

UNIVERSIDAD ANTONIO NARIÑO

**NOMBRE DEL PROYECTO: APLICATIVO DE GESTIÓN DE RESTAURANTES
(AGR)**

MATERIA: INGENIERÍA DE SOFTWARE

**ESTUDIANTES: PARRA YEISON, MURILLO ISABELLA, VARGAS JUAN, MELO
SHAYRA, LOZANO DAVID**

DOCENTE: JULIAN PRADO

02/OCTUBRE/2025

ÍNDICE

Sistema.....	3
Variables.....	3
Bocetos de diagramas.....	4
Interfaces.....	7
Funcionalidades de los actores.....	7
Modificación de la base de datos.....	12
Avances API backend.....	12
Creación de rutas fast api.....	13
Conexión con la base de datos y modelos según tablas de la base de datos.....	13
Inicio de desarrollo de Frontend en paralelo a Backend.....	15
Actividades para Octubre.....	17
Seguridad y Autenticación.....	17
Interfaz Frontend Inicial.....	17
Módulo de Pedidos (Versión Inicial).....	17
Seguridad Adicional.....	17
Pruebas Iniciales.....	17

Avances Segundo Corte:

Sistema

Variables

Cómo van a cambiar las variables

Plantillas:

Cuenta #n:	Cancelada
#mesa, → INT	→ X
Plato 1, → String	→ X
Plato 2, →	→ X
Plato 3, →	→ X
Plato n, →	→ X
Tiempo de ingreso, →	→ X
activa ++	→ activa --

Reserva #n:	Reservada	cancelada.
Nombre, Asistentes, día y horario, →	✓ activa ++	X
#mesa, →	✓	X activa --

Cuenta cobro #n:	Cuenta fracción #n:
#mesa, →	→
Plato 1, →	→
Plato 2, →	→
Plato n, →	→
Descuento, →	→
Total de la cuenta, →	→

Fraccionar Cuenta:

