

Evaluación 2: Las buenas prácticas como principio; la calidad como fundamento

Restaurante Los Naranjos

Cristobal Aguila

Angelo Millan

Oscar Muñoz

Profesor Álvaro Herrera

Ingeniería De Software

Mayo, 2025

**1. Introducción**

El restaurante “Los Naranjos”, ubicado en la comuna de La Florida, es un emprendimiento familiar que ha ido evolucionando en el tiempo y hoy es uno de los restaurantes que tiene una reputación importante en el sector.

A pesar de esta buena reputación, el año pasado la cantidad de reclamos relacionados con una deficiente atención, demora en la entrega de los productos, y cobros que no corresponden tuvo un aumento exponencial llegando a un 400%, generando molestias en los clientes, en los potenciales clientes, así como en la organización completa del restaurante.

Al revisar las causas raíz de la mala experiencia que están teniendo los clientes, se logró identificar y consensuar las siguientes:

* Existe demasiada manualidad en los procesos.
* Las reservas son anotadas en un cuaderno y la mayoría de las veces no están al momento que los clientes llegan.
* El precio de los productos no está correctamente establecido.
* Los roles de las personas del restaurante no están claros.
* Varias veces falta personal, debido a la inexistencia de un turno formal para operar.
* En la cocina las comandas llegan escritas sin un formato estándar, generando retrasos.
* Las comandas son escritas a puño y letra de la persona que atiende, por lo cual, muchas veces no se entiende el pedido.

**2. Objetivos del Equipo de Trabajo**

2.1 Objetivos Específicos

Una de las conclusiones de la planificación estratégica, es que se necesita incorporar tecnología a los procesos, y para ello se creó la “Unidad de Sistemas & Procesos” dependiendo directamente del Gerente General, para que lidere la implementación de un software que permita:

* **Optimizar la atención al cliente**, reduciendo tiempos de espera, mejorando la calidad del servicio.
* **Automatizar procesos** integrando el manejo de reservas, comandas, de cobros y de la disponibilidad de los productos en un solo sistema.
* Se pretende también **mejorar la precisión de la información** ingresada en las comandas, en pro de reducir los malos entendidos.
* Que el sistema sirva para la **organización interna de sus empleados**, evitando los problemas de falta de personal, por ejemplo.
* También el sistema buscará la **generación de reportes automatizados**, y con ello poder tomar decisiones con base, aportando en el desarrollo de la empresa.

2.2 Objetivos Generales.

**Mejorar la experiencia de los clientes** a través de la incorporación de tecnología a los procesos de venta y administración del restaurante Los Naranjos, permitiendo con ello recuperar la confianza perdida de los clientes, asegurando la continuidad y viabilidad del local comercial.

**3. Requerimientos del Proyecto**

3.1 Requerimientos Funcionales

Los requisitos funcionales describen las funciones y características que el sistema debe tener para poder cumplir a cabalidad con los Requisitos del Negocio.

* **Gestión de reservas**: El sistema debe permitir hacer, modificar y cancelar reservas en línea o telefónicamente, y en el local, con notificaciones en tiempo real para los anfitriones y mozos.
* **Sistema de comandas digital**: Los mozos podrán tomar los pedidos en un teléfono o tablet, lo que permitirá una comunicación más precisa con la cocina y el bar. Los pedidos deberán ser estructurados en base a una carta ya establecida dentro de la base de datos, impidiendo la ambigüedad de la escritura a mano total. De todas formas, debe existir un espacio pequeño para realizar anotaciones respecto a pequeñas variaciones que pudiera existir dentro de un pedido *(como por ejemplo: cliente no quiere pepinillos en la hamburguesa).*
* **Gestión de inventarios y disponibilidad de productos**: El sistema debe actualizar en tiempo real la disponibilidad de productos y enviar alertas cuando se acerque al límite del stock. También es necesario que pueda manejar los valores de los productos para su correcta cobranza al cliente.
* **Generación automática de detalle de venta**: Al finalizar la comida, se debe generar el detalle de los productos vendidos con su respectivo costo total, con la posibilidad de registrar el medio de pago elegido, sea este efectivo, tarjetas o cheques, y de seleccionar el monto de la propina para el mozo.
* **Lista de espera**: Si no hay mesas disponibles, el sistema debe gestionar la lista de espera y notificar a los anfitriones cuando el turno del siguiente cliente en fila ya esté próximo.

