</button>  
  
<h1 class="restaurante-title">Estadísticas de su restaurante</h1>  
  
<hr class="restaurante-divider">  
  
<div class="estadisticas-grid">  
  
    <div class="tarjeta estadistica-menu efecto-brillante">  
  
          
  
        <h3>DESCARGAS DE MENÚ</h3>  
  
        <p>Total acumulados<br><strong></strong></p>  
  
        <p class="porcentaje-subida"></p>  
  
          
  
    </div>  
  
  
    <div class="tarjeta estadistica-origen efecto-brillante">  
  
          
  
        <h3>INTERÉS POR ORIGEN</h3>  
  
        <p><span class="locales">● Locales</span></p>  
  
        <p><span class="extranjeros">● Extranjeros</span></p>  
  
        <div class="grafico-pastel">  
            <canvas id="graficoOrigen"></canvas>  
        </div>  
    </div>
```

```
</div>

<div class="tarjeta estadistica-aspectos efecto-brillante">

    <h3>ASPECTOS DESTACADOS</h3>

    <div class="barras-aspectos">

        <canvas id="graficoAspectos"></canvas>

    </div>

    </div>

    </div>

</div>

</main>

<footer class="rectangle-container">

    <div class="footer-contactos-redes">

        <div class="contctanos">Contáctanos</div>

        <div class="contactos-grid">

            <div class="gmail-1-1-parent">

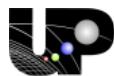
                <div class="sazonpatrimonial@gmail.com">sazonpatrimonial@gmail.com</div>

            </div>

        </div>

    </div>

</footer>
```



```
</div>

<div class="llamada-telefonica-2-parent">

    <div class="sazonpatrimonialgmailcom">+52 961 652 2093</div>

</div>

<div class="logotipo-de-instagram-2-parent">

    <div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

<div class="facebook-1-1-parent">

    <div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

</div>

<div class="footer-logos">

</div>
```

```
</footer>

</div>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script src="../Scripts/estadisticas.js"></script>

<script src="../Scripts/logoutRestaurantero.js"></script>

</body>

</html>
```

## Index

```
<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="utf-8">

<meta name="viewport" content="initial-scale=1, width=device-width">

<link rel="stylesheet" href="../styles/index.css"/>

<title>Sazón Patrimonial</title>

</head>

<body>

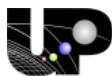
<div class="vista-principal">

<header class="header-principal">

<div class="rectangle-parent">

<div class="logo-blanco-parent">


```



```
<div class="catlogo-de-restaurantes">Restaurantes San Cristóbal</div>

</div>

<nav class="acciones-usuario">

    <button class="registro-usuario-parent" type="button"
    onclick="openRegistroModal()>

        <span class="registrarse">Registrarse</span>

    </button>

    <button class="log-in-parent" type="button" onclick="openLoginModal()>

        <span class="iniciar-sesion">Iniciar sesión</span>

    </button>

</nav>

</div>

</header>

<main>

<section class="calles-con-encanto-scaled-parent">

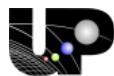
    

    <div class="contenido-superpuesto">

        <div class="descubre-la-magia">

            Descubre la Magia Culinaria de <br>San Cristóbal

        </div>
    </div>
</section>
</main>
```



```
</div>

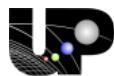
<div class="explora-los-mejores">
    Explora los mejores sabores de esta tierra
</div>

<div class="barra-busqueda">
    
    <input type="text" placeholder="Buscar restaurantes">
    <button class="buscar">Buscar</button>
</div>

</div>

<div class="barra-filtros-horizontal filtros-abajo">

    <div class="filtro-item">
        
        <select class="filtro-select" id="filtroTipoComida">
            <option value="">Tipo de comida</option>
            <option value="Comida Rápida">Comida Rápida</option>
            <option value="Gourmet">Gourmet</option>
            <option value="Vegetariano">Vegetariano</option>
            <option value="Económico">Económico</option>
        </select>
    </div>
</div>
```



```
<div class="filtro-item">

    <select class="filtro-select" id="filtroAmbiente">

        <option value="">Ambiente</option>

        <option value="Familiar">Familiar</option>

        <option value="Pet Friendly">Pet Friendly</option>

        <option value="Terraza">Terraza</option>

    </select>

</div>

<div class="filtro-item">

    <select class="filtro-select" id="filtroServicios">

        <option value="">Servicios</option>

        <option value="Delivery">Delivery</option>

        <option value="WiFi Gratuito">WiFi Gratuito</option>

        <option value="Estacionamiento">Estacionamiento</option>

    </select>

</div>

<button class="btn-filtrar" id="btnLimpiarFiltros">

     Limpiar Filtros

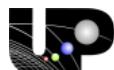
</button>
```

```
</div>

</section>

<div class="contenido-principal">
    <section class="line-parent">
        </section>
    </div>
</main>

<footer class="rectangle-container">
    <div class="footer-contactos-redes">
        <div class="contctanos">Contáctanos</div>
        <div class="contactos-grid">
            <div class="gmail-1-1-parent">
                
                <div
class="sazonpatrimonialgmailcom">sazonpatrimonial@gmail.com</div>
            </div>
            <div class="llamada-telefonica-2-parent">
                
                <div class="sazonpatrimonialgmailcom">+52 961 652 2093</div>
            </div>
        </div>
    </div>
</footer>
```



```
</div>

<div class="logotipo-de-instagram-2-parent">

    <div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

<div class="facebook-1-1-parent">

    <div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

</div>

<div class="footer-logos">

</div>

</footer>

</div>

<div id="registroModal" class="modal-overlay">

    <div class="modal-content">
```

```
<button class="close-modal" onclick="closeRegistroModal()">&times;</button>

<div id="registroContent"></div>

</div>

</div>

<div id="loginModal" class="modal-overlay">

<div class="modal-content">

<button class="close-modal" onclick="closeLoginModal()">&times;</button>

<div id="loginContent"></div>

</div>

</div>

<div id="loginAdminModal" class="modal-overlay">

<div class="modal-content">

<button class="close-modal" onclick="closeLoginAdminModal()">&times;</button>

<div id="loginAdminContent"></div>

</div>

</div>

<div id="loginRestModal" class="modal-overlay">

<div class="modal-content">

<button class="close-modal" onclick="closeLoginRestModal()">&times;</button>

<div id="loginRestContent"></div>
```

```
</div>

</div>

<script src="../Scripts/index.js"></script>

<script src="../Scripts/modals.js"></script>

<script src="../Scripts/vistaRestaurante.js"></script>

</body>

</html>
```

## **Login Admin**

```
<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="../styles/forms.css">

<title>Inicio de sesión de administrador</title>

</head>

<body>

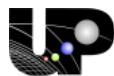
<section class="form">

<form id="loginForm">



<h4>Iniciar sesión</h4>

<label for="correo">Correo electrónico:</label>
```



```
<input class="controls" type="email" name="correo" id="correo"  
placeholder="Ingrese su correo">
```

```
<span class="error-msg" id="error-correo"></span>
```

```
<label for="contrasena">Contraseña:</label>
```

```
<input class="controls" type="password" name="contrasena" id="contrasena"  
placeholder="Ingrese su contraseña">
```

```
<span class="error-msg" id="error-contrasena"></span>
```

```
<input class="button" type="submit" value="Iniciar sesión">
```

```
</form>
```

```
</section>
```

```
<script src="../Scripts/login.js"></script>
```

```
</body>
```

```
</html>
```

## **Login Rest.**

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<link rel="stylesheet" href="../styles/forms.css">
```

```
<title>Inicio de sesión de restaurantero</title>
```

```
</head>

<body>

<section class="form">

    <form id="loginForm">

        <h4>Iniciar sesión</h4>

        <label for="correo">Correo electrónico:</label>

            <input class="controls" type="email" name="correo" id="correo" placeholder="Ingrese su correo">

        <span class="error-msg" id="error-correo"></span>

        <label for="contrasena">Contraseña:</label>

            <input class="controls" type="password" name="contrasena" id="contrasena" placeholder="Ingrese su contraseña">

        <span class="error-msg" id="error-contrasena"></span>

        <input class="button" type="submit" value="Iniciar sesión">

    </form>

</section>

<script src="../Scripts/login.js"></script>

</body>

</html>
```

## LoginSelect

```
<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="../styles/select.css">

<title>Seleccionar inicio de sesión</title>

</head>

<body>

<section class="choose">



<h4>Ingresar como</h4>

<button class="button admin-btn" type="button">



Administrador

</button>

<button class="button rest-btn" type="button">



Restaurantero

</button>

<a href="#" class="register-link">Registrarse</a>

</section>
```

</body>

</html>

## registro

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="../styles/forms.css">

<title>Registro de usuario</title>

</head>

<body>

<section class="form">

<form id="registroForm">



<h4>Registro de usuario</h4>

<label for="nombre">Nombre de usuario:</label>

<input class="controls" type="text" name="nombre" id="nombre" placeholder="Ingrese su nombre">

<span class="error-msg" id="error-nombre"></span>

<label for="correo">Correo electrónico:</label>

<input class="controls" type="email" name="correo" id="correo" placeholder="Ingrese su correo">

```
<span class="error-msg" id="error-correo"></span>
```

```
<label for="contraseña">Contraseña:</label>
```

```
    <input class="controls" type="password" name="contraseña" id="contraseña"  
placeholder="Ingrese su contraseña">
```

```
    <span class="error-msg" id="error-contraseña"></span>
```

```
<label for="confirmar">Confirmar contraseña:</label>
```

```
    <input class="controls" type="password" name="confirmar" id="confirmar"  
placeholder="Confirmar contraseña">
```

```
    <span class="error-msg" id="error-confirmar"></span>
```

```
    <input class="botons" type="submit" value="Registrarse" >
```

```
    </form>
```

```
</section>
```

```
<script src="../Scripts/registro.js"></script>
```

```
</body>
```

```
</html>
```

## solicitudRestaurante

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../styles/solicitud.css">

<title>Registro de usuario</title>

</head>

<body>

<div class="vista-principal">

<header class="header-principal">

<div class="rectangle-parent">

<div class="logo-blanco-parent">



<div class="catlogo-de-restaurantes">Restaurantes San Cristóbal</div>

</div>

</div>

</header>

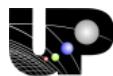
<main>

<div class="registro-container">

<button class="btn-back" aria-label="Regresar" onclick="window.history.back()">



</button>
```



```
<section class="form-section">

    <form id="registroForm">

        <h4>Solicitud de validación</h4>

        <p class="sub-text">
            Proporciona la siguiente información para verificar tu restaurante
        </p>

        <label for="restaurante">Restaurante</label>

        <input class="controls" type="text" id="restaurante" placeholder="Ingrese el
nombre del restaurante">

        <span class="error-msg" id="error-nombre-restaurante"></span>

        <label for="proprietario">Propietario</label>

        <input class="controls" type="text" id="proprietario" placeholder="Ingrese el
nombre del propietario">

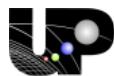
        <span class="error-msg" id="error-nombre-proprietario"></span>

        <label for="correo">Correo electrónico</label>

        <input class="controls" type="email" id="correo" placeholder="Ingrese su
correo">

        <span class="error-msg" id="error-correo"></span>

        <label for="numero">Número de celular</label>
```



<input class="controls" type="tel" id="numero" placeholder="Ingrese su número de celular">

<span class="error-msg" id="error-numero"></span>

<label for="facebook">Facebook</label>

<input class="controls" type="text" id="facebook" placeholder="Ingrese su Facebook">

<span class="error-msg" id="error-facebook"></span>

<label for="instagram">Instagram</label>

<input class="controls" type="text" id="instagram" placeholder="Ingrese su Instagram">

<span class="error-msg" id="error-instagram"></span>

<label for="direccion">Dirección</label>

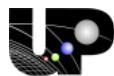
<input class="controls" type="text" id="direccion" placeholder="Ingrese la dirección del restaurante">

<span class="error-msg" id="error-direccion"></span>

<label for="horario">Horarios de cierre y apertura</label>

<input class="controls" type="text" id="horario" placeholder="Ingrese el horario">

<span class="error-msg" id="error-horario"></span>



```
<label>Imágenes (JPG o PNG)</label>

<div class="imagenes-group">

    <div class="input-file-group">

        <label class="input-file">

            Imagen principal

            <input type="file" name="imagen1" hidden accept="image/png,
image/jpeg">

        </label>

        <span class="error-msg" id="error-imagen1"></span>

    </div>

    <div class="input-file-group">

        <label class="input-file">

            Imagen secundaria

            <input type="file" name="imagen2" hidden accept="image/png,
image/jpeg">

        </label>

        <span class="error-msg" id="error-imagen2"></span>

    </div>

    <div class="input-file-group">

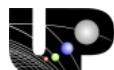
        <label class="input-file">

            Imagen de platillo

            <input type="file" name="imagen3" hidden accept="image/png,
image/jpeg">

        </label>

    </div>
```



<span class="error-msg" id="error-imagen3"></span>

</div>

</div>

<label>Archivos (PDF)</label>

<div class="imagenes-group">

<div class="input-file-group">

<label class="input-file">

Comprobante de domicilio

<input type="file" name="comprobante" hidden

accept="application/pdf">

</label>

<span class="error-msg" id="error-comprobante"></span>

</div>

<div class="input-file-group">

<label class="input-file">

Menú del restaurante

<input type="file" name="menu" hidden accept="application/pdf">

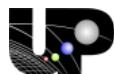
</label>

<span class="error-msg" id="error-menu"></span>

</div>

</div>

<input class="button" type="submit" value="Enviar solicitud">



```
</form>

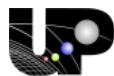
</section>

<section class="image-section">
    
</section>

</div>

</main>

<footer class="rectangle-container">
    <div class="footer-contactos-redes">
        <div class="contctanos">Contáctanos</div>
        <div class="contactos-grid">
            <div class="gmail-1-1-parent">
                
                <div
class="sazonpatrimonialgmailcom">sazonpatrimonial@gmail.com</div>
            </div>
            <div class="llamada-telefonica-2-parent">
                
                <div class="sazonpatrimonialgmailcom">+52 961 652 2093</div>
            </div>
        </div>
    </div>
</footer>
```



```
<div class="logotipo-de-instagram-2-parent">

    <div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

<div class="facebook-1-1-parent">

    <div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

</div>

<div class="footer-logos">

</div>

</footer>

</div>

<div id="modal" class="modal">

<div class="modal-contenido">

    <span id="cerrarModal" class="cerrar">&times;</span>
```

<p>¡Su solicitud ha sido enviada con éxito!<br>La información será validada muy pronto.<br>:)</p>

```
</div>

</div>

<script src="../Scripts/modalSolicitud.js"></script>

</body>

</html>
```

### vistaEdicionRest

```
<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="initial-scale=1, width=device-width">

    <link rel="stylesheet" href="../styles/vistaEdicionRest.css"/>

    <link rel="stylesheet" href="../styles/modal-membresia.css"/>

    <title>Edición de restaurante</title>

</head>

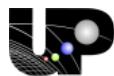
<body>

    <div class="vista-principal">

        <header class="header-principal">

            <div class="rectangle-parent">

                <div class="logo-blanco-parent">
```



```
  
  
<div class="catlogo-de-restaurantes">Restaurantes San Cristóbal</div>  
  
</div>  
  
<nav class="acciones-usuario">  
  
<button class="log-out-parent" type="button">  
  
  
  
<span class="log-out">Cerrar sesión</span>  
  
</button>  
  
</nav>  
  
<div class="icono-restaurantero">  
  
  
  
</div>  
  
</div>  
  
</header>  
  
  
  
  
<main>  
  
<div class="restaurante-edicion-container">  
  
<button class="btn-back" aria-label="Regresar" onclick="window.history.back()">  
  
  
  
</button>  
  
<h1 class="restaurante-nombre" id="restauranteNombre">Bienvenido</h1>
```

---

```
<section class="formulario-edicion">

    <div class="linea-interna">

        <div class="membresia-status">

            <div id="membresiaMensaje" class="membresia-mensaje"></div>

            <button id="btnEstadisticas" class="btn-estadisticas"
                onclick="window.location.href='estadisticas.html'">

                Estadísticas

            </button>

        </div>

    </div>

    <div class="linea-interna">

        <label class="campo-titulo">Galería de Imágenes</label>

        <div class="galeria-imagenes">

            <label class="imagen-slot">

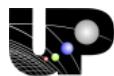
                <input type="file" id="imgGaleria1Input" name="Imagen1"
                    accept="image/png, image/jpeg" style="display:none;">

                <img id="imgGaleria1"
                    onclick="document.getElementById('imgGaleria1Input').click();">

            </label>

            <label class="imagen-slot">

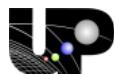

```



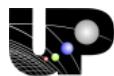
```
<input type="file" id="imgGaleria2Input" name="Imagen2"  
accept="image/png, image/jpeg" style="display:none;">  
  
    <img id="imgGaleria2"  
onclick="document.getElementById('imgGaleria2Input').click();">  
  
    </label>  
  
    <label class="Imagen-slot">  
  
        <input type="file" id="imgGaleria3Input" name="Imagen3"  
accept="image/png, image/jpeg" style="display:none;">  
  
        <img id="imgGaleria3"  
onclick="document.getElementById('imgGaleria3Input').click();">  
  
    </label>  
  
    </div>  
  
    </div>  
  
  


Menú en PDF:</label>  
  
<br/>  
  
Seleccionar archivo PDF</span>  
  
id="menuInput"/>  
  
</label>  
  
</div>


```



```
<div class="area ubicacion-area">  
  
    <label class="campo-titulo">Ubicación:</label>  
  
    <input id="direccionInput" type="text" placeholder="Ingrese la  
dirección">  
  
    </div>  
  
  
  
    <div class="area horarios-area">  
  
        <label class="campo-titulo">Horarios:</label>  
  
        <div class="horarios-inputs">  
  
            <input id="horarioLVInput" type="text" placeholder="Ingrese el  
horario">  
  
            </div>  
  
        </div>  
  
    </div>  
  
  
  
    <div class="linea-interna contacto-etiquetas">  
  
        <div class="restaurante-info-flex">  
  
            <div class="contactos">  
  
                <label class="campo-titulo">Contactos:</label>  
  
                <div class="input-con-icono">  
  
                      
  
                </div>  
  
            </div>  
  
        </div>  
  
    </div>
```



```
<input id="telefonoInput" type="text" placeholder="Ingrese su  
número celular">
```

```
</div>
```

```
<div class="input-con-icono">
```

```

```

```
<input id="facebookInput" type="text" placeholder="Ingrese su  
facebook">
```

```
</div>
```

```
<div class="input-con-icono">
```

```

```

```
<input id="instagramInput" type="text" placeholder="Ingrese su  
instagram">
```

```
</div>
```

```
</div>
```

```
<div class="etiquetas">
```

```
<label class="campo-titulo">Etiquetas:</label>
```

```
<div class="input-con-icono">
```

```

```

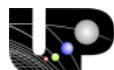
```
<select class="filtro-select">
```

```
<option value="" disabled selected>Seleccionar</option>
```

```
<option value=""></option>
```

```
<option value=""></option>
```

```
<option value=""></option>
```



```
</select>

</div>

<div class="input-con-icono">

    <select class="filtro-select">

        <option value="" disabled selected>Seleccionar</option>

        <option value=""></option>

        <option value=""></option>

        <option value=""></option>

    </select>

</div>

<div class="input-con-icono">

    <select class="filtro-select">

        <option value="" disabled selected>Seleccionar</option>

        <option value=""></option>

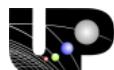
        <option value=""></option>

        <option value="ato"></option>

    </select>

</div>

</div>
```



```
</div>

</section>

<button class="btn-aplicar">

    Aplicar Cambios

</button>

</main>

<footer class="rectangle-container">

    <div class="footer-contactos-redes">

        <div class="contctanos">Contáctanos</div>

        <div class="contactos-grid">

            <div class="gmail-1-1-parent">

                <div
class="sazonpatrimonialgmailcom">sazonpatrimonial@gmail.com</div>

            </div>

            <div class="llamada-telefonica-2-parent">

                <div class="sazonpatrimonialgmailcom">+52 961 652 2093</div>

            </div>

            <div class="logotipo-de-instagram-2-parent">
```



<div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

<div class="facebook-1-1-parent">



<div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

</div>

</div>

<div class="footer-logos">







</div>

</footer>

</div>

<div id="membresiaModal" class="modal-overlay">

<div class="modal-content">

<button class="close-modal" onclick="closeMembresiaModal()">&times;</button>

<div id="membresiaContent"></div>

```
</div>

</div>

<div id="modalAviso" class="modal">

    <div class="modal-contenido">

        <span id="cerrarAviso" class="cerrar">&times;</span>

        <p>Solicitud enviada con éxito. Validaremos tu información y nos comunicaremos pronto para activar tu membresía :)</p>

    </div>

</div>

<script src="../Scripts/modals.js"></script>

<script src="../Scripts/logoutRestaurantero.js"></script>

<script src="../Scripts/vistaEdicionRest.js"></script>

</body>

</html>

vistaPrincipalAdmin

<!DOCTYPE html>

<html lang="es">

    <head>

        <meta charset="utf-8">

        <meta name="viewport" content="initial-scale=1, width=device-width">

        <link rel="stylesheet" href="../styles/vistaPrincipalAdmin.css"/>
```

<title>Edición de restaurante con membresía</title>

</head>

<body>

<div class="vista-principal">

<header class="header-principal">

<div class="rectangle-parent">

<div class="logo-blanco-parent">



<div class="catlogo-de-restaurantes">Restaurantes San Cristóbal</div>

</div>

<nav class="acciones-usuario">

<button class="log-out-parent" type="button">



<span class="log-out">Cerrar sesión</span>

</button>

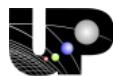
</nav>

</div>

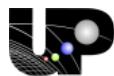
</header>

<div class="admin-panel">

<aside class="sidebar">



```
<button class="sidebar-item" type="button" data-vista="restaurantes">  
      
    <span class="restaurantes">Restaurantes</span>  
</button>  
  
<button class="sidebar-item" type="button" data-vista="solicitudes">  
      
    <span class="usuarios">Usuarios</span>  
</button>  
</aside>  
  
<main class="contenido">  
    <section id="vista-restaurantes" class="vista-admin">  
        <h2>Administrar Restaurantes</h2>  
        <table class="tabla-admin">  
            <thead>  
                <tr>  
                    <th>Nombre</th>  
                    <th>Ubicación</th>  
                    <th>Acciones</th>  
                </tr>  
            </thead>  
            <tbody>  
                <tr>
```



```
<td>Nombre 1</td>

<td>Ubicación 1</td>

<td>

    <button class="btn-eliminar" title="Eliminar"></button>

</td>

</tr>

</tbody>

</table>

</section>

<section id="vista-solicitudes" class="vista-admin" style="display:none;">

    <h2>Administrar Solicitudes de Usuarios</h2>

    <table class="tabla-admin">

        <thead>

            <tr>

                <th>Restaurante</th>

                <th>Propietario</th>

                <th>Correo</th>

                <th>Celular</th>

                <th>Dirección</th>

                <th>Horario</th>

                <th>Imágenes</th>

            </tr>

        <tbody>

            <tr>

                <td>Nombre 1</td>

                <td>Ubicación 1</td>

                <td>

                    <button class="btn-eliminar" title="Eliminar"></button>

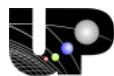
                </td>

            </tr>

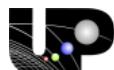
        </tbody>

    </table>

</section>
```



Comprobante	Acciones
Restaurante 1	<button class="btn-ver-imagenes" title="Ver imágenes">&lt;img src="../images/imagen.png" alt="Ver"/&gt;</button>
Propietario 1	<button class="btn-ver-comprobante" title="Ver comprobante">&lt;img src="../images/comprobante.png" alt="Ver"/&gt;</button>
correo1@ejemplo.com	
1234567890	
Dirección 1	
Horario 1	
	<button class="btn-aceptar" title="Aceptar">&lt;img src="../images/aceptar.png" alt="Aceptar"/&gt;</button>



```
<button class="btn-rechazar" title="Rechazar"></button>
```

```
</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

```
</section>
```

```
<section id="vista-membresias" class="vista-admin" style="display:none;">
```

```
<h2>Solicitudes de Membresía</h2>
```

```
<table class="tabla-admin">
```

```
<thead>
```

```
<tr>
```

```
<th>Restaurante</th>
```

```
<th>Propietario</th>
```

```
<th>Correo</th>
```

```
<th>Fecha de solicitud</th>
```

```
<th>Estado</th>
```

```
<th>Acciones</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<!-- Aquí se llenan dinámicamente las solicitudes -->
```

```
</tbody>

</table>

</section>

</main>

</div>

<footer class="rectangle-container">

<div class="footer-contactos-redes">

<div class="contctanos">Contáctanos</div>

<div class="contactos-grid">

<div class="gmail-1-1-parent">



<div
class="sazonpatrimonialgmailcom">sazonpatrimonial@gmail.com</div>

</div>

<div class="llamada-telefonica-2-parent">

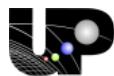


<div class="sazonpatrimonialgmailcom">+52 961 652 2093</div>

</div>

<div class="logotipo-de-instagram-2-parent">


```



```
<div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

<div class="facebook-1-1-parent">

    <div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

</div>

</div>

<div class="footer-logos">

</div>

</footer>

</div>

<script src="../Scripts/vistaPrincipalAdmin.js"></script>

<script src="../Scripts/logoutRestaurantero.js"></script>

</body>

</html>

vistaPrincipalRestaurantero.html
```

```
<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="initial-scale=1, width=device-width">

    <link rel="stylesheet" href="../styles/vistaPrincipalRestaurantero.css"/>

    <title>Vista principal de restaurantero</title>

</head>

<body>

    <div class="vista-principal">

        <header class="header-principal">

            <div class="rectangle-parent">

                <div class="logo-blanco-parent">

                    <div class="catlogo-de-restaurantes">Restaurantes San Cristóbal</div>

                </div>

            <nav class="acciones-usuario">

                <button class="mi-restaurante-parent" type="button">

                    <span class="miRestaurante">Mi restaurante</span>

                </button>

            </nav>

        </div>

    </header>

    <div class="contenedor-principal">

        <div class="cabecera-principal">

            <div class="cabecera-izquierda">

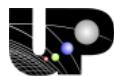
                <img alt="Logo de la Universidad Politécnica de Chiapas" data-bbox="114 89 185 115" />

                <div>
                    <h1>Universidad Politécnica de Chiapas</h1>
                    <h2>Sistemas de Gestión de Restaurantes</h2>
                </div>
            </div>

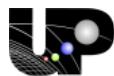
            <div class="cabecera-derecha">
                <img alt="Logo de TECNOLOGÍAS DE LA INFORMACIÓN E INNOVACIÓN DIGITAL (ITIID)" data-bbox="739 89 876 115" />
            </div>
        </div>

        <div class="menu-principal">
            <ul>
                <li>Inicio</li>
                <li>Reservaciones</li>
                <li>Reportes</li>
                <li>Ayuda</li>
            </ul>
        </div>

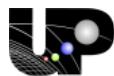
        <div class="contenido-principal">
            <div>
                <h3>Reservaciones</h3>
                <p>A continuación se presentan las reservaciones realizadas en el sistema.</p>
                <table border="1">
                    <thead>
                        <tr>
                            <th>Número de Reserva</th>
                            <th>Nombre del Cliente</th>
                            <th>Fecha de Reserva</th>
                            <th>Hora de Reserva</th>
                            <th>Estado</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>1</td>
                            <td>Juan Pérez</td>
                            <td>2023-10-15</td>
                            <td>19:00</td>
                            <td>Pendiente</td>
                        </tr>
                        <tr>
                            <td>2</td>
                            <td>Ana Gómez</td>
                            <td>2023-10-16</td>
                            <td>21:00</td>
                            <td>Pendiente</td>
                        </tr>
                        <tr>
                            <td>3</td>
                            <td>Pedro Martínez</td>
                            <td>2023-10-17</td>
                            <td>20:00</td>
                            <td>Pendiente</td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</body>
```



```
<button class="log-out-parent" type="button">  
      
    <span class="log-out">Cerrar sesión</span>  
</button>  
</nav>  
  
<div class="icono-restaurantero">  
      
</div>  
</div>  
</header>  
  
  
  
<main>  
    <section class="calles-con-encanto-scaled-parent">  
          
        <div class="contenido-superpuesto">  
            <div class="descubre-la-magia">  
                Descubre la Magia Culinaria de <br>San Cristóbal  
            </div>  
            <div class="explora-los-mejores">  
                Explora los mejores sabores de esta tierra  
            </div>  
        </div>  
    </section>  
</main>
```



```
<div class="nombre-restaurante">  
    Restaurante: <span id="nombre-restaurante">La mansión</span>  
</div>  
  
<div class="barra-busqueda">  
      
    <input type="text" placeholder="Buscar restaurantes">  
    <button class="buscar">Buscar</button>  
</div>  
</div>  
  
<div class="barra-filtros-horizontal filtros-abajo">  
    <div class="filtro-item">  
          
        <select class="filtro-select">  
            <option disabled selected>Precio</option>  
            <option value="bajo">Bajo</option>  
            <option value="medio">Medio</option>  
            <option value="alto">Alto</option>  
        </select>  
    </div>  
  
<div class="filtro-item">  
      
    <select class="filtro-select">
```



```
<option disabled selected>Horario</option>

<option value="manana">Mañana</option>

<option value="tarde">Tarde</option>

<option value="noche">Noche</option>

</select>

</div>

<div class="filtro-item">

    <select class="filtro-select">

        <option disabled selected>Tipo de comida</option>

        <option value="mexicana">Mexicana</option>

        <option value="cafe">Café</option>

        <option value="internacional">Internacional</option>

    </select>

</div>

<div class="filtro-item">

    <select class="filtro-select">

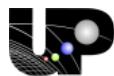
        <option disabled selected>Ubicación</option>

        <option value="centro">Centro</option>

        <option value="norte">Norte</option>

    </select>

</div>
```



```
<option value="sur">Sur</option>

</select>

</div>

<button class="btn-filtrar">

     Filtrar

</button>

</div>

</section>

<div class="contenido-principal">

    <section class="line-parent">

        <div class="line-item">
            
            <div>Nombre Apellido</div>
            <div>correo@upc.edu.mx</div>
        </div>
        <div class="line-item">
            
            <div>Nombre Apellido</div>
            <div>correo@upc.edu.mx</div>
        </div>
    </section>

    <div class="line-item">
        
        <div>Nombre Apellido</div>
        <div>correo@upc.edu.mx</div>
    </div>
</div>

</main>

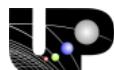
<footer class="rectangle-container">

    <div class="footer-contactos-redes">

        <div class="contctanos">Contáctanos</div>

        <div class="contactos-grid">

            <div class="gmail-1-1-parent">
```





<div  
class="sazonpatrimonialgmailcom">sazonpatrimonial@gmail.com</div>

</div>

<div class="llamada-telefonica-2-parent">



<div class="sazonpatrimonialgmailcom">+52 961 652 2093</div>

</div>

<div class="logotipo-de-instagram-2-parent">



<div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

<div class="facebook-1-1-parent">



<div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

</div>

<div class="footer-logos">

```
  
  
  
  
  
  
</div>  
  
</footer>  
  
</div>  
  
<script src="../Scripts/index.js"></script>  
  
<script src="../Scripts/vistaRestaurante.js"></script>  
  
<script src="../Scripts/vistaPrincipalRestaurantero.js"></script>  
  
<script src="../Scripts/logoutRestaurantero.js"></script>  
  
</body>  
  
</html>
```

## vistaRestaurante

```
<!DOCTYPE html>  
  
<html lang="es">  
  
<head>  
  
<meta charset="utf-8">  
  
<meta name="viewport" content="initial-scale=1, width=device-width">  
  
<link rel="stylesheet" href="../styles/vistaRestaurante.css"/>  
  
<link rel="stylesheet" href="../styles/modal-encuesta.css">  
  
<title>Sazón Patrimonial</title>  
  
</head>
```

```
<body>

<div class="vista-principal">

<header class="header-principal">

<div class="rectangle-parent">

<div class="logo-blanco-parent">



<div class="catlogo-de-restaurantes">Restaurantes San Cristóbal</div>

</div>

</div>

</header>

<main>

<div class="restaurante-vista-container">

<button class="btn-back" aria-label="Regresar" onclick="window.history.back()">



</button>

<h1 class="restaurante-nombre"></h1>

<hr class="restaurante-divider">

<div class="restaurante-galeria">

<img class="galeria-principal" src="" alt="Foto principal restaurante">

<div class="galeria-secundaria"></div>

</div>
```

```
<div class="restaurante-info-flex">

    <div class="restaurante-info-box restaurante-caracteristicas">

        <div class="caracteristicas-lista"></div>

    </div>

    <div class="restaurante-info-box restaurante-horarios"></div>

    <div class="restaurante-info-box restaurante-contactos"></div>

    <div class="restaurante-info-box restaurante-ubicacion"></div>

</div>

<button class="btn-menu" onclick="openEncuestaModal()">

     Menú

</button>

</div>

</main>

<footer class="rectangle-container">

    <div class="footer-contactos-redes">

        <div class="contctanos">Contáctanos</div>

        <div class="contactos-grid">

            <div class="gmail-1-1-parent">

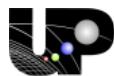
                <div class="sazonpatrimonial@gmail.com">sazonpatrimonial@gmail.com</div>

            </div>

        </div>

    </div>

</footer>
```



</div>

<div class="llamada-telefonica-2-parent">



<div class="sazonpatrimonialgmailcom">+52 961 652 2093</div>

</div>

<div class="logotipo-de-instagram-2-parent">



<div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

<div class="facebook-1-1-parent">



<div class="sazonpatrimonialgmailcom">@sazonpatrimonial</div>

</div>

</div>

<div class="footer-logos">







</div>

```
</footer>

</div>

<div id="encuestaModal" class="modal-overlay">

    <div class="modal-content">

        <button class="close-modal" onclick="closeEncuestaModal()">&times;</button>

        <div id="encuestaContent"></div>

    </div>

</div>

<script src="../Scripts/modals.js"></script>

<script src="../Scripts/encuesta.js"></script>

<script src="../Scripts/vistaRestaurante.js"></script>

</body>

</html>
```

## 5.2.2 CSS

### encuesta

```
.form {

    width: 350px;

    background: #fff;

    padding: 30px 24px;
```

```
margin: auto;  
border-radius: 8px;  
font-family: 'rubik', sans-serif;  
color: #912F2F;  
box-shadow: 7px 7px 5px #B64040;  
display: flex;  
flex-direction: column;  
align-items: center;  
}  
  
}
```

```
.form img.logo {  
display: block;  
margin: 0 auto 12px auto;  
}
```

```
.form h4 {  
text-align: left;  
font-size: 1.2rem;  
margin: 18px 0 6px 0;  
font-weight: 700;  
width: 100%;  
}  
  
}
```

```
.select-container {  
    position: relative;  
    width: 100%;  
    margin-top: 10px;  
    margin-bottom: 20px;  
}  
  
/*
```

```
.selec-filtro {  
    width: 100%;  
    padding: 10px 10px 10px 12px;  
    border: 1px solid #B64040;  
    border-radius: 20px;  
    font-size: 1rem;  
    color: #B64040;  
    background: #fff;  
    appearance: none;  
    outline: none;  
    margin-bottom: 0;  
    cursor: pointer;  
}  
  
/*
```

```
.keyboard-arrow-down-icon {  
    position: absolute;
```

right: 14px;  
top: 50%;  
transform: translateY(-50%);  
pointer-events: none;  
width: 18px;  
height: 18px;  
}