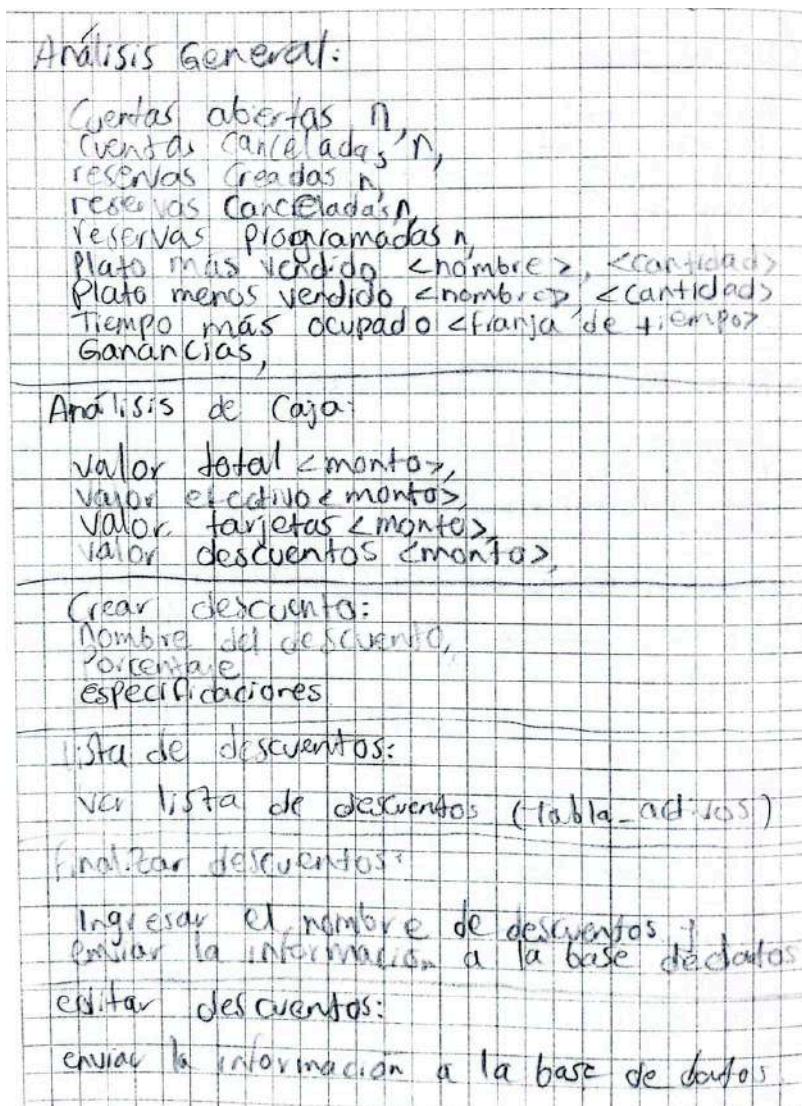
Plato <nombre>

#mesa

Especificaciones: ~~~~~

estado

Algunas funciones y las variables requeridas



Bocetos de diagramas

Diagrama de contexto 1

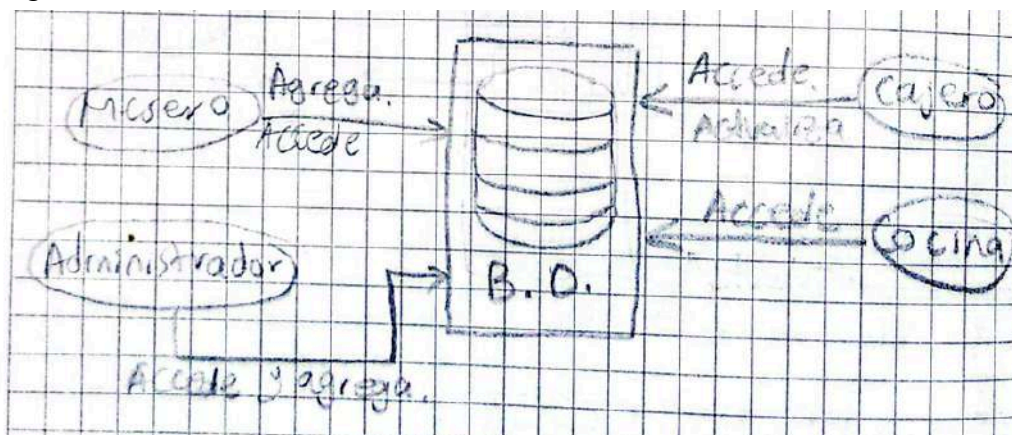
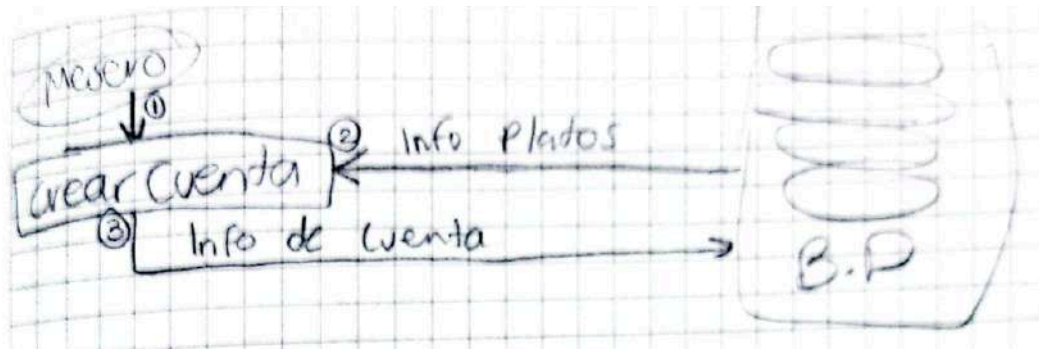
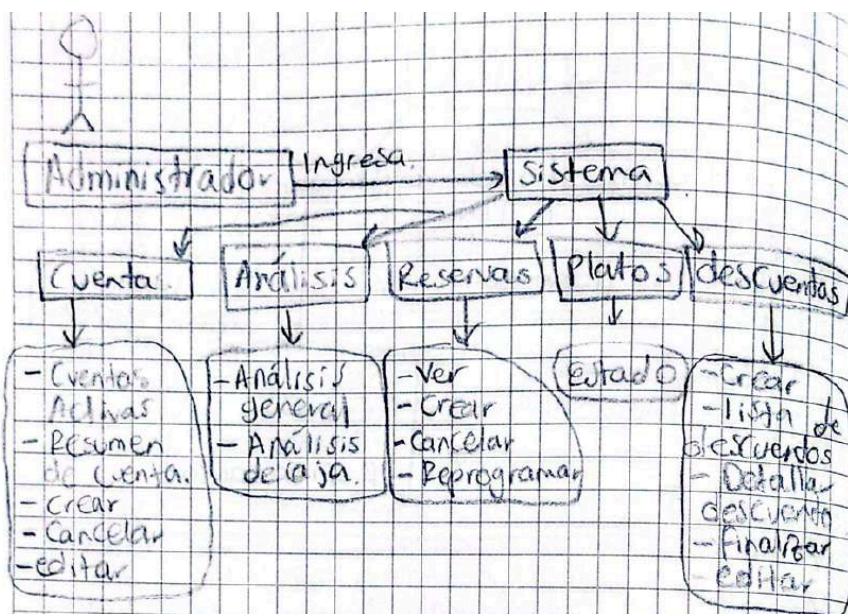
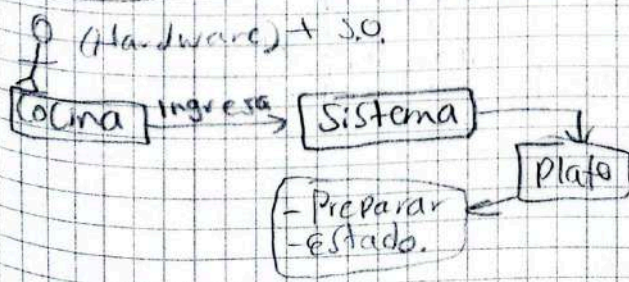
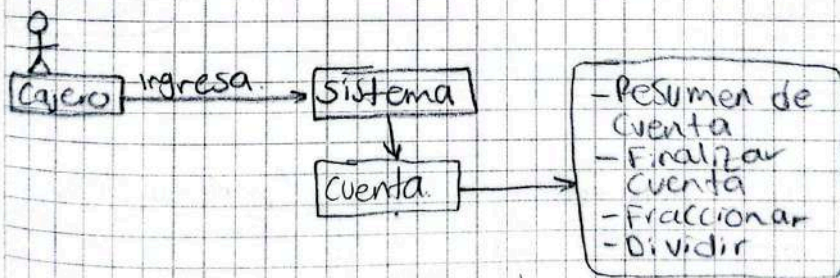
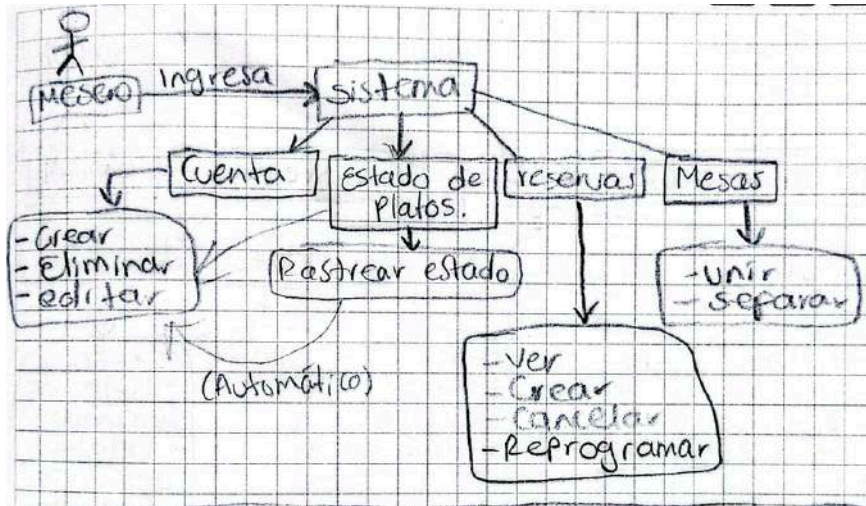


