



PLAN DE TRABAJO DEL ESTUDIANTE

DATOS DEL ESTUDIANTE

Apellidos y Nombres:	Moises condori ramos	ID:	001617659
Dirección Zonal/CFP:	Cusco		
Carrera:	Ingeniería de software		III
Curso/ Mód. Formativo:			
Tema de Trabajo Final:	Proyecto final		

1. INFORMACIÓN

- **Identifica la problemática del caso práctico propuesto.**

“Sabor Gourmet” es un restaurante con 20 mesas, atención presencial y autoservicio digital. Opera con varios empleados: meseros, cocineros, cajeros y un administrador. En las horas pico, el restaurante enfrenta desorganización: **órdenes mal tomadas, facturación lenta, y errores en cocina.** Para modernizar sus procesos, deciden implementar un **sistema digital** que gestione la operación interna y también permita **autoservicio a clientes desde una app o kiosko digital.**

- **Identifica propuesta de solución y evidencias.**

□ Propuesta de Solución

Con base en los problemas identificados en el restaurante “Sabor Gourmet”, se propone la implementación de un **Sistema Integral de Gestión para Restaurantes**, desarrollado bajo arquitectura modular, accesible vía web y compatible con dispositivos móviles y tablets. Este sistema permitirá automatizar y digitalizar todos los procesos operativos y administrativos del restaurante.

✓ Objetivos de la solución:

- Digitalizar la toma de pedidos y envío a cocina.
- Agilizar la facturación y cobro.
- Mejorar el control y trazabilidad de órdenes en cocina.
- Ofrecer al cliente una experiencia de autoservicio moderna.
- Brindar al administrador herramientas de gestión y reportes en tiempo real.
- Controlar accesos y roles mediante autenticación.

▪ Respuestas a preguntas guía

Durante el análisis y estudio del caso práctico, debes obtener las respuestas a las interrogantes:

Pregunta 01:	1) ¿Cómo definirías los requisitos funcionales y no funcionales del sistema propuesto para WEB-Cántabra?
□ Funcionales: Son las acciones que el sistema debe realizar: gestionar pedidos, tomar reservas, autenticar usuarios, facturar, mostrar el menú y generar reportes. □ No funcionales: Son las cualidades del sistema: debe ser rápido, seguro, accesible desde móviles, escalable y fácil de usar.	
Pregunta 02:	¿Qué métricas utilizarías para medir la eficiencia y el éxito del proceso ágil durante el desarrollo?
□ Velocidad del equipo (velocity): mide el avance por sprint. □ Burn-down chart: muestra el progreso real del equipo. □ Errores por entrega: para controlar la calidad. □ Satisfacción del cliente (CSAT/NPS): valida el valor entregado.	

☐ Tiempos de ciclo (lead/cycle time): miden eficiencia en las tareas.	
Pregunta 03:	¿Cuáles son las principales diferencias entre la metodología ágil y las tradicionales, y por qué elegiste ágil para este proyecto?
☐ Ágil es flexible, entrega rápido y escucha al usuario. ☐ Tradicional es rígida y lenta para adaptarse. ☐ Elegimos ágil porque el restaurante necesita mejorar y ajustar el sistema con rapidez, según la experiencia real del cliente y del personal.	
Pregunta 04:	¿Cómo garantizarías la calidad del software durante el ciclo de vida del proyecto, especialmente con entregas frecuentes?
☐ Usamos pruebas automatizadas, revisiones de código, entornos de prueba, y feedback del usuario en cada entrega. ☐ Cada versión se valida antes de subirla a producción.	
Pregunta 05:	¿Qué ventajas competitivas puede ofrecer el software propuesto en comparación con los competidores actuales de WEB-Cántabra?
☐ Pedidos rápidos y sin errores. ☐ Facturación automatizada. ☐ Autoservicio moderno para el cliente. ☐ Reportes inteligentes para el administrador. ☐ Mejor experiencia = más ventas y fidelidad.	