.button {  
width: 100%;  
background: #6A1E1E;  
border: none;  
padding: 12px;  
color: #fff;  
margin: 18px 0 0 0;  
font-size: 1rem;  
border-radius: 8px;  
cursor: pointer;  
font-weight: 600;  
letter-spacing: 1px;  
}

.button:hover, .button:focus {

```
background: #B64040;  
transform: translateY(-2px) scale(1.03);  
transition: background 0.2s, transform 0.2s;  
}  
  
}
```

```
.selec-filtro.error {  
  
border: 2px solid #e74c3c;  
  
background-color: #fdf2f2;  
  
}
```

```
.error-message {  
  
position: absolute;  
  
left: 0;  
  
bottom: -18px;  
  
color: #e74c3c;  
  
font-size: 12px;  
  
padding: 0 6px;  
  
display: block;  
  
z-index: 2;
```

```
pointer-events: none;  
  
width: max-content;  
  
max-width: 90%;  
  
margin: 0;
```

}

## estadísticas

```
html, body {  
    width: 100%;  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
    overflow-x: hidden;  
}
```

```
*, *::before, *::after {  
    box-sizing: inherit;  
}
```

```
.vista-principal {  
    width: 100%;  
}
```

```
.header-principal {  
    position: fixed;  
    top: 0;  
    z-index: 1000;  
    width: 100%;
```

```
background: #6b1e1e;  
color: #fff;  
box-shadow: 0 2px 8px rgba(0,0,0,0.04);  
}  
  
}
```

```
.rectangle-parent {  
    display: flex;  
    align-items: center;  
    justify-content: flex-start;  
    height: 100px;  
    padding-left: 40px;  
    padding-right: 40px;  
    gap: 32px;  
}
```

```
.logo-blanco-parent {  
    display: flex;  
    align-items: center;  
    gap: 10px;  
}
```

```
.logo-blanco {  
    height: 80px;
```

width: auto;

}

.icono-aumento {

width: 100px;

height: 100px;

margin-top: 10px;

}

.catalogo-de-restaurantes {

font-size: 1.6rem;

font-weight: 400;

color: #fff;

font-family: 'Montserrat', sans-serif;

max-width: 200px;

}

.acciones-usuario {

margin-left: auto;

display: flex;

gap: 18px;

}

```
.log-out-parent {  
    display: flex;  
    align-items: center;  
    gap: 8px;  
    background: #6A1E1E;  
    border: none;  
    border-radius: 50px;  
    color: #fff;  
    padding: 12px 22px;  
    font-size: 1rem;  
    font-family: 'Montserrat', sans-serif;  
    cursor: pointer;  
    transition: background 0.2s;  
}  
  
}
```

```
.log-out-parent:hover {  
    background: #912f2f;  
}
```

```
.log-out-parent img {  
    height: 22px;  
    width: 22px;  
}
```

```
.icono-restaurantero {
```

```
    display: flex;
```

```
    align-items: center;
```

```
}
```

```
.icono-restaurantero img {
```

```
    height: 48px;
```

```
    width: 48px;
```

```
    border-radius: 50%;
```

```
}
```

```
.restaurante-estadisticas-container {
```

```
    max-width: 90%;
```

```
    margin: 140px auto 40px auto;
```

```
    background: #fff;
```

```
    border-radius: 18px;
```

```
    padding: 32px 36px 36px 36px;
```

```
    font-family: 'Montserrat', sans-serif;
```

```
    position: relative;
```

```
}
```

```
.icon-32 {
```

```
width: 40px;  
  
height: 40px;  
  
}  
  
.icon-middle {  
  
vertical-align: middle;  
  
}
```

```
.btn-back {  
  
background: none;  
  
width: 10px;  
  
border: none;  
  
position: absolute;  
  
top: 24px;  
  
left: 24px;  
  
cursor: pointer;  
  
padding: 0;  
  
}
```

```
.restaurante-title {  
  
text-align: center;  
  
font-size: 3rem;  
  
font-weight: 700;  
  
color: #912f2f;
```

```
margin: 0 0 8px 0;  
letter-spacing: 1px;  
}
```

```
.restaurante-divider {  
border: none;  
border-top: 4px solid #912f2f;  
width: 80%;  
margin: 0 auto 24px auto;  
border-radius: 2px;  
}
```

```
.estadisticas-grid {  
display: flex;  
gap: 36px;  
justify-content: center;  
flex-wrap: wrap;  
margin-top: 40px;  
}
```

```
.tarjeta {  
border: 2px solid #912f2f;  
border-radius: 12px;
```

```
padding: 20px;  
width: 350px;  
font-family: 'Montserrat', sans-serif;  
text-align: center;  
color: #912f2f;  
background: #fff;  
box-shadow: 0 2px 4px rgba(0,0,0,0.05);  
}  
  
}
```

```
.icono-tarjeta {  
width: 40px;  
margin-bottom: 12px;  
}
```

```
.tarjeta h3 {  
font-size: 1.1rem;  
margin-bottom: 16px;  
}
```

```
.porcentaje-subida {  
font-size: 1.5rem;  
font-weight: bold;  
color: #c04242;
```

```
margin-top: 10px;  
}  
  
}
```

```
.locales {  
color: #6b1e1e;  
font-weight: bold;  
}
```

```
.extranjeros {  
color: #e07878;  
font-weight: bold;  
}
```

```
.grafico-pastel img {  
width: 120px;  
margin-top: 10px;  
}
```

```
.barras-aspectos {  
display: flex;  
flex-direction: column;  
align-items: flex-start;  
gap: 10px;
```

margin-top: 10px;

}

.barra {

width: 100%;

padding: 6px 12px;

color: #fff;

border-radius: 6px;

font-size: 0.9rem;

}

.barra.ambiente {

background-color: #6b1e1e;

}

.barra.comida {

background-color: #c06060;

}

.barra.higiene {

background-color: #efcfef;

color: #6b1e1e;

}

```
.efecto-brillante {  
    transition: transform 0.3s ease;  
}  
  
}
```

```
.efecto-brillante:hover {  
    transform: scale(1.05);  
    box-shadow: 0 8px 16px rgba(51, 34, 34, 0.2);  
}  
  
}
```

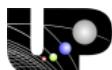
```
.rectangle-container {  
    width: 100%;  
    background: #912f2f;  
    color: #fff;  
    display: flex;  
    flex-wrap: wrap;  
    align-items: flex-start;  
    justify-content: space-between;  
    padding: 32px 40px 20px 40px;  
    box-sizing: border-box;  
    gap: 32px;  
    font-family: 'Montserrat', sans-serif;  
}  
  
}
```

```
.contctanos {  
    font-size: 1.2rem;  
    font-weight: 600;  
    margin-bottom: 16px;  
    grid-column: 1 / -1;  
}  
  
}
```

```
.contactos-grid {  
    display: grid;  
    grid-template-columns: repeat(2, 1fr);  
    gap: 18px 40px;  
}  
  
}
```

```
.gmail-1-1-parent,  
.llamada-telefonica-2-parent,  
.logotipo-de-instagram-2-parent,  
.facebook-1-1-parent {
```

```
    display: flex;  
    align-items: center;  
    gap: 10px;  
    font-size: 1rem;
```



Universidad Politécnica  
de Chiapas

}



.llamada-telefonica-2-icon {

width: 26px;

height: 26px;

object-fit: contain;

}

.sazonpatrimonialgmailcom {

font-size: 1rem;

color: #fff;

}

.footer-logos {

display: flex;

align-items: center;

gap: 18px;

}

.logo-blanco-2,

.logo-equipo,

.logo-universidad {

height: 120px;

```
width: auto;  
object-fit: contain;  
border-radius: 8px;  
}  
  
  
  
  

```

```
#graficoOrigen {  
width: 180px;  
height: 180px;  
max-width: 180px;  
max-height: 180px;  
display: block;  
margin: 0 auto;  
}
```

## forms

```
* {  
margin:0;  
padding: 0;  
box-sizing: border-box;  
}
```

```
body {  
height: 100vh;  
display: flex;
```

```
align-items: center;  
justify-content: center;  
background: #f9f9f9;  
}  
  
}
```

```
.form{  
width:400px;  
background: #ffffff;  
padding : 30px;  
margin: auto;  
border-radius: 4px;  
font-family: 'rubik', sans-serif;  
color: #912F2F;  
box-shadow: 7px 7px 5px #B64040;  
}  
  
}
```

```
.form h4{  
text-align: center;  
font-size: 30px;  
margin: 0 0 20px 0;  
}  
  
}
```

```
.controls {
```

```
width: 100%;  
  
background: #ffffff;  
  
padding: 10px;  
  
border-radius: 8px;  
  
margin-bottom: 2px;  
  
border: 1.5px solid #B64040;  
  
font-family: 'rubik', sans-serif;  
  
}
```

```
.button{  
  
width: 100%;  
  
background: #6A1E1E;  
  
border: none;  
  
padding: 12px;  
  
color: #ffffff;  
  
margin: 16px 0;  
  
font-size: 16px;  
  
border-radius: 8px;  
  
cursor: pointer;  
  
}
```

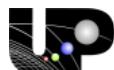
```
.button:hover, .button:focus {  
  
background: #B64040;  
  
transform: translateY(-2px) scale(1.03);
```

transition: background 0.2s, transform 0.2s;  
}  
  
}

.botons{  
  
width: 100%;  
  
background: #6A1E1E;  
  
border: none;  
  
padding: 12px;  
  
color: #ffffff;  
  
margin: 16px 0;  
  
font-size: 16px;  
  
border-radius: 8px;  
  
cursor: pointer;  
  
}

.botons:hover, .botons:focus {  
  
background: #B64040;  
  
transform: translateY(-2px) scale(1.03);  
  
transition: background 0.2s, transform 0.2s;  
  
}

.form img {  
  
display: block;  
  
margin: 0 auto 20px auto;



}

```
.controls::placeholder {
```

```
    color: #912F2F;
```

```
    opacity: 1;
```

```
}
```

```
label {
```

```
    display: block;
```

```
    margin-bottom: 6px;
```

```
    color: #912F2F;
```

```
    font-weight: 500;
```

```
    font-family: 'rubik', sans-serif;
```

```
}
```

```
.input-error {
```

```
    border: 2px solid #d65c5c;
```

```
    background-color: #ffe6e6;
```

```
}
```

```
.error-msg {
```

```
color: #d65c5c;  
font-size: 0.9em;  
margin-top: 2px;  
margin-bottom: 6px;  
display: block;  
}  
  
}
```

```
.modal {  
display: none;  
position: fixed;  
z-index: 9999;  
left: 0;  
top: 0;  
width: 100%;  
height: 100%;  
background-color: rgba(0,0,0,0.5);  
justify-content: center;  
align-items: center;  
}  
  
}
```

```
.modal-contenido {  
background-color: #fff;  
padding: 30px;
```

```
border-radius: 12px;  
text-align: center;  
max-width: 400px;  
width: 90%;  
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);  
border: 2px solid #912F2F;  
position: relative;  
font-family: 'Rubik', sans-serif;  
color: #912F2F;  
}
```

```
.modal-contenido p {  
font-size: 1rem;  
line-height: 1.6;  
}
```

```
.cerrar {  
position: absolute;  
top: 10px;  
right: 16px;  
font-size: 24px;  
font-weight: bold;  
color: #912F2F;
```

```
cursor: pointer;  
  
transition: color 0.2s;  
  
}  
  
.cerrar:hover {  
  
color: #B64040;  
  
}
```

## **index**

```
html, body {  
  
width: 100%;  
  
margin: 0;  
  
padding: 0;  
  
box-sizing: border-box;  
  
overflow-x: hidden;  
  
}
```

```
*, *::before, *::after {  
  
box-sizing: inherit;  
  
}
```

```
.vista-principal {  
  
width: 100%;  
  
}
```

```
.header-principal {  
    position: fixed;  
    top: 0;  
    z-index: 1000;  
    width: 100%;  
    background: #6b1e1e;  
    color: #fff;  
    box-shadow: 0 2px 8px rgba(0,0,0,0.04);  
}  
  
}
```

```
.rectangle-parent {  
    display: flex;  
    align-items: center;  
    justify-content: space-between;  
    height: 100px;  
    padding-left: 40px;  
    padding-right: 40px;  
    gap: 32px;  
}  
  
}
```

```
.logo-blanco-parent {  
    display: flex;  
    align-items: center;
```

gap: 10px;

}

.logo-blanco {

height: 80px;

width: auto;

}

.catlogo-de-restaurantes {

font-size: 1.6rem;

font-weight: 400;

color: #fff;

font-family: 'Montserrat', sans-serif;

max-width: 200px;

}

.acciones-usuario {

display: flex;

gap: 18px;

margin-left: auto;

}

.registro-usuario-parent,

```
.log-in-parent {  
    display: flex;  
    align-items: center;  
    gap: 8px;  
    background: #6A1E1E;  
    border: none;  
    border-radius: 50px;  
    color: #fff;  
    padding: 12px 22px;  
    font-size: 1rem;  
    font-family: 'Montserrat', sans-serif;  
    cursor: pointer;  
    transition: background 0.2s;  
}  
  
}
```

```
.registro-usuario-parent:hover,  
.log-in-parent:hover {  
    background: #912f2f;  
}
```

```
.registro-usuario-parent img,  
.log-in-parent img {  
    height: 22px;
```

width: 22px;

}

.calles-con-encanto-scaled-parent {

position: relative;

width: 100%;

height: 700px;

display: flex;

align-items: center;

justify-content: center;

overflow: hidden;

background: #000000;

}

.calles-con-encanto-scaled-icon {

position: absolute;

top: 0;

left: 0;

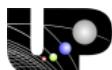
width: 100%;

height: 100%;

object-fit: cover;

z-index: 1;

opacity: 0.7;



}

.contenido-superpuesto {

position: relative;

z-index: 2;

width: 100%;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

.descubre-la-magia {

font-size: 2.8rem;

font-family: 'Montserrat', sans-serif;

font-weight: bold;

color: #fff;

text-shadow: 1px 1px 8px #000;

margin-bottom: 10px;

text-align: center;

}

.explora-los-mejores {

```
font-size: 1.2rem;  
font-family: 'Montserrat', sans-serif;  
color: #fff;  
text-shadow: 1px 1px 8px #000;  
margin-bottom: 24px;  
text-align: center;  
}
```

```
.barra-busqueda {  
display: flex;  
align-items: center;  
background: #fff;  
border-radius: 25px;  
padding: 8px 20px;  
box-shadow: 0 2px 8px rgba(0,0,0,0.08);  
width: 680px;  
max-width: 90%;  
margin: 0 auto;  
}
```

```
.barra-busqueda input::placeholder {  
color: #912F2F;  
opacity: 0.8;
```

{

.busqueda-icono {

width: 24px;

height: 24px;

margin-right: 10px;

}

.barra-busqueda input[type="text"] {

border: none;

outline: none;

font-size: 1rem;

flex: 1;

background: transparent;

padding: 8px 0;

color: #912F2F;

}

.barra-busqueda .buscar {

background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);

color: #fff;

border: none;

border-radius: 20px;

```
padding: 8px 22px;  
font-size: 1rem;  
margin-left: 10px;  
cursor: pointer;  
transition: background 0.8s;  
}  
  
}
```

```
.barra-busqueda .buscar:hover {  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
}
```

```
.barra-filtros-horizontal {  
display: flex;  
flex-wrap: wrap;  
justify-content: center;  
align-items: center;  
gap: 16px;  
padding: 16px;  
border-radius: 30px;  
  
width: 90%;  
max-width: 950px;  
}  
  
}
```

```
.filtro-item {  
    display: flex;  
    align-items: center;  
    background: #fff;  
    border: 1px solid #912f2f;  
    border-radius: 30px;  
    padding: 6px 14px;  
    gap: 8px;  
    min-width: 170px;  
}  
  
}
```

```
.icono-filtro {  
    width: 20px;  
    height: 20px;  
    opacity: 0.85;  
}  
  
}
```

```
.filtro-select {  
    border: none;  
    outline: none;  
    background: transparent;  
    color: #912f2f;
```

```
font-family: 'Rubik', sans-serif;  
font-size: 0.95rem;  
width: 100%;  
cursor: pointer;  
}
```

```
.btn-filtrar {  
    display: flex;  
    align-items: center;  
    gap: 10px;  
    background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
    color: #fff;  
    border: none;  
    border-radius: 30px;  
    padding: 10px 24px;  
    font-size: 1rem;  
    font-family: 'Montserrat', sans-serif;  
    font-weight: bold;  
    cursor: pointer;  
    transition: transform 0.2s, box-shadow 0.3s;  
}
```

```
.btn-filtrar:hover {
```

```
transform: scale(1.03);  
box-shadow: 0 6px 18px rgba(145, 47, 47, 0.2);  
}  
  
}
```

```
.icono-botón {
```

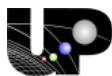
```
width: 18px;  
height: 18px;  
}
```

```
.filtros-abajo {
```

```
position: absolute;  
bottom: 30px;  
left: 50%;  
transform: translateX(-50%);  
z-index: 3;  
}
```

```
.line-parent {
```

```
display: flex;  
flex-wrap: wrap;  
gap: 32px;  
width: 100%;  
justify-content: center;
```



}

```
.contenido-principal {  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: center;  
    padding: 32px 40px;  
    width: 100%;  
    margin: 0;  
    padding: 32px 0;  
    box-sizing: border-box;  
    gap: 40px;  
}
```

```
.restaurante-card {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    background: #fff;  
    border-radius: 4px;  
    padding: 4px 4px 4px 4px;  
    cursor: pointer;  
    transition: box-shadow 0.3s ease, transform 0.3s ease;
```

width: 300px;

}

.restaurante-card:hover,

.restaurante-card:focus {

box-shadow: 0 6px 18px rgba(4, 4, 4, 0.166);

border-color: #912f2f;

transform: translateY(-4px) scale(1.03);

}

.restaurante-img {

width: 100%;

height: 200px;

border-radius: 4px;

object-fit: cover;

box-shadow: 0 2px 6px rgba(0,0,0,0.1);

margin-bottom: 8px;

}

.restaurante-info {

display: flex;

flex-direction: column;

gap: 6px;

```
justify-content: center;  
width: 100%;  
}
```

```
.restaurante-nombre {  
font-size: 1.3rem;  
font-weight: 600;  
margin: 0;  
color: #912f2f;  
font-family: 'Rubik', sans-serif;  
}
```

```
.restaurante-tipos,
```

```
.rectangle-container {  
width: 100%;  
background: #912f2f;  
color: #fff;  
display: flex;  
flex-wrap: wrap;  
align-items: flex-start;  
justify-content: space-between;  
padding: 32px 40px 20px 40px;
```

```
  box-sizing: border-box;  
  
  gap: 32px;  
  
  font-family: 'Montserrat', sans-serif;  
  
}
```

```
.contctanos {  
  
  font-size: 1.2rem;  
  
  font-weight: 600;  
  
  margin-bottom: 16px;  
  
  grid-column: 1 / -1;  
  
}
```

```
.contactos-grid {  
  
  display: grid;  
  
  grid-template-columns: repeat(2, 1fr);  
  
  gap: 18px 40px;  
  
}
```

```
.gmail-1-1-parent,  
.llamada-telefonica-2-parent,  
.logotipo-de-instagram-2-parent,  
.facebook-1-1-parent {
```

```
display: flex;  
align-items: center;  
gap: 10px;  
font-size: 1rem;  
}  
  
}
```

```
.llamada-telefonica-2-icon {  
width: 26px;  
height: 26px;  
object-fit: contain;  
}  
  
}
```

```
.sazonpatrimonialgmailcom {  
font-size: 1rem;  
color: #fff;  
}  
  
}
```

```
.footer-logos {  
display: flex;  
align-items: center;  
gap: 18px;  
}  
  
}
```

.logo-blanco-2,

.logo-equipo,

.logo-universidad {

height: 120px;

width: auto;

object-fit: contain;

border-radius: 8px;

}

.modal-overlay {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(0, 0, 0, 0.5);

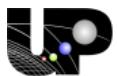
z-index: 9999;

justify-content: center;

align-items: center;

opacity: 0;

transition: opacity 0.3s ease;



}

.modal-overlay.show {

    opacity: 1;

}

.modal-content {

    position: relative;

    background: transparent;

    max-width: 90%;

    max-height: 95vh;

    overflow: visible;

    border-radius: 4px;

    display: flex;

    flex-direction: column;

    justify-content: center;

    transform: scale(0.7) translateY(-20px);

    transition: transform 0.3s ease, opacity 0.3s ease;

    opacity: 0;

}

.modal-overlay.show .modal-content {

    transform: scale(1) translateY(0);

opacity: 1;

}

.close-modal {

position: absolute;

top: 10px;

right: 15px;

background: none;

border: none;

font-size: 24px;

cursor: pointer;

color: #912F2F;

z-index: 10000;

width: 30px;

height: 30px;

display: flex;

align-items: center;

justify-content: center;

border-radius: 50%;

background: rgba(255, 255, 255, 0.8);

}

.close-modal:hover {

```
background: rgba(255, 255, 255, 1);  
  
color: #B64040;  
  
}
```

### **modal-encuesta**

```
.modal-overlay {  
  
display: none;  
  
position: fixed;  
  
z-index: 9999;  
  
left: 0;  
  
top: 0;  
  
width: 100vw;  
  
height: 100vh;  
  
background-color: rgba(0,0,0,0.5);  
  
justify-content: center;  
  
align-items: center;  
  
}
```

```
.modal-content {  
  
background: #fff;  
  
padding: 20px 20px;  
  
border-radius: 12px;  
  
max-width: 400px;  
  
width: 90%;
```

```
box-shadow: 7px 7px 5px #6A1E1E;  
font-family: 'rubik', sans-serif;  
color: #912F2F;  
position: relative;  
display: flex;  
flex-direction: column;  
align-items: center;  
}
```

```
.modal-content h2 {  
font-size: 1.4rem;  
margin-bottom: 18px;  
font-weight: 700;  
text-align: center;  
}
```

```
.modal-content img.logo {  
display: block;  
margin: 0 auto 0 auto;  
}
```

```
.modal-content label {  
display: block;
```

```
margin-bottom: 6px;  
  
color: #912F2F;  
  
font-weight: 500;  
  
font-family: 'rubik', sans-serif;  
  
text-align: left;  
  
width: 100%;  
  
}
```

```
.modal-content select {  
  
width: 100%;  
  
padding: 10px 10px 10px 12px;  
  
border: 1.5px solid #B64040;  
  
border-radius: 20px;  
  
font-size: 1rem;  
  
color: #912F2F;  
  
background: #fff;  
  
appearance: none;  
  
outline: none;  
  
margin-bottom: 10px;  
  
cursor: pointer;  
  
font-family: 'rubik', sans-serif;  
  
}
```

```
.modal-content select.error {  
  
    border: 2px solid #e74c3c;  
  
    background-color: #fdf2f2;  
  
}
```

```
.modal-content .button {  
  
    width: 100%;  
  
    background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
  
    border: none;  
  
    padding: 12px;  
  
    color: #fff;  
  
    margin: 18px 0 0 0;  
  
    font-size: 1rem;  
  
    border-radius: 8px;  
  
    cursor: pointer;  
  
    font-weight: 600;  
  
    letter-spacing: 1px;  
  
    transition: background 0.2s, transform 0.2s;  
  
}
```

```
.modal-content .button:hover, .modal-content .button:focus {  
  
    background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
  
    transform: translateY(-2px) scale(1.03);
```

}

.modal-content .error-message {

color: #e74c3c;

font-size: 12px;

margin-top: 2px;

margin-bottom: 6px;

display: block;

font-weight: 400;

letter-spacing: 0.01em;

}

.close-modal {

position: absolute;

top: 10px;

right: 16px;

font-size: 24px;

font-weight: bold;

color: #912F2F;

background: none;

border: none;

cursor: pointer;

transition: color 0.2s;

}

.close-modal:hover {

color: #B64040;

}

### **modal-login-admin**

#loginModal .error-msg {

color: #d32f2f;

font-size: 14px;

margin-bottom: 10px;

display: block;

min-height: 20px;

}

.form {

width: 400px;

background: #ffffff;

padding: 30px;

margin: auto;

border-radius: 4px;

font-family: 'Rubik', sans-serif;

color: #912F2F;

box-shadow: 7px 7px 5px #6A1E1E;

}

```
.form h4 {  
    text-align: center;  
    font-size: 36px;  
    margin: 0 0 20px 0;  
}  
  
}
```

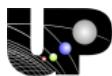
```
.form img {  
    display: block;  
    margin: 0 auto 0 auto;  
}  
  
}
```

```
.controls {  
    width: 100%;  
    background: #ffffff;  
    padding: 10px;  
    border-radius: 8px;  
    margin-bottom: 2px;  
    border: 1.5px solid #B64040;  
    font-family: 'Rubik', sans-serif;  
    color: #912F2F;  
}  
  
}
```

```
.controls::placeholder {  
    color: #912F2F;  
    opacity: 1;  
}  
  
label {
```

```
    display: block;  
    margin-bottom: 6px;  
    color: #912F2F;  
    font-weight: 500;  
    font-family: 'Rubik', sans-serif;  
}  
  
}
```

```
.button {  
    width: 100%;  
    background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
    border: none;  
    padding: 12px;  
    color: #ffffff;  
    margin: 16px 0;  
    font-size: 16px;  
    border-radius: 8px;  
    cursor: pointer;
```



}

```
.button:hover, .button:focus {  
  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
  
transform: translateY(-2px) scale(1.03);  
  
transition: background 0.2s, transform 0.2s;  
  
}
```

```
.input-error {  
  
border: 2px solid #d65c5c;  
  
background-color: #ffe6e6;  
  
}
```

```
.error-msg {  
  
color: #d65c5c;  
  
font-size: 0.9em;  
  
margin-top: 2px;  
  
margin-bottom: 6px;  
  
display: block;  
  
}
```

## modal-login

```
#loginModal .error-msg {  
  
color: #d32f2f;
```

```
font-size: 14px;  
  
margin-bottom: 10px;  
  
display: block;  
  
min-height: 20px;  
  
}  
  

```

```
.form {  
  
width: 400px;  
  
background: #ffffff;  
  
padding: 30px;  
  
margin: auto;  
  
border-radius: 4px;  
  
font-family: 'Rubik', sans-serif;  
  
color: #912F2F;  
  
box-shadow: 7px 7px 5px #6A1E1E;  
  
}  
  

```

```
.form h4 {  
  
text-align: center;  
  
font-size: 36px;  
  
margin: 0 0 20px 0;  
  
}
```

```
.form img {  
    display: block;  
    margin: 0 auto 0 auto;  
}  
  
}
```

```
.controls {  
    width: 100%;  
    background: #ffffff;  
    padding: 10px;  
    border-radius: 8px;  
    margin-bottom: 2px;  
    border: 1.5px solid #B64040;  
    font-family: 'Rubik', sans-serif;  
    color: #912F2F;  
}  
  
}
```

```
.controls::placeholder {  
    color: #912F2F;  
    opacity: 1;  
}  
  
}
```

```
label {  
    display: block;
```

```
margin-bottom: 6px;  
  
color: #912F2F;  
  
font-weight: 500;  
  
font-family: 'Rubik', sans-serif;  
  
}  
  
}
```

```
.button {  
  
width: 100%;  
  
background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
  
border: none;  
  
padding: 12px;  
  
color: #ffffff;  
  
margin: 16px 0;  
  
font-size: 16px;  
  
border-radius: 8px;  
  
cursor: pointer;  
  
}  
  
}
```

```
.button:hover, .button:focus {  
  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
  
transform: translateY(-2px) scale(1.03);  
  
transition: background 0.2s, transform 0.2s;  
  
}
```

```
.input-error {  
    border: 2px solid #d65c5c;  
    background-color: #ffe6e6;  
}  
  
}
```

```
.error-msg {  
    color: #d65c5c;  
    font-size: 0.9em;  
    margin-top: 2px;  
    margin-bottom: 6px;  
    display: block;  
}  
  
}
```

### **modal-registro**

```
#registroModal .form {  
    width: 450px;  
    background: #ffffff;  
    padding: 20px;  
    margin: auto;  
    border-radius: 4px;  
    font-family: 'Rubik', sans-serif;  
    color: #912F2F;  
    box-shadow: 5px 5px 5px #6A1E1E;
```

```
transform: scale(0.8) translateY(-30px);  
  
transition: transform 0.4s cubic-bezier(0.175, 0.885, 0.32, 1.275), opacity 0.3s ease;  
  
opacity: 0;  
  
box-sizing: border-box;  
  
}
```

```
#registroModal.show .form {  
  
transform: scale(1) translateY(0);  
  
opacity: 1;  
  
}
```

```
#registroModal .form h4 {  
  
text-align: center;  
  
font-size: 36px;  
  
margin: 0 0 20px 0;  
  
}
```

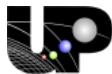
```
#registroModal .form img {  
  
display: block;  
  
margin: 0 auto 0 auto;  
  
}
```

```
#registroModal .form label {
```

```
display: block;  
  
margin-bottom: 4px;  
  
color: #912F2F;  
  
font-weight: 500;  
  
font-family: 'Rubik', sans-serif;  
  
}
```

```
#registroModal .controls {  
  
width: 100%;  
  
background: #ffffff;  
  
padding: 10px;  
  
border-radius: 8px;  
  
margin-bottom: 6px;  
  
border: 1px solid #912F2F;  
  
font-family: 'Rubik', sans-serif;  
  
font-size: 16px;  
  
color: #912F2F;  
  
box-sizing: border-box;  
  
}
```

```
#registroModal .controls::placeholder {  
  
color: #912F2F;  
  
opacity: 1;
```



}

```
#registroModal .controls:focus {
```

```
    outline: none;
```

```
    border-color: #B64040;
```

```
}
```

```
#registroModal .controls.input-error {
```

```
    border-color: #d32f2f;
```

```
    background-color: #ffeaea;
```

```
}
```

```
#registroModal .botons {
```

```
    width: 100%;
```

```
    background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);
```

```
    border: none;
```

```
    padding: 12px;
```

```
    color: #ffffff;
```

```
    margin: 16px 0;
```

```
    font-size: 16px;
```

```
    border-radius: 8px;
```

```
    cursor: pointer;
```

```
    font-family: 'Rubik', sans-serif;
```

```
transition: all 0.3s ease;  
box-shadow: 0 2px 8px rgba(182,64,64,0.08);  
box-sizing: border-box;  
}
```

```
#registroModal .botons:hover, #registroModal .botons:focus {  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
transform: translateY(-2px) scale(1.03);  
}
```

```
#registroModal .error-msg {  
color: #d32f2f;  
font-size: 14px;  
margin-bottom: 10px;  
display: block;  
min-height: 20px;  
}
```

## **select**

```
body {  
background: #fff;  
min-height: 100vh;  
margin: 0;  
font-family: 'Rubik', sans-serif;
```

```
color: #912F2F;  
  
display: flex;  
  
align-items: center;  
  
justify-content: center;  
  
}
```

```
.choose {  
  
background: #fff;  
  
border-radius: 8px;  
  
box-shadow: 7px 7px 5px #B64040;  
  
padding: 30px 30px 24px 30px;  
  
display: flex;  
  
flex-direction: column;  
  
align-items: center;  
  
}
```

```
.logo {  
  
margin-bottom: 20px;  
  
}
```

```
.choose h4 {  
  
font-size: 2.2rem;  
  
font-weight: 700;
```

```
margin: 0 0 28px 0;  
  
color: #912F2F;  
  
text-align: center;  
  
}  
  

```

```
.button {  
  
width: 100%;  
  
display: flex;  
  
align-items: center;  
  
gap: 12px;  
  
background: #6A1E1E;  
  
color: #fff;  
  
border: none;  
  
border-radius: 8px;  
  
padding: 14px 0px;  
  
font-size: 1.2rem;  
  
font-family: inherit;  
  
font-weight: 500;  
  
margin-bottom: 18px;  
  
cursor: pointer;  
  
transition: background 0.2s, transform 0.2s;  
  
box-shadow: 0 2px 8px rgba(182,64,64,0.08);  
  
justify-content: center;
```

}

.button:last-of-type {

margin-bottom: 24px;

}

.button:hover, .button:focus {

background: #B64040;

transform: translateY(-2px) scale(1.03);

}

.icon {

margin-right: 8px;

}

.register-link {

color: #912F2F;

text-decoration: underline;

font-size: 1rem;

margin-top: 8px;

transition: color 0.2s;

}

```
.register-link:hover, .register-link:focus {  
    color: #B64040;  
}  
}
```