Diagrama de contexto 2



Diagramas de casos de uso



Interfaces

Para el sistema se definieron las siguientes interfaces de acuerdo a los bocetos de diagramas anteriormente mencionados, con el fin de garantizar la facilidad e intuitividad de la interacción del usuario con el sistema final, para esto se empleó la herramienta gráfica Canva, con el objetivo de realizar un modelo visual de cómo se deberían ver.

- Login



- Meseros



MESEROS

PEDIDO

INGRESE EL NÚMERO DE MESA:

Numero de la mesa

CREAR PEDIDO

EDITAR PEDIDO

ELIMINAR PEDIDO

CONFIRMAR

ATRÁS

MESEROS

¡PRECAUCIÓN!

¿ESTÁS SEGURO DE ELIMINAR
ESTE PEDIDO?

CONFIRMAR

CANCELAR

MESEROS

PEDIDO

Empanaditas Mixtas X1 X

Pasta Alfredo con Camarones X1 X

Jugo de Mango X1 X

CERRAR SESIÓN

ENTRADAS

FUERTES

BEBIDAS

ENSALADAS

POSTRES

ADICIONES

DA CLICK A LO QUE DESEAS AGREGAR AL CARRITO

AGR

CONFIRMAR

MESEROS

PEDIDO

ACA VA LO QUE SE AGREGUE AL CARRITO

ENTRADAS

Bruschettas clásicas (pan artesanal, tomate, albahaca y aceite de oliva)

Croquetas de jamón serrano con salsa alioli

Ceviche de camarón al estilo peruano

Sopa crema de champiñones con crotones

Empanaditas mixtas (carne, pollo, queso)

AGR

DA CLICK A LO QUE DESEAS AGREGAR AL CARRITO

ATRÁS

CONFIRMAR

MESEROS

PEDIDO 

ACA VA LO QUE SE
AGREGUE AL
CARRITO

FUERTES

Lomo fino en salsa de
champiñones con papas
rústicas

Pollo a la plancha con
vegetales salteados

Pescado a la parrilla con
arroz de coco

Pasta Alfredo con
camarones

Risotto de setas y
parmesano

Hamburguesa artesanal
con papas en casco

Costillas BBQ con
ensalada de la casa

ATRÁS

DA CLICK A LO QUE DESEAS AGREGAR AL CARRITO

MESEROS

PEDIDO 

ACA VA LO QUE SE
AGREGUE AL
CARRITO

BEBIDAS

Jugo de mango en
agua

Jugo de mango en
lecho

Jugo de fresa en agua

Jugo de fresa en
lecho

Refrescos en lata

Té frío

Té caliente

Cerveza

Limonada de coco

Copa de vino (tinto,
blanco, rosé)

ATRÁS

DA CLICK A LO QUE DESEAS AGREGAR AL CARRITO

CONFIRMAR

MESEROS

PEDIDO 

ACA VA LO QUE SE
AGREGUE AL
CARRITO

ENSALADAS

Ensalada César con
pollo grillado

Ensalada
mediterránea

Ensalada fresca
tropical

Ensalada caprese

ATRÁS

DA CLICK A LO QUE DESEAS AGREGAR AL CARRITO

CONFIRMAR

MESEROS

PEDIDO 

ACA VA LO QUE SE
AGREGUE AL
CARRITO

POSTRES

Tarta de queso con
frutos rojos

Brownie con helado de
vainilla

Mousse de maracuyá

Tiramisu

Helado artesanal de
vainilla

Helado artesanal de
chocolate

ATRÁS

DA CLICK A LO QUE DESEAS AGREGAR AL CARRITO

CONFIRMAR

MESEROS

PEDIDO 

ACA VA LO QUE SE
AGREGUE AL
CARRITO

ADICIONES

Papas fritas

Papas on casco

Arroz Blanco

Arroz de Coco

Puré de papa

Vegetales salteados

Pan de ajo

Ensalada fresca
pequeña

ATRÁS

DA CLICK A LO QUE DESEAS AGREGAR AL CARRITO

CONFIRMAR



MESEROS

ESTADO DE PLATOS

INGRESE EL NÚMERO DE MESA:

Numero de la mesa

CONFIRMAR

PEDIDO:

Empanaditas Mixtas X1

Pasta Alfredo con Camarones X1

Jugo de Mango X1

PREPARANDOSE

ATRÁS

MESEROS

PRÓXIMAS RESERVAS

- Lunes 29 de Septiembre de 2025
 - Nombre: Lucía Hernández
 - Hora: 8:00pm
 - Mesa 2
- Lunes 29 de Septiembre de 2025
 - Nombre: Carlos Aponte
 - Hora: 8:00pm
 - Mesa 4
- Lunes 29 de Septiembre de 2025
 - Nombre: Alfredo Leyton
 - Hora: 9:00pm
 - Mesa 1

ATRÁS

- Cajeros

CAJEROS

PAGAR CUENTA

VALIDA EL NÚMERO DE MESA

Numero de la mesa

CONFIRMAR

CAJEROS

DE MESA

TOTAL A PAGAR
\$61.900
MONTO
\$70.000
SALDO
\$8.100

CERRAR SESIÓN

DETALLES DEL PEDIDO

Empanaditas Mixtas X1
Pasta Alfredo con Camarones X1
Jugo de Mango X1

MÉTODO DE PAGO:

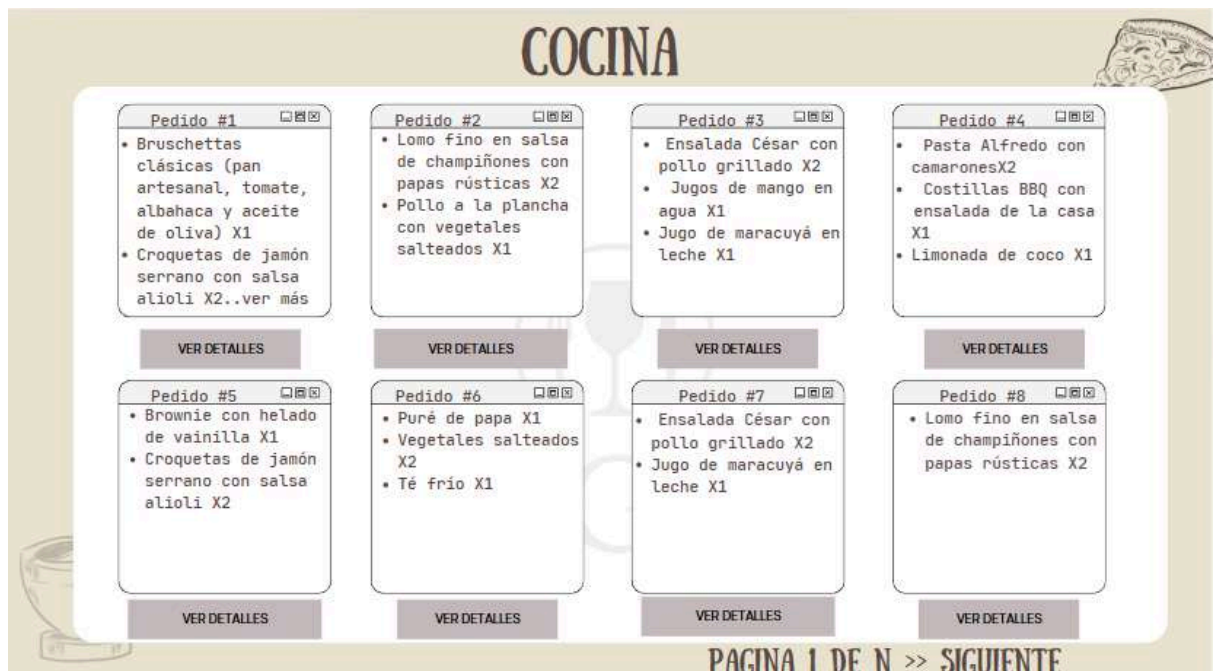
EFFECTIVO

TARJETA

CÓDIGO QR

CONFIRMAR PAGO

- Cocina



- Administrador



<https://www.canva.com/design/DAGygCQkL9U/goEMM0evQgPZ84bdOXgwyg/edit>

Funcionalidades de los actores

Al ingresar al sistema debe ingresar su código y/o su nombre (el cual debe estar verificado bajo su cargo). Al ser verificado en la base de datos, el programa da paso las funcionalidades que posee su cargo. Al finalizar su turno sale del sistema y la sesión se cierra.

A continuación explicamos el proceso que se lleva a cabo en cada funcionalidad.