2. PLANIFICACIÓN DEL TRABAJO

▪ **Cronograma de actividades:**

N°	ACTIVIDADES	CRONOGRAMA					

▪ **Lista de recursos necesarios:**

1. MÁQUINAS Y EQUIPOS	
Descripción	Cantidad

2. HERRAMIENTAS E INSTRUMENTOS	
Descripción	Cantidad

3. MATERIALES E INSUMOS	
Descripción	Cantidad

3. DECIDIR PROPUESTA

- Describe la propuesta determinada para la solución del caso práctico

PROPUESTA DE SOLUCIÓN
1. Introducción Este documento describe los requerimientos y especificaciones para el desarrollo del Sistema de Gestión de Restaurante, una solución integral que automatiza y optimiza las operaciones diarias de establecimientos gastronómicos. 1.1 Alcance del producto El Sistema de Gestión de Restaurante tiene como objetivo automatizar y optimizar las siguientes operaciones: - Gestión de mesas y reservas - Administración de pedidos y órdenes de cocina - Control de inventario y menú - Facturación y procesamiento de pagos - Generación de reportes administrativos - Control de usuarios y roles 1.2 Valor del producto El producto proporcionará los siguientes beneficios: - Reducción del tiempo de servicio al cliente en un 30% - Minimización de errores en pedidos y facturación - Optimización del flujo de trabajo entre meseros, cocina y caja - Mejora en la experiencia del cliente - Control financiero y operativo en tiempo real - Incremento de la eficiencia operativa del restaurante 1.3 Público objetivo Usuarios principales: Meseros: Personal de servicio al cliente Cocineros: Personal de cocina Cajeros: Personal de facturación Administradores: Gerentes y propietarios del restaurante

Usuarios secundarios:

- **Clientes:** Para funcionalidades de autoservicio (opcional)

1.4 Uso previsto

El sistema será utilizado para:

- **Meseros:** Registrar pedidos, consultar disponibilidad de mesas, gestionar el estado de las órdenes
- **Cocineros:** Visualizar pedidos pendientes, actualizar estado de preparación, gestionar tiempos de cocina
- **Cajeros:** Generar facturas, aplicar descuentos, procesar pagos, manejar devoluciones
- **Administradores:** Configurar menú, gestionar usuarios, generar reportes, administrar inventario
- **Clientes:** Realizar reservas, consultar menú, hacer pedidos de autoservicio

1.5 Descripción general

El Sistema de Gestión de Restaurante es una aplicación web responsiva con arquitectura cliente-servidor que integra módulos de gestión de pedidos, facturación, inventario y administración. El sistema incluye interfaces diferenciadas por rol de usuario, integración con sistemas de pago, y capacidades de reportería en tiempo real.

2. Requisitos funcionales**RF-001 Gestión de Autenticación**

- El sistema debe permitir login seguro con usuario y contraseña
- Debe implementar control de acceso basado en roles
- Debe mantener sesiones de usuario activas durante el turno de trabajo

RF-002 Gestión de Mesas

- El sistema debe mostrar el estado de las mesas (disponible, ocupada, reservada)
- Debe permitir asignar y liberar mesas
- Debe gestionar reservas con fecha y hora

RF-003 Gestión de Pedidos

- Los meseros deben poder registrar pedidos por mesa
- El sistema debe enviar automáticamente los pedidos a cocina
- Debe permitir modificar pedidos antes de enviar a cocina

RF-004 Gestión de Cocina

- Los cocineros deben visualizar pedidos pendientes en orden cronológico
- Debe permitir actualizar el estado de preparación de cada plato
- Debe notificar cuando los platos estén listos

RF-005 Facturación

- El sistema debe generar facturas automáticamente

- Debe permitir aplicar descuentos y promociones
- Debe soportar múltiples métodos de pago

RF-006 Administración de Menú

- Debe permitir agregar, modificar y eliminar productos del menú
- Debe gestionar precios y categorías
- Debe controlar disponibilidad de productos

RF-007 Reportería

- Debe generar reportes de ventas diarias, semanales y mensuales
- Debe mostrar estadísticas de productos más vendidos
- Debe generar reportes de eficiencia operativa

3. Requisitos de la interfaz externa**3.1 Requisitos de la interfaz de usuarios**

- Interfaz web responsiva compatible con tablets y dispositivos móviles
- Dashboard diferenciado por roles con navegación intuitiva
- Tema visual profesional con colores corporativos del establecimiento
- Iconografía clara para acciones principales
- Notificaciones visuales para alertas y confirmaciones
- Accesibilidad cumpliendo estándares WCAG 2.1

3.2 Requisitos de la interfaz de hardware

- **Dispositivos compatibles:** Tablets Android/iOS, computadoras Windows/Mac/Linux
- **Impresoras térmicas:** Para tickets de cocina y facturas
- **Lectores de códigos de barras:** Para productos de inventario
- **Terminal punto de venta (POS):** Para procesamiento de pagos
- **Red WiFi:** Estable con velocidad mínima de 10 Mbps

3.3 Requisitos de la interfaz de software

- **Base de datos:** MySQL 8.0 o PostgreSQL 12+
- **Framework backend:** Laravel 10+ (PHP 8.1+)
- **IDE de desarrollo:** Visual Studio Code con extensiones PHP y Laravel
- **Framework frontend:** Laravel Blade con Bootstrap 5 o Vue.js integrado
- **Servidor web:** Apache con mod_rewrite habilitado
- **API REST:** Para comunicación entre componentes
- **Integración con sistemas de pago:** PayPal, Stripe, o pasarelas locales