## solicitud

```
* , *::before, *::after {  
    box-sizing: border-box;  
}  
}
```

```
html, body {  
    width: 100%;  
    margin: 0;  
    padding: 0;  
    overflow-x: hidden;  
    font-family: 'Rubik', sans-serif;  
}  
}
```

```
.icon-32 {  
    width: 40px;  
    height: 40px;  
}  
}
```

```
.btn-back {  
    background: none;  
    border: none;
```

```
position: absolute;  
top: 65px;  
left: 65px;  
cursor: pointer;  
padding: 0;  
}
```

```
.registro-container {  
display: flex;  
gap: 24px;  
max-width: 100%;  
margin: 100px 0 0 0;  
width: 100%;  
flex-wrap: wrap;  
justify-content: center;  
position: relative;  
}
```

```
.form-section {  
background: #fff;  
padding: 40px;  
width: 600px;  
display: flex;
```

```
flex-direction: column;  
justify-content: center;  
}  
  
}
```

```
.form-section h4 {  
text-align: center;  
font-size: 2rem;  
margin-bottom: 4px;  
margin-top: 0;  
color: #912F2F;  
font-weight: 700;  
font-family: 'Rubik', sans-serif;  
}  
  
}
```

```
.form-section .sub-text {  
text-align: center;  
font-size: 0.95rem;  
color: #912F2F;  
margin-bottom: 24px;  
font-family: 'Rubik', sans-serif;  
}  
  
}
```

```
label {  
    display: block;  
    margin-bottom: 6px;  
    margin-top: 10px;  
    color: #912F2F;  
    font-weight: 500;  
    font-family: 'Rubik', sans-serif;  
    font-size: 1rem;  
}  
  
label::before {  
    content: attr(data-label);  
    font-size: 0.8em;  
    font-weight: 400;  
}
```

```
.controls {  
    width: 100%;  
    background: #fff;  
    padding: 10px;  
    border-radius: 8px;  
    margin-bottom: 0;  
    border: 1.5px solid #B64040;  
    font-family: 'Rubik', sans-serif;  
    font-size: 1rem;  
    color: #912F2F;  
    transition: border 0.2s;  
}  
  
.controls::before {  
    content: attr(data-label);  
    font-size: 0.8em;  
    font-weight: 400;  
}
```

```
.controls:focus {  
    border: 1.5px solid #912F2F;  
    outline: none;  
}  
  
}
```

```
.imagenes-group {  
    display: flex;  
    gap: 10px;  
    margin-top: 10px;  
    margin-bottom: 0;  
    justify-content: center;  
}  
  
}
```

```
.input-file-group {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    width: 50%;  
}  
  
}
```

```
.input-file {  
    background: #FFF;  
    color: #B64040;
```

```
text-align: center;  
  
border: 1.5px solid #B64040;  
  
border-radius: 8px;  
  
padding: 10px;  
  
font-weight: 500;  
  
  
  
cursor: pointer;  
  
transition: background 0.2s;  
  
font-family: 'Rubik', sans-serif;  
  
font-size: 1rem;  
  
}
```

```
.input-file:hover {  
  
background: #f9eaea;  
  
}
```

```
.button {  
  
width: 100%;  
  
background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
  
border: none;  
  
padding: 14px;  
  
color: #fff;  
  
margin-top: 16px;
```

```
font-size: 1.1rem;  
border-radius: 8px;  
cursor: pointer;  
font-family: 'Rubik', sans-serif;  
font-weight: 500;  
transition: background 0.2s, transform 0.2s;  
}  
  
}
```

```
.button:hover, .button:focus {  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
transform: translateY(-2px) scale(1.03);  
}
```

```
.image-section {  
display: flex;  
align-items: stretch;  
justify-content: center;  
width: 700px;  
padding: 40px;  
box-sizing: border-box;  
}
```

```
.registro-img {
```

```
width: 100%;  
height: 100%;  
max-width: 700px;  
border-radius: 12px;  
object-fit: cover;  
box-shadow: 0 2px 12px rgba(145,47,47,0.10);  
}  
  
::placeholder {
```

```
color: #912F2F;  
opacity: 0.8;  
font-weight: 400;  
}
```

```
.vista-principal {  
width: 100%;  
}
```

```
.header-principal {  
position: fixed;  
top: 0;  
z-index: 1000;  
width: 100%;
```

```
background: #6b1e1e;  
color: #fff;  
box-shadow: 0 2px 8px rgba(0,0,0,0.04);  
}  
  
}
```

```
.rectangle-parent {  
    display: flex;  
    align-items: center;  
    justify-content: space-between;  
    height: 100px;  
    padding-left: 40px;  
    padding-right: 40px;  
    gap: 32px;  
}
```

```
.logo-blanco-parent {  
    display: flex;  
    align-items: center;  
    gap: 10px;  
}
```

```
.logo-blanco {  
    height: 80px;
```

```
width: auto;  
}  
  
width: auto;
```

```
.catlogo-de-restaurantes {  
  
font-size: 1.6rem;  
  
font-weight: 400;  
  
color: #fff;  
  
font-family: 'Montserrat', sans-serif;  
  
max-width: 200px;  
  
}
```

```
.rectangle-container {  
  
width: 100%;  
  
background: #912f2f;  
  
color: #fff;  
  
display: flex;  
  
flex-wrap: wrap;  
  
align-items: flex-start;  
  
justify-content: space-between;  
  
padding: 32px 40px 20px 40px;  
  
box-sizing: border-box;  
  
gap: 32px;  
  
font-family: 'Montserrat', sans-serif;
```

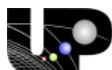
}

```
.contctanos {  
  
    font-size: 1.2rem;  
  
    font-weight: 600;  
  
    margin-bottom: 16px;  
  
    grid-column: 1 / -1;  
  
}
```

```
.contactos-grid {  
  
    display: grid;  
  
    grid-template-columns: repeat(2, 1fr);  
  
    gap: 18px 40px;  
  
}
```

```
.gmail-1-1-parent,  
.llamada-telefonica-2-parent,  
.logotipo-de-instagram-2-parent,  
.facebook-1-1-parent {
```

```
    display: flex;  
  
    align-items: center;  
  
    gap: 10px;  
  
    font-size: 1rem;
```



Universidad Politécnica  
de Chiapas

}



.llamada-telefonica-2-icon {

width: 26px;

height: 26px;

object-fit: contain;

}

.sazonpatrimonialgmailcom {

font-size: 1rem;

color: #fff;

}

.footer-logos {

display: flex;

align-items: center;

gap: 18px;

}

.logo-blanco-2,

.logo-equipo,

.logo-universidad {

height: 120px;

```
width: auto;  
object-fit: contain;  
border-radius: 8px;  
}  
  
}
```

```
.modal {  
position: fixed;  
top: 0; left: 0; right: 0; bottom: 0;  
background: rgba(0,0,0,0.5);  
z-index: 2000;  
display: none;  
align-items: center;  
justify-content: center;  
}
```

```
.modal-contenido {  
background: #fff;  
border-radius: 12px;  
padding: 32px 24px 24px 24px;  
box-shadow: 0 4px 32px rgba(0,0,0,0.18);  
position: relative;  
min-width: 300px;  
max-width: 90vw;
```

text-align: center;

}

.modal-contenido p {

font-size: 1rem;

line-height: 1.6;

color:#912f2f;

}

.cerrar {

position: absolute;

top: 12px;

right: 18px;

font-size: 2rem;

color: #912F2F;

cursor: pointer;

}

.cerrar:hover {

color: #B64040;

}

.input-error {

```
border: 2px solid #d65c5c;  
background-color: #ffe6e6;  
}  
  
}
```

```
.error-msg {  
color: #d65c5c;  
font-size: 0.95rem;  
margin-top: 2px;  
margin-bottom: 6px;  
display: block;  
}
```

```
.input-group {  
display: flex;  
flex-direction: column;  
align-items: flex-start;  
}
```

## solicitudMembresia

```
.form{  
width:400px;  
background: #ffffff;  
padding : 30px;  
margin: auto;
```

```
border-radius: 4px;  
  
font-family: 'rubik', sans-serif;  
  
color: #912F2F;  
  
box-shadow: 7px 7px 5px #B64040;  
  
}
```

```
.form h4{  
  
text-align: center;  
  
font-size: 30px;  
  
margin: 0 0 20px 0;  
  
}
```

```
.form p{  
  
text-align: center;  
  
font-size: 12px;  
  
margin: 0 0 20px 0;  
  
color: #912F2F;  
  
}
```

```
.controls {  
  
width: 100%;  
  
background: #ffffff;  
  
padding: 10px;  
  
border-radius: 8px;
```

```
margin-bottom: 2px;  
  
border: 1.5px solid #B64040;  
  
font-family: 'rubik', sans-serif;  
  
}
```

```
.button{  
  
width: 100%;  
  
background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
  
border: none;  
  
padding: 12px;  
  
color: #ffffff;  
  
margin: 16px 0;  
  
font-size: 16px;  
  
border-radius: 8px;  
  
cursor: pointer;  
  
}
```

```
.form img {  
  
display: block;  
  
margin: 0 auto 20px auto;  
  
}
```

```
.controls::placeholder {
```

```
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
opacity: 1;  
}
```

```
label {  
display: block;  
margin-bottom: 6px;  
color: #912F2F;  
font-weight: 500;  
font-family: 'rubik', sans-serif;  
}
```

```
.button:hover, .button:focus {  
background: #B64040;  
transform: translateY(-2px) scale(1.03);  
transition: background 0.2s, transform 0.2s;  
}
```

```
.input-error {  
border: 2px solid #d65c5c;  
background-color: #fff;  
outline: none;  
box-shadow: 0 0 0 2px rgba(214,92,92,0.08);
```

transition: border 0.2s, box-shadow 0.2s;  
}  
  
}

.error-msg {  
  
color: #d65c5c;  
  
font-size: 0.85em;  
  
margin-top: 2px;  
  
margin-bottom: 6px;  
  
display: block;  
  
font-weight: 400;  
  
letter-spacing: 0.01em;  
  
}

.modal {  
  
display: none;  
  
position: fixed;  
  
z-index: 9999;  
  
left: 0;  
  
top: 0;  
  
width: 100%;  
  
height: 100%;  
  
background-color: rgba(0,0,0,0.5);  
  
justify-content: center;

```
    align-items: center;  
}  
  
 .modal-contenido {
```

```
background-color: #fff;  
padding: 30px;  
border-radius: 12px;  
text-align: center;  
max-width: 400px;  
width: 90%;  
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);  
border: 2px solid #912F2F;  
position: relative;  
font-family: 'Rubik', sans-serif;  
color: #912F2F;  
}
```

```
.modal-contenido p {  
font-size: 1rem;  
line-height: 1.6;  
}
```

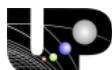
```
.cerrar {
```

```
position: absolute;  
top: 10px;  
right: 16px;  
font-size: 24px;  
font-weight: bold;  
color: #912F2F;  
cursor: pointer;  
transition: color 0.2s;  
}  
.cerrar:hover {  
color: #B64040;  
}
```

### vistaEdicionRest

```
html, body {  
width: 100%;  
margin: 0;  
padding: 0;  
box-sizing: border-box;  
overflow-x: hidden;  
}  
}
```

```
*, *::before, *::after {  
box-sizing: inherit;
```



}

.vista-principal {

width: 100%;

}

.header-principal {

position: fixed;

top: 0;

z-index: 1000;

width: 100%;

background: #6b1e1e;

color: #fff;

box-shadow: 0 2px 8px rgba(0,0,0,0.04);

}

.rectangle-parent {

display: flex;

align-items: center;

justify-content: space-between;

height: 100px;

padding-left: 40px;

padding-right: 40px;

gap: 32px;

}

.logo-blanco-parent {

display: flex;

align-items: center;

gap: 10px;

}

.logo-blanco {

height: 80px;

width: auto;

}

.catlogo-de-restaurantes {

font-size: 1.6rem;

font-weight: 400;

color: #fff;

font-family: 'Montserrat', sans-serif;

max-width: 200px;

}

.acciones-usuario {

margin-left: auto;

```
display: flex;  
gap: 18px;  
}
```

```
.subscription-parent,
```

```
.log-out-parent {
```

```
    display: flex;
```

```
    align-items: center;
```

```
    gap: 8px;
```

```
    background: #6A1E1E;
```

```
    border: none;
```

```
    border-radius: 50px;
```

```
    color: #fff;
```

```
    padding: 12px 22px;
```

```
    font-size: 1rem;
```

```
    font-family: 'Montserrat', sans-serif;
```

```
    cursor: pointer;
```

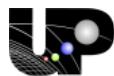
```
    transition: background 0.2s;
```

```
}
```

```
.subscription-parent:hover,
```

```
.log-out-parent:hover {
```

```
    background: #912f2f;
```



}

.subscripcion-parent img,

.log-out-parent img {

height: 22px;

width: 22px;

}

.icono-restaurantero {

display: flex;

align-items: center;

}

.icono-restaurantero img {

height: 48px;

width: 48px;

border-radius: 50%;

}

.restaurante-edicion-container {

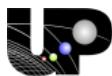
max-width: 90%;

margin: 140px auto 40px auto;

```
background: #fff;  
border-radius: 18px;  
padding: 32px 36px 36px 36px;  
font-family: 'Montserrat', sans-serif;  
position: relative;  
}  
  
}
```

```
.icon-32 {  
width: 40px;  
height: 40px;  
}  
  
.icon-middle {  
vertical-align: middle;  
}
```

```
.btn-back {  
background: none;  
border: none;  
position: absolute;  
top: 24px;  
left: 24px;  
cursor: pointer;  
padding: 0;
```



}

.restaurante-nombre {

text-align: center;

font-size: 3rem;

font-weight: 700;

color: #912f2f;

margin: 0 0 8px 0;

letter-spacing: 1px;

}

.btn-estadisticas {

display: flex;

justify-content: center;

align-items: center;

gap: 8px;

background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);

color: #fff;

border: none;

border-radius: 8px;

width: 20%;

padding: 12px 0;

font-size: 1.1rem;

```
font-weight: 600;  
cursor: pointer;  
margin: 0 auto;  
box-shadow: 0 2px 8px rgba(0,0,0,0.08);  
transition: background 0.2s;  
}  
}
```

```
.btn-estadisticas:hover, .btn-estadisticas:focus {  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
transform: translateY(-2px) scale(1.03);  
transition: background 0.2s, transform 0.2s;  
}
}
```

```
.restaurante-divider {  
border: none;  
border-top: 4px solid #912f2f;  
width: 80%;  
margin: 0 auto 24px auto;  
border-radius: 2px;  
}
}
```

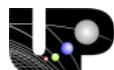
```
.formulario-edicion {  
border: 2px solid #912f2f;  
border-radius: 10px;
}
```

```
padding: 0;  
  
font-family: 'Montserrat', sans-serif;  
  
color: #912f2f;  
  
overflow: hidden;  
  
margin: 0 auto 40px;  
  
width: 80%;  
  
}
```

```
.linea-interna {  
  
padding: 20px 24px;  
  
border-bottom: 2px solid #912f2f;  
  
display: flex;  
  
flex-direction: column;  
  
gap: 12px;  
  
}
```

```
.linea-interna:last-child {  
  
border-bottom: none;  
  
}
```

```
.campo-titulo {  
  
font-weight: 600;  
  
font-size: 1.1rem;
```

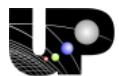


}

```
.galeria-imagenes {  
    display: flex;  
    gap: 16px;  
    justify-content: flex-start;  
}  
  
}
```

```
.imagen-slot {  
    border: 1.5px dashed #912f2f;  
    border-radius: 4px;  
    width: 40%;  
    height: 200px;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    background: #fdfdfd;  
    cursor: pointer;  
    transition: background 0.3s;  
}  
  
}
```

```
.imagen-slot:hover {  
    background: #f9eaea;
```



}

```
.imagen-slot img {  
    width: 100%;  
    height: 100%;  
    pointer-events: none;  
}  
  
input[type="text"]::placeholder {  
    color: #912f2f;  
    opacity: 1;  
}  
  
input[type="text"] {  
    border: 1px solid #912f2f;  
    border-radius: 6px;  
    padding: 6px 12px;  
    font-size: 0.95rem;  
    color: #912f2f;  
    width: 100%;  
    max-width: 100%;  
    height: 40px;  
}
```

```
.restaurante-info-flex {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-template-rows: auto auto;  
    grid-template-areas:  
        "menu menu"  
        "ubicacion horarios";  
    gap: 24px;  
}
```

```
.horarios-area { grid-area: horarios; }  
.ubicacion-area { grid-area: ubicacion; }  
.menu-area { grid-area: menu; }
```

```
.contacto-etiquetas {  
    display: flex;  
    justify-content: space-between;  
    gap: 40px;  
    flex-wrap: wrap;  
}
```

```
.filtro-select {  
    width: 100%;  
    padding: 10px 38px 10px 36px;  
    border: 1px solid #912f2f;  
    border-radius: 20px;  
    font-size: 1rem;  
    color: #912f2f;  
    background: #fff;  
    appearance: none;  
    outline: none;  
    margin-bottom: 0;  
    cursor: pointer;  
    font-family: 'rubik', sans-serif;  
}  
}
```

```
.contactos,  
.etiquetas {  
    flex: 1 1 45%;  
    display: flex;  
    flex-direction: column;  
    gap: 12px;  
}  
}
```

```
.input-con-icono {
```

```
    display: flex;
```

```
    align-items: center;
```

```
    gap: 10px;
```

```
}
```

```
.input-con-icono img {
```

```
    width: 26px;
```

```
    height: 26px;
```

```
    object-fit: contain;
```

```
}
```

```
.contactos .input-con-icono input,
```

```
.etiquetas .input-con-icono input {
```

```
    flex: 1;
```

```
}
```

```
.btn-menu {
```

```
    display: flex;
```

```
    justify-content: center;
```

```
    align-items: center;
```

```
    gap: 8px;
```

```
    background: #fff;
```

```
color: #912F2F;  
  
border: 1px solid #912F2F;  
  
border-radius: 6px;  
  
width: 100%;  
  
padding: 6px 12px;  
  
font-size: 0.95rem;  
  
font-weight: normal;  
  
cursor: pointer;  
  
margin: 0 auto;  
  
box-shadow: none;  
  
transition: background 0.2s;  
  
height: 40px;
```

```
}
```

```
.btn-menu:hover {  
  
background: #f9eaea;  
  
}
```

```
.btn-menu span {  
  
font-size: 0.95rem;  
  
font-weight: normal;  
  
color: #912F2F;  
  
}
```

```
.btn-menu .file-name {  
    color: #912F2F;  
}  
}
```

```
.btn-aplicar {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    gap: 8px;  
    width: 30%;  
    background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
    color: #fff;  
    border: none;  
    border-radius: 8px;  
    padding: 12px 28px;  
    font-size: 1.1rem;  
    font-weight: 600;  
    cursor: pointer;  
    margin: 0 auto;  
    box-shadow: 0 2px 8px rgba(0,0,0,0.08);  
    transition: background 0.2s;  
}  
}
```

```
.btn-aplicar:hover, .btn-aplicar:focus {  
  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
  
transform: translateY(-2px) scale(1.03);  
  
transition: background 0.2s, transform 0.2s;  
  
}  
  
}
```

```
.icon-20 {  
  
width: 20px;  
  
height: 20px;  
  
}
```

```
.rectangle-container {  
  
width: 100%;  
  
background: #912f2f;  
  
color: #fff;  
  
display: flex;  
  
flex-wrap: wrap;  
  
align-items: flex-start;  
  
justify-content: space-between;  
  
padding: 32px 40px 20px 40px;  
  
box-sizing: border-box;  
  
gap: 32px;
```

font-family: 'Montserrat', sans-serif;  
}

.contctanos {  
  
font-size: 1.2rem;  
  
font-weight: 600;  
  
margin-bottom: 16px;  
  
grid-column: 1 / -1;  
  
}

.contactos-grid {  
  
display: grid;  
  
grid-template-columns: repeat(2, 1fr);  
  
gap: 18px 40px;  
  
}

.gmail-1-1-parent,  
.llamada-telefonica-2-parent,  
.logotipo-de-instagram-2-parent,  
.facebook-1-1-parent {  
  
display: flex;  
  
align-items: center;

gap: 10px;

font-size: 1rem;

}

.llamada-telefonica-2-icon {

width: 26px;

height: 26px;

object-fit: contain;

}

.sazonpatrimonialgmailcom {

font-size: 1rem;

color: #fff;

}

.footer-logos {

display: flex;

align-items: center;

gap: 18px;

}

.logo-blanco-2,

.logo-equipo,

```
.logo-universidad {  
    height: 120px;  
    width: auto;  
    object-fit: contain;  
    border-radius: 8px;  
}  
  
.modal-overlay {  
    display: none;  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    background: rgba(0, 0, 0, 0.5);  
    z-index: 9999;  
    justify-content: center;  
    align-items: center;  
    opacity: 0;  
    transition: opacity 0.3s ease;  
}
```

```
.modal-overlay.show {  
    opacity: 1;  
}  
  
.modal-content {  
    position: relative;  
    background: transparent;  
    max-width: 90%;  
    max-height: 95vh;  
    overflow: visible;  
    border-radius: 4px;  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
    transform: scale(0.7) translateY(-20px);  
    transition: transform 0.3s ease, opacity 0.3s ease;  
    opacity: 0;  
}  
  
.modal-overlay.show .modal-content {  
    transform: scale(1) translateY(0);  
    opacity: 1;  
}
```

```
.close-modal {  
    position: absolute;  
    top: 10px;  
    right: 15px;  
    background: none;  
    border: none;  
    font-size: 24px;  
    cursor: pointer;  
    color: #912F2F;  
    z-index: 10000;  
    width: 30px;  
    height: 30px;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    border-radius: 50%;  
    background: rgba(255, 255, 255, 0.8);  
}  
  
.close-modal:hover {  
    background: rgba(255, 255, 255, 1);  
    color: #B64040;
```

{

```
.membresia-status {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    justify-content: center;  
    width: 100%;  
}
```

## vistaPrincipalAdmin.

```
.modal-contenido-visualizador {  
    color: #6b1e1e;  
}
```

```
.modal-contenido-visualizador h3,  
.modal-contenido-visualizador p,  
.modal-contenido-visualizador label,  
.modal-contenido-visualizador span,  
.modal-contenido-visualizador a {  
    color: #6b1e1e;  
}
```

```
.modal-visualizador {  
    position: fixed;
```

```
top: 0;  
left: 0;  
width: 100vw;  
height: 100vh;  
background: rgba(98, 98, 98, 0.85);  
display: flex;  
align-items: center;  
justify-content: center;  
z-index: 9999;  
}
```

```
.modal-contenido-visualizador {  
background: #fff8f8;  
padding: 24px 32px 32px 32px;  
border-radius: 14px;  
max-width: 90vw;  
max-height: 90vh;  
overflow: auto;  
position: relative;  
box-shadow: 0 4px 32px rgba(107,30,30,0.18);  
}
```

```
.cerrar-modal-visualizador {
```

```
position: absolute;  
top: 10px;  
right: 10px;  
font-size: 1.5em;  
background: none;  
border: none;  
cursor: pointer;  
color: #912F2F;  
}  
  
.tabla-admin td input[type="text"] {  
width: 50%;  
padding: 2px 12px;  
font-size: 1rem;  
border: 2px solid #912F2F;  
border-radius: 6px;  
outline: none;  
font-family: 'Montserrat', sans-serif;  
color: #6b1e1e;  
background: #fff8f8;  
box-sizing: border-box;  
transition: border 0.2s, box-shadow 0.2s;  
}  
  
.tabla-admin td input[type="text"]:focus {
```

```
border-color: #6A1E1E;  
box-shadow: 0 0 0 2px #e0bc0c;  
}
```

```
html, body {  
width: 100%;  
margin: 0;  
padding: 0;  
box-sizing: border-box;  
overflow-x: hidden;  
}
```

```
*, *::before, *::after {  
box-sizing: inherit;  
}
```

```
.vista-principal {  
width: 100%;  
}
```

```
.header-principal {  
position: fixed;  
top: 0;  
z-index: 1000;
```

```
width: 100%;  
  
background: #6b1e1e;  
  
color: #fff;  
  
box-shadow: 0 2px 8px rgba(0,0,0,0.04);  
  
}
```

```
.rectangle-parent {  
  
display: flex;  
  
align-items: center;  
  
justify-content: space-between;  
  
height: 100px;  
  
padding-left: 40px;  
  
padding-right: 40px;  
  
gap: 32px;  
  
}
```

```
.logo-blanco-parent {  
  
display: flex;  
  
align-items: center;  
  
gap: 10px;  
  
}
```

```
.logo-blanco {
```

```
height: 80px;  
width: auto;  
}  
  
height: 80px;  
width: auto;  
}
```

```
.catalogo-de-restaurantes {  
    font-size: 1.6rem;  
    font-weight: 400;  
    color: #fff;  
    font-family: 'Montserrat', sans-serif;  
    max-width: 200px;  
}  
  
height: 80px;  
width: auto;  
}
```

```
.acciones-usuario {  
    margin-left: auto;  
    display: flex;  
    gap: 18px;  
}  
  
height: 80px;  
width: auto;  
}
```

```
.log-out-parent {  
    display: flex;  
    align-items: center;  
    gap: 8px;  
    background: #6A1E1E;  
}
```

```
border: none;  
  
border-radius: 50px;  
  
color: #fff;  
  
padding: 12px 22px;  
  
font-size: 1rem;  
  
font-family: 'Montserrat', sans-serif;  
  
cursor: pointer;  
  
transition: background 0.2s;  
  
}  
  
}
```

```
.log-out-parent:hover {  
  
background: #912f2f;  
  
}
```

```
.log-out-parent img {  
  
height: 22px;  
  
width: 22px;  
  
}
```

```
.admin-panel {  
  
display: flex;  
  
min-height: 70vh;
```

```
margin-top: 100px;  
  
font-family: 'Montserrat', sans-serif;  
  
}
```

```
.tabla-admin {  
  
width: 100%;  
  
border-collapse: separate;  
  
border-spacing: 0;  
  
background: #fff;  
  
border-radius: 12px;  
  
overflow: hidden;  
  
box-shadow: 0 2px 8px rgba(145, 30, 30, 0.08);  
  
font-family: 'Montserrat', sans-serif;  
  
margin-bottom: 32px;  
  
}
```

```
.tabla-admin thead {  
  
background: #6b1e1e;  
  
}
```

```
.tabla-admin th, .tabla-admin td {  
  
padding: 14px 18px;  
  
text-align: center;
```

```
font-size: 1rem;  
}  
  
}
```

```
.tabla-admin th {  
color: #FFF;  
font-weight: 700;  
letter-spacing: 0.5px;  
border-bottom: 2px solid #e0bcbc;  
}  
  
}
```

```
.tabla-admin tbody tr {  
border-bottom: 1px solid #f1dada;  
transition: background 0.2s;  
}  
  
}
```

```
.tabla-admin tbody tr:nth-child(even) {  
background: #f8eaea;  
}  
  
}
```

```
.tabla-admin tbody tr:hover {  
background: #f3d6d6;  
}  
  
}
```

```
.tabla-admin td {  
    color: #6b1e1e;  
    font-weight: 500;  
    vertical-align: middle;  
}  
  
}
```

```
.tabla-admin button {  
    background: none;  
    border: none;  
    cursor: pointer;  
    margin-right: 6px;  
    padding: 4px;  
    transition: transform 0.1s;  
}  
  
}
```

```
.tabla-admin button:last-child {  
    margin-right: 0;  
}
```

```
.tabla-admin button:hover {  
    transform: scale(1.15);  
}
```

```
.tabla-admin button img {  
    width: 22px;  
    height: 22px;  
    vertical-align: middle;  
}  
  
}
```

```
.btn-agregar-etiqueta {  
    display: flex;  
    align-items: center;  
    background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
    color: #fff;  
    border: none;  
    border-radius: 8px;  
    padding: 10px 24px;  
    font-size: 1rem;  
    font-weight: 600;  
    cursor: pointer;  
    gap: 8px;  
    transition: background 0.2s, transform 0.2s;  
}  
  
.btn-agregar-etiqueta img {  
    height: 22px;  
    width: 22px;
```

{

```
.btn-agregar-etiqueta:hover, btn-agregar-etiqueta:focus {  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
transform: scale(1.05);  
}  
}
```

```
.header-etiquetas {  
display: flex;  
align-items: center;  
justify-content: space-between;  
margin-bottom: 18px;  
}  
}
```

```
.sidebar {  
background: #6b1e1e;  
width: 60px;  
min-width: 60px;  
max-width: 180px;  
margin-top: 0;  
padding: 30px 0 32px 0;
```

```
display: flex;  
flex-direction: column;  
align-items: stretch;  
gap: 18px;  
box-shadow: 2px 0 8px rgba(0,0,0,0.04);  
position: sticky;  
top: 100px;  
transition: width 0.6s cubic-bezier(.4,0,.2,1) , min-width 0.6s cubic-bezier(.4,0,.2,1);  
z-index: 120;  
overflow: visible;  
}  
  
}
```

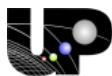
```
.sidebar-item {  
display: flex;  
align-items: center;  
gap: 8px;  
background: #6A1E1E;  
border: none;  
border-radius: 50px;  
color: #fff;  
padding: 12px 16px;  
font-size: 1rem;  
font-family: 'Montserrat', sans-serif;
```

```
cursor: pointer;  
transition: background 0.2s;  
white-space: nowrap;  
overflow: hidden;  
}  
  
}
```

```
.sidebar-item span {  
    opacity: 0;  
    width: 0;  
    margin-left: 0;  
    transition: opacity 0.5s, width 0.5s, margin-left 0.5s;  
    pointer-events: none;  
}  
  
}
```

```
.sidebar:hover {  
    width: 180px;  
    min-width: 180px;  
}  
  
}
```

```
.sidebar:hover .sidebar-item span {  
    opacity: 1;  
    width: auto;  
    margin-left: 8px;  
    pointer-events: auto;
```



}

```
.sidebar-item.active,  
.sidebar-item:hover {  
background: #912f2f;  
}
```

```
.sidebar-item img {  
height: 22px;  
width: 22px;  
flex-shrink: 0;  
}
```

```
.contenido {  
flex: 1;  
background: #fff;  
padding: 24px 48px 40px 48px;  
border-top-right-radius: 18px;  
border-bottom-right-radius: 18px;  
min-height: 600px;  
box-shadow: 0 2px 8px rgba(0,0,0,0.04);  
}
```

```
.sidebar h1 {  
    font-size: 1.5rem;  
    color: #fff;  
    margin: 20px;  
    text-align: center;  
    font-weight: 600;  
    max-width: 170px;  
    white-space: normal;  
    line-height: 1.2;  
    word-break: break-word;  
    margin-left: auto;  
    margin-right: auto;  
}  
  
}
```

```
.vista-admin {  
    display: block;  
    animation: fadeln 0.3s;  
}  
  
}
```

```
@keyframes fadeln {  
    from { opacity: 0; }  
    to { opacity: 1; }
```

}

```
.vista-admin h2 {  
    color: #6b1e1e;  
    font-size: 2rem;  
    margin-bottom: 28px;  
    font-weight: 600;  
    letter-spacing: 1px;  
}
```

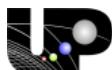
```
.rectangle-container {  
    width: 100%;  
    background: #912f2f;  
    color: #fff;  
    display: flex;  
    flex-wrap: wrap;  
    align-items: flex-start;  
    justify-content: space-between;  
    padding: 32px 40px 20px 40px;  
    box-sizing: border-box;  
    gap: 32px;  
    font-family: 'Montserrat', sans-serif;  
}
```

```
.contctanos {  
    font-size: 1.2rem;  
    font-weight: 600;  
    margin-bottom: 16px;  
    grid-column: 1 / -1;  
}  
  
}
```

```
.contactos-grid {  
    display: grid;  
    grid-template-columns: repeat(2, 1fr);  
    gap: 18px 40px;  
}  
  
}
```

```
.gmail-1-1-parent,  
.llamada-telefonica-2-parent,  
.logotipo-de-instagram-2-parent,  
.facebook-1-1-parent {
```

```
    display: flex;  
    align-items: center;  
    gap: 10px;  
    font-size: 1rem;
```



Universidad Politécnica  
de Chiapas

}



.llamada-telefonica-2-icon {

width: 26px;

height: 26px;

object-fit: contain;

}

.sazonpatrimonialgmailcom {

font-size: 1rem;

color: #fff;

}

.footer-logos {

display: flex;

align-items: center;

gap: 18px;

}

.logo-blanco-2,

.logo-equipo,

.logo-universidad {

height: 120px;

```
width: auto;  
object-fit: contain;  
border-radius: 8px;  
}
```

## vistaPrincipalRestaurantero

```
html, body {  
width: 100%;  
margin: 0;  
padding: 0;  
box-sizing: border-box;  
overflow-x: hidden;  
}
```

```
*, *::before, *::after {  
box-sizing: inherit;  
}
```

```
.vista-principal {  
width: 100%;  
}
```

```
.header-principal {  
position: fixed;
```

```
top: 0;  
z-index: 1000;  
width: 100%;  
background: #6b1e1e;  
color: #fff;  
box-shadow: 0 2px 8px rgba(0,0,0,0.04);  
}
```

```
.rectangle-parent {  
display: flex;  
align-items: center;  
justify-content: space-between;  
height: 100px;  
padding-left: 40px;  
padding-right: 40px;  
gap: 32px;  
}
```

```
.logo-blanco-parent {  
display: flex;  
align-items: center;  
gap: 10px;  
}
```

```
.logo-blanco {  
    height: 80px;  
    width: auto;  
}  
  
}
```

```
.catalogo-de-restaurantes {  
    font-size: 1.6rem;  
    max-width: 200px;  
    font-weight: 400;  
    color: #fff;  
    font-family: 'Montserrat', sans-serif;  
}  
  
}
```

```
.acciones-usuario {  
    margin-left: auto;  
    display: flex;  
    gap: 18px;  
}  
  
}
```

```
.mi-restaurante-parent,  
.log-out-parent {  
    display: flex;
```

```
align-items: center;  
  
gap: 8px;  
  
background: #6A1E1E;  
  
border: none;  
  
border-radius: 50px;  
  
color: #fff;  
  
padding: 12px 22px;  
  
font-size: 1rem;  
  
font-family: 'Montserrat', sans-serif;  
  
cursor: pointer;  
  
transition: background 0.2s;  
  
}  
  
  
  
.mi-restaurante-parent:hover,  
.log-out-parent:hover {  
  
background: #912f2f;  
  
}  
  
  
  
.mi-restaurante-parent img,  
.log-out-parent img {  
  
height: 22px;  
  
width: 22px;  
  
}
```

```
.icono-restaurantero {
```

```
    display: flex;
```

```
    align-items: center;
```

```
}
```

```
.icono-restaurantero img {
```

```
    height: 48px;
```

```
    width: 48px;
```

```
    border-radius: 50%;
```

```
}
```

```
.calles-con-encanto-scaled-parent {
```

```
    position: relative;
```

```
    width: 100%;
```

```
    height: 700px;
```

```
    display: flex;
```

```
    align-items: center;
```

```
    justify-content: center;
```

```
    overflow: hidden;
```

```
    background: #000000;
```

```
}
```

```
.calles-con-encanto-scaled-icon {  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    object-fit: cover;  
    z-index: 1;  
    opacity: 0.7;  
}
```

```
.contenido-superpuesto {  
    position: relative;  
    z-index: 2;  
    width: 100%;  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    justify-content: center;  
}
```

```
.descubre-la-magia {  
    font-size: 2.8rem;
```

```
font-family: 'Montserrat', sans-serif;  
font-weight: bold;  
color: #fff;  
text-shadow: 1px 1px 8px #000;  
margin-bottom: 10px;  
text-align: center;  
}
```

```
.explora-los-mejores {  
font-size: 1.2rem;  
font-family: 'Montserrat', sans-serif;  
color: #fff;  
text-shadow: 1px 1px 8px #000;  
margin-bottom: 24px;  
text-align: center;  
}
```

```
.barra-busqueda {  
display: flex;  
align-items: center;  
background: #fff;  
border-radius: 25px;  
padding: 8px 20px;
```

```
box-shadow: 0 2px 8px rgba(0,0,0,0.08);  
width: 680px;  
max-width: 90%;  
margin: 0 auto;  
}
```

```
.barra-busqueda input::placeholder {  
color: #912F2F;  
opacity: 0.8;  
}
```

```
.busqueda-icono {  
width: 24px;  
height: 24px;  
margin-right: 10px;  
}
```

```
.barra-busqueda input[type="text"] {  
border: none;  
outline: none;  
font-size: 1rem;  
flex: 1;  
background: transparent;
```

```
padding: 8px 0;  
color: #912F2F;  
}
```

```
.barra-busqueda .buscar {  
background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
color: #fff;  
border: none;  
border-radius: 20px;  
padding: 8px 22px;  
font-size: 1rem;  
margin-left: 10px;  
cursor: pointer;  
transition: background 0.8s;  
}
```

```
.barra-busqueda .buscar:hover {  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
}
```

```
.barra-filtros-horizontal {  
display: flex;
```

```
flex-wrap: wrap;  
justify-content: center;  
align-items: center;  
gap: 16px;  
padding: 16px;  
border-radius: 30px;  
  
  
width: 90%;  
max-width: 950px;  
}
```

```
.filtro-item {  
display: flex;  
align-items: center;  
background: #fff;  
border: 1px solid #912f2f;  
border-radius: 30px;  
padding: 6px 14px;  
gap: 8px;  
min-width: 170px;  
}
```

```
.icono-filtro {
```

```
width: 20px;  
  
height: 20px;  
  
opacity: 0.85;  
  
}
```

```
.filtro-select {  
  
border: none;  
  
outline: none;  
  
background: transparent;  
  
color: #912f2f;  
  
font-family: 'Rubik', sans-serif;  
  
font-size: 0.95rem;  
  
width: 100%;  
  
cursor: pointer;  
  
}
```

```
.btn-filtrar {  
  
display: flex;  
  
align-items: center;  
  
gap: 10px;  
  
background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
  
color: #fff;  
  
border: none;
```

```
border-radius: 30px;  
  
padding: 10px 24px;  
  
font-size: 1rem;  
  
font-family: 'Montserrat', sans-serif;  
  
font-weight: bold;  
  
cursor: pointer;  
  
transition: transform 0.2s, box-shadow 0.3s;  
  
}
```

```
.btn-filtrar:hover {  
  
transform: scale(1.03);  
  
box-shadow: 0 6px 18px rgba(145, 47, 47, 0.2);  
  
}
```

```
.icono-botón {  
  
width: 18px;  
  
height: 18px;  
  
}
```

```
.filtros-abajo {  
  
position: absolute;  
  
bottom: 30px;  
  
left: 50%;
```

```
transform: translateX(-50%);  
  
z-index: 3;  
  
}
```

```
.line-parent {  
  
display: flex;  
  
flex-wrap: wrap;  
  
gap: 32px;  
  
width: 100%;  
  
justify-content: center;  
  
}
```

```
.contenido-principal {  
  
display: flex;  
  
flex-wrap: wrap;  
  
justify-content: center;  
  
padding: 32px 40px;  
  
width: 100%;  
  
margin: 0;  
  
padding: 32px 0;  
  
box-sizing: border-box;  
  
gap: 40px;  
  
}
```

```
.restaurante-card {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    background: #fff;  
    border-radius: 4px;  
    padding: 4px 4px 4px 4px;  
    cursor: pointer;  
    transition: box-shadow 0.3s ease, transform 0.3s ease;  
    width: 300px;  
}  
  
width: 300px;
```

```
.restaurante-card:hover,  
.restaurante-card:focus {  
    box-shadow: 0 6px 18px rgba(4, 4, 4, 0.166);  
    border-color: #912f2f;  
    transform: translateY(-4px) scale(1.03);  
}
```

```
.restaurante-img {  
    width: 100%;  
    height: 200px;
```

```
border-radius: 4px;  
object-fit: cover;  
box-shadow: 0 2px 6px rgba(0,0,0,0.1);  
margin-bottom: 8px;  
}
```

```
.restaurante-info {  
display: flex;  
flex-direction: column;  
gap: 6px;  
justify-content: center;  
width: 100%;  
}
```

```
.restaurante-nombre {  
font-size: 1.3rem;  
font-weight: 600;  
margin: 0;  
color: #912f2f;  
font-family: 'Rubik', sans-serif;  
}
```

```
.restaurante-tipos,
```

```
.restaurante-horario {  
    font-size: smaller;  
    color: #912f2f;  
    font-family: 'Rubik', sans-serif;  
}  
  
.rectangle-container {  
    width: 100%;  
    background: #912f2f;  
    color: #fff;  
    display: flex;  
    flex-wrap: wrap;  
    align-items: flex-start;  
    justify-content: space-between;  
    padding: 32px 40px 20px 40px;  
    box-sizing: border-box;  
    gap: 32px;  
    font-family: 'Montserrat', sans-serif;  
}  
  
.contctanos {  
    font-size: 1.2rem;
```

```
font-weight: 600;  
  
margin-bottom: 16px;  
  
grid-column: 1 / -1;  
  
}
```

```
.contactos-grid {  
  
display: grid;  
  
grid-template-columns: repeat(2, 1fr);  
  
gap: 18px 40px;  
  
}
```

```
.gmail-1-1-parent,  
.llamada-telefonica-2-parent,  
.logotipo-de-instagram-2-parent,  
.facebook-1-1-parent {
```

```
display: flex;  
  
align-items: center;  
  
gap: 10px;  
  
font-size: 1rem;  
  
}
```

```
.llamada-telefonica-2-icon {  
  
width: 26px;
```

```
height: 26px;  
object-fit: contain;  
}  
  
.
```

```
.sazonpatrimonialgmailcom {  
font-size: 1rem;  
color: #fff;  
}
```

```
.footer-logos {  
display: flex;  
align-items: center;  
gap: 18px;  
}
```

```
.logo-blanco-2,  
.logo-equipo,  
.logo-universidad {  
height: 120px;  
width: auto;  
object-fit: contain;  
border-radius: 8px;  
}
```