Mesero

Al ingresar al sistema se despliega un menú donde tenemos los siguientes apartados (opciones):

Apartado de Cuentas:

- Crear Cuenta
 1. Registrar número de mesa,
 2. Registrar platos,
 3. Registrar Bebidas,
 4. Realizar especificaciones,
 5. Finalizar.
- Eliminar cuenta
 1. Ingresar el número de mesa,
 2. Hacer validación,
 3. Eliminar cuenta de la base de datos,
 4. Eliminar lo que pidieron de la base de datos,
 5. Finalizar
- Editar cuenta
 1. Ingresar el número de mesa,
 2. OPCIÓN MODIFICAR PLATO,
 3. Modificar en la base de datos,
 4. OPCIÓN MODIFICAR NÚMERO DE MESA.
 5. Modificar en la base de datos,
 6. Finalizar.

Apartado de Estado de platos:

- Rastrear estado de plato
 1. Ingresar mesa,
 2. Seleccionar el plato,
 3. Recibir estado.
 4. Finalizar.

Apartado de Reservas:

- Ver reservas
 1. Ver listado de reservas,
 2. Seleccionar una reserva,
 3. Mostrar información de reserva,
 4. Finalizar.
- Crear reserva
 1. Registrar usuario,

2. Registrar cantidad de asistentes,
 3. Registrar día y horario,
 4. Registrar mesa,
 5. Inutilizar el número de mesa en el sistema (Por lo menos unas 3 horas antes).
 6. Modificar en la base de datos,
 7. Finalizar.
- Cancelar reserva
 1. Ingresar nombre de usuario,
 2. Eliminar reserva,
 3. Borrar de la base de datos,
 4. Finalizar.
 - Reprogramar reserva
 1. Ingresar nombre del usuario,
 2. Buscar en la base de datos,
 3. OPCIÓN CAMBIAR HORARIO,
 4. OPCIÓN CAMBIAR DÍA,,
 5. Modificar la base de datos.
 6. Finalizar.

Apartado de Mesas:

- Unir mesas
 1. Ingresar una mesa inicial,
 2. Ingresar una mesa secundaria,
 3. Inhabilitar la mesa secundaria en el sistema,
 4. Modificar el número de mesa en la base de datos de la cuenta,
 5. Finalizar.
- Separar mesas
 1. Ingresar número de mesa,
 2. Modificar en la base de datos de mesas disponibles,
 3. Habilitar la mesa en el sistema,
 4. Modificar en la base de datos de la cuenta,
 5. Finalizar.

Cajero

Al ingresar al sistema se despliega un menú donde tenemos los siguientes apartados (opciones):

Apartado de cuentas:

- Resumen de cuenta
 1. Ingresar número de mesa,
 2. Ver listado de los pedidos que han hecho,
 3. Ver cuánto tiempo llevan en el restaurante,
 4. Finalizar.
- Finalizar cuenta
 1. Ingresar el número de mesa,

2. Ver el resumen,
 3. Calcular total,
 4. Aplicar descuentos,
 5. Recibir el pago,
 6. Elimina la cuenta de la base de datos,
 7. Habilita la mesa en el sistema,
 8. Finalizar.
- Fraccionar cuenta
 1. Ingresar el número de mesa,
 2. OPCIÓN DIVIDIR,
 3. Calcula el total y lo divide entre la cantidad seleccionada,
 4. OPCIÓN FRACCIONAR,
 5. Ingresar platos en una tabla de apoyo,
 6. Calcular el total,
 7. Recibir el pago,
 8. Finalizar.

Administrador

Al ingresar al sistema se despliega un menú donde tenemos los siguientes apartados (opciones):

Apartado de cuentas:

- Cuentas activas
 1. Ver listado de cuentas,
 2. Seleccionar cuenta,
 3. Ver listado de los pedidos que han hecho,
 4. Estado de los pedidos,
 5. Ver cuánto tiempo llevan en el restaurante,
 6. Finalizar.
- Resumen de cuenta
 1. Ingresar número de mesa,
 2. Ver listado de los pedidos que han hecho,
 3. Ver cuánto tiempo llevan en el restaurante,
 4. Finalizar.
- Crear cuenta
 1. Registrar número de mesa,
 2. Registrar platos,
 3. Registrar Bebidas,
 4. Realizar especificaciones,
 5. Finalizar.
- Cancelar cuenta
 1. Ingresar el número de mesa,
 2. Hacer validación,
 3. Eliminar cuenta de la base de datos,

4. Eliminar lo que pidieron de la base de datos,
5. Finalizar
- Editar cuenta
 1. Ingresar el número de mesa,
 2. OPCIÓN MODIFICAR PLATO,
 3. Modificar en la base de datos,
 4. OPCIÓN MODIFICAR NÚMERO DE MESA.
 5. Modificar en la base de datos,
 6. Finalizar.

Apartado de análisis:

- Análisis general
 1. Ver número de cuentas totales abiertas,
 2. Ver número de cuentas totales cerradas,
 3. Ver cantidad de reservas creadas,
 4. Ver cantidad de reservas canceladas,
 5. Ver cantidad de reservas programadas,
 6. Ver plato más pedido,
 7. Ver plato menos pedido,
 8. Ver horario más transcurrido,
 9. Ver flujo de caja,
- Análisis de caja
 1. Ver total vendido,
 2. Ver dinero en efectivo,
 3. Ver dinero en tarjetas,
 4. Ver dinero que se fue en descuentos,
 5. Finalizar.