3.4 Requisitos de la interfaz de comunicación

- Protocolo HTTPS para todas las comunicaciones
- API REST con formato JSON para intercambio de datos
- WebSockets para notificaciones en tiempo real
- Integración por email para confirmaciones y reportes
- SMS opcional para notificaciones de reservas

4. Requisitos no funcionales

4.1 Seguridad

- Encriptación de contraseñas con bcrypt o similar
- Implementación de HTTPS con certificados SSL/TLS
- Protección contra ataques SQL injection, XSS y CSRF
- Backup automático de datos cada 24 horas
- Logs de auditoría para acciones críticas
- Cumplimiento con normativas de protección de datos locales

4.2 Capacidad

- **Usuarios concurrentes:** Hasta 50 usuarios simultáneos
- **Almacenamiento:** 100GB inicial con capacidad de expansión
- **Transacciones:** 1000 pedidos por día
- **Base de datos:** Hasta 1 millón de registros históricos
- **Tiempo de respuesta:** Máximo 3 segundos para consultas complejas

4.3 Compatibilidad

- **Navegadores web:** Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
- **Sistemas operativos:** Windows 10+, macOS 10.15+, Ubuntu 20.04+
- **Dispositivos móviles:** Android 8.0+, iOS 13+
- **Resoluciones de pantalla:** Desde 1024x768 hasta 4K

4.4 Confiabilidad

- **Disponibilidad:** 99.5% de uptime (máximo 3.6 horas de inactividad mensual)
- **MTBF (Mean Time Between Failures):** 720 horas
- **MTTR (Mean Time To Recovery):** Máximo 30 minutos
- Tolerancia a fallos con recuperación automática de sesiones

4.5 Escalabilidad

- **Escalabilidad horizontal:** Soporte para múltiples servidores

- **Carga máxima:** 200% de la capacidad normal durante picos
- **Crecimiento de datos:** Manejo eficiente hasta 10x el volumen inicial
- **Nuevas funcionalidades:** Arquitectura modular para expansiones futuras

4.6 Mantenibilidad

- Código documentado con comentarios y documentación técnica
- Integración continua con pipelines automatizados
- Versionado de código con Git
- Monitoreo automatizado de rendimiento y errores
- Actualizaciones sin interrupciones del servicio

4.7 Facilidad de uso

- **Curva de aprendizaje:** Máximo 2 horas de entrenamiento por usuario
- **Interfaz intuitiva:** Máximo 3 clics para acciones comunes
- Ayuda contextual y tooltips informativos
- Manual de usuario en español con capturas de pantalla
- Soporte técnico durante horarios comerciales

4.8 Otros requisitos no funcionales

- **Rendimiento:** Tiempo de carga inicial menor a 5 segundos
- **Usabilidad móvil:** Optimización para uso con una mano
- **Internacionalización:** Preparado para múltiples idiomas
- **Cumplimiento legal:** Normativas fiscales y comerciales locales

5. Definiciones y acrónimos

- **API:** Application Programming Interface - Interfaz de Programación de Aplicaciones
- **CRUD:** Create, Read, Update, Delete - Operaciones básicas de base de datos
- **HTTPS:** HyperText Transfer Protocol Secure - Protocolo seguro de transferencia
- **JSON:** JavaScript Object Notation - Formato de intercambio de datos
- **MTBF:** Mean Time Between Failures - Tiempo medio entre fallos
- **MTTR:** Mean Time To Recovery - Tiempo medio de recuperación
- **POS:** Point of Sale - Punto de venta
- **REST:** Representational State Transfer - Arquitectura de servicios web
- **SSL/TLS:** Secure Sockets Layer/Transport Layer Security - Protocolos de seguridad
- **UML:** Unified Modeling Language - Lenguaje de modelado unificado

- **WCAG:** Web Content Accessibility Guidelines - Pautas de accesibilidad web

6. Diagramas de Procesos

[Los diagramas específicos serán desarrollados durante la fase de diseño detallado]

7. Casos de Uso del Negocio

Actores del Negocio

1. **Mesero:** Atiende clientes, registra pedidos, consulta estado de mesas
2. **Cocinero:** Consulta órdenes pendientes, actualiza estado de preparación
3. **Cajero:** Genera factura, aplica descuentos, recibe pagos
4. **Administrador:** Administra menú, usuarios, configuración del restaurante, y reportes
5. **Cliente (opcional):** Reserva mesa, consulta menú, hace pedidos (autoservicio)
6. **Sistema de Autenticación:** Maneja Login, registro, roles, y control de acceso