3.2 Requerimientos No Funcionales

Pensados fundamentalmente en el rendimiento del sistema, la seguridad y la disponibilidad del servicio. Los requerimientos No Funcionales detectados son:

* **Escalabilidad**: El sistema debe ser capaz de manejar un aumento de clientes sin perder eficiencia. De todas formas el sistema debe ser capaz de manejar sin problemas la máxima capacidad de clientes, el cual no varía, pues las mesas tampoco varían en su cantidad. Donde sí es necesaria la completa escalabilidad es durante los momentos en que se realizan más reservas: épocas festivas, años nuevos, navidades. Es fundamental garantizar que todas las personas puedan tener la posibilidad de reservar en el restaurante, y lograrlo a través de un sistema de cola de espera es una de las más factibles soluciones.
* **Seguridad**: El sistema debe garantizar la protección de datos personales y transacciones financieras.
* **Interfaz intuitiva**: La interfaz del sistema debe ser fácil de usar tanto para el personal del restaurante como para los clientes al momento de realizar sus reservas.
* **Disponibilidad**: El sistema debe estar operativo en los horarios establecidos (lunes a sábado de 9:30 am a 12:00 pm y domingo de 11:00 am a 3:00 pm

**4. Presupuesto Global**

Tomando en consideración que el caso entregado carece de las cantidades de personas atendidas en un periodo determinado, ni de la cantidad de mesas, ni de la cantidad de mozos que trabajan en el lugar, es que es necesario hacer ciertas suposiciones respecto a dichas cantidades en pro de obtener los costos asociados.

Podemos suponer que para un restaurante que lleva poco tiempo en el mercado, pero que ha construido una importante reputación, la cantidad de mesas podría rondar las 20 - 25 unidades.

Por otro lado, los mozos necesarios podrían ser 5, de tal forma de atender 5 mesas cada uno *(esto podría suceder un día típico de total confluencia de público)*

¿Por qué es necesario conocer estos números? Pues el restaurante tendrá que usar tablets o teléfonos para poder usar el sistema, además de una nueva caja registradora, y la respectiva base de datos asociada. Todos estos dispositivos encarecen la puesta en marcha del sistema solicitado.

Cabe destacar que estos costos pueden ser omitidos de nuestra oferta, pues queda a discreción de nuestro cliente si los adquiere a través de nuestro servicio, o por su propia cuenta. De todas formas los dispositivos ofrecidos cuentan con las especificaciones estrictamente necesarias para el óptimo funcionamiento del sistema a implementar.

Para garantizar la **alta disponibilidad** del sistema y cumplir con los **requerimientos no funcionales de disponibilidad y escalabilidad**, se implementarán las siguientes estrategias, las cuales están consideradas dentro del costo de desarrollo y soporte anual:

* **Infraestructura de Hosting en la Nube:** El sistema será desplegado en una infraestructura de **nube** (por ejemplo, AWS, Azure, Google Cloud) que permite la **replicación de servicios y bases de datos en múltiples zonas de disponibilidad**. Esto asegura que si una ubicación geográfica tiene un problema, el sistema pueda seguir operando desde otra, minimizando el riesgo de interrupciones.
* **Capacidad para Picos de Demanda:** Si bien la operación diaria puede ser manejada por un servidor eficiente, se asegurará que la infraestructura subyacente de la nube tenga la **capacidad elástica** para manejar picos de concurrencia, especialmente durante los momentos de mayor demanda de reservas (épocas festivas como Año Nuevo o Navidad). Esto podría incluir el **escalado automático de recursos** (CPU, memoria) del servidor web y de la base de datos para absorber el aumento de solicitudes sin degradación del rendimiento. No necesariamente se requerirá un balanceador de carga dedicado para el uso diario, sino que la arquitectura en la nube permitirá adaptar los recursos ante eventos de alta demanda de reservas.
* **Backups Automatizados y Replicación de Base de Datos:** Se implementarán **copias de seguridad (backups)** diarias y automatizadas de la base de datos, almacenadas en una ubicación segura y distinta a la principal. Adicionalmente, se establecerá una **replicación de la base de datos** para minimizar la pérdida de datos en caso de un incidente.
* **Plan de Recuperación ante Desastres (DRP):** Se definirá un plan de recuperación que incluye los procedimientos y tiempos estimados para restaurar el servicio en caso de una falla mayor. Esto incluye la automatización de la restauración de backups y el despliegue rápido del sistema en un entorno de contingencia. Los tiempos objetivos de recuperación (RTO) y puntos objetivos de recuperación (RPO) serán definidos en la fase de diseño técnico detallado.
* **Monitoreo Continuo:** Se establecerán herramientas de monitoreo proactivo para detectar cualquier anomalía o posible falla en la infraestructura y el sistema, permitiendo una intervención rápida para prevenir interrupciones.
* Desarrollo del proyecto: $10.000.000 (Diez millones de pesos)
* Capacitación de los usuarios del software: $500.000 (Quinientos mil pesos)
* Soporte y mantenimiento (cobrado anualmente): $700.000 (Setecientos mil pesos)
* Contingencias: $1.000.000 (Un millón de pesos).
* Dispositivos:

- x5 Tablet Lenovo Tab M11 $179.990

- x1 Gaveta Registradora $60.000

- x1 HP Eliteone 800 G4 $359.990

- x1 Pistola Lectora de Codigo de Barras 3nStar SC050 $43.050

**Total costo del proyecto** *(con dispositivos incluidos + año de soporte + capacitación de empleados):* $13.562.990 IVA incl.

**Total costo del proyecto** *(sin dispositivos):* $12.200.000 IVA incl.

**5. Planificación de actividades**

5.1 Cronograma General

Es imperativo conocer los problemas que aquejan a los clientes y trabajadores del restaurante desde su propio punto de vista, y esto lo logramos a través de las siguientes Historias de Usuario, las cuales fueron clasificadas de acuerdo al área afectada.