Apartado de Gestión de reservas:

- Ver reservas
 1. Ver listado de reservas (Base de Datos),
 2. Seleccionar una reserva,
 3. Mostrar información de reserva,
 4. Finalizar.
- Crear reserva
 1. Registrar usuario,
 2. Registrar cantidad de asistentes,
 3. Registrar día y horario,
 4. Registrar mesa,
 5. Inutilizar el número de mesa en el sistema (Por lo menos unas 3 horas antes).
 6. Modificar en la base de datos,
 7. Finalizar.
- Cancelar reserva
 1. Ingresar nombre de usuario,
 2. Eliminar reserva,
 3. Borrar de la base de datos,

4. Finalizar.
- Reprogramar reserva
 1. Ingresar nombre del usuario,
 2. Buscar en la base de datos,
 3. OPCIÓN CAMBIAR HORARIO,
 4. OPCIÓN CAMBIAR DÍA,
 5. Modificar la base de datos,
 6. Finalizar.

Apartado de Estado de platos:

- Rastrear estado de plato
 1. Ingresar mesa,
 2. Seleccionar el plato,
 3. Recibir estado.
 4. Finalizar.

Apartado de descuentos:

- Crear descuento
 1. Registrar nombre,
 2. Se guarda en una tabla,
 3. Registrar Porcentaje,
 4. Se guarda en una tabla,
 5. Agregar especificaciones,
 6. Se guarda en una tabla de descuentos activos,
 7. Finalizar.
- Lista de descuentos
 1. Ver la lista de descuentos,
 2. finalizar.
- Detallar descuento
 1. Ingresar el nombre,
 2. Ver porcentaje de descuento,
 3. Ver especificaciones,
 4. finalizar.
- Finalizar descuento
 1. Ingresar nombre,
 2. Confirmar finalización,
 3. Inhabilitar el uso en el sistema,
 4. Finalizar.
- Editar descuento
 1. Ingresar nombre,
 2. Ver información detallada del descuento,
 3. Los cambios se guardan en la base de datos,
 4. finalizar.

Cocina

Al ingresar al sistema se despliega un menú donde tenemos los siguientes apartados (opciones):

Apartado de platos:

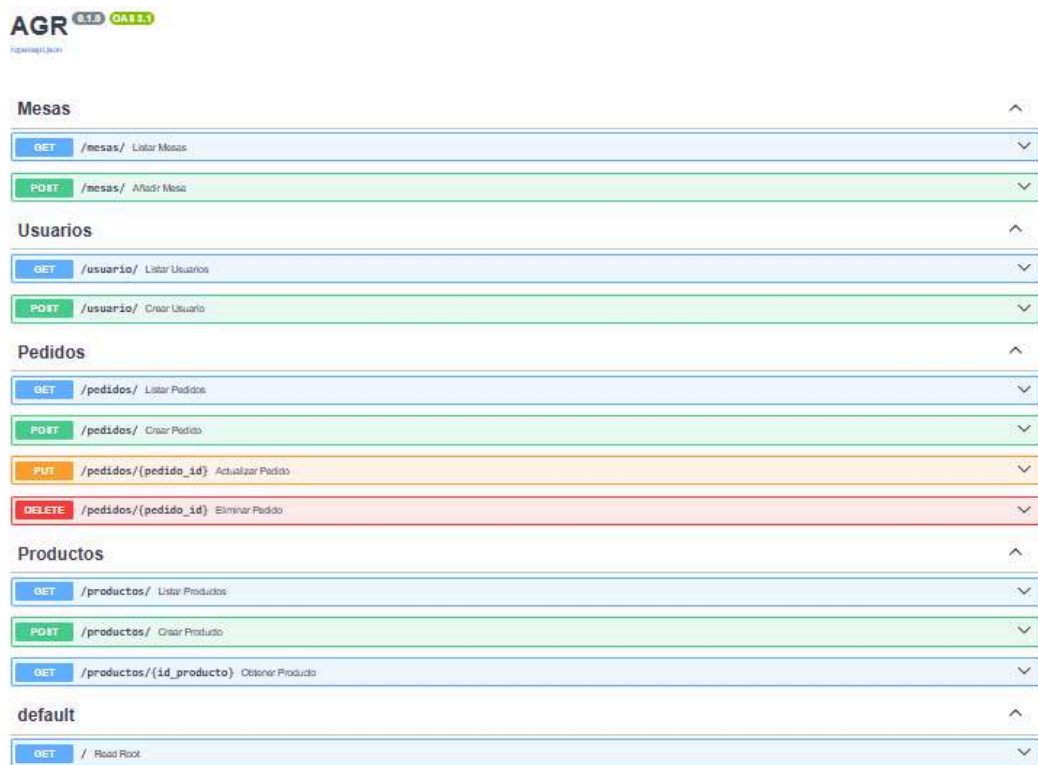
- Preparar
 1. Ver nombre del plato,
 2. Ver especificaciones.
 3. Finalizar.
- Estado del plato
 1. Al recibir el pedido se marca como “Preparándose”,
 2. Lo marca como “Entregado” y el mesero recibe la notificación,
 3. Finalizar.

Modificación de la base de datos

Se agregaron unos triggers para que se genere un detalle cada que se hace un pedido y además se calcule el subtotal desde base de datos y se actualice, cada pedido va a tener una lista de detalles, esto se maneja desde los esquemas de pydantic, se decide dejar así, debido a que la idea original de que un producto tenga un solo detalle y este detalle unos productos, iba pero requería tablas extras para manejar las relaciones N a N por y podría llegar a generar problemas en las consultas joined, por lo que se decidió que un pedido tuviera varios detalles donde van los productos de ese pedido.