Casos de Uso Principales

CU-001: Gestionar Pedido

- **Actor:** Mesero
- **Descripción:** El mesero registra un nuevo pedido para una mesa específica
- **Flujo principal:** Login → Seleccionar mesa → Agregar productos → Confirmar pedido → Enviar a cocina

CU-002: Preparar Orden

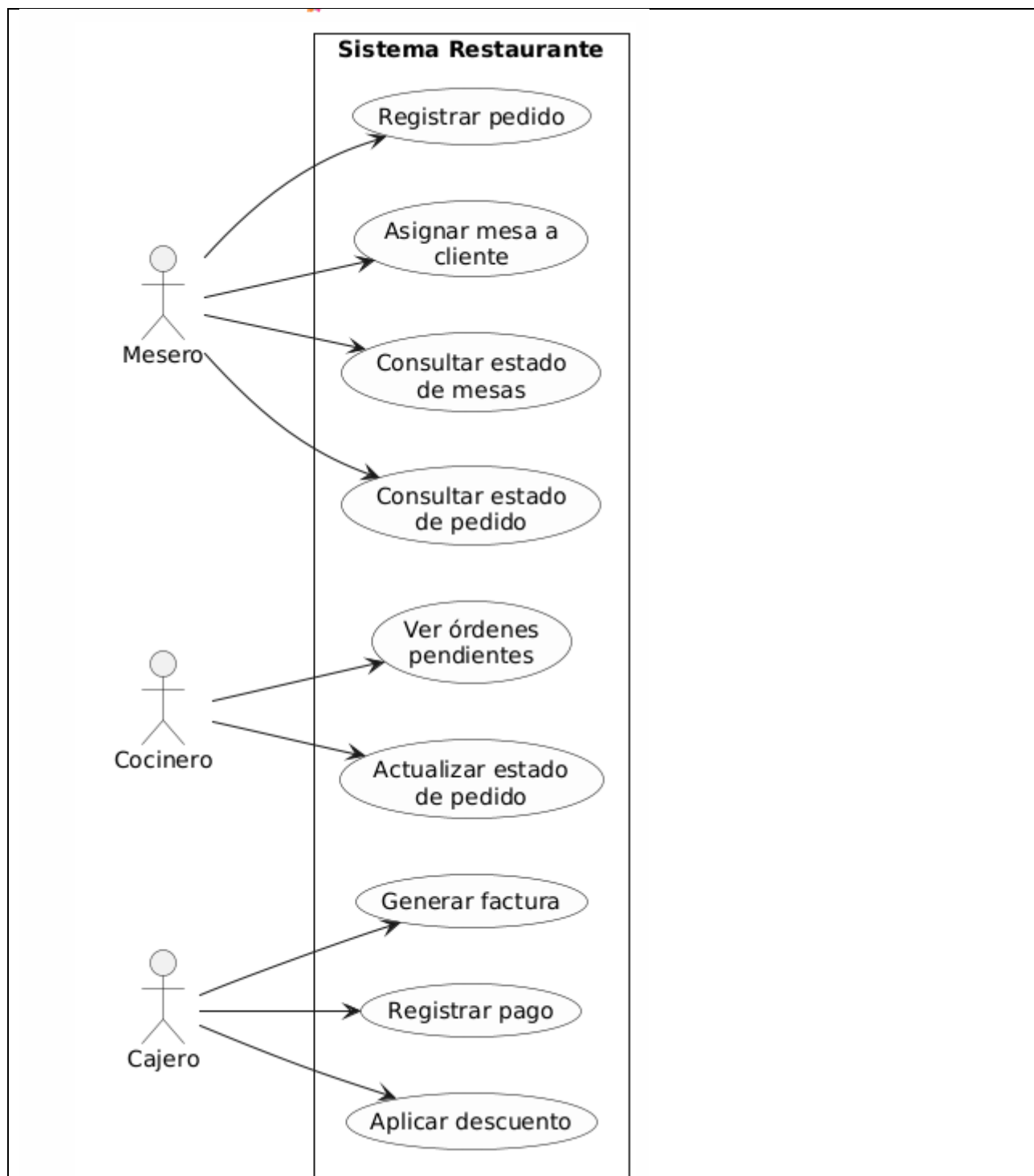
- **Actor:** Cocinero
- **Descripción:** El cocinero visualiza y actualiza el estado de preparación de las órdenes
- **Flujo principal:** Login → Ver pedidos pendientes → Seleccionar pedido → Actualizar estado → Notificar finalización