## vistaRestaurante

```
.icon-16, .icon-20 {
```

```
    width: 20px;
```

```
    height: 20px;
```

```
}
```

```
.icon-32 {
```

```
    width: 40px;
```

```
    height: 40px;
```

```
}
```

```
.icon-middle {
```

```
    vertical-align: middle;
```

```
}
```

```
.restaurante-vista-container {
```

```
    max-width: 90%;
```

```
    margin: 140px auto 40px auto;
```

```
    background: #fff;
```

```
    border-radius: 18px;
```

```
    padding: 32px 36px 36px 36px;
```

```
    font-family: 'Montserrat', sans-serif;
```

```
position: relative;  
}  
  
}
```

```
.btn-back {  
  
background: none;  
  
border: none;  
  
position: absolute;  
  
top: 24px;  
  
left: 24px;  
  
cursor: pointer;  
  
padding: 0;  
  
}
```

```
.restaurante-nombre {  
  
text-align: center;  
  
font-size: 3rem;  
  
font-weight: 700;  
  
color: #912f2f;  
  
margin: 0 0 8px 0;  
  
letter-spacing: 1px;  
  
}
```

```
.restaurante-divider {
```

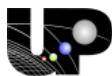
```
border: none;  
  
border-top: 4px solid #912f2f;  
  
width: 80%;  
  
margin: 0 auto 24px auto;  
  
border-radius: 2px;  
  
}
```

```
.restaurante-galeria {  
  
display: flex;  
  
gap: 2%;  
  
margin-bottom: 28px;  
  
justify-content: center;  
  
flex-wrap: wrap;  
  
align-items: flex-start;  
  
}
```

```
.galeria-principal {  
  
width: 60%;  
  
height: 400px;  
  
object-fit: cover;  
  
border-radius: 8px;  
  
box-shadow: 0 2px 8px rgba(0,0,0,0.08);  
  
}
```

```
.galeria-secundaria {  
    display: flex;  
    flex-direction: column;  
    gap: 15px;  
    justify-content: flex-start;  
    width: 35%;  
    height: 400px;  
}  
  
.galeria-secundaria img {  
    width: 100%;  
    height: calc(50% - 7.5px);  
    object-fit: cover;  
    border-radius: 8px;  
    box-shadow: 0 1px 4px rgba(0,0,0,0.07);  
}
```

```
.restaurante-info-flex {  
    display: grid;  
    grid-template-columns: repeat(4, 1fr);  
    gap: 24px;  
    width: 90%;  
    max-width: 1100px;  
    margin: 0 auto 32px auto;
```



}

.restaurante-info-box h2 {

font-size: 2rem;

color: #912f2f;

margin: 0 0 8px 0;

font-weight: 600;

}

.restaurante-contactos div {

margin-bottom: 6px;

font-size: 1.3rem;

display: flex;

align-items: center;

gap: 6px;

color:#912f2f;

}

.restaurante-caracteristicas {

display: flex;

align-items: flex-start;

gap: 14px;

```
background: transparent;  
box-shadow: none;  
padding: 0;  
}  
  
}
```

```
.caracteristicas-izq {  
display: flex;  
align-items: flex-start;  
gap: 12px;  
}
```

```
.icon-etiqueta {  
width: 32px;  
height: 32px;  
margin-top: 2px;  
flex-shrink: 0;  
}
```

```
.caracteristicas-lista {  
display: flex;  
flex-direction: column;  
gap: 10px;  
align-items: flex-start;
```

}

.caracteristica-item {

display: flex;

align-items: center;

gap: 8px;

}

.caracteristica {

background: #912f2f;

color: #fff;

border-radius: 18px;

padding: 6px 16px;

min-width: 160px;

max-width: 160px;

text-align: center;

white-space: nowrap;

box-sizing: border-box;

display: flex;

justify-content: center;

align-items: center;

}

```
.restaurante-info-box {  
background: #fff;  
padding: 20px;  
border-radius: 12px;  
display: flex;  
flex-direction: column;  
align-items: flex-start;  
font-size: 1.3rem;  
color: #912f2f;  
}  
  
-----  
  
-----
```

```
.btn-menu {  
display: flex;  
align-items: center;  
gap: 8px;  
background: linear-gradient(90deg, #912F2F 60%, #6A1E1E 100%);  
color: #fff;  
border: none;  
border-radius: 8px;  
padding: 12px 28px;  
font-size: 1.1rem;  
font-weight: 600;
```

```
cursor: pointer;  
  
margin: 0 auto;  
  
box-shadow: 0 2px 8px rgba(0,0,0,0.08);  
  
transition: background 0.2s;  
  
}  
  
.btn-menu:hover {  
  
background: linear-gradient(90deg, #6A1E1E 60%, #912F2F 100%);  
  
}
```

```
.restaurante-info-flex,  
  
.restaurante-caracteristicas,  
  
.btn-menu {  
  
max-width: 1100px  
  
}
```

```
html, body {  
  
width: 100%;  
  
margin: 0;  
  
padding: 0;  
  
box-sizing: border-box;  
  
overflow-x: hidden;  
  
}
```

```
* , *::before, *::after {
```

```
  box-sizing: inherit;
```

```
}
```

```
.vista-principal {
```

```
  width: 100%;
```

```
}
```

```
.header-principal {
```

```
  position: fixed;
```

```
  top: 0;
```

```
  z-index: 1000;
```

```
  width: 100%;
```

```
  background: #6b1e1e;
```

```
  color: #fff;
```

```
  box-shadow: 0 2px 8px rgba(0,0,0,0.04);
```

```
}
```

```
.rectangle-parent {
```

```
  display: flex;
```

```
  align-items: center;
```

```
  justify-content: space-between;
```

```
  height: 100px;
```

```
padding-left: 40px;  
padding-right: 40px;  
gap: 32px;  
}  
  
}
```

```
.logo-blanco-parent {  
    display: flex;  
    align-items: center;  
    gap: 10px;  
}
```

```
.logo-blanco {  
    height: 80px;  
    width: auto;  
}
```

```
.catlogo-de-restaurantes {  
    font-size: 1.6rem;  
    font-weight: 400;  
    color: #fff;  
    font-family: 'Montserrat', sans-serif;  
    max-width: 200px;  
}
```

```
.rectangle-container {  
    width: 100%;  
    background: #912f2f;  
    color: #fff;  
    display: flex;  
    flex-wrap: wrap;  
    align-items: flex-start;  
    justify-content: space-between;  
    padding: 32px 40px 20px 40px;  
    box-sizing: border-box;  
    gap: 32px;  
    font-family: 'Montserrat', sans-serif;  
}
```

```
.contctanos {  
    font-size: 1.2rem;  
    font-weight: 600;  
    margin-bottom: 16px;  
    grid-column: 1 / -1;  
}
```

```
.contactos-grid {  
    display: grid;  
    grid-template-columns: repeat(2, 1fr);  
    gap: 18px 40px;  
}  
  
.
```

```
.gmail-1-1-parent,  
.llamada-telefonica-2-parent,  
.logotipo-de-instagram-2-parent,  
.facebook-1-1-parent {
```

```
    display: flex;  
    align-items: center;  
    gap: 10px;  
    font-size: 1rem;  
}  
  
.
```

```
.llamada-telefonica-2-icon {  
    width: 26px;  
    height: 26px;  
    object-fit: contain;  
}
```

```
.sazonpatrimonialgmailcom {
```

```
font-size: 1rem;  
  
color: #fff;  
  
}
```

```
.footer-logos {  
  
display: flex;  
  
align-items: center;  
  
gap: 18px;  
  
}
```

```
.logo-blanco-2,  
  
.logo-equipo,  
  
.logo-universidad {  
  
height: 120px;  
  
width: auto;  
  
object-fit: contain;  
  
border-radius: 8px;  
  
}
```

```
.modal-overlay {  
  
display: none;  
  
position: fixed;  
  
top: 0;
```

```
left: 0;  
width: 100%;  
height: 100%;  
background: rgba(0, 0, 0, 0.5);  
z-index: 9999;  
justify-content: center;  
align-items: center;  
opacity: 0;  
transition: opacity 0.3s ease;  
}
```

```
.modal-overlay.show {  
    opacity: 1;  
}
```

```
.modal-content {  
    position: relative;  
    background: transparent;  
    max-width: 90%;  
    max-height: 95vh;  
    overflow: visible;  
    border-radius: 4px;  
    display: flex;
```

```
flex-direction: column;  
justify-content: center;  
transform: scale(0.7) translateY(-20px);  
transition: transform 0.3s ease, opacity 0.3s ease;  
opacity: 0;  
}  
  
}
```

```
.modal-overlay.show .modal-content {  
    transform: scale(1) translateY(0);  
    opacity: 1;  
}
```

```
.close-modal {  
    position: absolute;  
    top: 10px;  
    right: 15px;  
    background: none;  
    border: none;  
    font-size: 24px;  
    cursor: pointer;  
    color: #912F2F;  
    z-index: 10000;  
    width: 30px;
```

```
height: 30px;  
  
display: flex;  
  
align-items: center;  
  
justify-content: center;  
  
border-radius: 50%;  
  
background: rgba(255, 255, 255, 0.8);  
  
}
```

```
.close-modal:hover {  
  
background: rgba(255, 255, 255, 1);  
  
color: #B64040;  
  
}
```

### 5.2.3 JavaScript

api

```
const API_CONFIG = {  
  
BASE_URL: 'http://52.23.26.163:7070',  
  
ENDPOINTS: {  
  
RESTAURANTEROS: '/api/restauranteros',  
  
USUARIOS: '/api/usuarios',  
  
ADMIN: '/api/admin'  
  
}  
  
};
```

```
class SazonAPI {  
  
    static async request(endpoint, options = {}) {  
  
        const url = `${API_CONFIG.BASE_URL}${endpoint}`;  
  
        const defaultOptions = {  
  
            headers: {  
  
                'Content-Type': 'application/json',  
  
            },  
  
        };  
  
        try {  
  
            const response = await fetch(url, { ...defaultOptions, ...options });  
  
            if (!response.ok) {  
  
                throw new Error(`HTTP ${response.status}: ${response.statusText}`);  
  
            }  
  
            const contentType = response.headers.get('content-type');  
  
            if (contentType && contentType.includes('application/json')) {  
  
                return await response.json();  
  
            } else {  
  
                return await response.text();  
  
            }  
  
        } catch (error) {  
  
        }  
    }  
}
```

```
        console.error('API Error:', error);

        throw error;

    }

}

static async getAllRestauranteros() {

    return this.request(API_CONFIG.ENDPOINTS.RESTAURANTEROS);

}

static async getRestauranteroById(id) {

    return this.request(` ${API_CONFIG.ENDPOINTS.RESTAURANTEROS}/${id}`);

}

static async createRestaurantero(data) {

    return this.request(API_CONFIG.ENDPOINTS.RESTAURANTEROS, {

        method: 'POST',

        body: JSON.stringify(data)

    });

}

static async updateRestaurantero(id, data) {

    return this.request(` ${API_CONFIG.ENDPOINTS.RESTAURANTEROS}/${id}` , {

        method: 'PUT',
```

```
        body: JSON.stringify(data)

    });

}

static async deleteRestaurantero(id) {

    return this.request(` ${API_CONFIG.ENDPOINTS.RESTAURANTEROS}/${id}` , {

        method: 'DELETE'

    });

}

static async getAllUsuarios() {

    return this.request(API_CONFIG.ENDPOINTS.USUARIOS);

}

static async createUsuario(data) {

    return this.request(API_CONFIG.ENDPOINTS.USUARIOS, {

        method: 'POST',

        body: JSON.stringify(data)

    });

}

static async loginUsuario(credentials) {

    return this.request(` ${API_CONFIG.ENDPOINTS.USUARIOS}/login` , {
```

```
    method: 'POST',  
  
    body: JSON.stringify(credentials)  
});
```

```
}
```

```
static async getAllSolicitudes() {  
  
    return this.request('/solicitudes');  
  
}
```

```
class UIUtils {  
  
    static showMessage(message, type = 'info') {  
  
        const toast = document.createElement('div');  
  
        toast.className = `toast toast-${type}`;  
  
        toast.textContent = message;
```

```
        Object.assign(toast.style, {
```

```
            position: 'fixed',  
  
            top: '20px',  
  
            right: '20px',  
  
            padding: '15px 20px',  
  
            borderRadius: '5px',  
  
            color: 'white',
```

```
zIndex: '9999',
fontWeight: 'bold',
backgroundColor: type === 'error' ? '#e74c3c' : type === 'success' ? '#27ae60' :
'#3498db'
});
```

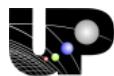
```
document.body.appendChild(toast);
```

```
setTimeout(() => {
  if (toast.parentNode) {
    toast.parentNode.removeChild(toast);
  }
}, 3000);
}
```

```
static showError(message) {
  this.showMessage(message, 'error');
}
```

```
static showSuccess(message) {
  this.showMessage(message, 'success');
}
```

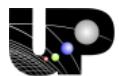
```
static showLoading(show = true) {  
  
    let loader = document.getElementById('global-loader');  
  
  
    if (show && !loader) {  
  
        loader = document.createElement('div');  
  
        loader.id = 'global-loader';  
  
        loader.innerHTML = '<div class="spinner"></div>';  
  
  
  
  
        Object.assign(loader.style, {  
  
            position: 'fixed',  
  
            top: '0',  
  
            left: '0',  
  
            width: '100%',  
  
            height: '100%',  
  
            backgroundColor: 'rgba(0,0,0,0.5)',  
  
            display: 'flex',  
  
            justifyContent: 'center',  
  
            alignItems: 'center',  
  
            zIndex: '9998'  
        });  
  
  
  
  
        const spinner = loader.querySelector('.spinner');  
  
        Object.assign(spinner.style, {
```



```
width: '50px',  
  
height: '50px',  
  
border: '5px solid #f3f3f3',  
  
borderTop: '5px solid #3498db',  
  
borderRadius: '50%',  
  
animation: 'spin 1s linear infinite'  
});
```

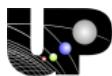
```
document.body.appendChild(loader);  
  
} else if (!show && loader) {  
  
    loader.remove();  
  
}  
  
}
```

```
const style = document.createElement('style');  
  
style.textContent = `  
  
@keyframes spin {  
  
    0% { transform: rotate(0deg); }  
  
    100% { transform: rotate(360deg); }  
  
}  
  
`;  
  
document.head.appendChild(style);
```



## auth

```
class AuthManager {  
  
    static currentUser = null;  
  
    static async login(email, password) {  
  
        try {  
            UIUtils.showLoading(true);  
  
            const credentials = {  
                correo: email,  
                contrasena: password  
            };  
  
            const response = await SazonAPI.loginUsuario(credentials);  
  
            if (response && response.user) {  
                this.currentUser = response.user;  
                localStorage.setItem('sazon_user', JSON.stringify(response.user));  
                UIUtils.showSuccess('¡Bienvenido a SazonPT!');  
            }  
        } catch (error) {  
            UIUtils.showError(error.message);  
        } finally {  
            UIUtils.hideLoading();  
        }  
    }  
}
```



```
        return true;

    } else {

        throw new Error('Credenciales inválidas');

    }

} catch (error) {

    UIUtils.showError('Error al iniciar sesión: ' + error.message);

    return false;

} finally {

    UIUtils.showLoading(false);

}

}

static async register(userData) {

    try {

        UIUtils.showLoading(true);

        if (!userData.nombreU || !userData.correo || !userData.contrasena) {

            throw new Error('Todos los campos son obligatorios');

        }

        if (!userData.correo.includes('@')) {

            throw new Error('Formato de correo inválido');

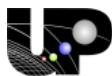
        }

    }

}
```

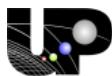
```
if (userData.contrasena.length < 6) {  
  
    throw new Error('La contraseña debe tener al menos 6 caracteres');  
  
}  
  
  
const response = await SazonAPI.createUser(userData);  
  
UIUtils.showSuccess('¡Registro exitoso! Ya puedes iniciar sesión');  
  
return true;  
  
  
} catch (error) {  
  
    UIUtils.showError('Error en el registro: ' + error.message);  
  
    return false;  
  
} finally {  
  
    UIUtils.showLoading(false);  
  
}  
  
}  
  
  
static logout() {  
  
    this.currentUser = null;  
  
    localStorage.removeItem('sazon_user');  
  
    UIUtils.showSuccess('Sesión cerrada exitosamente');  
  
    window.location.href = '../index.html';  
  
}
```

```
static getCurrentUser() {  
  
    if (!this.currentUser) {  
  
        const stored = localStorage.getItem('sazon_user');  
  
        if (stored) {  
  
            this.currentUser = JSON.parse(stored);  
  
        }  
  
    }  
  
    return this.currentUser;  
  
}  
  
  
static isLoggedIn() {  
  
    return this.getCurrentUser() !== null;  
  
}  
  
  
static requireAuth() {  
  
    if (!this.isLoggedIn()) {  
  
        UIUtils.showError('Debes iniciar sesión para acceder a esta página');  
  
        window.location.href = './pages/loginSelect.html';  
  
        return false;  
  
    }  
  
    return true;  
  
}
```



}

```
function initLoginForm() {  
  
    const loginForm = document.getElementById('login-form');  
  
    if (loginForm) {  
  
        loginForm.addEventListener('submit', async (e) => {  
  
            e.preventDefault();  
  
            const email = document.getElementById('email').value;  
  
            const password = document.getElementById('password').value;  
  
            const success = await AuthManager.login(email, password);  
  
            if (success) {  
  
                const user = AuthManager.getCurrentUser();  
  
                if (user.tipo === 'restaurantero') {  
  
                    window.location.href = 'vistaPrincipalRestaurantero.html';  
  
                } else if (user.tipo === 'admin') {  
  
                    window.location.href = 'vistaPrincipalAdmin.html';  
  
                } else {  
  
                    window.location.href = '../index.html';  
  
                }  
            }  
        });  
    }  
}
```



}

}

```
function initRegisterForm() {  
  
    const registerForm = document.getElementById('register-form');  
  
    if (registerForm) {  
  
        registerForm.addEventListener('submit', async (e) => {  
  
            e.preventDefault();  
  
            const userData = {  
  
                nombreU: document.getElementById('nombre').value,  
  
                correo: document.getElementById('email').value,  
  
                contraseña: document.getElementById('password').value,  
  
                telefono: document.getElementById('telefono')?.value || null,  
  
                dirección: document.getElementById('dirección')?.value || null  
            };  
  
            const success = await AuthManager.register(userData);  
  
            if (success) {  
  
                setTimeout(() => {  
  
                    window.location.href = 'loginSelect.html';  
  
                }, 2000);  
  
            }  
        }  
    }  
}
```

```
    });
}

}

document.addEventListener('DOMContentLoaded', function() {

  initLoginForm();
  initRegisterForm();

  const user = AuthManager.getCurrentUser();

  if (user) {
    const userInfo = document.getElementById('user-info');
    if (userInfo) {
      userInfo.textContent = `Bienvenido, ${user.nombreU}`;
    }
  }
});

encuesta

function initEncuestaForm() {

  const form = document.getElementById('encuestaForm');
  const atraccion = document.getElementById('atraccion');
  const origen = document.getElementById('origen');

  if (!form || !atraccion || !origen) {
```

```
    return;
```

```
}
```

```
form.addEventListener('submit', async function(e) {
```

```
    if (!validateForm()) {
```

```
        e.preventDefault();
```

```
        return;
```

```
}
```

```
    e.preventDefault();
```

```
const restaurante = window.restauranteActual;
```

```
const params = new URLSearchParams(window.location.search);
```

```
const restaurantId = params.get('id');
```

```
const correo = params.get('correo');
```

```
const data = {
```

```
    atraccion: atraccion.value,
```

```
    origen: origen.value,
```

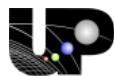
```
    restaurantId: restaurantId || restaurante?.id,
```

```
    correoRestaurante: correo || restaurante?.correo,
```

```
    nombreRestaurante: restaurante?.restaurante
```

```
};
```

```
try {  
  
    try {  
  
        const estadisticasPayload = {  
  
            correo: correo || restaurante?.correo,  
  
            accion: 'incrementar'  
  
        };  
  
  
        if (origen.value === 'nacional') {  
  
            estadisticasPayload.incrementar_nacional = 1;  
  
        } else if (origen.value === 'extranjero') {  
  
            estadisticasPayload.incrementar_extranjero = 1;  
  
        }  
  
  
        estadisticasPayload.incrementar_descargas = 1;  
  
  
  
        const getResponse = await  
fetch(`http://52.23.26.163:7070/estadisticas?correo=${encodeURIComponent(correo ||  
restaurante?.correo)}`);  
  
  
        if (!getResponse.ok) {  
  
            const errorText = await getResponse.text();  
  
            throw new Error(`Error al obtener estadística: ${getResponse.status} -  
${errorText}`);  
    }  
}
```



}

```
const responseText = await getResponse.text();
```

```
let estadisticaExistente;
```

```
try {
```

```
    estadisticaExistente = JSON.parse(responseText);
```

```
} catch (parseError) {
```

```
    throw new Error(`Respuesta no es JSON válido:
```

```
 ${responseText.substring(0, 100)}');
```

```
}
```

```
let estadisticaId;
```

```
let valoresActuales = {
```

```
    nacional: 0,
```

```
    extranjero: 0,
```

```
    descargas: 0,
```

```
    comida: 0,
```

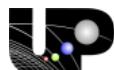
```
    vista: 0,
```

```
    horario: 0,
```

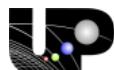
```
    recomendacion: 0,
```

```
    ubicacion: 0
```

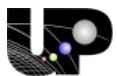
```
};
```



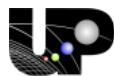
```
const correoObjectivo = correo || restaurante?.correo;  
  
console.log('🔴 Buscando estadística para correo:', correoObjectivo);  
  
if (Array.isArray(estadisticaExistente)) {  
  
    const estadisticaCorrecta = estadisticaExistente.find(est => est.correo ===  
correoObjectivo);  
  
    if (estadisticaCorrecta) {  
  
        estadisticaId = estadisticaCorrecta.id_estadistica;  
  
        valoresActuales.correoOriginal = estadisticaCorrecta.correo;  
  
        valoresActuales.nacional = estadisticaCorrecta.nacional || 0;  
  
        valoresActuales.extranjero = estadisticaCorrecta.extranjero || 0;  
  
        valoresActuales.descargas = estadisticaCorrecta.descargas || 0;  
  
        valoresActuales.comida = estadisticaCorrecta.comida || 0;  
  
        valoresActuales.vista = estadisticaCorrecta.vista || 0;  
  
        valoresActuales.horario = estadisticaCorrecta.horario || 0;  
  
        valoresActuales.recomendacion = estadisticaCorrecta.recomendacion ||  
0;  
  
        valoresActuales.ubicacion = estadisticaCorrecta.ubicacion || 0;  
  
        console.log('✅ Estadística encontrada para correo:', correoObjectivo);  
  
    } else {  
  
        console.error('🔴 No se encontró estadística para el correo:',  
correoObjectivo);  
    }  
}
```



```
        console.error('📋 Correos disponibles:', estadisticaExistente.map(est =>
est.correo));  
  
            throw new Error(`No se encontró estadística para el correo:  
${correoObjectivo}`);  
  
    }  
  
} else if (estadisticaExistente.id_estadistica) {  
  
    if (estadisticaExistente.correo === correoObjectivo) {  
  
        estadisticaId = estadisticaExistente.id_estadistica;  
  
        valoresActuales.correoOriginal = estadisticaExistente.correo;  
  
        valoresActuales.nacional = estadisticaExistente.nacional || 0;  
  
        valoresActuales.extranjero = estadisticaExistente.extranjero || 0;  
  
        valoresActuales.descargas = estadisticaExistente.descargas || 0;  
  
        valoresActuales.comida = estadisticaExistente.comida || 0;  
  
        valoresActuales.vista = estadisticaExistente.vista || 0;  
  
        valoresActuales.horario = estadisticaExistente.horario || 0;  
  
        valoresActuales.recomendacion = estadisticaExistente.recomendacion ||  
0;  
  
        valoresActuales.ubicacion = estadisticaExistente.ubicacion || 0;  
  
        console.log('✅ Estadística directa encontrada para correo:',  
correoObjectivo);  
  
    } else {  
  
        console.error('❗ El correo no coincide:', {  
  
            esperado: correoObjectivo,  
  
            recibido: estadisticaExistente.correo  
        });  
    }  
}
```



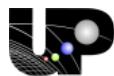
```
});  
  
        throw new Error(`Correo no coincide. Esperado: ${correoObjetivo},  
Recibido: ${estadisticaExistente.correo}`);  
  
    }  
  
} else {  
  
    console.error('✖ Estructura de estadística no reconocida.',  
estadisticaExistente);  
  
    throw new Error('No se encontró id_estadistica en la respuesta');  
  
}  
  
  
if (!estadisticaId) {  
  
    throw new Error('ID de estadística es null o undefined');  
  
}  
  
  
console.log('📊 Valores actuales:', valoresActuales);  
  
  
  
const nuevosValores = {  
  
    correo: valoresActuales.correoOriginal || (correo || restaurante?.correo),  
  
    nacional: valoresActuales.nacional,  
  
    extranjero: valoresActuales.extranjero,  
  
    descargas: valoresActuales.descargas + 1,  
  
    comida: valoresActuales.comida,  
  
    vista: valoresActuales.vista,
```



```
horario: valoresActuales.horario,  
recomendacion: valoresActuales.recomendacion,  
ubicacion: valoresActuales.ubicacion  
};
```

```
if (origen.value === 'nacional') {  
    nuevosValores.nacional = valoresActuales.nacional + 1;  
}  
else if (origen.value === 'extranjero') {  
    nuevosValores.extranjero = valoresActuales.extranjero + 1;  
}
```

```
if (atraccion.value === 'comida') {  
    nuevosValores.comida = valoresActuales.comida + 1;  
}  
else if (atraccion.value === 'vista') {  
    nuevosValores.vista = valoresActuales.vista + 1;  
}  
else if (atraccion.value === 'horario') {  
    nuevosValores.horario = valoresActuales.horario + 1;  
}  
else if (atraccion.value === 'recomendacion') {  
    nuevosValores.recomendacion = valoresActuales.recomendacion + 1;  
}  
else if (atraccion.value === 'ubicacion') {  
    nuevosValores.ubicacion = valoresActuales.ubicacion + 1;  
}
```



```
console.log('📈 Valores incrementados:', {
```

```
    antes: valoresActuales,
```

```
    despues: nuevosValores,
```

```
    incremento_origen: origen.value,
```

```
    incremento_atraccion: atraccion.value
```

```
});
```

```
console.log('>ID ID de estadística encontrado:', estadisticaId);
```

```
        console.log('🔗 URL PUT final:',
```

```
`http://52.23.26.163:7070/estadisticas/${estadisticaId}`);
```

```
const encuestaResponse = await  
fetch(`http://52.23.26.163:7070/estadisticas/${estadisticaId}` , {
```

```
    method: 'PUT',
```

```
    headers: { 'Content-Type': 'application/json' },
```

```
    body: JSON.stringify(nuevosValores)
```

```
});
```

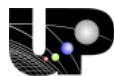
```
console.log('📈 Respuesta del servidor de estadísticas:', {
```

```
    status: encuestaResponse.status,
```

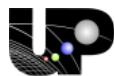
```
    statusText: encuestaResponse.statusText,
```

```
    ok: encuestaResponse.ok
```

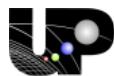
```
});
```



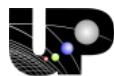
```
if (encuestaResponse.ok) {  
  
    const resultado = await encuestaResponse.text();  
  
    console.log('✅ Respuesta PUT cruda:', resultado);  
  
  
try {  
  
    const resultadoParsed = JSON.parse(resultado);  
  
    console.log('✅ Estadísticas actualizadas:', resultadoParsed);  
  
} catch (parseError) {  
  
    console.log('✅ Respuesta PUT (texto):', resultado);  
  
}  
  
} else {  
  
try {  
  
    const errorText = await encuestaResponse.text();  
  
    console.error('❌ Error del servidor (estadísticas):', {  
  
        status: encuestaResponse.status,  
  
        error: errorText  
  
    });  
  
} catch (e) {  
  
    console.warn('⚠️ Error al actualizar estadísticas:',  
    encuestaResponse.status);  
  
}  
  
}
```



```
        } catch (estadisticasError) {  
  
            console.error('✖ Error al enviar estadísticas:', estadisticasError);  
  
        }  
  
        if (restaurante && restaurante.menu) {  
  
            console.log('📄 Descargando menú desde:', restaurante.menu);  
  
            let menuUrl = restaurante.menu;  
  
            if (menuUrl.includes('localhost:7070')) {  
  
                menuUrl = menuUrl.replace('localhost:7070', '52.23.26.163:7070');  
  
                console.log('🔄 URL corregida para descarga:', menuUrl);  
  
            }  
  
            if (!menuUrl.startsWith('http')) {  
  
                menuUrl = `http://52.23.26.163:7070${menuUrl}`;  
  
            }  
  
            try {  
  
                console.log('⬇ Intentando descargar desde:', menuUrl);  
  
                const menuResponse = await fetch(menuUrl);  
  
            } catch (error) {  
  
                console.error('⚠ Error al descargar el menú:', error);  
  
            }  
  
        }  
  
    } catch (error) {  
  
        console.error('⚠ Error al enviar estadísticas:', error);  
  
    }  
  
}
```



```
console.log('📥 Respuesta de descarga:', {  
    status: menuResponse.status,  
    statusText: menuResponse.statusText,  
    ok: menuResponse.ok  
});  
  
if (menuResponse.ok) {  
    const blob = await menuResponse.blob();  
    const downloadUrl = window.URL.createObjectURL(blob);  
  
    const link = document.createElement('a');  
    link.href = downloadUrl;  
    link.download = `menu_${restaurante.restaurante || 'restaurante'}.pdf`;  
    document.body.appendChild(link);  
    link.click();  
    document.body.removeChild(link);  
  
    window.URL.revokeObjectURL(downloadUrl);  
  
    console.log('✅ Menú descargado exitosamente');  
} else {  
    console.error('❌ Error al descargar el menú:', menuResponse.status);  
    alert('No se pudo descargar el menú. Por favor, contacta al restaurante.');
```



```
    }

} catch (downloadError) {

    console.error(X Error en la descarga:', downloadError);

        if (downloadError.message.includes('CORS') ||

downloadError.message.includes('NetworkError')) {

            alert('Error de conexión al descargar el menú. El servidor puede no estar

disponible.');

        } else {

            alert('Error al descargar el menú.');

        }

    }

} else {

    console.warn(! No hay menú disponible para este restaurante');

    alert('Este restaurante no tiene menú disponible para descarga.');

}

if (typeof closeEncuestaModal === "function") {

    closeEncuestaModal();

} else if (typeof cerrarModalEncuesta === "function") {

    cerrarModalEncuesta();

}
```

```
        } catch (error) {  
  
            console.error('✖ Error al procesar la encuesta:', error);  
  
            alert('Ocurrió un error al enviar la encuesta.');
```