Avances API backend

Creación de rutas fast api



The screenshot displays the AGR API documentation interface. At the top left, the logo 'AGR' is shown with version '0.1.0' and 'QA11.1' in a green box. Below the logo is the URL 'http://api.agr.com'. The main content is organized into sections, each with a title and a list of endpoints. Each endpoint is represented by a colored bar indicating the HTTP method: GET (blue), POST (green), PUT (orange), and DELETE (red). The endpoints are as follows:

Section	Method	Endpoint	Description
Mesas	GET	/mesas/	Listar Mesas
	POST	/mesas/	Añadir Mesa
Usuarios	GET	/usuario/	Listar Usuarios
	POST	/usuario/	Crear Usuario
Pedidos	GET	/pedidos/	Listar Pedidos
	POST	/pedidos/	Crear Pedido
	PUT	/pedidos/{pedido_id}	Actualizar Pedido
	DELETE	/pedidos/{pedido_id}	Eliminar Pedido
Productos	GET	/productos/	Listar Productos
	POST	/productos/	Crear Producto
	GET	/productos/{id_producto}	Obtener Producto
default	GET	/	Root Root

Conexión con la base de datos y modelos según tablas de la base de datos

En el apartado de [database.py](#) se gestiona la conexión a la base de datos que definimos en la entrega anterior.

```

from sqlalchemy.orm import sessionmaker, declarative_base
from sqlalchemy import create_engine

DATABASE_URL= "mysql+pymysql://root:1234@localhost:3306/db_restaurante"

engine =create_engine(DATABASE_URL, echo=True)

SessionLocal =sessionmaker(autocommit=False, autoflush=False, bind=engine)
# (Session maker)=Este es un generador de sesiones, que cuando se llama se obtiene la clase
# Cancela el autocommit para que los cambios en la base de datos solo se hagan cuando hacemos
# El auto flush evita que los modelos se sincronicen automaticamente con la base de datos
#
Base = declarative_base() #Esta es una clase que le dice a SQLALCHEMY como covertir los datos
| | | | | | | | | | #python a la base de datos, como si se pasara con un diccionario,
|
def get_db():
    db=SessionLocal() #Aqui se crea una nueva sesion db.
    try:
        yield db #Aqui le entrega la sesion al codigo que la llamo, el yield permite que
    finally:
        db.close() # sin importar lo que pase, si hay error o no la sesion se cierra, pa

```

Con la librería SQLAlchemy de Python definimos los modelos o las clases que mapean las tablas de la base de datos para que de esta manera se pueda interactuar con los datos usando POO en lugar de consultas SQL directas.

```

from sqlalchemy import Column, Integer, String, ForeignKey, Text, Enum, Boolean, Numeric, TII
from sqlalchemy.orm import relationship
from app.database import Base

class Usuario(Base):
    __tablename__="usuarios"
    id_usuario= Column(Integer, primary_key=True, index=True)
    nombre= Column(String(100), nullable=False)
    correo= Column(String(100), unique= False )
    usuario= Column(String(100), nullable=False, unique=True)
    contraseña= Column(String(100), nullable=False, unique=True)
    rol=Column(Enum('administrador', 'cajero', 'mesero', name="rol_usuario"), nullable=False)

    pedidos= relationship("Pedido", back_populates= "usuario")

class Mesa(Base):
    __tablename__="mesas"
    id_mesa= Column(Integer, primary_key=True, index=True, autoincrement=True)
    numero= Column(Integer, unique=True, nullable=False)
    tipo= Column(String(100), nullable=False)
    estado= Column(Enum('libre', 'ocupada', 'reservada'), nullable=False, default="libre")

    pedidos= relationship("Pedido", back_populates="mesa")

```


Inicio de desarrollo de Frontend en paralelo a Backend

Con el fin de inicializar la conexiones del backend con el frontend al que va a servir se crearon unos scripts básicos en html y javascript inicialmente, estos scripts son básicos debido a los constantes cambios en el backend.

En html se definieron formularios básicos con botones y en js se conectó esos botones y opciones a las rutas correspondientes en el backend.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Gestión de Usuarios</title>
</head>
<body>
  <h1>Usuarios</h1>

  <!-- Formulario para crear usuario -->
  <form id="formUsuario"> <!-- le da un identificador unico para que pueda ser manipulado p
    <input type="text" id="nombre" placeholder="Nombre" required><br>
    <!-- Le da un identificador unico, para el js. el placeholder es el texto en gris o temp
    <input type="email" id="correo" placeholder="Correo"><br>
    <input type="text" id="usuario" placeholder="Usuario" required><br>
    <input type="password" id="contraseña" placeholder="Contraseña" required><br>
    <select id="rol">
      <option value="administrador">Administrador</option> <!-- el value va a ser lo que se
      <option value="cajero">Cajero</option>
      <option value="mesero">Mesero</option>
    </select><br>
    <button type="submit">Crear</button>
  </form>

  <hr> <!-- la linea horizontal que separa -->

  <!-- Lista de usuarios -->
  <h2>Usuarios registrados</h2>
  <button onclick="cargarUsuarios()">Ver usuarios</button> <!-- Este crea un boton que al p

```

```

const API_URL = "http://127.0.0.1:8000"; // tu backend FastAPI

// Capturar el formulario y crear usuario; busca el elemento detro del html que tiene el id
// En este caso el <form> de la pagina, asigna un escuchador de eventos al formulario para q
//cuando el usuario intente eviar el formulario en este caso al hacer clic en crear.

document.getElementById("formUsuario").addEventListener("submit", async (e) => { //async es
  //que se va a ejecutar cuando el evento ocurra en este caso el crear, asincronica
  //quiere decir que no para la ejecucion del programa hasta que se termine de ejecutar sin
  //sigue ejecutando o respondiendo otros eventos.