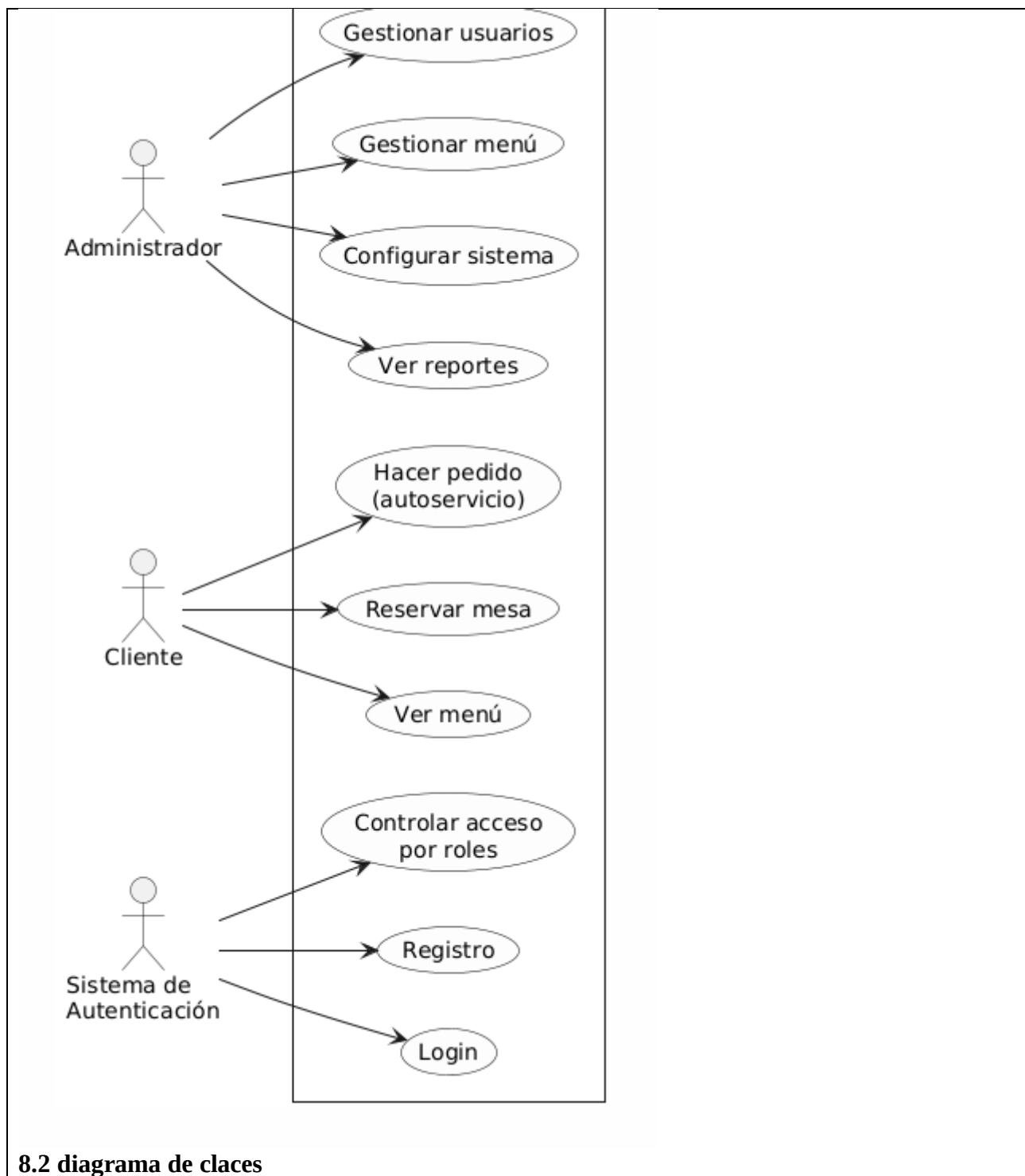
CU-003: Procesar Pago

- **Actor:** Cajero
- **Descripción:** El cajero genera la factura y procesa el pago del cliente
- **Flujo principal:** Login → Seleccionar mesa → Generar factura → Aplicar descuentos → Procesar pago → Emitir comprobante