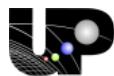
}

});

```
function validateForm() {  
  
    let isValid = true;  
  
    if (!atraccion.value) {  
  
        showError(atraccion, 'Por favor selecciona una respuesta');  
  
        isValid = false;  
  
    } else {  
  
        removeError(atraccion);  
  
    }  
  
    if (!origen.value) {  
  
        showError(origen, 'Por favor selecciona una respuesta');  
  
        isValid = false;  
  
    } else {  
  
        removeError(origen);  
  
    }  
  
    return isValid;  
}
```

```
function showError(element, message) {  
  
    removeError(element);  
  
    element.classList.add('error');  
  
    let errorSpan = null;  
  
    if (element.id === 'atraccion') {  
  
        errorSpan = document.getElementById('error-atraccion');  
  
    } else if (element.id === 'origen') {  
  
        errorSpan = document.getElementById('error-origen');  
  
    }  
  
    if (errorSpan) {  
  
        errorSpan.textContent = message;  
  
    }  
  
}
```

```
function removeError(element) {  
  
    element.classList.remove('error');  
  
    let errorSpan = null;  
  
    if (element.id === 'atraccion') {  
  
        errorSpan = document.getElementById('error-atraccion');  
  
    } else if (element.id === 'origen') {  
  
        errorSpan = document.getElementById('error-origen');  
  
    }  
  
    if (errorSpan) {
```



```
errorSpan.textContent = "";

}

}

function handleSuccessfulSubmit() {

}

function getOptionText(selectElement) {

    const selectedOption = selectElement.options[selectElement.selectedIndex];

    return selectedOption.text;

}

function descargarMenuPDF(restaurantId) {

    const pdfUrl = `./menus/menu_${restaurantId}.pdf`;

    const link = document.createElement('a');

    link.href = pdfUrl;

    link.download = `menu_${restaurantId}.pdf`;

    document.body.appendChild(link);

    link.click();

    document.body.removeChild(link);

}
```

```
window.initEncuestaForm = initEncuestaForm;
```

## estadísticas

```
document.addEventListener('DOMContentLoaded', function() {  
  
    const correoRestaurantero = sessionStorage.getItem('correo') ||  
localStorage.getItem('correo');  
  
    const idUsuario = sessionStorage.getItem('id') || localStorage.getItem('id');  
  
  
    if (!correoRestaurantero && !idUsuario) {  
  
        alert('Error: No se pudo identificar al restaurantero. Por favor, inicie sesión  
nuevamente.');//  
  
        window.location.href = 'loginRest.html';  
  
        return;  
  
    }  
  
  
    const parametros = new URLSearchParams();  
  
    if (correoRestaurantero) {  
  
        parametros.append('correo', correoRestaurantero);  
  
    }  
  
    if (idUsuario) {  
  
        parametros.append('id', idUsuario);  
  
    }  
  
  
    fetch(`http://52.23.26.163:7070/estadisticas?${parametros.toString()}`)
```

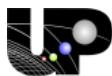
```
.then(res => {
    if (!res.ok) {
        throw new Error(`HTTP ${res.status}: ${res.statusText}`);
    }
    return res.json();
})

.then(estadisticas => {
    if (!estadisticas || typeof estadisticas !== 'object') {
        throw new Error('Datos de estadísticas inválidos');
    }
}

let datosEstadisticas = estadisticas;
if (Array.isArray(estadisticas)) {
    if (estadisticas.length === 0) {
        throw new Error('No se encontraron estadísticas para este restaurante');
    }
}

const estadisticaCorrecta = estadisticas.find(est => est.correo ===
correoRestaurantero);

if (estadisticaCorrecta) {
    datosEstadisticas = estadisticaCorrecta;
    console.log('✓ Estadística encontrada para correo:', correoRestaurantero);
```



```
console.log('📊 Datos específicos del restaurante:', datosEstadisticas);

} else {

    console.error('❌ No se encontró estadística para el correo:', correoRestaurantero);

    console.error('✉️ Correos disponibles:', estadisticas.map(est => est.correo));

    throw new Error(`No se encontraron estadísticas para el correo: ${correoRestaurantero}`);
}

}

const descargasSemanaEl = document.querySelector('.estadistica-menu p strong');

const descargasTotalesEl = document.querySelectorAll('.estadistica-menu p strong')[1];

const porcentajeSubidaEl = document.querySelector('.porcentaje-subida');

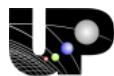
const totalDescargas = datosEstadisticas.descargas || 0;

if (descargasSemanaEl) {

    descargasSemanaEl.textContent = datosEstadisticas.descargasSemana || totalDescargas;

}

if (descargasTotalesEl) {
```



```
descargasTotalesEl.textContent = totalDescargas;

}

if (porcentajeSubidaEl) {

    const porcentaje = datosEstadisticas.porcentajeSubida || (totalDescargas > 0
? 15 : 0);

    porcentajeSubidaEl.textContent = `+${porcentaje}%`;

}

console.log('📊 Datos de descargas procesados:', {

    totalDescargas,

    descargasSemana: datosEstadisticas.descargasSemana || totalDescargas,

    porcentaje: datosEstadisticas.porcentajeSubida || 0

});

const nacional = datosEstadisticas.nacional || 0;

const extranjero = datosEstadisticas.extranjero || 0;

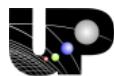
const total = nacional + extranjero;

let porcentajeNacional = 0;

let porcentajeExtranjero = 0;

if (total > 0) {

    porcentajeNacional = Math.round((nacional / total) * 100);
```



```
    porcentajeExtranjero = Math.round((extranjero / total) * 100);  
}  

```

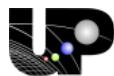
```
console.log('🌐 Datos de origen calculados:', {  
    nacional,  
    extranjero,  
    total,  
    porcentajeNacional,  
    porcentajeExtranjero  
});
```

```
const localesEl = document.querySelector('.estadistica-origen .locales');  
const extranjerosEl = document.querySelector('.estadistica-origen .extranjeros');
```

```
if (localesEl) {  
    localesEl.textContent = `• Locales ${porcentajeNacional}% (${nacional})`;  
}  
if (extranjerosEl) {  
    extranjerosEl.textContent = `• Extranjeros ${porcentajeExtranjero}%  
    (${extranjero})`;  
}
```

```
const ctxPie = document.getElementById('graficoOrigen');
```

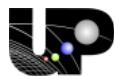
```
if (ctxPie) {  
  
    new Chart(ctxPie, {  
  
        type: 'pie',  
  
        data: {  
  
            labels: ['Locales', 'Extranjeros'],  
  
            datasets: [{  
  
                data: [porcentajeNacional, porcentajeExtranjero],  
  
                backgroundColor: ['#6b1e1e', '#e07878']  
            }]  
        },  
        options: {  
            plugins: {  
                legend: { display: true, position: 'bottom' }  
            }  
        }  
    });  
  
}  
  
const comida = datosEstadisticas.comida || 0;  
const ubicacion = datosEstadisticas.ubicacion || 0;  
const recomendacion = datosEstadisticas.recomendacion || 0;  
const horario = datosEstadisticas.horario || 0;  
const vista = datosEstadisticas.vista || 0;
```



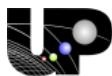
```
const totalAspectos = comida + ubicacion + recomendacion + horario + vista;
```

```
let aspectosData = [];
```

```
if (totalAspectos > 0) {  
    aspectosData = [  
        { nombre: 'Comida', porcentaje: Math.round((comida / totalAspectos) * 100),  
         cantidad: comida },  
        { nombre: 'Ubicación', porcentaje: Math.round((ubicacion / totalAspectos) *  
          100), cantidad: ubicacion },  
        { nombre: 'Recomendación', porcentaje: Math.round((recomendacion /  
          totalAspectos) * 100), cantidad: recomendacion },  
        { nombre: 'Horario', porcentaje: Math.round((horario / totalAspectos) * 100),  
         cantidad: horario },  
        { nombre: 'Vista', porcentaje: Math.round((vista / totalAspectos) * 100),  
         cantidad: vista }  
    ];  
}  
} else {  
    aspectosData = [  
        { nombre: 'Comida', porcentaje: 0, cantidad: 0 },  
        { nombre: 'Ubicación', porcentaje: 0, cantidad: 0 },  
        { nombre: 'Recomendación', porcentaje: 0, cantidad: 0 },  
        { nombre: 'Horario', porcentaje: 0, cantidad: 0 },  
        { nombre: 'Vista', porcentaje: 0, cantidad: 0 }  
    ];  
}
```



```
{ nombre: 'Vista', porcentaje: 0, cantidad: 0 }  
];  
}  
  
const ctxBar = document.getElementById('graficoAspectos');  
  
if (ctxBar) {  
  
    new Chart(ctxBar, {  
  
        type: 'bar',  
  
        data: {  
  
            labels: aspectosData.map(a => a.nombre),  
  
            datasets: [ {  
  
                label: 'Porcentaje',  
  
                data: aspectosData.map(a => a.porcentaje),  
  
                backgroundColor: ['#6b1e1e', '#c06060', '#efcfef', '#d4a4a4', '#b8b8b8']  
            }]  
        },  
        options: {  
  
            indexAxis: 'y',  
  
            scales: {  
  
                x: { beginAtZero: true, max: 100 }  
            },  
            plugins: {  
  
                legend: { display: false }  
            }  
        }  
    });  
}
```



```
        }
```

```
    }
```

```
});
```

```
}
```

```
})
```

```
.catch(error => {
```

```
    console.error('✖ Error al cargar estadísticas:', error);
```

```
    alert(`No se pudieron cargar las estadísticas: ${error.message}`);
```

```
});
```

```
});
```

**index**

```
let todasLasSolicitudes = [];
```

```
function crearTarjetaRestaurante(solicitud) {
```

```
    const card = document.createElement('div');
```

```
    card.className = 'restaurante-card';
```

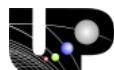
```
    const identificador = solicitud.correo || solicitud.id || 'sin-identificador';
```

```
    const tipold = solicitud.correo ? 'correo' : 'id';
```

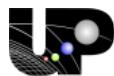
  
  

```
    card.onclick = () => {
```

```
        if (tipold === 'correo') {
```



```
window.location.href =  
`vistaRestaurante.html?correo=${encodeURIComponent(identificador)}`;  
  
} else {  
  
    window.location.href = `vistaRestaurante.html?id=${identificador}`;  
  
}  
  
};  
  
  
card.innerHTML = `  
  
      
  
    <div class="restaurante-info">  
  
        <h3 class="restaurante-nombre">${solicitud.restaurante}</h3>  
  
    </div>  
  
`;  
  
  
return card;  
  
}  
  
  
  
  
function mostrarRestaurantes(solicitudes) {  
  
    const container = document.querySelector('.line-parent');  
  
    if (!container) return;  
  
  
  
  
    container.innerHTML = ";
```



```
if (solicitudes.length === 0) {  
  
    container.innerHTML = '<p>No se encontraron restaurantes.</p>';  
  
    return;  
  
}  
  
  
solicitudes.forEach(solicitud => {  
  
    const card = crearTarjetaRestaurante(solicitud);  
  
    container.appendChild(card);  
  
});  
  
}  
  
  
function buscarRestaurantes(termino) {  
  
    if (!termino.trim()) {  
  
        aplicarFiltros();  
  
        return;  
  
    }  
  
  
    const restaurantesFiltrados = todasLasSolicitudes.filter(solicitud =>  
  
        solicitud.restaurant &&  
  
        solicitud.restaurant.toLowerCase().includes(termino.toLowerCase())  
  
    );
```

```
        mostrarRestaurantes(restaurantesFiltrados);

    }
```

```
function restauranteTieneEtiqueta(restaurante, etiqueta) {
```

```
    if (!etiqueta || etiqueta === "") return true;
```

```
    const etiquetasRestaurante = [
```

```
        restaurante.etiqueta1,
```

```
        restaurante.etiqueta2,
```

```
        restaurante.etiqueta3
```

```
    ].filter(e => e && e !== "" && e !== 'Seleccionar');
```

```
    return etiquetasRestaurante.some(e =>
```

```
        e && e.toLowerCase() === etiqueta.toLowerCase()
```

```
    );
```

```
}
```

```
function aplicarFiltros() {
```

```
    const filtroTipoComida = document.getElementById('filtroTipoComida')?.value || "";
```

```
    const filtroAmbiente = document.getElementById('filtroAmbiente')?.value || "";
```

```
    const filtroServicios = document.getElementById('filtroServicios')?.value || "";
```

```
        const terminoBusqueda = document.querySelector('.barra-busqueda input[type="text"]')?.value || "";
```

```
let restaurantesFiltrados = todasLasSolicitudes.filter(solicitud => {  
  
    const coincideBusqueda = !terminoBusqueda.trim() ||  
  
        (solicitud.restaurantе &&  
solicitud.restaurantе.toLowerCase().includes(terminoBusqueda.toLowerCase()));  
  
  
    const coincideTipoComida = restauranteTieneEtiqueta(solicitud, filtroTipoComida);  
  
    const coincideAmbiente = restauranteTieneEtiqueta(solicitud, filtroAmbiente);  
  
    const coincideServicios = restauranteTieneEtiqueta(solicitud, filtroServicios);  
  
  
    return coincideBusqueda && coincideTipoComida && coincideAmbiente &&  
coincideServicios;  
  
});  
  
  
mostrarRestaurantes(restaurantesFiltrados);  
}  
  
  
function limpiarFiltros() {  
  
    const filtros = ['filtroTipoComida', 'filtroAmbiente', 'filtroServicios'];  
  
    filtros.forEach(filtroId => {  
  
        const filtro = document.getElementById(filtroId);  
  
        if (filtro) filtro.value = "";  
  
    });
```

```
const inputBusqueda = document.querySelector('.barra-busqueda input[type="text"]');

if (inputBusqueda) inputBusqueda.value = "";

mostrarRestaurantes(todasLasSolicitudes);

}

function configurarBusqueda() {

const inputBusqueda = document.querySelector('.barra-busqueda input[type="text"]');

const botonBuscar = document.querySelector('.barra-busqueda .buscar');

if (inputBusqueda && botonBuscar) {

botonBuscar.addEventListener('click', () => {

aplicarFiltros();

});

inputBusqueda.addEventListener('keypress', (e) => {

if (e.key === 'Enter') {

aplicarFiltros();

}

});

inputBusqueda.addEventListener('input', (e) => {
```

```
        aplicarFiltros();  
    });  
  
    }  
  
  
const filtros = ['filtroTipoComida', 'filtroAmbiente', 'filtroServicios'];  
  
filtros.forEach(filtroId => {  
  
    const filtro = document.getElementById(filtroId);  
  
    if (filtro) {  
  
        filtro.addEventListener('change', () => {  
  
            console.log(`💡 Filtro ${filtroId} cambiado a: ${filtro.value}`);  
  
            aplicarFiltros();  
  
        });  
  
    }  
  
});  
  
  
const btnLimpiarFiltros = document.getElementById('btnLimpiarFiltros');  
  
if (btnLimpiarFiltros) {  
  
    btnLimpiarFiltros.addEventListener('click', () => {  
  
        limpiarFiltros();  
  
    });  
  
}  
  
}
```

```
document.addEventListener('DOMContentLoaded', async function() {  
  
    const container = document.querySelector('.line-parent');  
  
    if (!container) return;  
  
    try {  
  
        console.log('Intentando cargar solicitudes...');  
  
        const response = await fetch('http://52.23.26.163:7070/solicitudes', {  
            method: 'GET',  
  
            headers: { 'Content-Type': 'application/json' }  
        });  
  
        if (!response.ok) {  
  
            throw new Error(`HTTP ${response.status}: ${response.statusText}`);  
        }  
  
        const solicitudes = await response.json();  
        console.log('Solicitudes recibidas:', solicitudes);  
  
        todasLasSolicitudes = solicitudes.filter(solicitud =>  
            solicitud.estado === 'aprobado' ||  
            solicitud.estado === 'Aprobado' ||  
            solicitud.estado === 'APROBADO'
```

);

```
console.log('Solicitudes aprobadas:', todasLasSolicitudes);
```

```
if (todasLasSolicitudes.length === 0) {
```

```
    container.innerHTML = '<p>No hay restaurantes aprobados disponibles.</p>';
```

```
    return;
```

```
}
```

```
mostrarRestaurantes(todasLasSolicitudes);
```

```
configurarBusqueda();
```

```
} catch (error) {
```

```
    console.error('Error al cargar restaurantes:', error);
```

```
    container.innerHTML = '<p>Error al cargar restaurantes. Revisa la consola para  
más detalles.</p>';
```

```
}
```

```
});
```

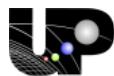
## login

```
document.addEventListener("DOMContentLoaded", function () {
```

```
    const form = document.getElementById("registroForm");
```

```
form.addEventListener("submit", function (event) {  
    event.preventDefault();  
  
    const correo = document.getElementById("correo");  
    const contrasena = document.getElementById("contrasena");  
  
    const errorCorreo = document.getElementById("error-correo");  
    const errorContrasena = document.getElementById("error-contrasena");  
  
    let formularioValido = true;  
  
    errorCorreo.textContent = "";  
    errorContrasena.textContent = "";  
  
    correo.classList.remove("input-error");  
    contrasena.classList.remove("input-error");  
  
    if (correo.value.trim() === "") {  
        errorCorreo.textContent = "*Este campo es obligatorio.";  
        correo.classList.add("input-error");  
        formularioValido = false;  
    }  
})
```

```
if (contrasena.value.trim() === "") {  
    errorContrasena.textContent = "*Este campo es obligatorio.";  
    contrasena.classList.add("input-error");  
    formularioValido = false;  
}  
  
});  
  
});  
  
  
function initializeLoginForm() {  
    const form = document.getElementById("loginForm");  
  
  
  
    if (!form) {  
        return;  
    }  
  
  
  
    const newForm = form.cloneNode(true);  
    form.parentNode.replaceChild(newForm, form);  
  
  
  
    newForm.addEventListener("submit", function (event) {  
        event.preventDefault();  
  
  
  
        const correo = document.getElementById("correo");  
    });
```



```
const contrasena = document.getElementById("contrasena");

const errorCorreo = document.getElementById("error-correo");

const errorContrasena = document.getElementById("error-contrasena");

if (!correo || !contrasena || !errorCorreo || !errorContrasena) {

    return;

}

let formularioValido = true;

errorCorreo.textContent = "";

errorContrasena.textContent = "";

correo.classList.remove("input-error");

contrasena.classList.remove("input-error");

if (correo.value.trim() === "") {

    errorCorreo.textContent = "*Este campo es obligatorio./";

    correo.classList.add("input-error");

    formularioValido = false;

} else if (!isValidEmail(correo.value)) {

    errorCorreo.textContent = "*Ingrese un correo electrónico válido./";

}
```

```
correo.classList.add("input-error");

formularioValido = false;

}

if (contrasena.value.trim() === "") {

    errorContrasena.textContent = "*Este campo es obligatorio.";

    contrasena.classList.add("input-error");

    formularioValido = false;

} else if (contrasena.value.length < 6) {

    errorContrasena.textContent = "*La contraseña debe tener al menos 6
caracteres./";

    contrasena.classList.add("input-error");

    formularioValido = false;

}

if (formularioValido) {

    let rol = 'restaurantero';

    let parent = newForm.parentElement;

    while (parent) {

        if (parent.id === 'loginAdminContent') {

            rol = 'administrador';

        }

    }

    let loginAdminContent = document.getElementById('loginAdminContent');

    loginAdminContent.innerHTML = `

        <div>
            <input type="text" value="${rol}" name="rol" />
        </div>

    `;

}

}

let loginAdminContent = document.getElementById('loginAdminContent');

loginAdminContent.innerHTML = `

<div>
    <input type="text" value="restaurantero" name="rol" />
</div>

<div>
    <input type="text" value="administrador" name="rol" />
</div>

`;
```

```
        break;

    } else if (parent.id === 'loginRestContent') {

        rol = 'restaurantero';

        break;

    }

    parent = parent.parentElement;

}
```

```
fetch('http://52.23.26.163:7070/login', {

    method: 'POST',

    headers: { 'Content-Type': 'application/json' },

    body: JSON.stringify({

        correo: correo.value,

        contrasena: contrasena.value,

        rol: rol

    })

})

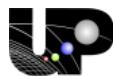
.then(res => res.json())

.then(data => {

    if (data.success) {

        let usuarioData = null;

        if (data.usuario) {
```



```
usuarioData = data.usuario;

}

else if (data.id) {

    usuarioData = {

        id: data.id,

        correo: data.correo || correo.value,

        rol: data.rol,

        nombre: data.nombre || "",

        telefono: data.telefono || ""

    };

}

else {

    usuarioData = {

        id: data.id || data.userId || Date.now(),

        correo: correo.value,

        rol: data.rol || rol,

        nombre: data.nombre || "",

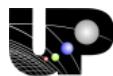
        telefono: data.telefono || ""

    };

}

if (usuarioData) {

    sessionStorage.setItem('id', usuarioData.id);
```



```
sessionStorage.setItem('correo', usuarioData.correo);

sessionStorage.setItem('rol', usuarioData.rol);

sessionStorage.setItem('nombre', usuarioData.nombre);

sessionStorage.setItem('telefono', usuarioData.telefono || "");

console.log('✅ Datos guardados en sessionStorage:', {

    id: usuarioData.id,

    correo: usuarioData.correo,

    rol: usuarioData.rol,

    nombre: usuarioData.nombre,

    telefono: usuarioData.telefono

});

} else {

    console.warn('⚠️ No se pudieron extraer datos del usuario, guardando datos básicos');

    sessionStorage.setItem('correo', correo.value);

    sessionStorage.setItem('rol', data.rol || rol);

    sessionStorage.setItem('loginSuccess', 'true');

}

if (data.rol === 'administrador') {

    window.location.href = 'vistaPrincipalAdmin.html';

} else {
```

```
window.location.href = 'vistaPrincipalRestaurantero.html';

}

} else {

    errorContrasena.textContent = data.message || "Credenciales incorrectas";

}

.errorContrasena);

.catch(() => {

    errorContrasena.textContent = "Error de conexión con el servidor";

});

} else {

    console.log('Formulario inválido, errores mostrados');

}

});

console.log('Validaciones configuradas correctamente');

}

function isValidEmail(email) {

    const emailRegex = /^[^@\s]+@[^\s@]+\.[^\s@]+$/;

    return emailRegex.test(email);

}

document.addEventListener('DOMContentLoaded', function() {
```

```
console.log('DOM cargado, inicializando login...');

initializeLoginForm();

});
```

```
if (document.readyState === 'complete' || document.readyState === 'interactive') {

    console.log('Documento ya cargado, inicializando login...');

    initializeLoginForm();

}
```

## **loginRestaurantero**

```
document.addEventListener('DOMContentLoaded', function() {
```

```
    const logoutBtn = document.querySelector('.log-out-parent');
```

```
    if (logoutBtn) {

        logoutBtn.addEventListener('click', function() {
```

```
            window.location.href = 'index.html';

        });
    }
```

```
}
```

```
const miRestauranteBtn = document.querySelector('.mi-restaurante-parent');
```

```
if (miRestauranteBtn) {
```

```
    miRestauranteBtn.addEventListener('click', function() {
```

```
        window.location.href = 'vistaEdicionRest.html';

    });
}
```

```
});
```

## modalSolicitud

```
document.addEventListener('DOMContentLoaded', function() {  
  
    document.querySelectorAll('input[type="file"]').forEach(function(input) {  
  
        input.addEventListener('change', function(event) {  
  
            const fileList = event.target.files;  
  
            let fileNameSpan = input.nextElementSibling;  
  
            if (!fileNameSpan || !fileNameSpan.classList.contains('file-name')) {  
  
                fileNameSpan = document.createElement('span');  
  
                fileNameSpan.className = 'file-name';  
  
                input.parentNode.insertBefore(fileNameSpan, input.nextSibling);  
  
            }  
  
            if (fileList.length > 0) {  
  
                if (input.hasAttribute('multiple')) {  
  
                    const names = Array.from(fileList).map(f => f.name).join(', ');  
  
                    fileNameSpan.textContent = `Archivos seleccionados: ${names}`;  
  
                } else {  
  
                    fileNameSpan.textContent = `Archivo seleccionado: ${fileList[0].name}`;  
  
                }  
  
            } else {  
  
                fileNameSpan.textContent = "";  
  
            }  
  
        });  
  
    });  

```

});

```
document.addEventListener("DOMContentLoaded", function () {
```

```
    const form = document.getElementById("registroForm");
```

```
    const modal = document.getElementById("modal");
```

```
    const cerrarModal = document.getElementById("cerrarModal");
```

```
function validarCampoVacio(campo, spanError, mensaje) {
```

```
    if (campo.value.trim() === "") {
```

```
        campo.classList.add("input-error");
```

```
        spanError.textContent = mensaje;
```

```
        return false;
```

```
    } else {
```

```
        campo.classList.remove("input-error");
```

```
        spanError.textContent = "";
```

```
        return true;
```

```
}
```

```
}
```

```
function comprimirImagen(file, maxWidth = 400, quality = 0.3) {
```

```
    return new Promise((resolve) => {
```

```
        const canvas = document.createElement('canvas');
```

```
        const ctx = canvas.getContext('2d');
```

```
        const img = new Image();
```

```
img.onload = function() {  
  
    const ratio = Math.min(maxWidth / img.width, maxWidth / img.height);  
  
    canvas.width = img.width * ratio;  
  
    canvas.height = img.height * ratio;  
  
  
    ctx.drawImage(img, 0, 0, canvas.width, canvas.height);  
  
    canvas.toBlob(resolve, 'image/jpeg', quality);  
  
};  
  
  
img.src = URL.createObjectURL(file);  
  
});  
  
}  
  
  
form.addEventListener("submit", async function (event) {  
  
    event.preventDefault();  
  
  
    const restaurante = document.getElementById("restaurante");  
  
    const propietario = document.getElementById("propietario");  
  
    const correo = document.getElementById("correo");  
  
    const numero = document.getElementById("numero");  
  
    const facebook = document.getElementById("facebook");  
  
    const instagram = document.getElementById("instagram");
```

```
const direccion = document.getElementById("direccion");

const horario = document.getElementById("horario");

const imagen1 = document.getElementsByName("imagen1")[0];

const imagen2 = document.getElementsByName("imagen2")[0];

const imagen3 = document.getElementsByName("imagen3")[0];

const comprobante = document.getElementsByName("comprobante")[0];

const menu = document.getElementsByName("menu")[0];

const errorRestaurante = document.getElementById("error-nombre-restaurante");

const errorPropietario = document.getElementById("error-nombre-propietario");

const errorCorreo = document.getElementById("error-correo");

const errorNumero = document.getElementById("error-numero");

const errorFacebook = document.getElementById("error-facebook");

const errorInstagram = document.getElementById("error-instagram");

const errorDireccion = document.getElementById("error-direccion");

const errorHorario = document.getElementById("error-horario");

const errorImagen1 = document.getElementById("error-imagen1");

const errorImagen2 = document.getElementById("error-imagen2");

const errorImagen3 = document.getElementById("error-imagen3");

const errorComprobante = document.getElementById("error-comprobante");

const errorMenu = document.getElementById("error-menu");
```

```
let formularioValido = true;

formularioValido &= validarCampoVacio(restaurante, errorRestaurante, "*Este campo es obligatorio.");

formularioValido &= validarCampoVacio(propietario, errorPropietario, "*Este campo es obligatorio.");

formularioValido &= validarCampoVacio(correo, errorCorreo, "*Este campo es obligatorio.");

formularioValido &= validarCampoVacio(numero, errorNumero, "*Este campo es obligatorio.");

formularioValido &= validarCampoVacio(direccion, errorDireccion, "*Este campo es obligatorio.");

formularioValido &= validarCampoVacio(horario, errorHorario, "*Este campo es obligatorio.");

formularioValido &= validarCampoVacio(facebook, errorFacebook, "*Este campo es obligatorio.");

formularioValido &= validarCampoVacio(instagram, errorInstagram, "*Este campo es obligatorio.");

function validarImagen(input, errorSpan) {

    if (!input.files || input.files.length !== 1) {

        errorSpan.textContent = "*Seleccione una imagen.";

        formularioValido = false;

    } else {

        const tipo = input.files[0].type;
```

```
if (tipo !== "image/png" && tipo !== "image/jpeg") {  
  
    errorSpan.textContent = "*La imagen debe ser PNG o JPEG.>";  
  
    formularioValido = false;  
  
} else {  
  
    errorSpan.textContent = "";  
  
}  
  
}  
  
validarImagen(imagen1, errorImagen1);  
  
validarImagen(imagen2, errorImagen2);  
  
validarImagen(imagen3, errorImagen3);
```

```
if (!comprobante.files || comprobante.files.length === 0) {  
  
    errorComprobante.textContent = "*Suba un comprobante de domicilio.>";  
  
    formularioValido = false;  
  
} else {  
  
    errorComprobante.textContent = "";  
  
}
```

```
if (!menu.files || menu.files.length === 0) {  
  
    errorMenu.textContent = "*Suba el menú del restaurante.>";  
  
    formularioValido = false;  
  
} else {
```

```
errorMenu.textContent = "";  
}  
  
if (formularioValido) {  
    const formData = new FormData();  
  
    formData.append('restaurante', restaurante.value.trim());  
    formData.append('propietario', propietario.value.trim());  
    formData.append('correo', correo.value.trim());  
    formData.append('numero', numero.value.replace(/\D/g, " "));  
    formData.append('direccion', direccion.value.trim());  
    formData.append('horario', horario.value.trim());  
    formData.append('facebook', facebook.value.trim());  
    formData.append('instagram', instagram.value.trim());  
    formData.append('estado', 'pendiente');  
    formData.append('etiqueta1', 'Seleccionar');  
    formData.append('etiqueta2', 'Seleccionar');  
    formData.append('etiqueta3', 'Seleccionar');  
  
    if (imagen1?.files && imagen1.files.length === 1) {  
        const imagen1Comprimida = await comprimirImagen(imagen1.files[0]);  
        formData.append('imagen1', imagen1Comprimida, imagen1.files[0].name ||  
        'imagen1.jpg');
```

}

```
if (imagen2?.files && imagen2.files.length === 1) {  
  
    const imagen2Comprimida = await comprimirImagen(imagen2.files[0]);  
  
    formData.append('imagen2', imagen2Comprimida, imagen2.files[0].name ||  
'imagen2.jpg');
```

}

```
if (imagen3?.files && imagen3.files.length === 1) {  
  
    const imagen3Comprimida = await comprimirImagen(imagen3.files[0]);  
  
    formData.append('imagen3', imagen3Comprimida, imagen3.files[0].name ||  
'imagen3.jpg');
```

}

```
const comprobanteArchivo = comprobante?.files[0];
```

```
if (comprobanteArchivo) {
```

```
    if (comprobanteArchivo.type.startsWith('image/')) {
```

```
        const comprobanteComprimido = await comprimirImagen(comprobanteArchivo,  
800, 0.4);
```

```
        formData.append('comprobante', comprobanteComprimido,  
comprobanteArchivo.name || 'comprobante.jpg');
```

```
    } else {
```

```
        formData.append('comprobante', comprobanteArchivo,  
comprobanteArchivo.name);
```

}

}

```
const menuArchivo = menu?.files && menu.files.length > 0 ? menu.files[0] : null;

if (menuArchivo) {

    if (menuArchivo.type.startsWith('image/')) {

        const menuComprimido = await comprimirImagen(menuArchivo, 800, 0.4);

        formData.append('menu', menuComprimido, menuArchivo.name || 'menu.jpg');

    } else {

        formData.append('menu', menuArchivo, menuArchivo.name);

    }

}

try {

    const response = await fetch('http://52.23.26.163:7070/solicitudes/with-files', {

        method: 'POST',

        body: formData

    });

}

if (response.ok) {

    try {

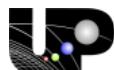
        console.log('Creando registro de estadísticas para:', correo.value.trim());

        const estadisticasPayload = {

            correo: correo.value.trim(),
```

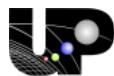
nacional: 0,  
extranjero: 0,  
descargas: 0,  
comida: 0,  
vista: 0,  
horario: 0,  
recomendacion: 0,  
ubicacion: 0  
};

```
const estadisticasResponse = await fetch('http://52.23.26.163:7070/estadisticas',  
{  
    method: 'POST',  
    headers: {  
        'Content-Type': 'application/json'  
    },  
    body: JSON.stringify(estadisticasPayload)  
});  
  
if (estadisticasResponse.ok) {  
    console.log('✅ Registro de estadísticas creado exitosamente');  
} else {
```



```
        console.warn('⚠️ Error al crear registro de estadísticas:',  
estadisticasResponse.status);  
  
    }  
  
} catch (estadisticasError) {  
  
    console.error('❌ Error al crear estadísticas:', estadisticasError);  
  
}  
  
  
modal.style.display = "flex";  
  
form.reset();  
  
} else {  
  
    const errorText = await response.text();  
  
    console.error('Error del servidor:', response.status, errorText);  
  
    alert(`Error al enviar la solicitud: ${response.status} - ${errorText}`);  
  
}  
  
} catch (error) {  
  
    console.error('Error de conexión:', error);  
  
    alert("Error de conexión.");  
  
}  
  
});  
  
  
cerrarModal.addEventListener("click", function () {  
  
    modal.style.display = "none";
```

```
window.location.href = "index.html";  
});  
  
window.addEventListener("click", function (event) {  
    if (event.target === modal) {  
        modal.style.display = "none";  
        window.location.href = "index.html";  
    }  
});  
});  
  
modal  
  
function openEncuestaModal() {  
    fetch('encuesta.html')  
        .then(response => response.text())  
        .then(html => {  
            const parser = new DOMParser();  
            const doc = parser.parseFromString(html, 'text/html');  
            const form = doc.querySelector('form');  
            if (form) {  
                document.getElementById('encuestaContent').innerHTML = form.outerHTML;  
                if (!window.initEncuestaForm) {  
                    const script = document.createElement('script');  
                    script.src = '../Scripts/encuesta.js';  
                }  
            }  
        })  
};
```

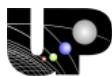


```
script.onload = function() {  
  
    if (window.initEncuestaForm) window.initEncuestaForm();  
  
};  
  
document.body.appendChild(script);  
  
} else {  
  
    window.initEncuestaForm();  
  
}  
  
}  
  
const modal = document.getElementById('encuestaModal');  
  
modal.style.display = 'flex';  
  
setTimeout(() => {  
  
    modal.classList.add('show');  
  
}, 10);  
  
});  
  
}  
  
  
function closeEncuestaModal() {  
  
const modal = document.getElementById('encuestaModal');  
  
if (modal) {  
  
    modal.classList.remove('show');  
  
    setTimeout(() => {  
  
        modal.style.display = 'none';  
  
    }, 300);  
}
```