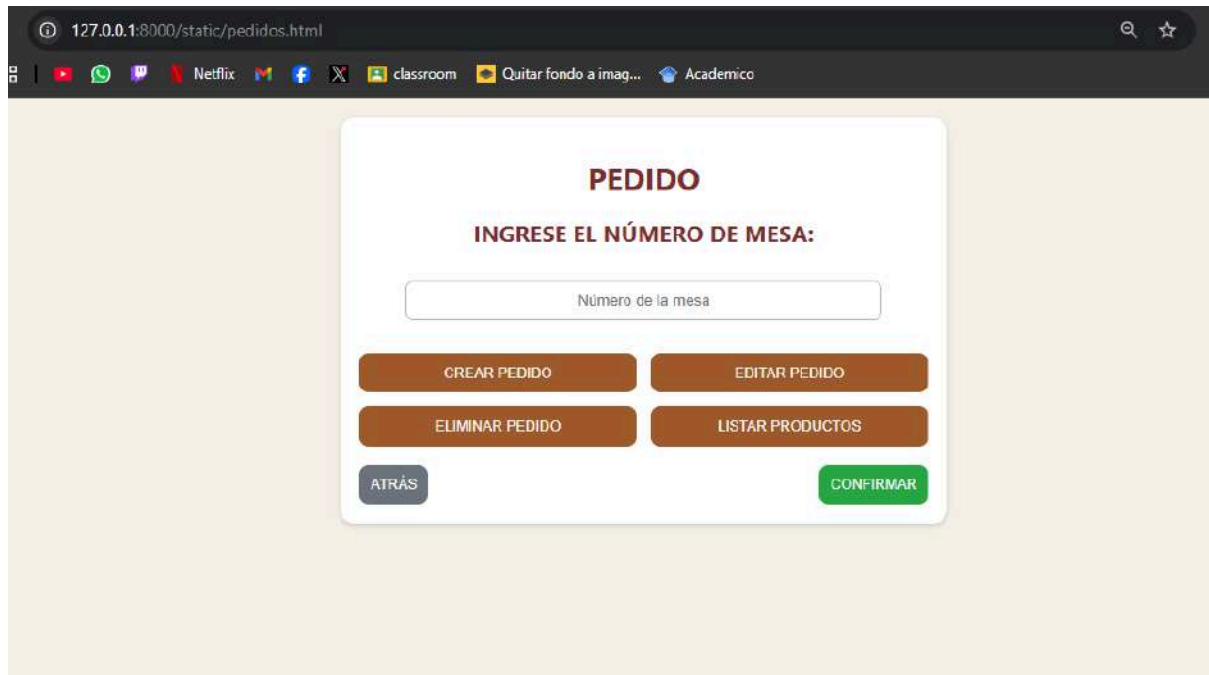
  e.preventDefault(); //esto es un metodo del evento, hace que la pagina no se recargue cuan
  //formulario, para js pueda manejar la solicitud de forma asincronica.

  const data = { //Aquí se crea un objeto para almacenar la informacion del form

    nombre: document.getElementById("nombre").value, //Busca el campo del form con el id "noi
    //a la propiedad nombre del objeto data.
    correo: document.getElementById("correo").value,
    usuario: document.getElementById("usuario").value,
    contraseña: document.getElementById("contraseña").value,
    rol: document.getElementById("rol").value
  };

  const resp = await fetch(`${API_URL}/usuario/`, { //Aquí se envia una solicitud fetch o de
    //para obtener un recurso del servidor o en este caso la api usando la api fetch integra
    //promesa de solicitud en este caso post de forma asincronica, el await lo que hace es q
    //la funcion espere a que la respuesta del servidor llegue.

```



Actividades para Octubre

Seguridad y Autenticación

- Implementar encriptación de contraseñas.
- Pasar el login al frontend y conectarlo con el backend.
- Establecer roles de usuario (cajero, mesero, cocina, administrador).

Interfaz Frontend Inicial

- Crear la pantalla de login y registro de usuarios para el personal del restaurante con base las plantillas ya realizadas en canva.
- Mockups o prototipos de la interfaz del restaurante (mesas, pedidos, reservas).

Módulo de Pedidos (Versión Inicial)

- Conectar el backend en Python con la base de datos para registrar pedidos desde una interfaz básica.
- Permitir que un mesero seleccione una mesa y asigne productos.
- Que la cocina pueda visualizar en tiempo real los pedidos registrados.

Seguridad Adicional

- Validaciones en el frontend (que el cliente o usuario no ingrese datos inválidos).
- Restricción de accesos según rol en el backend.
- Plan inicial de copias de seguridad de la base de datos.

Pruebas Iniciales

- Probar flujo completo: login → pedido → cocina → caja.
- Documentar errores encontrados y posibles soluciones.