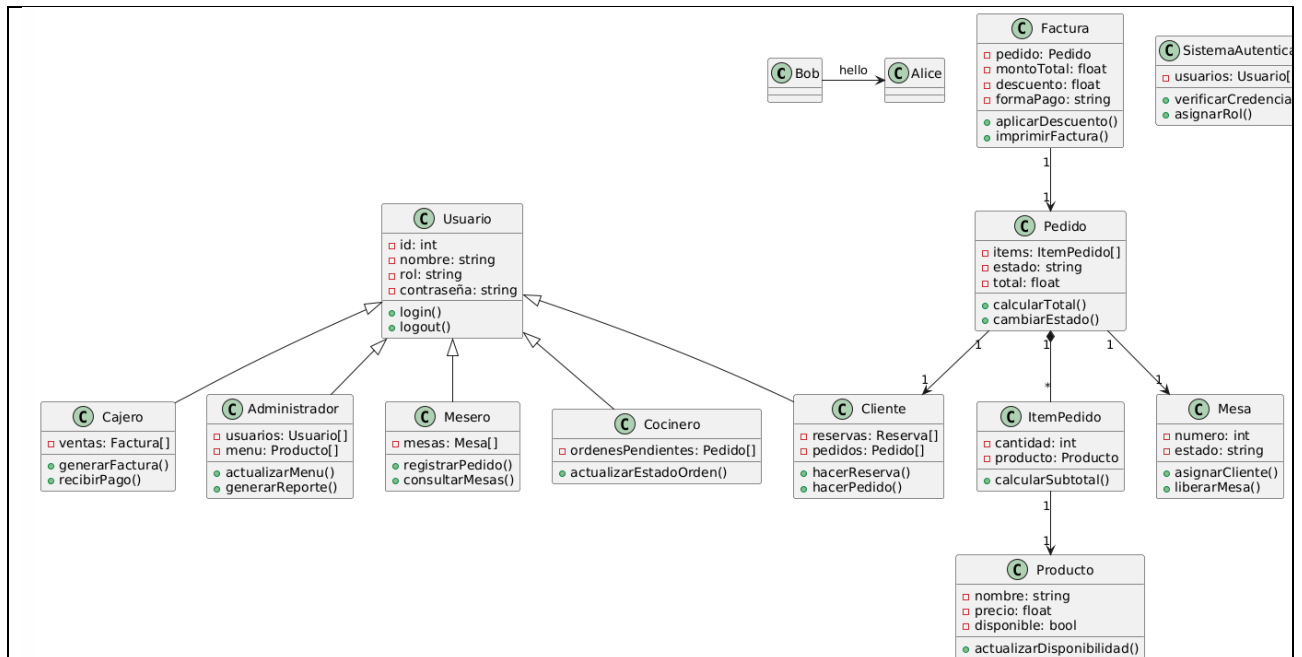
8USO DE DIAGRAMAS UML







8.2 diagrama de clases



1. Usuario (clase base)

- Representa cualquier persona que interactúe con el sistema.
- Atributos: ID, nombre, rol, contraseña.
- Métodos: iniciar y cerrar sesión.
- Subclases: Cliente, Mesero, Cocinero, Cajero, Administrador.

2. Cliente

- Hereda de Usuario.
- Puede hacer reservas y pedidos.
- Relación con Pedido (uno a muchos).

3. Mesero

- Hereda de Usuario.
- Asocia mesas a clientes y registra pedidos.

4. Cocinero

- Hereda de Usuario.
- Consulta y actualiza órdenes pendientes.

5. Cajero

- Hereda de Usuario.
- Genera facturas y registra pagos.

6. Administrador

- Hereda de Usuario.
- Gestiona el menú, usuarios del sistema y reportes.

□ Mesa

- Representa una mesa del restaurante.
- Tiene un número, estado (disponible, ocupada).
- Se asocia con clientes y pedidos.

□ Pedido

- Contiene los items solicitados por un cliente.
- Tiene estado (pendiente, en preparación, entregado).
- Se relaciona con Cliente, Mesa y ItemPedido.

□ ItemPedido

- Representa un ítem dentro de un pedido.
- Tiene cantidad y referencia a un Producto.
- Se utiliza para calcular subtotales.

Producto (del Menú)

- Contiene nombre, precio y disponibilidad.
- Referenciado por los ítems de pedido.

Factura

- Se genera al finalizar un pedido.
- Incluye total, forma de pago y descuentos.
- Se asocia a un Pedido.