}

}

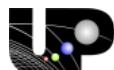
```
function openMembresiaModal() {  
  
    fetch('solicitudMembresia.html')  
  
        .then(response => response.text())  
  
        .then(html => {  
  
            const parser = new DOMParser();  
  
            const doc = parser.parseFromString(html, 'text/html');  
  
            const formSection = doc.querySelector('section.form');  
  
            if (formSection) {  
  
                document.getElementById('membresiaContent').innerHTML =  
formSection.outerHTML;  
  
                loadModalCSS('membresia');  
  
                const modal = document.getElementById('membresiaModal');  
  
                modal.style.display = 'flex';  
  
                setTimeout(() => {  
  
                    modal.classList.add('show');  
  
                }, 10);  
  
                loadMembresiaScript();  
  
            }  
  
        })  
  
.catch(error => console.error('Error cargando membresía:', error));
```



}

```
function closeMembresiaModal() {  
  
    const modal = document.getElementById('membresiaModal');  
  
    modal.classList.remove('show');  
  
    setTimeout(() => {  
  
        modal.style.display = 'none';  
  
        document.getElementById('membresiaContent').innerHTML = '';  
  
        const membresiaCSS = document.getElementById('modal-membresia-css');  
  
        if (membresiaCSS) {  
  
            membresiaCSS.remove();  
  
            loadedCSS.delete('modal-membresia-css');  
  
        }  
  
    }, 300);  
  
}
```

```
function loadMembresiaScript() {  
  
    loadScript('../Scripts/membresia.js');  
  
    setTimeout(() => {  
  
        if (typeof initializeMembresiaForm === 'function') {  
  
            initializeMembresiaForm();  
  
        }  
  
    }, 100);  
  
}
```



}

```
let loadedCSS = new Set();
```

```
function openRegistroModal() {
```

```
    fetch('registro.html')
```

```
        .then(response => response.text())
```

```
        .then(html => {
```

```
            const parser = new DOMParser();
```

```
            const doc = parser.parseFromString(html, 'text/html');
```

```
            const formSection = doc.querySelector('section.form');
```

```
            if (formSection) {
```

```
                document.getElementById('registroContent').innerHTML =  
                formSection.outerHTML;
```

```
                loadModalCSS('registro');
```

```
            const modal = document.getElementById('registroModal');
```

```
            modal.style.display = 'flex';
```

```
            setTimeout(() => {
```

```
                modal.classList.add('show');
```

```
            }, 10);
```

```
        loadRegistroScript();  
    }  
  
})  
  
.catch(error => alert('Error cargando registro: ' + error.message));  
  
}  
  
  
  
function closeRegistroModal() {  
  
    const modal = document.getElementById('registroModal');  
  
    modal.classList.remove('show');  
  
  
  
    setTimeout(() => {  
  
        modal.style.display = 'none';  
  
        document.getElementById('registroContent').innerHTML = "";  
  
        const registroCSS = document.getElementById('modal-registro-css');  
  
        if (registroCSS) {  
  
            registroCSS.remove();  
  
            loadedCSS.delete('modal-registro-css');  
  
        }  
  
    }, 300);  
  
}  
  
  
  
function loadRegistroScript() {
```

```
loadScript('../Scripts/registro.js');

setTimeout(() => {

    if (typeof initializeRegistroForm === 'function') {

        initializeRegistroForm();

    }

}, 100);

}

function openLoginAdminModal() {

    fetch('loginAdmin.html')

        .then(response => response.text())

        .then(html => {

            const parser = new DOMParser();

            const doc = parser.parseFromString(html, 'text/html');

            const formSection = doc.querySelector('section.form');

            if (formSection) {

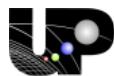
                document.getElementById('loginAdminContent').innerHTML =
formSection.outerHTML;

                loadModalCSS('login-admin');

            }

        });

}
```



```
const modal = document.getElementById('loginAdminModal');

modal.style.display = 'flex';

setTimeout(() => {

    modal.classList.add('show');

}, 10);

loadScript('../Scripts/login.js', () => {

    console.log('Script de login cargado');

    let attempts = 0;

    const maxAttempts = 5;

    function tryInitialize() {

        attempts++;

        console.log(`Intento ${attempts} de inicialización...`);

        if (typeof initializeLoginForm === 'function') {

            initializeLoginForm();

        } else if (attempts < maxAttempts) {

            setTimeout(tryInitialize, 100);

        } else {

    }

});
```

```
        console.log('No se pudo inicializar después de ' + maxAttempts + '  
intentos');  
  
    }  
  
}  
  
setTimeout(tryInitialize, 100);  
  
});  
  
}  
  
})  
  
.catch(error => alert('Error cargando loginAdmin: ' + error.message));  
  
}  
  
  
function closeLoginAdminModal() {  
  
const modal = document.getElementById('loginAdminModal');  
  
modal.classList.remove('show');  
  
  
  
setTimeout(() => {  
  
modal.style.display = 'none';  
  
document.getElementById('loginAdminContent').innerHTML = "";  
  
const loginAdminCSS = document.getElementById('modal-login-admin-css');  
  
if (loginAdminCSS) {  
  
loginAdminCSS.remove();  
  
loadedCSS.delete('modal-login-admin-css');
```

}

, 300);

}

```
function openLoginRestModal() {  
  
    fetch('loginRest.html')  
  
        .then(response => response.text())  
  
        .then(html => {  
  
            const parser = new DOMParser();  
  
            const doc = parser.parseFromString(html, 'text/html');  
  
            const formSection = doc.querySelector('section.form');  
  
  
            if (formSection) {  
  
                document.getElementById('loginRestContent').innerHTML =  
formSection.outerHTML;  
  
  
                loadModalCSS('login-rest');  
  
  
                const modal = document.getElementById("loginRestModal");  
  
                modal.style.display = 'flex';  
  
  
                setTimeout(() => {  
  
                    modal.classList.add('show');  
                }, 300);  
            }  
        })  
}  
};
```

```
    }, 10);  
  
    loadLoginScript();  
  
}  
  
})  
  
.catch(error => alert('Error cargando loginRest: ' + error.message));  
  
}  
  
  
  
function closeLoginRestModal() {  
  
    const modal = document.getElementById('loginRestModal');  
  
    modal.classList.remove('show');  
  
  
  
    setTimeout(() => {  
  
        modal.style.display = 'none';  
  
        document.getElementById('loginRestContent').innerHTML = "";  
  
        const loginRestCSS = document.getElementById('modal-login-rest-css');  
  
        if (loginRestCSS) {  
  
            loginRestCSS.remove();  
  
            loadedCSS.delete('modal-login-rest-css');  
  
        }  
  
        }, 300);  
  
}
```

```
function loadLoginScript() {  
  
    loadScript('../Scripts/login.js');  
  
  
    let attempts = 0;  
  
    const maxAttempts = 10;  
  
  
  
  
    function tryInitialize() {  
  
        attempts++;  
  
        if (typeof initializeLoginForm === 'function') {  
  
            console.log('Ejecutando initializeLoginForm... (intento ' + attempts + ')');  
  
            initializeLoginForm();  
  
        } else if (attempts < maxAttempts) {  
  
            setTimeout(tryInitialize, 100);  
  
        } else {  
  
            console.log('No se pudo inicializar initializeLoginForm después de ' +  
maxAttempts + ' intentos');  
  
        }  
  
    }  
  
  
  
  
    setTimeout(tryInitialize, 200);  
  
}  
  
  
  
  
function openLoginModal() {
```

```
fetch('loginSelect.html')

.then(response => response.text())

.then(html => {

    const parser = new DOMParser();

    const doc = parser.parseFromString(html, 'text/html');

    const chooseSection = doc.querySelector('section.choose');




    if (chooseSection) {

        document.getElementById('loginContent').innerHTML =
chooseSection.outerHTML;






        loadModalCSS('login');



        const modal = document.getElementById('loginModal');

        modal.style.display = 'flex';



        setTimeout(() => {

            modal.classList.add('show');

        }, 10);






        setupLoginModalButtons();

    }

})
```

```
.catch(error => alert('Error cargando login: ' + error.message));  
}
```

```
function closeLoginModal() {  
  
    const modal = document.getElementById('loginModal');  
  
    modal.classList.remove('show');  
  
  
  
    setTimeout(() => {  
  
        modal.style.display = 'none';  
  
        document.getElementById('loginContent').innerHTML = "";  
  
        const loginCSS = document.getElementById('modal-login-css');  
  
        if (loginCSS) {  
  
            loginCSS.remove();  
  
            loadedCSS.delete('modal-login-css');  
  
        }  
  
    }, 300);  
}
```

```
function setupLoginModalButtons() {  
  
    const registerLink = document.querySelector('.register-link');  
  
    if (registerLink) {  
  
        registerLink.addEventListener('click', function(e) {  
  
            e.preventDefault();  
        })  
    }  
}
```

```
closeLoginModal();
```

```
setTimeout(() => {
```

```
    openRegistroModal();
```

```
}, 100);
```

```
});
```

```
}
```

```
const buttons = document.querySelectorAll('.button');
```

```
if (buttons[0]) {
```

```
    buttons[0].addEventListener('click', function(e) {
```

```
        e.preventDefault();
```

```
        closeLoginModal();
```

```
        setTimeout(() => {
```

```
            openLoginAdminModal();
```

```
        }, 100);
```

```
    });
```

```
}
```

```
if (buttons[1]) {
```

```
    buttons[1].addEventListener('click', function(e) {
```

```
        e.preventDefault();
```

```
closeLoginModal();

setTimeout(() => {
    openLoginRestModal();
}, 100);

});

}

function loadModalCSS(modalType) {
    const cssId = `modal-${modalType}-css`;

    if (!loadedCSS.has(cssId)) {
        const link = document.createElement('link');
        link.rel = 'stylesheet';
        link.id = cssId;

        switch(modalType) {
            case 'login-admin':
                link.href = './styles/modal-login-admin.css';
                break;
            case 'login-rest':
                link.href = './styles/modal-login-rest.css';
                break;
        }
    }
}
```

```
case 'login':  
  
    link.href = './styles/modal-login.css';  
  
    break;  
  
case 'registro':  
  
    link.href = './styles/modal-registro.css';  
  
    break;  
  
default:  
  
    link.href = `./styles/modal-${modalType}.css`;  
  
}  
  
  
document.head.appendChild(link);  
  
loadedCSS.add(cssId);  
  
}  
  
}  
  
  
function loadScript(src, callback) {  
  
    if (!document.querySelector(`script[src="${src}"]`)) {  
  
        const script = document.createElement('script');  
  
        script.src = src;  
  
  
  
        if (callback) {  
  
            script.onload = callback;  
  
        }  
    }  
}
```

```
document.head.appendChild(script);

} else if (callback) {

    callback();

}

}

document.addEventListener('click', function(event) {

    if (event.target.classList.contains('modal-overlay')) {

        if (event.target.id === 'encuestaModal') {

            closeEncuestaModal();

        }

        if (event.target.id === 'registroModal') {

            closeRegistroModal();

        } else if (event.target.id === 'loginModal') {

            closeLoginModal();

        } else if (event.target.id === 'loginAdminModal') {

            closeLoginAdminModal();

        } else if (event.target.id === 'loginRestModal') {

            closeLoginRestModal();

        } else if (event.target.id === 'membresiaModal') {

            closeMembresiaModal();

        }

    }

})
```

```
        }
```

```
    }
```

```
    if (event.target.id === 'modalAviso') {
```

```
        event.target.style.display = 'none';
```

```
    }
```

```
});
```

```
document.addEventListener('keydown', function(event) {
```

```
    if (event.key === 'Escape') {
```

```
        closeRegistroModal();
```

```
        closeLoginModal();
```

```
        closeLoginAdminModal();
```

```
        closeLoginRestModal();
```

```
    }
```

```
});
```

```
registro
```

```
function initializeRegistroForm() {
```

```
    const form = document.getElementById("registroForm");
```

```
    if (!form) {
```

```
        console.log("Formulario de registro no encontrado");
```

```
        return;
```

```
    }
```

```
form.addEventListener("submit", function (event) {  
    event.preventDefault();  
  
    const nombre = document.getElementById("nombre");  
    const correo = document.getElementById("correo");  
    const contrasena = document.getElementById("contrasena");  
    const confirmar = document.getElementById("confirmar");  
  
    const errorNombre = document.getElementById("error-nombre");  
    const errorCorreo = document.getElementById("error-correo");  
    const errorContrasena = document.getElementById("error-contrasena");  
    const errorConfirmar = document.getElementById("error-confirmar");  
  
    let formularioValido = true;  
  
    errorNombre.textContent = "";  
    errorCorreo.textContent = "";  
    errorContrasena.textContent = "";  
    errorConfirmar.textContent = "";  
  
    nombre.classList.remove("input-error");  
    correo.classList.remove("input-error");
```

```
contraseña.classList.remove("input-error");

confirmar.classList.remove("input-error");

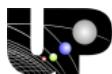
if (nombre.value.trim() === "") {
    errorNombre.textContent = "*Este campo es obligatorio.";
    nombre.classList.add("input-error");
    formularioValido = false;
}

if (correo.value.trim() === "") {
    errorCorreo.textContent = "*Este campo es obligatorio.";
    correo.classList.add("input-error");
    formularioValido = false;
} else if (!isValidEmail(correo.value)) {
    errorCorreo.textContent = "*Ingrese un correo electrónico válido.";
    correo.classList.add("input-error");
    formularioValido = false;
}

if (contraseña.value.trim() === "") {
    errorContrasena.textContent = "*Este campo es obligatorio.";
    contraseña.classList.add("input-error");
    formularioValido = false;
}
```

```
    } else if (contrasena.value.length < 6) {  
  
        errorContrasena.textContent = "*La contraseña debe tener al menos 6  
caracteres.>";  
  
        contrasena.classList.add("input-error");  
  
        formularioValido = false;  
  
    }  
  
  
if (confirmar.value.trim() === "") {  
  
    errorConfirmar.textContent = "*Este campo es obligatorio.";  
  
    confirmar.classList.add("input-error");  
  
    formularioValido = false;  
  
} else if (contrasena.value !== confirmar.value) {  
  
    errorConfirmar.textContent = "*Las contraseñas no coinciden.";  
  
    confirmar.classList.add("input-error");  
  
    formularioValido = false;  
  
}  
  
  
if (formularioValido) {  
  
    fetch('http://52.23.26.163:7070/users', {  
  
        method: 'POST',  
  
        headers: { 'Content-Type': 'application/json' },  
  
        body: JSON.stringify({  
  
            nombre: nombre.value,  
            apellidoPaterno: apellidoPaterno.value,  
            apellidoMaterno: apellidoMaterno.value,  
            correoElectronico: correoElectronico.value,  
            contraseña: contrasena.value  
        })  
    }).then(response => response.json()).then(data => {  
        console.log(data);  
    })  
}
```

```
correo: correo.value,  
contrasena: contrasena.value,  
tipo: "restaurantero"  
})  
})  
.then(async res => {  
let data;  
try {  
data = await res.json();  
} catch {  
data = {};  
}  
if (res.ok && data.success) {  
window.location.href = "solicitudRestaurante.html";  
} else if (data.message) {  
alert(data.message);  
} else {  
alert("Error en el registro");  
}  
})  
.catch(() => alert("Error de conexión con el servidor"));  
}  
});
```



}

```
function isValidEmail(email) {  
  
    const emailRegex = /^[^@\s]+@[^\s@]+\.[^\s@]+$/;  
  
    return emailRegex.test(email);  
  
}  
  
  
if (document.readyState === 'loading') {  
  
    document.addEventListener('DOMContentLoaded', initializeRegistroForm);  
  
} else {  
  
    initializeRegistroForm();  
  
}
```

### **restaurantero**

```
class RestauranterosManager {
```

```
    static currentRestauranteros = [];
```

```
    static currentEditingId = null;
```

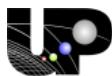
```
    static async loadAllRestauranteros() {
```

```
        try {
```

```
            UIUtils.showLoading(true);
```

```
            this.currentRestauranteros = await SazonAPI.getAllRestauranteros();
```

```
            this.displayRestauranteros();
```



```
        UIUtils.showSuccess(`#${this.currentRestauranteros.length} restauranteros
cargados`);

    } catch (error) {

        UIUtils.showError('Error al cargar restauranteros: ' + error.message);

        this.currentRestauranteros = [];

    } finally {

        UIUtils.showLoading(false);

    }

}

static displayRestauranteros() {

    const container = document.getElementById('restauranteros-list');

    if (!container) return;

    if (this.currentRestauranteros.length === 0) {

        container.innerHTML = '<p class="no-data">No hay restauranteros
disponibles</p>';

        return;

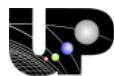
    }

    container.innerHTML = this.currentRestauranteros.map(r => `

        <div class="restaurantero-card" data-id="${r.codigorestaurantero ||

r.id_usuario}">

            <div class="card-header">
```



```
<h3>${r.nombreU}</h3>

<span class="status-badge ${r.status === 1 ? 'active' : 'inactive'}">
    ${r.status === 1 ? 'Activo' : 'Inactivo'}
</span>

</div>

<div class="card-body">

    <p><strong>Correo:</strong> ${r.correo}</p>
    <p><strong>RFC:</strong> ${r.rfc || 'No especificado'}</p>
    <p><strong>Tipo:</strong> ${r.tipo || 'restaurantero'}</p>

</div>

<div class="card-actions">

    <button class="btn-edit"
        onclick="RestauranterosManager.editRestaurantero(${r.codigorestaurantero
        || r.id_usuario})">
        Editar
    </button>

    <button class="btn-delete"
        onclick="RestauranterosManager.deleteRestaurantero(${r.codigorestaurantero
        || r.id_usuario})">
        Eliminar
    </button>
</div>

`).join(");
```

}

```
static async createRestaurantero(data) {  
  
    try {  
  
        UIUtils.showLoading(true);  
  
        await SazonAPI.createRestaurantero(data);  
  
        UIUtils.showSuccess('Restaurantero creado exitosamente');  
  
        await this.loadAllRestauranteros();  
  
        this.clearForm();  
  
    } catch (error) {  
  
        UIUtils.showError('Error al crear restaurantero: ' + error.message);  
  
    } finally {  
  
        UIUtils.showLoading(false);  
  
    }  
  
}
```

```
static async updateRestaurantero(id, data) {  
  
    try {  
  
        UIUtils.showLoading(true);  
  
        await SazonAPI.updateRestaurantero(id, data);  
  
        UIUtils.showSuccess('Restaurantero actualizado exitosamente');  
  
        await this.loadAllRestauranteros();  
  
        this.clearForm();  
  
    } catch (error) {  
  
        UIUtils.showError('Error al actualizar restaurantero: ' + error.message);  
  
    } finally {  
  
        UIUtils.showLoading(false);  
  
    }  
  
}
```

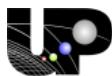
```
        } catch (error) {  
  
            UIUtils.showError('Error al actualizar restaurantero: ' + error.message);  
  
        } finally {  
  
            UIUtils.showLoading(false);  
  
        }  
  
    }  
  
  
  
    static async deleteRestaurantero(id) {  
  
        if (!confirm('¿Estás seguro de que quieres eliminar este restaurantero?')) {  
  
            return;  
  
        }  
  
  
  
        try {  
  
            UIUtils.showLoading(true);  
  
            await SazonAPI.deleteRestaurantero(id);  
  
            UIUtils.showSuccess('Restaurantero eliminado exitosamente');  
  
            await this.loadAllRestauranteros();  
  
        } catch (error) {  
  
            UIUtils.showError('Error al eliminar restaurantero: ' + error.message);  
  
        } finally {  
  
            UIUtils.showLoading(false);  
  
        }  
  
    }  

```

```
static async editRestaurantero(id) {  
  try {  
    const restaurantero = await SazonAPI.getRestauranteroById(id);  
    this.fillForm(restaurantero);  
    this.currentEditingId = id;  
  
    const form = document.getElementById('restaurantero-form-container');  
    if (form) {  
      form.scrollIntoView({ behavior: 'smooth' });  
    }  
  } catch (error) {  
    UIUtils.showError('Error al cargar datos del restaurantero: ' + error.message);  
  }  
  
  static fillForm(restaurantero) {  
    const form = document.getElementById('restaurantero-form');  
    if (!form) return;  
  
    const nombreField = document.getElementById('nombre');  
    const correoField = document.getElementById('correo');  
    const rfcField = document.getElementById('rfc');
```

```
if (nombreField) nombreField.value = restaurantero.nombreU || ";\n\nif (correoField) correoField.value = restaurantero.correo || ";\n\nif (rfcField) rfcField.value = restaurantero.rfc || ";\n\n\nconst submitBtn = form.querySelector('button[type="submit"]');\n\nif (submitBtn) {\n\n    submitBtn.textContent = 'Actualizar Restaurantero';\n\n}\n\n}\n\n\nstatic clearForm() {\n\n    const form = document.getElementById('restaurantero-form');\n\n    if (form) {\n\n        form.reset();\n\n        this.currentEditingId = null;\n\n\n        const submitBtn = form.querySelector('button[type="submit"]');\n\n        if (submitBtn) {\n\n            submitBtn.textContent = 'Crear Restaurantero';\n\n        }\n\n    }\n\n}\n\n}
```

```
static initForm() {  
  
    const form = document.getElementById('restaurantero-form');  
  
    if (!form) return;  
  
  
    form.addEventListener('submit', async (e) => {  
  
        e.preventDefault();  
  
  
        const formData = {  
  
            nombreU: document.getElementById('nombre').value.trim(),  
  
            correo: document.getElementById('correo').value.trim(),  
  
            contrasena: document.getElementById('contrasena').value,  
  
            rfc: document.getElementById('rfc')?.value.trim() || null,  
  
            tipo: 'restaurantero'  
  
        };  
  
  
        if (!formData.nombreU || !formData.correo || !formData.contrasena) {  
  
            UIUtils.showError('Nombre, correo y contraseña son obligatorios');  
  
            return;  
  
        }  
  
  
        if (!formData.correo.includes('@')) {  
  
            UIUtils.showError('Formato de correo inválido');  
  
        }  
  
    })  
}
```



```
        return;  
    }  
  
    if (this.currentEditingId) {  
        await this.updateRestaurantero(this.currentEditingId, formData);  
    } else {  
        await this.createRestaurantero(formData);  
    }  
});  
  
const cancelBtn = document.getElementById('cancel-btn');  
  
if (cancelBtn) {  
    cancelBtn.addEventListener('click', () => {  
        this.clearForm();  
    });  
}  
}  
  
}  
  
document.addEventListener('DOMContentLoaded', function() {  
    if (document.body.classList.contains('require-auth')) {  
        if (!AuthManager.requireAuth()) return;  
    }  
})
```

```
RestauranterosManager.initForm();  
RestauranterosManager.loadAllRestauranteros();  
});
```

### vistaEdicionRest

```
document.addEventListener('DOMContentLoaded', function() {  
    document.querySelectorAll('label.imagen-slot')  
    input[type="file"][accept*="image"]).forEach(function(input) {  
        input.addEventListener('change', function(event) {  
            const fileList = event.target.files;  
            const label = input.closest('label.imagen-slot');  
            const placeholderImg = label ? label.querySelector('img') : null;  
            if (fileList.length > 0 && placeholderImg) {  
                const reader = new FileReader();  
                reader.onload = function(e) {  
                    placeholderImg.src = e.target.result;  
                    placeholderImg.style.objectFit = 'cover';  
                    placeholderImg.style.width = '100%';  
                    placeholderImg.style.height = '100%';  
                };  
                reader.readAsDataURL(fileList[0]);  
            }  
        });  
    });  
});
```

});

```
async function verificarBackend(reintentos = 3) {  
  for (let i = 0; i < reintentos; i++) {  
    try {  
      const response = await fetch('http://52.23.26.163:7070/solicitudes', {  
        method: 'GET',  
        headers: { 'Content-Type': 'application/json' },  
        timeout: 10000  
      });  
  
      if (response.status < 500) {  
        return true;  
      }  
  
      if (i < reintentos - 1) {  
        await new Promise(resolve => setTimeout(resolve, 2000));  
      }  
    } catch (error) {  
      if (i < reintentos - 1) {  
        await new Promise(resolve => setTimeout(resolve, 2000));  
      }  
    }  
  }  
}
```

}

return false;

}

```
async function cargarDatosRestauranteUsuario() {
```

```
    try {
```

```
        const backendDisponible = await verificarBackend();
```

```
        if (!backendDisponible) {
```

```
            throw new Error('Backend no disponible en http://52.23.26.163:7070');
```

```
    }
```

```
    const idUsuario = sessionStorage.getItem('id') || localStorage.getItem('id');
```

```
    const correoUsuario = sessionStorage.getItem('correo') ||  
localStorage.getItem('correo');
```

```
    const loginSuccess = sessionStorage.getItem('loginSuccess') ||  
localStorage.getItem('loginSuccess');
```

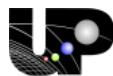
```
    if (!idUsuario && !correoUsuario && !loginSuccess) {
```

```
        alert('✖ Sesión no válida. Redirigiendo al login...');
```

```
        window.location.href = 'loginRest.html';
```

```
        return;
```

```
}
```



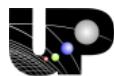
```
let telefonoUsuario = null;

if (correoUsuario) {
    try {
        const responseSolicitudes = await
fetch(`http://52.23.26.163:7070/solicitudes`, {
            method: 'GET',
            headers: { 'Content-Type': 'application/json' }
        });

        if (responseSolicitudes.ok) {
            const solicitudes = await responseSolicitudes.json();

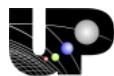
            const solicitudUsuario = solicitudes.find(s =>
                s.correo && s.correo.toLowerCase() === correoUsuario.toLowerCase()
            );
        }

        if (solicitudUsuario) {
            telefonoUsuario = solicitudUsuario.numero ||
                solicitudUsuario.telefono ||
                solicitudUsuario.telefonoContacto ||
                solicitudUsuario.celular ||
                solicitudUsuario.phone ||
        }
    }
}
```

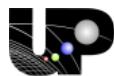


solicitudUsuario.numeroContacto;

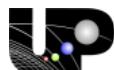
```
if (telefonoUsuario) {  
    sessionStorage.setItem('telefono', telefonoUsuario);  
}  
  
window.solicitudUsuario = solicitudUsuario;  
}  
}  
}  
}  
}  
  
try {  
    response = await fetch('http://52.23.26.163:7070/solicitudes', {  
        method: 'GET',  
        headers: { 'Content-Type': 'application/json' }  
    });  
  
    if (response.ok) {  
        const restaurantes = await response.json();  
        let restauranteUsuario = null;  
    }  
}
```



```
if (Array.isArray(restaurantes)) {  
  
    if (telefonoUsuario && telefonoUsuario.trim() !== "") {  
  
        const telefonoLimpio = telefonoUsuario.replace(/[\s\-\(\)]/g, "");  
  
  
        restauranteUsuario = restaurantes.find(r => {  
  
            const telefonosAComparar = [  
  
                r.numero,  
  
                r.telefono,  
  
                r.telefonoContacto,  
  
                r.celular,  
  
                r.phone,  
  
                r.numeroContacto,  
  
                r.numeroTelefono  
  
            ];  
  
  
  
            return telefonosAComparar.some(telefono => {  
  
                if (!telefono) return false;  
  
                const telefonoRestauranteLimpio =  
                    telefono.toString().replace(/[\s\-\(\)]/g, "");  
  
                return telefonoRestauranteLimpio === telefonoLimpio;  
  
            });  
  
        });  
  
    }  
}
```



```
if (!restauranteUsuario && correoUsuario) {  
  
    restauranteUsuario = restaurantes.find(r => {  
  
        const correosAComparar = [  
  
            r.correo,  
  
            r.correoContacto,  
  
            r.email,  
  
            r.correoElectronico,  
  
            r.emailContacto,  
  
            r.contactoCorreo  
  
        ];  
  
  
        return correosAComparar.some(correo =>  
  
            correo && correo.toLowerCase() ===  
correoUsuario.toLowerCase()  
  
        );  
  
    });  
  
}  
  
  
if (!restauranteUsuario && idUsuario) {  
  
    restauranteUsuario = restaurantes.find(r => {  
  
        const idsAComparar = [  
  
            r.idUsuario,  
  
        ];  
  
    });  
  
}
```

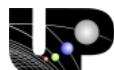


```
r.usuariold,  
r.id,  
r.restauranterold,  
r.propietariold  
];  
  
return idsAComparar.some(id =>  
    id && id.toString() === idUsuario.toString()  
);  
});  
}  
}
```

```
if (restauranteUsuario) {  
    window.restauranteActual = restauranteUsuario;
```

```
cargarDatosEnFormulario(restauranteUsuario, window.solicitudUsuario);
```

```
const etiquetasIndividuales = [  
    restauranteUsuario.etiqueta1,  
    restauranteUsuario.etiqueta2,  
    restauranteUsuario.etiqueta3  
].filter(etiqueta =>
```



```
etiqueta &&

etiqueta !== " &&

etiqueta !== 'Seleccionar' &&

etiqueta.trim() !== ""

);

if (etiquetasIndividuales.length > 0) {

    cargarEtiquetasRestaurante(etiquetasIndividuales.join(', '));

} else if (restauranteUsuario.etiquetas) {

    cargarEtiquetasRestaurante(restauranteUsuario.etiquetas);

}

return;

} else {

    throw new Error('No se encontró ningún restaurante asociado al

usuario');

}

} else {

    throw new Error(`Error al obtener restaurantes: ${response.status}`);

}

}

} catch (fetchError) {

    throw fetchError;
```

}

```
} catch (error) {  
  
    if (error.message.includes('fetch')) {  
  
        alert('✖ Error de conexión: No se puede conectar al servidor.');//  
  
    } else if (error.message.includes('404')) {  
  
        alert('✖ Endpoint no encontrado: Verifique la configuración del servidor.');//  
  
    } else if (error.message.includes('500')) {  
  
        alert('✖ Error del servidor: Contacte al administrador.');//  
  
    } else {  
  
        alert(`✖ Error al cargar los datos del restaurante: ${error.message}`);  
  
    }  
  
}
```

```
function cargarDatosEnFormulario(restaurante) {  
  
    const nombreElement = document.getElementById('restauranteNombre');  
  
    if (nombreElement) {  
  
        nombreElement.textContent = restaurante.restaurante || 'Mi Restaurante';  
  
    }  
  
    const inputUbicacion = document.querySelector('input[placeholder="Ingrese la  
dirección"]');
```

```
if (inputUbicacion && restaurante.direccion) {  
  
    inputUbicacion.value = restaurante.direccion;  
  
}  
  
  
const inputTelefono = document.querySelector('input[placeholder="Ingrese su  
número celular"]');  
  
if (inputTelefono && restaurante.numero) {  
  
    inputTelefono.value = restaurante.numero;  
  
}  
  
  
const horariosInputs = document.querySelectorAll('.horarios-inputs input');  
  
if (restaurante.horario && horariosInputs.length > 0) {  
  
    horariosInputs[0].value = restaurante.horario;  
  
}  
  
  
const inputInstagram = document.querySelector('input[placeholder="Ingrese su  
instagram"]');  
  
if (inputInstagram && restaurante.instagram) {  
  
    inputInstagram.value = restaurante.instagram;  
  
}  
  
  
const inputFacebook = document.querySelector('input[placeholder="Ingrese su  
facebook"]');  
  
if (inputFacebook && restaurante.facebook) {  
  
    inputFacebook.value = restaurante.facebook;
```

}

```
const img1 = document.getElementById('imgGaleria1');

if (img1) img1.src = restaurante.imagen1 || '../images/imagen.png';

const img2 = document.getElementById('imgGaleria2');

if (img2) img2.src = restaurante.imagen2 || '../images/imagen.png';

const img3 = document.getElementById('imgGaleria3');

if (img3) img3.src = restaurante.imagen3 || '../images/imagen.png';

const btnMenu = document.querySelector('label.btn-menu input[type="file"]');

if (btnMenu && restaurante.menu) {

    const label = btnMenu.closest('label.btn-menu');

    const menuSpan = label.querySelector('span');

    if (menuSpan) {

        const nombreArchivo = restaurante.menu.split('/').pop() || restaurante.menu;

        menuSpan.textContent = nombreArchivo;

    }

}

const estadoMembresia = window.estadoMembresiaRestaurante || "activa";

const btnHeaderSubscripcion = document.getElementById('btnHeaderSubscripcion');
```

```
const btnEstadisticas = document.getElementById('btnEstadisticas');

const mensaje = document.getElementById('membresiaMensaje');

if (btnEstadisticas && mensaje) {

    if (estadoMembresia === "activa") {

        btnEstadisticas.style.display = "inline-block";

        btnEstadisticas.disabled = false;

        if (btnHeaderSubscripcion) btnHeaderSubscripcion.style.display = "none";

    } else if (estadoMembresia === "pendiente") {

        btnEstadisticas.style.display = "none";

        if (btnHeaderSubscripcion) btnHeaderSubscripcion.style.display = "inline-block";

    } else {

        btnEstadisticas.style.display = "none";

        mensaje.textContent = "";

        if (btnHeaderSubscripcion) btnHeaderSubscripcion.style.display = "inline-block";

    }

}

cargarDatosRestauranteUsuario();

const ETIQUETAS_DISPONIBLES = [
    'Comida Rápida',
    'Pet Friendly',
]
```

'Familiar',  
'Económico',  
'Gourmet',  
'Vegetariano',  
'Delivery',  
'Terraza',  
'WiFi Gratuito',  
'Estacionamiento'  
];

```
const selectsEtiquetas = document.querySelectorAll('.etiquetas .filtro-select');
```

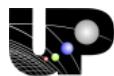
```
selectsEtiquetas.forEach(select => {  
    while (select.children.length > 1) {  
        select.removeChild(select.lastChild);  
    }  
}
```

```
ETIQUETAS_DISPONIBLES.forEach(etiqueta => {  
    const option = document.createElement('option');  
    option.value = etiqueta;  
    option.textContent = etiqueta;  
    select.appendChild(option);  
});
```