SistemaAutenticación

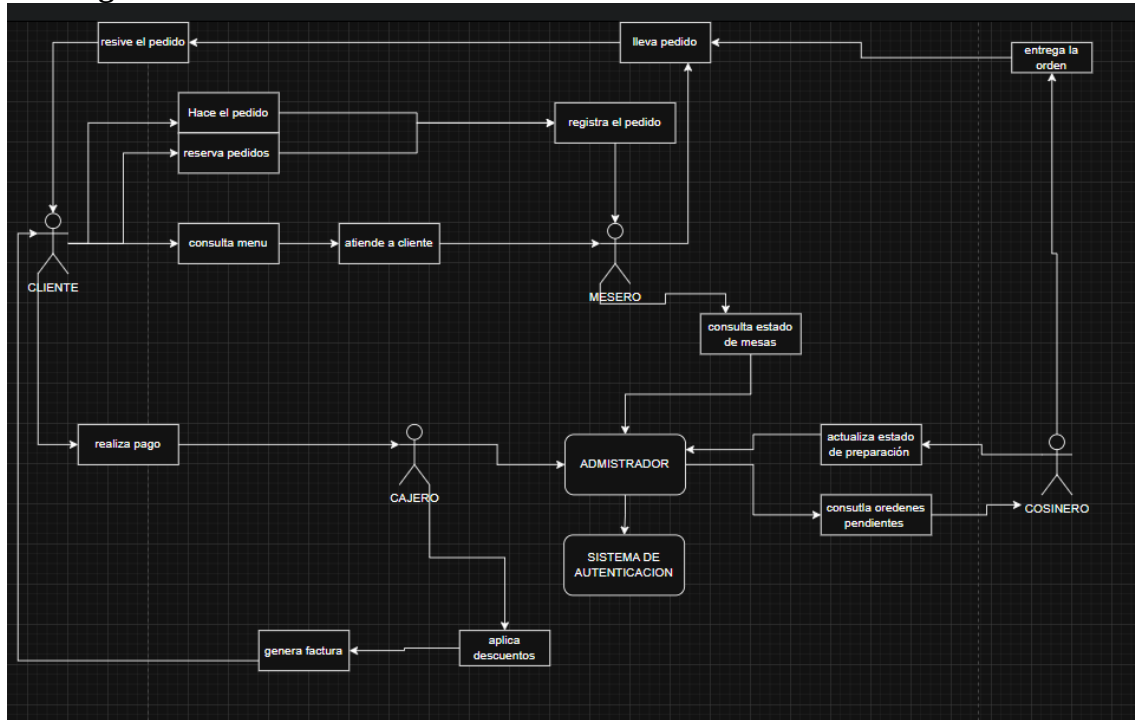
- Maneja la verificación de credenciales y asignación de roles.
- Se relaciona con la clase Usuario.

Relaciones Clave

- **Herencia:** Usuario es la clase base de todos los roles del sistema.
- **Asociaciones:**
 - Cliente → Pedido (1:N)
 - Pedido → ItemPedido (1:N)

- ItemPedido → Producto (1:1)
- Pedido → Factura (1:1)
- Pedido → Mesa (1:1)

8.3 diagrama de actividades



9. Desarrollo Final del Prototipado

El prototipo final debe incluir:

- Mockups de interfaces para cada tipo de usuario
- Flujos de navegación principales del sistema
- Casos de prueba para validar funcionalidades críticas

4. EJECUTAR

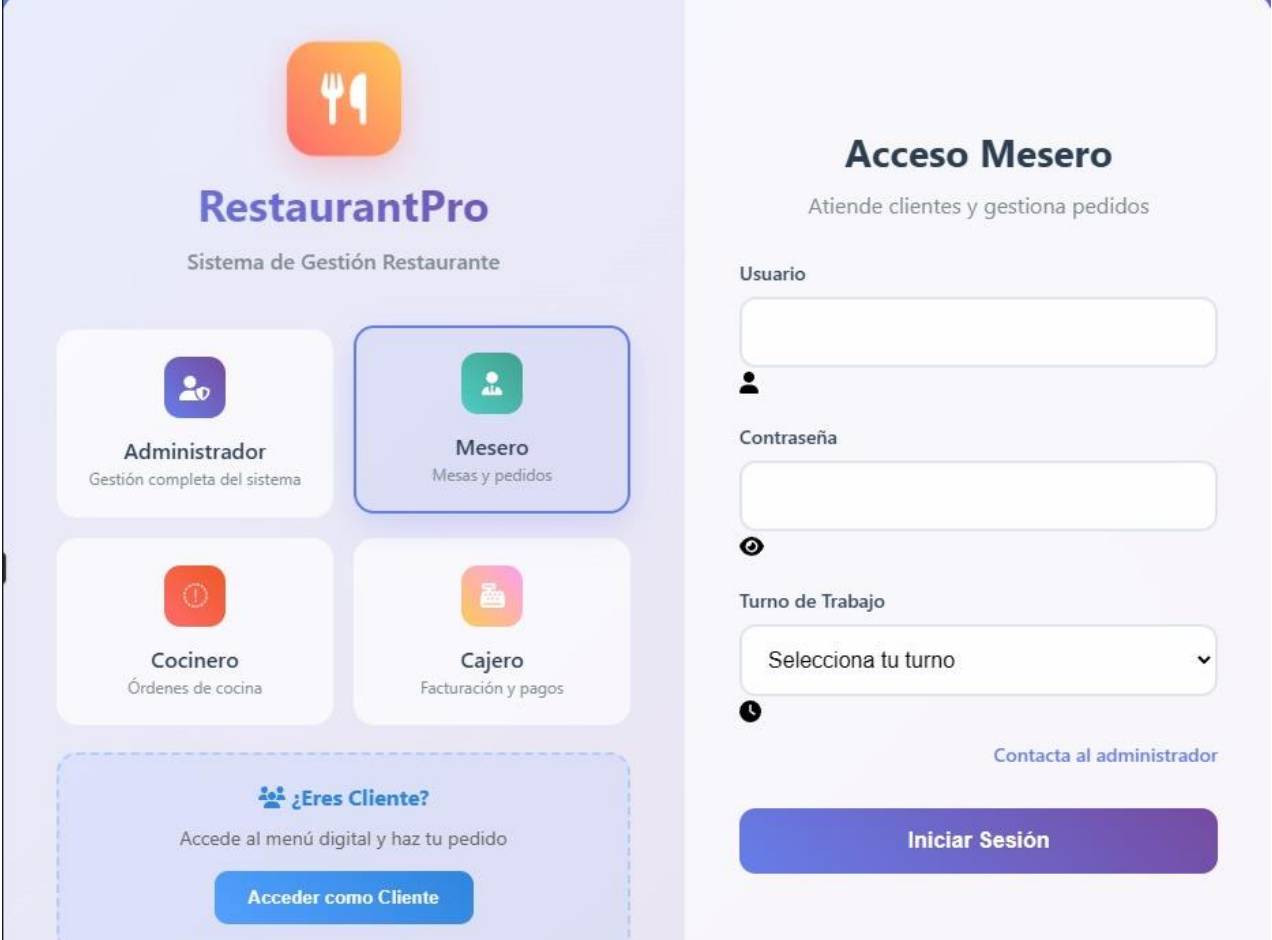
- Resolver el caso práctico, utilizando como referencia el problema propuesto y las preguntas guía proporcionadas para orientar el desarrollo.
- Fundamentar sus propuestas en los conocimientos adquiridos a lo largo del curso, aplicando lo aprendido en las tareas y operaciones descritas en los contenidos curriculares.