});

```
window.obtenerEtiquetasSeleccionadas = function() {  
  
    const etiquetasSeleccionadas = [];  
  
    selectsEtiquetas.forEach((select, index) => {  
  
        if (select.value && select.value !== "" && select.value !== 'Seleccionar') {  
  
            etiquetasSeleccionadas.push(select.value);  
  
        }  
  
    });  
  
};
```

```
window.cargarEtiquetasRestaurante = function(etiquetasString) {  
  
    if (!etiquetasString) {  
  
        return;  
  
    }  
  
    const etiquetasArray = etiquetasString.split(',').map(e => e.trim()).filter(e => e !== "");  
  
    etiquetasArray.forEach((etiqueta, index) => {  
  
    });  
  
};
```



```
if (index < selectsEtiquetas.length) {  
  
    console.log(`💡 Asignando etiqueta ${index + 1}: "${etiqueta}" al select`);  
  
    selectsEtiquetas[index].value = etiqueta;  
  
  
    if (selectsEtiquetas[index].value === etiqueta) {  
  
        console.log(`✅ Etiqueta ${index + 1} asignada correctamente`);  
  
    } else {  
  
        console.warn(`❌ Error al asignar etiqueta ${index + 1}: "${etiqueta}". Valor  
actual: "${selectsEtiquetas[index].value}"`);  
  
        // Verificar si la opción existe en el select  
  
        const opcionExiste =  
Array.from(selectsEtiquetas[index].options).some(option => option.value === etiqueta);  
  
        console.log(`🔍 ¿La opción "${etiqueta}" existe en el select?`,  
opcionExiste);  
  
    }  
  
};  
  
});  
  
selectsEtiquetas.forEach((select, index) => {  
  
});  
  
};  
  
function establecerEtiquetasPorDefecto() {  
  
const etiquetasPorDefecto = ['Seleccionar', 'Seleccionar', 'Seleccionar'];
```

```
etiquetasPorDefecto.forEach((etiqueta, index) => {  
    if (index < selectsEtiquetas.length) {  
        selectsEtiquetas[index].value = etiqueta;  
    }  
});  
  
}  
  
establecerEtiquetasPorDefecto();  
  
  
const menuInput = document.getElementById('menuInput');  
if (menuInput) {  
    menuInput.addEventListener('change', function(event) {  
        const file = event.target.files[0];  
        const btnMenu = document.querySelector('.btn-menu');  
        const menuSpan = btnMenu.querySelector('span');  
  
        if (file) {  
            menuSpan.textContent = file.name;  
        } else {  
            menuSpan.textContent = 'Seleccionar archivo PDF';  
        }  
    });  
}
```

```
selectsEtiquetas.forEach((select, index) => {
    select.addEventListener('change', function() {
        const etiquetaSeleccionada = this.value;

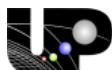
        if (etiquetaSeleccionada && etiquetaSeleccionada !== "") {
            selectsEtiquetas.forEach(otroSelect, otroIndex) => {
                if (otroIndex !== index) {
                    const opcionARemover =
                        otroSelect.querySelector(`option[value="${etiquetaSeleccionada}"]`);

                    if (opcionARemover && otroSelect.value !== etiquetaSeleccionada) {
                        opcionARemover.style.display = 'none';
                    }
                }
            });
        }

        actualizarOpcionesDisponibles();
    });
});

function actualizarOpcionesDisponibles() {
    const etiquetasYaSeleccionadas = [];
}
```

```
selectsEtiquetas.forEach(select => {  
  
    if (select.value && select.value !== "") {  
  
        etiquetasYaSeleccionadas.push(select.value);  
  
    }  
  
});  
  
selectsEtiquetas.forEach(select => {  
  
    const etiquetaActual = select.value;  
  
    Array.from(select.options).forEach(option => {  
  
        if (option.value === "" || option.value === etiquetaActual) {  
  
            option.style.display = "";  
  
        } else if (etiquetasYaSeleccionadas.includes(option.value)) {  
  
            option.style.display = 'none';  
  
        } else {  
  
            option.style.display = "";  
  
        }  
  
    });  
  
});  
  
}  
  
  
const btnAplicar = document.querySelector('.btn-aplicar');  
  
if (btnAplicar) {  
  
    btnAplicar.addEventListener('click', async function() {  
  
        if (!window.restauranteActual) {
```



```
    alert('✖ No se han cargado los datos del restaurante');

    return;

}

const etiquetasSeleccionadas = obtenerEtiquetasSeleccionadas();

console.log('🏷️ Etiquetas obtenidas:', etiquetasSeleccionadas);

const direccion = document.querySelector('input[placeholder="Ingrese la dirección"])?.value || "";

const telefono = document.querySelector('input[placeholder="Ingrese su número celular"])?.value || "";

const horario = document.querySelector('.horarios-inputs input')?.value || "";

const instagram = document.querySelector('input[placeholder="Ingrese su instagram"])?.value || "";

const facebook = document.querySelector('input[placeholder="Ingrese su facebook"])?.value || "";

const img1Input = document.getElementById('imgGaleria1Input');

const img2Input = document.getElementById('imgGaleria2Input');

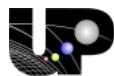
const img3Input = document.getElementById('imgGaleria3Input');

console.log('img1Input:', img1Input, 'files:', img1Input && img1Input.files);

console.log('img2Input:', img2Input, 'files:', img2Input && img2Input.files);

console.log('img3Input:', img3Input, 'files:', img3Input && img3Input.files);

const menuInput = document.getElementById('menuInput');
```



```
const formData = new FormData();

    formData.append('restaurante', window.restauranteActual.restaurante || window.restauranteActual.nombre || "");

    formData.append('correo', window.restauranteActual.correo || "");

    formData.append('direccion', direccion);

    formData.append('numero', telefono);

    formData.append('horario', horario);

    formData.append('facebook', facebook);

    formData.append('instagram', instagram);

    const etiquetasArray = etiquetasSeleccionadas.split(',').map(e => e.trim()).filter(e => e !== "");

    console.log('🏷️ Array de etiquetas procesado:', etiquetasArray);

    formData.append('etiqueta1', etiquetasArray[0] || 'Seleccionar');

    formData.append('etiqueta2', etiquetasArray[1] || 'Seleccionar');

    formData.append('etiqueta3', etiquetasArray[2] || 'Seleccionar');

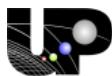
    console.log('📦 Etiquetas que se enviarán:', {
        etiqueta1: etiquetasArray[0] || 'Seleccionar',
        etiqueta2: etiquetasArray[1] || 'Seleccionar',
        etiqueta3: etiquetasArray[2] || 'Seleccionar'
    })
```

});

```
if (img1Input && img1Input.files && img1Input.files[0]) {  
  
    formData.append('imagen1', img1Input.files[0]);  
  
}  
  
if (img2Input && img2Input.files && img2Input.files[0]) {  
  
    formData.append('imagen2', img2Input.files[0]);  
  
}  
  
if (img3Input && img3Input.files && img3Input.files[0]) {  
  
    formData.append('imagen3', img3Input.files[0]);  
  
}  
  
if (menuInput && menuInput.files && menuInput.files[0]) {  
  
    formData.append('menu', menuInput.files[0]);  
  
}
```

```
const formDataPreview = {};  
  
formData.forEach((value, key) => {  
  
    if (value instanceof File) {  
  
        formDataPreview[key] = {  
  
            name: value.name,  
  
            size: value.size,  
  
            type: value.type  
  
    };
```

```
    } else {  
  
        formDataPreview[key] = value;  
  
    }  
  
});  
  
console.log('📦 Datos a enviar al backend:', formDataPreview);  
  
  
  
try {  
  
    const idUsuario = sessionStorage.getItem('id') || localStorage.getItem('id');  
  
        const correoUsuario = sessionStorage.getItem('correo') ||  
localStorage.getItem('correo');  
  
        const idSolicitud = (window.restauranteActual &&  
(window.restauranteActual.id_solicitud || window.restauranteActual.id ||  
window.restauranteActual.idSolicitud))  
  
        || (window.solicitudUsuario && (window.solicitudUsuario.id_solicitud ||  
window.solicitudUsuario.id));  
  
    if (!idSolicitud) {  
  
        alert('❌ No se puede actualizar: No se tiene el ID de la solicitud');  
  
        return;  
  
    }  
  
    const endpointUrl = `http://52.23.26.163:7070/solicitudes/${idSolicitud}`;  
  
    const endpointUrlWithFiles =  
`http://52.23.26.163:7070/solicitudes/${idSolicitud}/with-files`;  
  
    const response = await fetch(endpointUrlWithFiles, {  
  
        method: 'PUT',  
    }
```



```
body: formData

});

if (response.ok) {

    const responseData = await response.json();

    console.log('✅ Respuesta del servidor:', responseData);

    alert('✅ Datos del restaurante actualizados correctamente con archivos');

    await cargarDatosRestauranteUsuario();

} else {

    const errorText = await response.text();

    console.error('❌ Error del servidor:', response.status, errorText);

    alert('❌ Error al actualizar (${response.status}): ${errorText}');

}

} catch (error) {

    console.error('❌ Error al guardar:', error);

    if (error.name === 'TypeError' && error.message.includes('fetch')) {

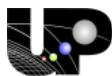
        alert('❌ Error de conexión: Verifique que el backend esté ejecutándose en  
http://52.23.26.163:7070');

    } else if (error.message.includes('CORS')) {

        alert('❌ Error CORS: El backend no permite conexiones desde este  
origen');

    }

}
```



```
        alert(` Error de conexión al guardar los datos: ${error.message}`);
    }
}

});

}

});
```

### **vistaPrincipalAdmin**

```
document.addEventListener('DOMContentLoaded', function () {

    const botones = document.querySelectorAll('.sidebar-item');

    const vistas = {

        restaurantes: document.getElementById('vista-restaurantes'),
        solicitudes: document.getElementById('vista-solicitudes'),
        membresias: document.getElementById('vista-membresias')
    };

});
```

```
botones.forEach(boton => {

    boton.addEventListener('click', function () {

        botones.forEach(b => b.classList.remove('active'));

        this.classList.add('active');
```

```
Object.values(vistas).forEach(vista => vista.style.display = 'none');

const vistaSeleccionada = this.getAttribute('data-vista');
```

```
if (vistas[vistaSeleccionada]) {  
  
    vistas[vistaSeleccionada].style.display = 'block';  
  
    // Cargar datos reales según la vista seleccionada  
  
    if (vistaSeleccionada === 'solicitudes') {  
  
        cargarSolicitudes();  
  
    } else if (vistaSeleccionada === 'restaurantes') {  
  
        cargarRestaurantes();  
  
    }  
  
};  
  
});
```

```
if (botones && botones.length > 0) {  
  
    botones[0].click();  
  
}
```

```
function cargarMembresias() {
```

```
const tbody = document.querySelector('#vista-membresias tbody');  
  
tbody.innerHTML = "";  
  
membresias.forEach(m => {  
  
    const tr = document.createElement('tr');
```

```
let acciones = "";

if (m.estado.toLowerCase() !== 'activa') {

    acciones += `<button class="btn-aceptar" title="Aceptar"></button>`;

}

acciones += `<button class="btn-rechazar" title="Rechazar"></button>`;

tr.innerHTML = `

<td>${m.restaurante}</td>

<td>${m.propietario}</td>

<td>${m.correo}</td>

<td>${m.fecha}</td>

<td>${m.estado.charAt(0).toUpperCase() + m.estado.slice(1)}</td>

<td>${acciones}</td>

`;

tbody.appendChild(tr);

});

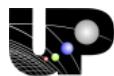
}

const btnSidebarMembresias = document.querySelector('[data-vista="membresias"]');

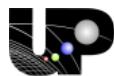
if (btnSidebarMembresias) {

    btnSidebarMembresias.addEventListener('click', cargarMembresias);

}
```

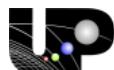


```
function agregarEventosTabla(selectorTabla, tipo, editable = false) {  
  
    const tabla = document.querySelector(selectorTabla);  
  
    if (!tabla) return;  
  
    tabla.addEventListener('click', function (e) {  
  
        if (e.target.closest('.btn-editar')) {  
  
            const fila = e.target.closest('tr');  
  
            if (!fila) return;  
  
            const celda = fila.querySelector('td');  
  
            if (!celda) return;  
  
            if (editable) {  
  
                if (celda.querySelector('input')) return;  
  
                const valorActual = celda.textContent;  
  
                celda.innerHTML = '';  
  
                const input = document.createElement('input');  
  
                input.type = 'text';  
  
                input.value = valorActual;  
  
                input.style.width = '80%';  
  
                celda.appendChild(input);  
  
                input.focus();  
  
                input.addEventListener('blur', function () {  
  
                    celda.textContent = input.value || valorActual;  
  
                });  
            }  
        }  
    });  
}
```



```
input.addEventListener('keydown', function (ev) {  
    if (ev.key === 'Enter') {  
        input.blur();  
    }  
});  
  
} else {  
  
    const nombre = celda.textContent;  
  
    alert('Editar ' + tipo + ': ' + nombre);  
  
}  
  
}  
  
if (e.target.closest('.btn-eliminar')) {  
  
    const fila = e.target.closest('tr');  
  
    const nombre = fila ? fila.children[0].textContent : '';  
  
    if (confirm('¿Seguro que deseas eliminar este ' + tipo + ': ' + nombre + '?')) {  
  
        fila.remove();  
    }  
  
};  
  
});  
  
}  
  
  
const tablaRestaurantesAdmin = document.querySelector('#vista-restaurantes  
.tabla-admin');  
  
if (tablaRestaurantesAdmin) {
```

```
tablaRestaurantesAdmin.addEventListener('click', async function (e) {  
  
    if (e.target.closest('.btn-eliminar')) {  
  
        const fila = e.target.closest('tr');  
  
        const nombre = fila ? fila.children[0].textContent : '';  
  
        const idRestaurante = fila ? fila.getAttribute('data-id-restaurante') : '';  
  
  
        if (!confirm(`¿Seguro que deseas eliminar este restaurante: ${nombre}?`))  
            return;  
  
  
        try {  
  
            console.log('🗑️ Eliminando restaurante:', { nombre, idRestaurante });  
  
  
            const response = await  
                fetch(`http://52.23.26.163:7070/solicitudes/${idRestaurante}` , {  
  
                    method: 'DELETE',  
  
                    headers: {  
  
                        'Content-Type': 'application/json'  
  
                    }  
  
                });  
  
  
            console.log('📡 Respuesta del servidor:', {  
  
                status: response.status,  
  
                statusText: response.statusText,  
  
            });  
  
        } catch (error) {  
            console.error(error);  
        }  
    }  
});
```



```
ok: response.ok

});

if (response.ok) {

    fila.remove();

    alert(`✅ Restaurante "${nombre}" eliminado correctamente`);

    await cargarRestaurantes();

} else {

    const errorText = await response.text();

    console.error('❌ Error del servidor:', errorText);

    let errorMessage = 'Error al eliminar el restaurante';

    try {

        const errorObj = JSON.parse(errorText);

        errorMessage = errorObj.message || errorMessage;

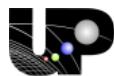
    } catch (e) {

        errorMessage = errorText || errorMessage;

    }

    alert(`❌ ${errorMessage}`);

} catch (err) {
```



```
        console.error('✖ Error de conexión:', err);

        alert('✖ Error de conexión al eliminar el restaurante. Verifica que el
servidor esté funcionando.');

    }
}
```

```
});
```

```
}
```

```
agregarEventosTabla('#vista-solicitudes .tabla-admin', 'solicitud de usuario');
```

```
const tablaSolicitudes = document.querySelector('#vista-solicitudes .tabla-admin
tbody');
```

```
const tablaRestaurantesBody = document.querySelector('#vista-restaurantes
.tabla-admin tbody');
```

```
if (tablaSolicitudes && tablaRestaurantesBody) {
```

```
    tablaSolicitudes.addEventListener('click', async function (e) {
```

```
        const btnImg = e.target.closest('.btn-ver-imagenes');
```

```
        if (btnImg) {
```

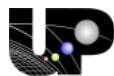
```
            const imagenes = JSON.parse(btnImg.getAttribute('data-imagenes'));
```

```
            verImagenes(imagenes);
```

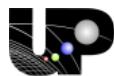
```
            return;
```

```
}
```

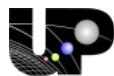
```
        const btnComp = e.target.closest('.btn-ver-comprobante');
```



```
if (btnComp) {  
  
    const comprobante = btnComp.getAttribute('data-comprobante');  
  
    verComprobante(comprobante);  
  
    return;  
}  
  
if (e.target.closest('.btn-aceptar')) {  
  
    const fila = e.target.closest('tr');  
  
    if (!fila) return;  
  
  
    const id = fila.getAttribute('data-id-solicitud');  
  
    if (!id) {  
  
        alert('No se pudo obtener el ID de la solicitud.');//  
  
        return;  
    }  
  
  
    if (!confirm('¿Seguro que deseas aceptar esta solicitud? El restaurante será  
agregado al sistema.')) return;  
  
  
    try {  
  
        const tds = fila.querySelectorAll('td');//  
  
        const datosRestaurante = {  
  
            nombre: tds[0] ? .textContent || "",  
  
            propietario: tds[1] ? .textContent || "",  
            ...  
        };  
  
        // Agregar los datos al restaurante en la base de datos  
        // ...  
    } catch (error) {  
        console.error(error);  
    }  
}
```



```
correo: tds[2]?.textContent || "",  
numero: tds[3]?.textContent || "",  
direccion: tds[4]?.textContent || "",  
horario: tds[5]?.textContent || ""  
};  
  
console.log('Datos de la solicitud a aprobar:', datosRestaurante);  
  
const response = await  
fetch(`http://52.23.26.163:7070/solicitudes/aprobar/${id}` , {  
    method: 'PUT',  
    headers: {  
        'Content-Type': 'application/json'  
    },  
    body: JSON.stringify({ estado: 'aprobado' })  
});  
  
  
  
if (response.ok) {  
    const imagenesBsns = fila.querySelector('.btn-ver-imagenes');  
    let imagen1 = "", imagen2 = "", imagen3 = "";  
    if (imagenesBsns && imagenesBsns.dataset.dataset.imagenes) {  
        try {  
            const imagenesArr =  
JSON.parse(imagenesBsns.dataset.dataset.imagenes);  
            imagen1 = imagenesArr[0] || "";
```



```
        imagen2 = imagenesArr[1] || "";

        imagen3 = imagenesArr[2] || "";

    } catch (e) {}

}

const datosRestauranteInsert = {

    nombre: datosRestaurante.nombre,

    direccion: datosRestaurante.direccion,

    horario: datosRestaurante.horario,

    telefono: datosRestaurante.numero,

    imagen1,

    imagen2,

    imagen3

};

try {

    await fetch('http://52.23.26.163:7070/restaurantes', {

        method: 'POST',

        headers: { 'Content-Type': 'application/json' },

        body: JSON.stringify(datosRestauranteInsert)

    });

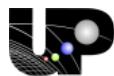
    console.log('Restaurante insertado en tabla restaurante:',

datosRestauranteInsert);

} catch (e) {

    console.error('Error al insertar en restaurante:', e);

}
```



}

```
fila.remove();
```

```
const vistaRestaurantesActiva =  
document.getElementById('vista-restaurantes').style.display !== 'none';
```

```
if (vistaRestaurantesActiva) {  
  
    await cargarRestaurantes();  
  
}
```

```
    alert(`✅ Solicitud aprobada exitosamente!\n\nRestaurante  
"${datosRestaurante.nombre}" ha sido agregado al sistema.`);
```

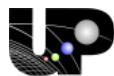
```
} else {  
  
    const errorText = await response.text();  
  
    const errorObj = JSON.parse(errorText);
```

```
        if (errorObj.message && errorObj.message.includes("doesn't have a  
default value")) {
```

```
            alert(`❌ Error en el backend: Falta configurar valores por defecto en  
la base de datos.\n\nDetalles técnicos: ${errorObj.message}\n\n💡 Solución: El  
desarrollador del backend necesita asegurar que todos los campos requeridos se estén  
copiando correctamente de la solicitud al crear el restaurante.`);
```

```
} else {
```

```
        alert(`❌ Error al aprobar la solicitud:\n\n${errorObj.message} ||  
errorText`);
```



```
        }

    }

} catch (err) {

    console.error('Error de conexión:', err);

    alert('✖ Error de conexión al procesar la solicitud. Verifica que el servidor  
esté funcionando.');

}

if (e.target.closest('.btn-rechazar')) {

    const fila = e.target.closest('tr');

    if (!fila) return;

    const id = fila.getAttribute('data-id-solicitud');

    if (!id) {

        alert('No se pudo obtener el ID de la solicitud.');

        return;

    }

    if (!confirm('¿Seguro que deseas rechazar esta solicitud?')) return;

    try {

        const response = await fetch(`http://52.23.26.163:7070/solicitudes/${id}` , {

            method: 'DELETE'

        });

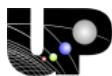
        if (response.ok) {

            fila.remove();

        }

    }

}
```



```
        alert('Solicitud rechazada y eliminada correctamente');

    } else {

        const errorText = await response.text();

        alert('Error al rechazar: ' + errorText);

    }

} catch (err) {

    alert('Error de conexión al rechazar');

}

}

});

}

async function cargarSolicitudes() {

try {

    const response = await fetch('http://52.23.26.163:7070/solicitudes');

    if (!response.ok) throw new Error('No se pudo obtener las solicitudes');

    const solicitudes = await response.json();

    const tbody = document.querySelector('#vista-solicitudes tbody');

    tbody.innerHTML = "";

    const solicitudesPendientes = solicitudes.filter(solicitud =>

        solicitud.estado && solicitud.estado.toLowerCase() === 'pendiente'
```

);

```
solicitudesPendientes.forEach(solicitud => {

    const imagenes = [solicitud.imagen1, solicitud.imagen2,
solicitud.imagen3].filter(Boolean);

    const tr = document.createElement('tr');

    tr.setAttribute('data-id-solicitud', solicitud.id_solicitud || solicitud.id || "");

    tr.innerHTML = `

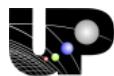
        <td>${solicitud.restaurante || ""}</td>
        <td>${solicitud.propietario || ""}</td>
        <td>${solicitud.correo || ""}</td>
        <td>${solicitud.numero || ""}</td>
        <td>${solicitud.direccion || ""}</td>
        <td>${solicitud.horario || ""}</td>
        <td>

            ${imagenes.length > 0 ? `<button class="btn-ver-imagenes" title="Ver
imágenes" data-imagenes='${JSON.stringify(imagenes)}'>

</button>` : 'Sin imágenes'}
        </td>
        <td>

            ${solicitud.comprobante ? `<button class="btn-ver-comprobante"
title="Ver comprobante" data-comprobante='${solicitud.comprobante}'>

</button>` : 'Sin comprobante'}
        </td>
    
```



```
</button>` : 'Sin comprobante'>

</td>

<td>

    <button class="btn-aceptar" title="Aceptar"></button>

    <button class="btn-rechazar" title="Rechazar"></button>

</td>

`;

tbody.appendChild(tr);

});

if (solicitudesPendientes.length === 0) {

    const tr = document.createElement('tr');

    tr.innerHTML = `

        <td colspan="9" style="text-align: center; padding: 20px; color: #666;">

             No hay solicitudes pendientes

        </td>

`;

    tbody.appendChild(tr);

}

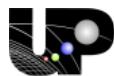
} catch (error) {

    alert("Error al cargar solicitudes.");
}
```

}

}

```
async function cargarRestaurantes() {  
    try {  
        console.log('📋 Cargando lista de restaurantes...');  
  
        const response = await fetch('http://52.23.26.163:7070/solicitudes');  
        if (!response.ok) throw new Error('No se pudo obtener las solicitudes');  
        const solicitudes = await response.json();  
  
        const restaurantesAprobados = solicitudes.filter(s => s.estado &&  
            s.estado.toLowerCase() === 'aprobado');  
        console.log(`📍 Restaurantes aprobados encontrados: ${  
            restaurantesAprobados.length}`);  
  
        const tbody = document.querySelector('#vista-restaurantes tbody');  
        tbody.innerHTML = '';  
  
        if (restaurantesAprobados.length === 0) {  
            const tr = document.createElement('tr');  
            tr.innerHTML = `<td colspan="3" style="text-align: center; padding: 20px; color: #666;">
```



No hay restaurantes registrados

</td>

`;

tbody.appendChild(tr);

return;

}

restaurantesAprobados.forEach(restaurante => {

const tr = document.createElement('tr');

const idSolicitud = restaurante.id\_solicitud || restaurante.id ||  
restaurante.idSolicitud || ";

tr.setAttribute('data-id-restaurante', idSolicitud);

tr.innerHTML = `

<td>\${restaurante.restaurante || 'Sin nombre'}</td>

<td>\${restaurante.direccion || 'Sin dirección'}</td>

<td>

<button class="btn-eliminar" title="Eliminar"

data-nombre="\${restaurante.restaurante || 'Sin nombre'}">



</button>

</td>

`;

tbody.appendChild(tr);

```
});  
  
} catch (error) {  
  
    console.error('✖ Error al cargar restaurantes:', error);  
  
    alert("✖ Error al cargar restaurantes: " + error.message);  
  
}  
  
}  
  
  
  
function crearModal(contenidoHtml) {  
  
    let modal = document.getElementById('modal-visualizador');  
  
    if (!modal) {  
  
        modal = document.createElement('div');  
  
        modal.id = 'modal-visualizador';  
  
        modal.className = 'modal-visualizador';  
  
        document.body.appendChild(modal);  
  
    }  
  
    modal.innerHTML = `<div id="modal-contenido-visualizador"  
class="modal-contenido-visualizador">  
  
        <button id="cerrar-modal-visualizador"  
class="cerrar-modal-visualizador">&times;</button>  
  
        ${contenidoHtml}  
  
    </div>`;  
  
    modal.style.display = 'flex';  
  
    document.getElementById('cerrar-modal-visualizador').onclick = function() {
```

```
modal.style.display = 'none';

};

modal.onclick = function(e) {

    if (e.target === modal) {

        modal.style.display = 'none';

    }

};

}

window.verImagenes = function(imagenes) {

    if (!Array.isArray(imagenes)) imagenes = [imagenes];

    let html = '<h3>Imágenes</h3><div style="display:flex;gap:10px;flex-wrap:wrap;">';

    imagenes.forEach(url => {

        html += ``;

    });

    html += '</div>';

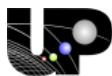
    crearModal(html);

}

window.verComprobante = function(url) {

    let html = '<h3>Comprobante</h3>';

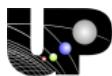
    if (url && !/^https?:\/\/.test(url) && url.endsWith('.pdf')) {
```



```
if (!url.includes('/uploads/documents/')) {  
  
    url = '/uploads/documents/' + url;  
  
}  
  
url = 'http://52.23.26.163:7070' + url;  
  
}  
  
if (url && url.endsWith('.pdf')) {  
  
    html += `<iframe src="${url}"  
style="width:600px;height:600px;border:none;"></iframe>`;  
  
} else if (url) {  
  
    html += `<a href="${url}" target="_blank">Abrir comprobante</a>`;  
  
} else {  
  
    html += '<p>No hay comprobante disponible.</p>';  
  
}  
  
crearModal(html);  
  
}  
  
});
```

### vistaPrincipalRestaurante

```
document.addEventListener('DOMContentLoaded', function() {  
  
    document.querySelectorAll('label.imagen-slot  
input[type="file"][accept*="image"]').forEach(function(input) {  
  
        input.addEventListener('change', function(event) {  
  
            const fileList = event.target.files;
```



```
const label = input.closest('label.imagen-slot');

const placeholderImg = label ? label.querySelector('img') : null;

if (fileList.length > 0 && placeholderImg) {

    const reader = new FileReader();

    reader.onload = function(e) {

        placeholderImg.src = e.target.result;
        placeholderImg.style.objectFit = 'cover';
        placeholderImg.style.width = '100%';
        placeholderImg.style.height = '100%';

    };

    reader.readAsDataURL(fileList[0]);

}

});

});

document.querySelectorAll('label.btn-menu input[type="file"]')[accept="application/pdf"].forEach(function(input) {

    input.addEventListener('change', function(event) {

        const fileList = event.target.files;
        const label = input.closest('label.btn-menu');

        Array.from(label.children).forEach(function(child) {

            if (child.tagName === 'IMG') {

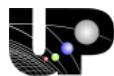
                child.style.display = (fileList.length > 0) ? 'none' : '';

            }

        });

    });

});
```



```
    }

});

let fileNameSpan = label.querySelector('.file-name');

if (!fileNameSpan) {

    fileNameSpan = document.createElement('span');

    fileNameSpan.className = 'file-name';

    fileNameSpan.style.marginLeft = '10px';

    fileNameSpan.style.fontWeight = 'normal';

    label.appendChild(fileNameSpan);

}

if (fileList.length > 0) {

    fileNameSpan.textContent = `Archivo seleccionado: ${fileList[0].name}`;

    fileNameSpan.style.display = 'inline-block';

} else {

    fileNameSpan.textContent = "";

    fileNameSpan.style.display = 'none';

    Array.from(label.children).forEach(function(child) {

        if (child.tagName === 'IMG') {

            child.style.display = "";

        }

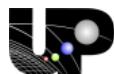
    });

}

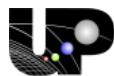
});
```

});

```
async function verificarBackend(reintentos = 3) {  
  
    for (let i = 0; i < reintentos; i++) {  
  
        try {  
  
            const response = await fetch('http://52.23.26.163:7070/solicitudes', {  
  
                method: 'GET',  
  
                headers: { 'Content-Type': 'application/json' },  
  
                timeout: 10000  
  
            });  
  
  
            if (response.status < 500) {  
  
                return true;  
  
            }  
  
  
            console.warn(`Intento ${i + 1}: Servidor responde pero con error  
${response.status}`);  
  
            if (i < reintentos - 1) {  
  
                await new Promise(resolve => setTimeout(resolve, 2000));  
  
            }  
  
        } catch (error) {  
  
            console.warn(`Intento ${i + 1}: Error de conexión:`, error.message);  
  
            if (i < reintentos - 1) {  
  
            }  
  
        }  
  
    }  
  
}
```



```
        await new Promise(resolve => setTimeout(resolve, 2000));  
  
    }  
  
}  
  
}  
  
return false;  
  
}  
  
  
  
async function cargarDatosRestauranteUsuario() {  
  
try {  
  
    const backendDisponible = await verificarBackend();  
  
    if (!backendDisponible) {  
  
        throw new Error('Backend no disponible en http://52.23.26.163:7070');  
  
    }  
  
  
  
    const idUsuario = sessionStorage.getItem('id') || localStorage.getItem('id');  
  
        const correoUsuario = sessionStorage.getItem('correo') ||  
localStorage.getItem('correo');  
  
        const loginSuccess = sessionStorage.getItem('loginSuccess') ||  
localStorage.getItem('loginSuccess');  
  
  
  
if (!idUsuario && !correoUsuario && !loginSuccess) {  
  
    alert('✖ Sesión no válida. Redirigiendo al login...');  
  
    window.location.href = 'loginRest.html';  
}
```



```
    return;  
}
```

```
let telefonoUsuario = null;
```

```
if (correoUsuario) {
```

```
    try {
```

```
        const responseSolicitudes = await
```

```
fetch(`http://52.23.26.163:7070/solicitudes`, {
```

```
    method: 'GET',
```

```
    headers: { 'Content-Type': 'application/json' }
```

```
});
```

```
if (responseSolicitudes.ok) {
```

```
    const solicitudes = await responseSolicitudes.json();
```

```
    const solicitudUsuario = solicitudes.find(s =>
```

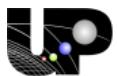
```
        s.correo && s.correo.toLowerCase() === correoUsuario.toLowerCase()
```

```
);
```

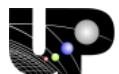
```
if (solicitudUsuario) {
```

```
    telefonoUsuario = solicitudUsuario.numero ||
```

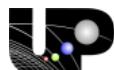
```
    solicitudUsuario.telefono ||
```



```
solicitudUsuario.telefonoContacto ||  
  
solicitudUsuario.celular ||  
  
solicitudUsuario.phone ||  
  
solicitudUsuario.numeroContacto;  
  
  
  
  
if (telefonoUsuario) {  
  
    sessionStorage.setItem('telefono', telefonoUsuario);  
  
}  
  
  
  
window.solicitudUsuario = solicitudUsuario;  
  
}  
  
}  
  
} catch (errorSolicitudes) {  
  
    console.warn('Error al obtener solicitudes:', errorSolicitudes.message);  
  
}  
  
}  
  
  
  
try {  
  
    response = await fetch('http://52.23.26.163:7070/solicitudes', {  
  
        method: 'GET',  
  
        headers: { 'Content-Type': 'application/json' }  
  
   });
```



```
if (response.ok) {  
  
    const restaurantes = await response.json();  
  
    let restauranteUsuario = null;  
  
  
    if (Array.isArray(restaurantes)) {  
  
        if (telefonoUsuario && telefonoUsuario.trim() !== "") {  
  
            const telefonoLimpio = telefonoUsuario.replace(/\s\-\(\)/g, " ");  
  
  
            restauranteUsuario = restaurantes.find(r => {  
  
                const telefonosAComparar = [  
  
                    r.numero,  
  
                    r.telefono,  
  
                    r.telefonoContacto,  
  
                    r.celular,  
  
                    r.phone,  
  
                    r.numeroContacto,  
  
                    r.numeroTelefono  
  
                ];  
  
  
                return telefonosAComparar.some(telefono => {  
  
                    if (!telefono) return false;  
  
                    const telefonoRestauranteLimpio =  
telefono.toString().replace(/\s\-\(\)/g, " ");  
                });  
            });  
        }  
    }  
}
```



```
return telefonoRestauranteLimpio === telefonoLimpio;

});

};

}

if (!restauranteUsuario && correoUsuario) {

    restauranteUsuario = restaurantes.find(r => {

        const correosAComparar = [

            r.correo,

            r.correoContacto,

            r.email,

            r.correoElectronico,

            r.emailContacto,

            r.contactoCorreo

        ];

        return correosAComparar.some(correo =>

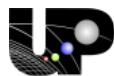
            correo && correo.toLowerCase() ===

correoUsuario.toLowerCase()

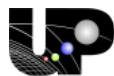
        );

    });

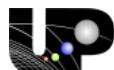
}
```



```
if (!restauranteUsuario && idUsuario) {  
  
    restauranteUsuario = restaurantes.find(r => {  
  
        const idsAComparar = [  
  
            r.idUsuario,  
  
            r.usuariold,  
  
            r.id,  
  
            r.restauranterold,  
  
            r.propietariold  
  
        ];  
  
  
        return idsAComparar.some(id =>  
  
            id && id.toString() === idUsuario.toString()  
  
        );  
  
    });  
  
}  
  
  
if (restauranteUsuario) {  
  
    window.restauranteActual = restauranteUsuario;  
  
  
  
  
    console.log('✅ Datos del restaurante encontrado:', restauranteUsuario);  
  
    console.log('🔍 ID del restaurante encontrado:', {  
  
        id: restauranteUsuario.id,  
  
    })  
}
```



```
idRestaurante: restauranteUsuario.idRestaurante,  
restaurantId: restauranteUsuario.restaurantId,  
id_restaurante: restauranteUsuario.id_restaurante  
});  
  
console.log('🔑 Foreign Keys del restaurante:', {  
    id_solicitud_aprobada: restauranteUsuario.id_solicitud_aprobada,  
    id_zona: restauranteUsuario.id_zona  
});  
  
if (window.solicitudUsuario) {  
    console.log('📄 Datos de la solicitud encontrada:',  
    window.solicitudUsuario);  
  
    console.log('🔍 ID de la solicitud:', {  
        id: window.solicitudUsuario.id,  
        id_solicitud: window.solicitudUsuario.id_solicitud  
    });  
}  
  
cargarDatosEnFormulario(restauranteUsuario, window.solicitudUsuario);  
  
if (restauranteUsuario.etiquetas) {  
    cargarEtiquetasRestaurante(restauranteUsuario.etiquetas);  
}
```



```
        return;

    } else {

        console.log('No se encontró restaurante específico para el usuario');

        console.log('Restaurantes disponibles:', restaurantes.map(r => ({
            id: r.id || r.idRestaurante || r.restaurantId,
            nombre: r.nombre,
            telefono: r.telefono || r.numero,
            correo: r.correo
        })));
        console.log('Datos del usuario buscado:', { correoUsuario, idUsuario,
telefonoUsuario });

        throw new Error('No se encontró ningún restaurante asociado al
usuario');

    }

} else {

    throw new Error(`Error al obtener restaurantes: ${response.status}`);

}

}

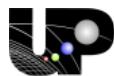
} catch (fetchError) {

    console.error('Error al obtener restaurantes:', fetchError);

    throw fetchError;

}
```

```
        } catch (error) {  
  
            console.error('Error al cargar datos del restaurante:', error);  
  
            if (error.message.includes('fetch')) {  
  
                alert('✖ Error de conexión: No se puede conectar al servidor.');//  
            } else if (error.message.includes('404')) {  
  
                alert('✖ Endpoint no encontrado: Verifique la configuración del servidor.');//  
            } else if (error.message.includes('500')) {  
  
                alert('✖ Error del servidor: Contacte al administrador.');//  
            } else {  
  
                alert('✖ Error al cargar los datos del restaurante: ${error.message}');//  
            }  
        }  
  
    }  
  
function cargarDatosEnFormulario(restaurante) {  
  
    const nombreElement = document.getElementById('restauranteNombre');  
  
    if (nombreElement) {  
  
        nombreElement.textContent = restaurante.restaurante || 'Mi Restaurante';  
    }  
}
```



```
const inputUbicacion = document.querySelector('input[placeholder="Ingrese la dirección"]');
```

```
if (inputUbicacion && restaurante.direccion) {  
    inputUbicacion.value = restaurante.direccion;  
}
```

```
const inputTelefono = document.querySelector('input[placeholder="Ingrese su número celular"]');
```

```
if (inputTelefono && restaurante.numero) {  
    inputTelefono.value = restaurante.numero;  
}
```

```
const horariosInputs = document.querySelectorAll('.horarios-inputs input');
```

```
if (restaurante.horario && horariosInputs.length > 0) {  
    horariosInputs[0].value = restaurante.horario;  
}
```

```
const inputInstagram = document.querySelector('input[placeholder="Ingrese su instagram"]');
```

```
if (inputInstagram && restaurante.instagram) {  
    inputInstagram.value = restaurante.instagram;  
}
```

```
const inputFacebook = document.querySelector('input[placeholder="Ingrese su facebook"]');
```

```
if (inputFacebook && restaurante.facebook) {  
  
    inputFacebook.value = restaurante.facebook;  
  
}  
  
  
  
const img1 = document.getElementById('imgGaleria1');  
  
if (img1) img1.src = restaurante.imagen1 || '../images/imagen.png';  
  
const img2 = document.getElementById('imgGaleria2');  
  
if (img2) img2.src = restaurante.imagen2 || '../images/imagen.png';  
  
const img3 = document.getElementById('imgGaleria3');  
  
if (img3) img3.src = restaurante.imagen3 || '../images/imagen.png';  
  
  
  
const btnMenu = document.querySelector('label.btn-menu input[type="file"]');  
  
if (btnMenu && restaurante.menu) {  
  
    const label = btnMenu.closest('label.btn-menu');  
  
    let fileNameSpan = label.querySelector('.file-name');  
  
  
  
    if (!fileNameSpan) {  
  
        fileNameSpan = document.createElement('span');  
  
        fileNameSpan.className = 'file-name';  
  
        fileNameSpan.style.marginLeft = '10px';  
  
        fileNameSpan.style.fontWeight = 'normal';  
  
        fileNameSpan.style.color = '#912F2F';  
  
        label.appendChild(fileNameSpan);  
    }  
}
```

}

```
const nombreArchivo = restaurante.menu.split('/').pop() || restaurante.menu;

fileNameSpan.textContent = `Menú guardado: ${nombreArchivo}`;

fileNameSpan.style.display = 'inline-block';

}

}

if (btnEstadisticas && mensaje) {

  if (estadoMembresia === "activa") {

    btnEstadisticas.style.display = "inline-block";

    btnEstadisticas.disabled = false;

    if (btnHeaderSubscripcion) btnHeaderSubscripcion.style.display = "none";

  } else if (estadoMembresia === "pendiente") {

    btnEstadisticas.style.display = "none";

    if (btnHeaderSubscripcion) btnHeaderSubscripcion.style.display = "inline-block";

  } else {

    btnEstadisticas.style.display = "none";

    mensaje.textContent = "";

    if (btnHeaderSubscripcion) btnHeaderSubscripcion.style.display = "inline-block";

  }

}
```

```
cargarDatosRestauranteUsuario();
```

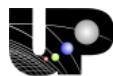
```
const ETIQUETAS_DISPONIBLES = [  
    'Comida Rápida',  
    'Pet Friendly',  
    'Familiar',  
    'Económico',  
    'Gourmet',  
    'Vegetariano',  
    'Delivery',  
    'Terraza',  
    'WiFi Gratuito',  
    'Estacionamiento'  
];
```

```
const selectsEtiquetas = document.querySelectorAll('.etiquetas .filtro-select');
```

```
selectsEtiquetas.forEach(select => {  
    while (select.children.length > 1) {  
        select.removeChild(select.lastChild);  
    }  
}
```

```
ETIQUETAS_DISPONIBLES.forEach(etiqueta => {  
    const option = document.createElement('option');  
    option.value = etiqueta;  
    option.textContent = etiqueta;  
    select.appendChild(option);  
});  
  
});  
  
window.obtenerEtiquetasSeleccionadas = function() {  
    const etiquetasSeleccionadas = [];  
    selectsEtiquetas.forEach(select => {  
        if (select.value && select.value !== "") {  
            etiquetasSeleccionadas.push(select.value);  
        }  
    });  
    return etiquetasSeleccionadas.join(', ');  
};  
  
window.cargarEtiquetasRestaurante = function(etiquetasString) {  
    if (!etiquetasString) return;  
  
    const etiquetasArray = etiquetasString.split(',').map(e => e.trim());  
};
```

```
etiquetasArray.forEach((etiqueta, index) => {  
  
    if (index < selectsEtiquetas.length) {  
  
        selectsEtiquetas[index].value = etiqueta;  
  
    }  
  
});  
  
};  
  
  
selectsEtiquetas.forEach((select, index) => {  
  
    select.addEventListener('change', function() {  
  
        const etiquetaSeleccionada = this.value;  
  
  
        if (etiquetaSeleccionada && etiquetaSeleccionada !== "") {  
  
            selectsEtiquetas.forEach((otroSelect, otroIndex) => {  
  
                if (otroIndex !== index) {  
  
                    const opcionARemover =  
otroSelect.querySelector(`option[value="${etiquetaSeleccionada}"]`);  
  
                    if (opcionARemover && otroSelect.value !== etiquetaSeleccionada) {  
  
                        opcionARemover.style.display = 'none';  
  
                    }  
  
                }  
  
            });  
  
        }  
  
    });  
  
});  
}
```



```
    actualizarOpcionesDisponibles();  
});  
});
```

```
function actualizarOpcionesDisponibles() {  
  
    const etiquetasYaSeleccionadas = [];  
  
    selectsEtiquetas.forEach(select => {  
  
        if (select.value && select.value !== "") {  
  
            etiquetasYaSeleccionadas.push(select.value);  
        }  
    });  
  
    selectsEtiquetas.forEach(select => {  
  
        const etiquetaActual = select.value;  
  
        Array.from(select.options).forEach(option => {  
  
            if (option.value === "" || option.value === etiquetaActual) {  
  
                option.style.display = "";  
            } else if (etiquetasYaSeleccionadas.includes(option.value)) {  
  
                option.style.display = 'none';  
            } else {  
  
                option.style.display = "";  
            }  
        });  
    });  
};
```

}

```
const btnAplicar = document.querySelector('.btn-aplicar');
```

```
if (btnAplicar) {
```

```
    btnAplicar.addEventListener('click', async function() {
```

```
        if (!window.restauranteActual) {
```

```
            alert('✖ No se han cargado los datos del restaurante');
```

```
            return;
```

```
}
```

```
const etiquetasSeleccionadas = obtenerEtiquetasSeleccionadas();
```

```
const datosActualizados = {
```

```
    nombre: window.restauranteActual.nombre,
```

```
    direccion: document.querySelector('input[placeholder="Ingrese la  
dirección"]')?.value || "",
```

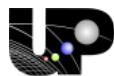
```
    telefono: document.querySelector('input[placeholder="Ingrese su número  
celular"]')?.value || "",
```

```
    horario: document.querySelector('.horarios-inputs input')?.value || "",
```

```
    etiquetas: etiquetasSeleccionadas,
```

```
};
```

```
try {
```



```
const idUsuario = sessionStorage.getItem('id') || localStorage.getItem('id');

const correoUsuario = sessionStorage.getItem('correo') ||
localStorage.getItem('correo');
```

```
if (!window.restauranteActual.id) {

    alert('🚫 No se puede actualizar: No se tiene el ID del restaurante');

    return;

}
```

```
const endpointUrl = `http://52.23.26.163:7070/restaurantes/${window.restauranteActual.id}`;
```

```
const response = await fetch(endpointUrl, {
```

```
    method: 'PUT',
```

```
    headers: {
```

```
        'Content-Type': 'application/json'
```

```
    },
```

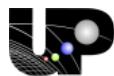
```
    body: JSON.stringify(datosActualizados)
```

```
});
```

```
if (response.ok) {
```

```
    const responseData = await response.json();
```

```
    console.log('✅ Respuesta del servidor:', responseData);
```



```
        alert('✅ Datos del restaurante actualizados correctamente');

        await cargarDatosRestauranteUsuario();

    } else {

        const errorText = await response.text();

        console.error('❌ Error del servidor:', response.status, errorText);

        alert('❌ Error al actualizar (${response.status}): ${errorText}');

    }

}

} catch (error) {

    console.error('❌ Error al guardar:', error);

}

if (error.name === 'TypeError' && error.message.includes('fetch')) {

    alert('❌ Error de conexión: Verifique que el backend esté ejecutándose en
http://52.23.26.163:7070');

} else if (error.message.includes('CORS')) {

    alert('❌ Error CORS: El backend no permite conexiones desde este
origen');

} else {

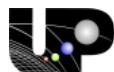
    alert('❌ Error de conexión al guardar los datos: ${error.message}');

}

}

});

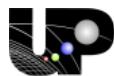
}
```



});

### vistaRestaurante

```
document.addEventListener('DOMContentLoaded', async function() {  
  
    const params = new URLSearchParams(window.location.search);  
  
    const id = params.get('id');  
  
    const correo = params.get('correo');  
  
    if (!id && !correo) {  
  
        console.warn('⚠️ No se proporcionó ID ni correo del restaurante - usando  
fallback');  
  
    }  
  
  
    async function verificarBackend(reintentos = 3) {  
  
        for (let i = 0; i < reintentos; i++) {  
  
            try {  
  
                const response = await fetch('http://52.23.26.163:7070/solicitudes', {  
  
                    method: 'GET',  
  
                    headers: { 'Content-Type': 'application/json' },  
  
                    timeout: 10000  
  
                });  
  
  
                if (response.status < 500) {  
  
                    return true;  
  
                }  
  
            } catch (error) {  
  
                if (i === reintentos - 1) {  
  
                    console.error(`Error en la solicitud ${i + 1}: ${error.message}`);  
  
                } else {  
  
                    console.log(`Error en la solicitud ${i + 1}: ${error.message}. Intentando de nuevo...`);  
  
                }  
  
            }  
  
        }  
  
    }  
  
    verificarBackend().then(() => {  
  
        // Lógica para manejar el resultado del backend  
  
    }).catch(error => {  
  
        console.error(`Error general: ${error.message}`);  
  
    });  
  
});
```



}

```
        console.warn(`Intento ${i + 1}: Servidor responde pero con error
${response.status}`);

        if (i < reintentos - 1) {

            await new Promise(resolve => setTimeout(resolve, 2000));

        }

    } catch (error) {

        console.warn(`Intento ${i + 1}: Error de conexión:`, error.message);

        if (i < reintentos - 1) {

            await new Promise(resolve => setTimeout(resolve, 2000));

        }

    }

}

return false;

}

try {

    console.log('🔍 Buscando restaurante con:', { id, correo });

}

const backendDisponible = await verificarBackend();

if (!backendDisponible) {

    throw new Error('Backend no disponible en http://52.23.26.163:7070');

}
```

}

```
const response = await fetch('http://52.23.26.163:7070/solicitudes', {  
  method: 'GET',  
  headers: { 'Content-Type': 'application/json' }  
});  
  
if (!response.ok) {  
  throw new Error(`HTTP ${response.status}: ${response.statusText}`);  
}  
  
const solicitudes = await response.json();  
console.log('📋 Solicitudes obtenidas:', solicitudes);  
console.log('📊 Total de solicitudes:', solicitudes.length);  
  
if (solicitudes.length > 0) {  
  console.log('⚡ Estructura de datos (primer elemento):',  
  Object.keys(solicitudes[0]));  
  console.log('>ID Datos disponibles:', solicitudes.map(s => ({  
    id: s.id,  
    restaurante: s.restaurante,  
    estado: s.estado,  
    correo: s.correo
```

```
        }));  
    }  
  
let restauranteEncontrado = null;
```

```
if (correo) {  
  
    restauranteEncontrado = solicitudes.find(s =>  
  
        s.correo && s.correo.toLowerCase() === correo.toLowerCase()  
    );  
  
    if (restauranteEncontrado) {  
  
        console.log('✓ Restaurante encontrado por correo:', correo);  
  
    }  
  
}
```

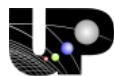
```
if (!restauranteEncontrado && id && id !== 'undefined' && id !== 'null') {  
  
    restauranteEncontrado = solicitudes.find(s =>  
  
        s.id == id ||  
  
        s.idRestaurante == id ||  
  
        s.restaurantId == id ||  
  
        s.id_restaurante == id  
    );  
  
    if (restauranteEncontrado) {  
  
        console.log('✓ Restaurante encontrado por ID:', id);  
  
    }
```

```
        console.log('✓ Restaurante encontrado por ID:', id);  
  
    }
```

```
}
```

```
}
```

```
if (!restauranteEncontrado && id && !isNaN(parseInt(id))) {  
  
    const index = parseInt(id);  
  
    if (index >= 0 && index < solicitudes.length) {  
  
        restauranteEncontrado = solicitudes[index];  
  
        console.log('✓ Restaurante encontrado por índice:', index);  
  
    }  
  
}  
  
  
  
if (!restauranteEncontrado) {  
  
    const solicitudesAprobadas = solicitudes.filter(s =>  
  
        s.estado === 'aprobado' ||  
  
        s.estado === 'Aprobado' ||  
  
        s.estado === 'APROBADO'  
  
    );  
  
  
  
    if (solicitudesAprobadas.length > 0) {  
  
        restauranteEncontrado = solicitudesAprobadas[0];  
  
        console.log('✓ Usando primer restaurante aprobado disponible');  
  
    }  
  
}
```



```
if (!restauranteEncontrado && solicitudes.length > 0) {  
  
    restauranteEncontrado = solicitudes[0];  
  
    console.log('✅ Usando primer restaurante disponible como fallback');  
  
}  
  
  
if (!restauranteEncontrado) {  
  
    console.error('❌ No se encontró ningún restaurante');  
  
    console.error('🔍 Parámetros buscados:', { id, correo });  
  
    console.error('📋 Solicitudes disponibles:', solicitudes.map(s => ({ id: s.id, correo: s.correo })));  
  
    throw new Error(`No hay restaurantes disponibles para mostrar`);  
  
}  
  
  
console.log('✅ Restaurante encontrado:', restauranteEncontrado);  
  
  
  
window.restauranteActual = restauranteEncontrado;  
  
  
  
cargarDatosRestaurante(restauranteEncontrado);  
  
  
  
} catch (error) {  
  
    console.error('❌ Error al cargar restaurante:', error);  
}
```

```

if (error.message.includes('fetch')) {

    alert('✖ Error de conexión: No se puede conectar al servidor.);

} else if (error.message.includes('404')) {

    alert('✖ Endpoint no encontrado: Verifique la configuración del servidor.);

} else if (error.message.includes('500')) {

    alert('✖ Error del servidor: Contacte al administrador.);

} else {

    alert('✖ Error al cargar la información del restaurante: ${error.message}`);

}

}

function cargarDatosRestaurante(restaurante) {

try {

    console.log('🎨 Cargando datos en la vista:', restaurante);

    const nombreElement = document.querySelector('.restaurante-nombre');

    if (nombreElement) {

        nombreElement.textContent = restaurante.restaurante || 'Restaurante sin
nombre';

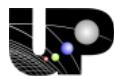
    }

}

const imgPrincipal = document.querySelector('.galeria-principal');

if (imgPrincipal) {

```



```
const imagenPrincipal = restaurante.imagen1 || './images/img_rest2.jpg';

imgPrincipal.src = imagenPrincipal;

imgPrincipal.alt = `Imagen principal de ${restaurante.restaurante}`;

console.log('🖼 Imagen principal cargada:', imagenPrincipal);

}

const galeriaSecundaria = document.querySelector('.galeria-secundaria');

if (galeriaSecundaria) {

    const imagenesSecundarias = [];

    if (restaurante.imagen2) imagenesSecundarias.push(restaurante.imagen2);

    if (restaurante.imagen3) imagenesSecundarias.push(restaurante.imagen3);

    galeriaSecundaria.innerHTML = imagenesSecundarias.length > 0

        ? imagenesSecundarias.map(img => `<img src='${img}' alt='Imagen
restaurante' class='galeria-thumb' />`).join("")

        : '<span style="color:#888">No hay imágenes adicionales</span>';

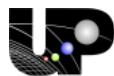
    console.log('🖼 Imágenes secundarias cargadas:',

    imagenesSecundarias.length);

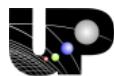
}

const caracteristicas = document.querySelector('.caracteristicas-lista');

if (caracteristicas) {
```



```
let etiquetasArray = [];  
  
const etiquetasIndividuales = [  
    restaurante.etiqueta1,  
    restaurante.etiqueta2,  
    restaurante.etiqueta3  
].filter(etiqueta =>  
    etiqueta &&  
    etiqueta !== "" &&  
    etiqueta !== 'Seleccionar' &&  
    etiqueta.trim() !== "  
);  
  
  
if (etiquetasIndividuales.length === 0 && restaurante.etiquetas) {  
    if (typeof restaurante.etiquetas === 'string') {  
        etiquetasArray = restaurante.etiquetas.split(',').map(e => e.trim()).filter(e  
=> e && e !== 'Seleccionar');  
    } else if (Array.isArray(restaurante.etiquetas)) {  
        etiquetasArray = restaurante.etiquetas.filter(e => e && e !==  
'Seleccionar');  
    }  
}  
etiquetasArray = etiquetasIndividuales;
```



}

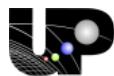
```
características.innerHTML = etiquetasArray.length > 0  
  
? etiquetasArray.map(etiqueta =>  
  
`<div class='caracteristica-item'>  
  
    <img src='./images/etiqueta.png' alt='Etiqueta' class='icon-etiqueta' />  
  
    <span class='caracteristica'>${etiqueta}</span>  
  
</div>`  
  
).join("")  
  
: '<span style="color:#888">No hay características disponibles</span>;
```

```
console.log('📌 Etiquetas cargadas:', etiquetasArray);  
  
console.log('🔍 Etiquetas individuales encontradas:', {  
  
    etiqueta1: restaurante.etiqueta1,  
  
    etiqueta2: restaurante.etiqueta2,  
  
    etiqueta3: restaurante.etiqueta3  
  
});  
  
}
```

```
const horarios = document.querySelector('.restaurante-horarios');  
  
if (horarios) {  
  
    horarios.innerHTML = `  
  


## Horarios


```



```
<div><img src='..../images/reloj.png' alt='Horario' class='icon-16' />
${restaurante.horario} || 'No especificado'</div>

';

console.log('⌚ Horarios cargados:', restaurante.horario);

}

const contactos = document.querySelector('.restaurante-contactos');

if (contactos) {

    const telefono = restaurante.numero || restaurante.telefono || 'No
especificado';

    const facebook = restaurante.facebook || 'No especificado';

    const instagram = restaurante.instagram || 'No especificado';

    contactos.innerHTML = `

        <h2>Contactos</h2>

        <div><img src='..../images/llamada.png' alt='Teléfono' class='icon-16' />
${telefono}</div>

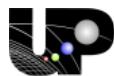
        <div><img src='..../images/facebook.png' alt='Facebook' class='icon-16' />
${facebook}</div>

        <div><img src='..../images/instagram.png' alt='Instagram' class='icon-16' />
${instagram}</div>

`;

    console.log('📞 Contactos cargados:', { telefono, facebook, instagram });

}
```



```
const ubicacion = document.querySelector('.restaurante-ubicacion');

if (ubicacion) {

    const direccion = restaurante.direccion || 'No especificada';

    ubicacion.innerHTML = `

        <h2>Ubicación</h2>

        <div><img src='..../images/ubicacion.png' alt='Ubicación' class='icon-16' />
${direccion}</div>

`;

    console.log('📍 Ubicación cargada:', direccion);
}

if (restaurante.descripcion) {

    console.log('📝 Descripción disponible:', restaurante.descripcion);
}

if (restaurante.menu) {

    console.log('📋 Menú disponible:', restaurante.menu);
}

console.log('✅ Todos los datos cargados correctamente');

} catch (error) {

    console.error('❌ Error al cargar datos en la vista:', error);
}
```

}

}

});

## 6.- Vistas Fronted

### 6.1 Usuario



Restaurante San Cristóbal

Regístrate Iniciar sesión

Descubre la Magia Culinaria de San Cristóbal

Explora los mejores sabores de esta tierra

Buscar restaurantes Buscar

Precio Horario Tipo de comida Ubicación Filtrar

**La Mansión**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00

**Tarumba**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00

**Taniperla**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00

**El secreto**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00

Contáctanos

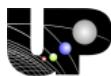
 sazonpatrimonial@gmail.com  +52 961 652 2093

 @sazonpatrimonial  @sazonpatrimonial

 SAZÓN PATRIMONIAL

 DEVBOX

**Universidad Politécnica de Chiapas** 



## La mansión



Libre de gluten

Comida local

Pet-friendly

### Horarios

#### MORARIOS

Lunes a viernes  
8:00-20:00  
Sábado y domingo  
10:00-20:00

### Contactos

#### CONTACTOS

+52 123 467 2543  
@LaMansion  
@la\_mansion

### Ubicación

Av. 4 de junio N°20,  
San Cristóbal de las  
Casas, Chiapas,  
México.

Menú

### Contáctanos

sazonpatrimonial@gmail.com

+52 961 652 2093

@sazonpatrimonial

@sazonpatrimonial



**Encuesta**

¿Qué te atrajo del restaurante?

Selecciona una opción

¿De dónde nos visitas?

Selecciona una opción

**Enviar**

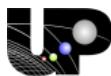


Libre de gluten

Horarios

Contactos

Ubicación



Restaurante  
San Cristóbal

Registrarse Iniciar sesión

Descubrir

Ingresar como

Administrador

Restaurantero

Registrarse

Buscar restaurante

Precio Horario Tipo de comida Ubicación Filtrar

## 6.2 Restaurantero

Restaurante  
San Cristóbal

Mi restaurante Cerrar sesión

Descubre la Magia Culinaria de San Cristóbal

Explora los mejores sabores de esta tierra

Buscar restaurantes Buscar

Precio Horario Tipo de comida Ubicación Filtrar



**La Mansión**

Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**Tarumba**

Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**Taniperla**

Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**El secreto**

Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00

### Contáctanos

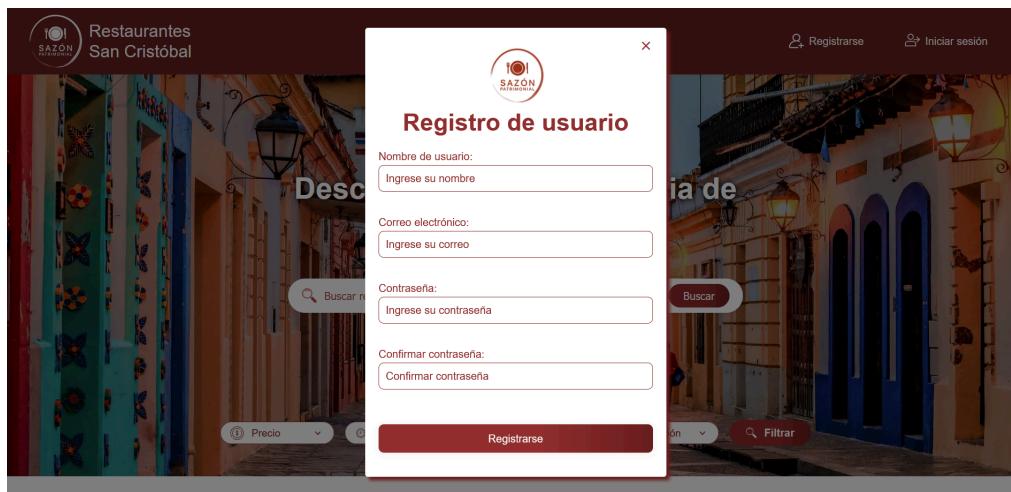
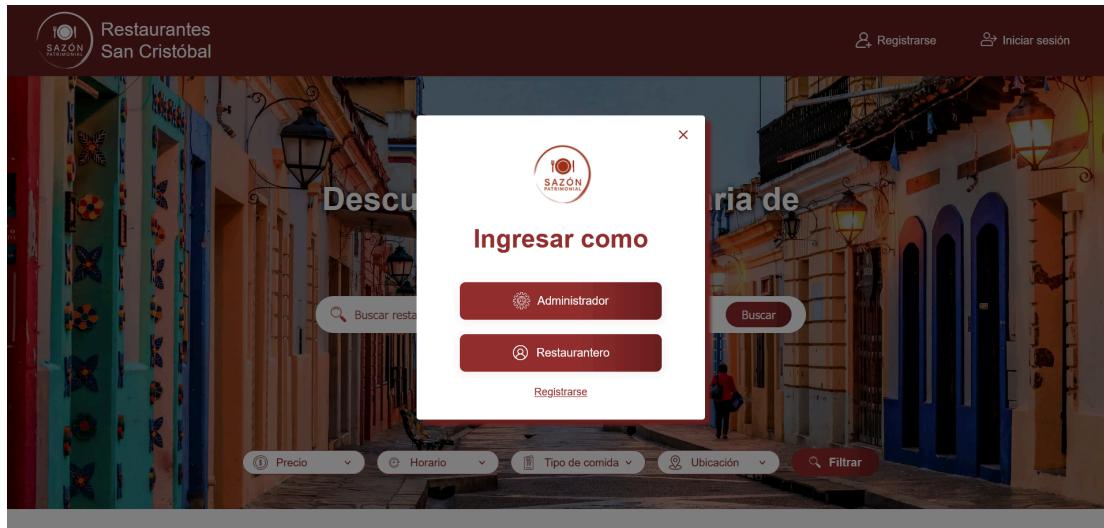
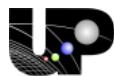
sazonpatrimonial@gmail.com

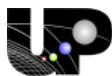
+52 961 652 2093

@sazonpatrimonial

@sazonpatrimonial







## Solicitud de validación

Proporciona la siguiente información para verificar tu restaurante

Restaurante

Ingrese el nombre del restaurante

Propietario

Ingrese el nombre del propietario

Correo electrónico

Ingrese su correo

Número de celular

Ingrese su número de celular

Facebook

Ingrese su Facebook

Instagram

Ingrese su Instagram

Dirección

Ingrese la dirección del restaurante

Horarios de cierre y apertura

Ingrese el horario

Imágenes (JPG o PNG)

Imagen principal

Imagen secundaria

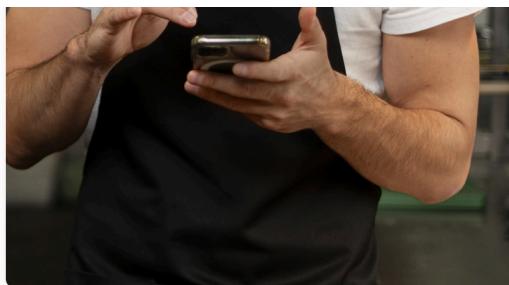
Imagen de platillo

Archivos (PDF)

Comprobante de domicilio

Menú del restaurante

Enviar solicitud



### Contáctanos

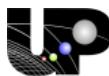
sazonpatrimonial@gmail.com

+52 961 652 2093

@sazonpatrimonial

@sazonpatrimonial





## La mansión



Libre de gluten

Comida local

Pet-friendly

### Horarios

#### MORARIOS

Lunes a viernes  
8:00-20:00  
Sábado y domingo  
10:00-20:00

### Contactos

#### CONTACTOS

+52 123 467 2543  
@LaMansion  
@la\_mansion

### Ubicación

Av. 4 de junio N°20,  
San Cristóbal de las  
Casas, Chiapas,  
México.

Menú

### Contáctanos

sazonpatrimonial@gmail.com

+52 961 652 2093

@sazonpatrimonial

@sazonpatrimonial



**Encuesta**

¿Qué te atrajo del restaurante?

Selecciona una opción

¿De dónde nos visitas?

Selecciona una opción

**Enviar**

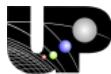


Libre de gluten

Horarios

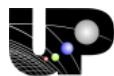
Contactos

Ubicación



The screenshot shows a login modal window titled "Iniciar sesión" (Log in) centered over a blurred background image of a colorful street in San Cristóbal. The modal contains fields for "Correo electrónico:" (Email address) and "Contraseña:" (Password), both with placeholder text "Ingrese su correo" and "Ingrese su contraseña". A "Buscar" button is also present. Below the modal, a navigation bar includes filters for "Tipo de comida", "Ambiente", "Servicios", and a "Limpiar Filtros" button.

The screenshot shows the user's profile icon and a "Cerrar sesión" (Logout) link in the top right corner. The main content area displays a "Bienvenido" (Welcome) message above three empty square placeholders labeled "Galería de Imágenes". Below this is a section for "Menú en PDF:" with a file selection input field labeled "Seleccionar archivo PDF".



<b>Ubicación:</b> <input type="text" value="Ingrese la dirección"/>	<b>Horarios:</b> <input type="text" value="Ingrese el horario"/>
<b>Contactos:</b>	
<input type="text" value="Ingrese su número celular"/>	<input type="text"/>
<input type="text" value="Ingrese su facebook"/>	<input type="text"/>
<input type="text" value="Ingrese su instagram"/>	<input type="text"/>

Aplicar Cambios

Restaurantes  
San Cristóbal Cerrar sesión

**Estadísticas de su restaurante**

---

**DESCARGAS DE MENÚ**

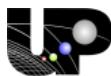
Total acumulados  
3  
**+15%**

**INTERÉS POR ORIGEN**

- Locales 67% (2)
- Extranjeros 33% (1)

**ASPECTOS DESTACADOS**

Aspecto	Porcentaje
Comida	67%
Ubicación	20%
Recomendación	15%
Horario	10%
Vista	10%



## 6.3 Administrador

Restaurante  
San Cristóbal

Registrarse Iniciar sesión

## Descubre la Magia Culinaria de San Cristóbal

Explora los mejores sabores de esta tierra

Buscar restaurantes Buscar

Precio Horario Tipo de comida Ubicación Filtrar



**La Mansión**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**Tarumba**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**Taniperla**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00



**El secreto**  
Cocina regional mexicana  
Lunes a viernes 8:00 a 20:00

Contáctanos

**M** sazonpatrimonial@gmail.com **✉** +52 961 652 2093  
**Instagram** @sazonpatrimonial **Facebook** @sazonpatrimonial

**SAZÓN PATRIMONIAL** **DEVBOX** **Universidad Politécnica de Chiapas** **UP**

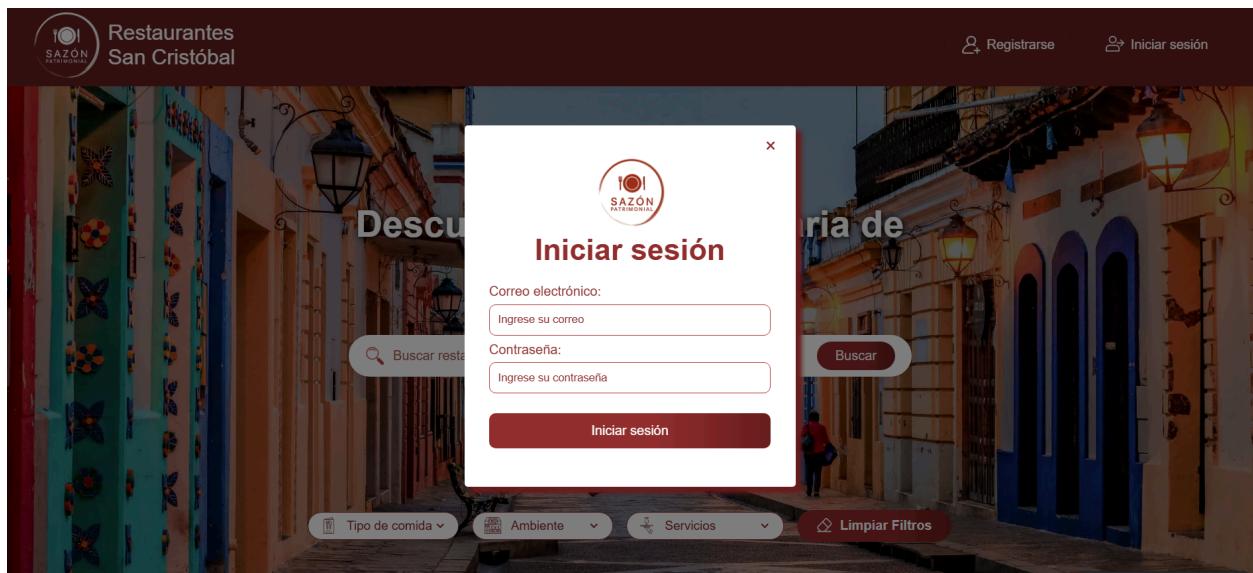
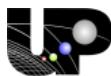
Restaurante  
San Cristóbal

Registrarse Iniciar sesión

## Ingresar como

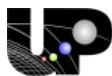
Administrador Restaurantero Registrarse

Precio Horario Tipo de comida Ubicación Filtrar



The screenshot shows a table titled "Administrar Restaurantes" (Manage Restaurants) with the following data:

Nombre	Ubicación	Etiquetas	Acciones
Nombre 1	Ubicación 1	Etiqueta 1	



Restaurante Sazón Patrimonial

Restaurante San Cristóbal

Cerrar sesión

Restaurante Etiquetas Usuarios

## Administrar Etiquetas

Agregar etiquetas

Nombre	Acciones
Etiqueta 1	

Restaurante Sazón Patrimonial

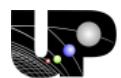
Restaurante San Cristóbal

Cerrar sesión

Restaurante Etiquetas Usuarios

## Administrar Solicitudes de Usuarios

Restaurante	Propietario	Correo	Celular	Dirección	Horario	Imágenes	Comprobante	Acciones
Restaurante 1	Propietario 1	correo1@ejemplo.com	1234567890	Dirección 1	Horario 1			



Universidad Politécnica  
de Chiapas