INSTRUCCIONES: Ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. Tomar en cuenta los aspectos de calidad, medio ambiente y SHI.

OPERACIONES / PASOS / SUBPASOS	NORMAS TÉCNICAS - ESTANDARES / SEGURIDAD / MEDIO AMBIENTE

DIBUJO / ESQUEMA / DIAGRAMA DE PROPUESTA

(Adicionar las páginas que sean necesarias)



DISEÑO(BOSQUEJO) DE LA PAGIANA WEB

 SENATI	[NOMBRE DEL TEMA DEL TRABAJO FINAL]	
	[APELLIDOS Y NOMBRES]	[ESCALA]

5. CONTROLAR

- Verificar el cumplimiento de los procesos desarrollados en la propuesta de solución del caso práctico.

EVIDENCIAS	CUMPLE	NO CUMPLE
• ¿Se identificó claramente la problemática del caso práctico?	☐	☐
• ¿Se desarrolló las condiciones de los requerimientos solicitados?	☐	☐
• ¿Se formularon respuestas claras y fundamentadas a todas las preguntas guía?	☐	☐
• ¿Se elaboró un cronograma claro de actividades a ejecutar?	☐	☐
• ¿Se identificaron y listaron los recursos (máquinas, equipos, herramientas, materiales) necesarios para ejecutar la propuesta?	☐	☐
• ¿Se ejecutó la propuesta de acuerdo con la planificación y cronograma establecidos?	☐	☐
• ¿Se describieron todas las operaciones y pasos seguidos para garantizar la correcta ejecución?	☐	☐
• ¿Se consideran las normativas técnicas, de seguridad y medio ambiente en la propuesta de solución?	☐	☐
• ¿La propuesta es pertinente con los requerimientos solicitados?	☐	☐
• ¿Se evaluó la viabilidad de la propuesta para un contexto real?	☐	☐

6. VALORAR

- Califica el impacto que representa la propuesta de solución ante la situación planteada en el caso práctico.

CRITERIO DE EVALUACIÓN	DESCRIPCIÓN DEL CRITERIO	PUNTUACIÓN MÁXIMA	PUNTAJE CALIFICADO POR EL ESTUDIANTE
Identificación del problema	Claridad en la identificación del problema planteado.	3	
Relevancia de la propuesta de solución	La propuesta responde adecuadamente al problema planteado y es relevante para el contexto del caso práctico.	8	
Viabilidad técnica	La solución es técnicamente factible, tomando en cuenta los recursos y conocimientos disponibles.	6	
Cumplimiento de Normas	La solución cumple con todas las normas técnicas de seguridad, higiene y medio ambiente.	3	
PUNTAJE TOTAL		20	