**Reservas:**

HU-1: Como **cliente**, quiero poder hacer una reserva de mesa fácilmente para asegurar mi lugar en el restaurante.

HU-2: Como **anfitrión**, quiero tener un sistema de reservas digitalizado para poder gestionar las reservas de manera eficiente y evitar pérdidas.

HU-3: Como **anfitrión**, quiero poder acceder rápidamente a la información de las reservas para atender a los clientes al llegar.

**Precios:**

HU-4: Como **cliente**, quiero ver los precios de los productos claramente indicados en el menú para saber cuánto voy a pagar.

HU-5: Como **cajero**, quiero tener un sistema de precios actualizado y preciso para evitar errores al cobrar a los clientes.

HU-6: Como **administrador**, quiero poder actualizar los precios de los productos de manera sencilla y rápida en el sistema.

**Roles del Personal:**

HU-7: Como **empleado del restaurante**, quiero tener claridad sobre mis responsabilidades y tareas para poder desempeñar mi trabajo de manera efectiva.

HU-8: Como **administrador**, quiero poder definir y comunicar claramente los roles y responsabilidades de cada miembro del personal para asegurar una operación fluida.

**Turnos y Personal:**

HU-9: Como **empleado del restaurante**, quiero tener un horario de trabajo formal y predecible para poder planificar mi tiempo.

HU-10: Como **administrador**, quiero poder gestionar los turnos del personal de manera eficiente para asegurar que siempre haya suficiente personal para atender a los clientes.

HU-11: Como **administrador**, quiero poder anticipar y cubrir la falta de personal para evitar afectar la calidad del servicio.

**Comandas en la Cocina:**

HU-12: Como **cocinero**, quiero recibir las comandas de manera clara y legible para poder preparar los platos correctamente y sin retrasos.

HU-13: Como **cocinero**, quiero que las comandas tengan un formato estándar para facilitar su lectura y comprensión.

HU-14: Como **mesero**, quiero poder enviar las comandas a la cocina de forma rápida y precisa para evitar errores en los pedidos.

**Procesos Generales:**

HU-15: Como **empleado del restaurante**, quiero que los procesos sean más eficientes para poder brindar un mejor servicio a los clientes.

HU-16: Como **administrador**, quiero identificar y optimizar los procesos manuales para reducir errores y mejorar la eficiencia operativa del restaurante.

Desde la lista anterior es que se han definido y priorizado las historias de usuario que serán el motor principal de los primeros Sprints a ejecutar.

5.2 Sprints Agendados

Tomando en consideración que buscamos ser una solución sólida para nuestro cliente, es que se decidió que los Sprint tengan una duración de 1 mes. Esto con el fin de poder tener un desarrollo y pruebas lo suficientemente robustas. Se busca superar las expectativas del cliente.

**Sprint 1**: Para este Sprint las historias de usuario priorizadas fueron las HU-1, HU-2, HU-3, HU-12, HU-13 y HU-14.

Tiene por objetivo la implementación de los sistemas que permiten gestionar las reservas del local, así como también el sistema que permite manejar lo relativo a las comandas por parte de los mozos.

En este Sprint las actividades a desarrollar son:

* Desarrollo de funcionalidad de gestión de reservas.
* Desarrollo de interfaz de toma de comandas.
* Integración de la base de datos de los productos disponibles.
* Pruebas de funcionamiento.

**Se pretende entregar el Módulo de Reservas y el Sistema de Comandas Digital**

**Sprint 2:** En este Sprint las historias de usuario priorizadas fueron las HU-4, HU-5, HU-6, HU-15, HU-16.

Dicha priorización tiene por objetivo la implementación de los sistemas que permiten la facturación (total a pagar) de los clientes, también del sistema que permite la comunicación entre las comandas generadas por los mozos y su posterior utilización por parte de los chefs en la cocina, o por los barman en el bar.

En este Sprint las actividades a desarrollar son:

* Desarrollo de la funcionalidad de generación de facturas (total a pagar).
* Integración con el sistema de cocina y bar para el envío automático de comandas.
* Pruebas de integración entre los sistemas.

**Se pretende entregar el Módulo de Generación de Facturas, y el Módulo del Sistema de Cocina, y la integración de este último con el Sistema de Comandas**

**6. Roles del Equipo**

* **Nuestro Cliente (Gerente General)**: Responsable de la visión del producto, priorización de las funcionalidades y toma de decisiones estratégicas.
* **Product Owner:** responsable de llevar a cabo los deseos del Cliente, ser el puente de información entre el desarrollo y el Cliente. Acepta o rechaza los desarrollos en conjunto con el Cliente.
* **Scrum Master**: Facilita la metodología ágil, asegura que el equipo cumpla con los objetivos del sprint y elimina los obstáculos.
* **Desarrolladores**: Encargados de construir las funcionalidades del sistema según los requerimientos.
* **Diseñadores UX/UI**: Crean la interfaz gráfica y aseguran que la experiencia del usuario sea óptima.
* **Equipo de pruebas**: Verifica el funcionamiento correcto del sistema y realiza pruebas de usabilidad.

**7. Sprints**

7.1

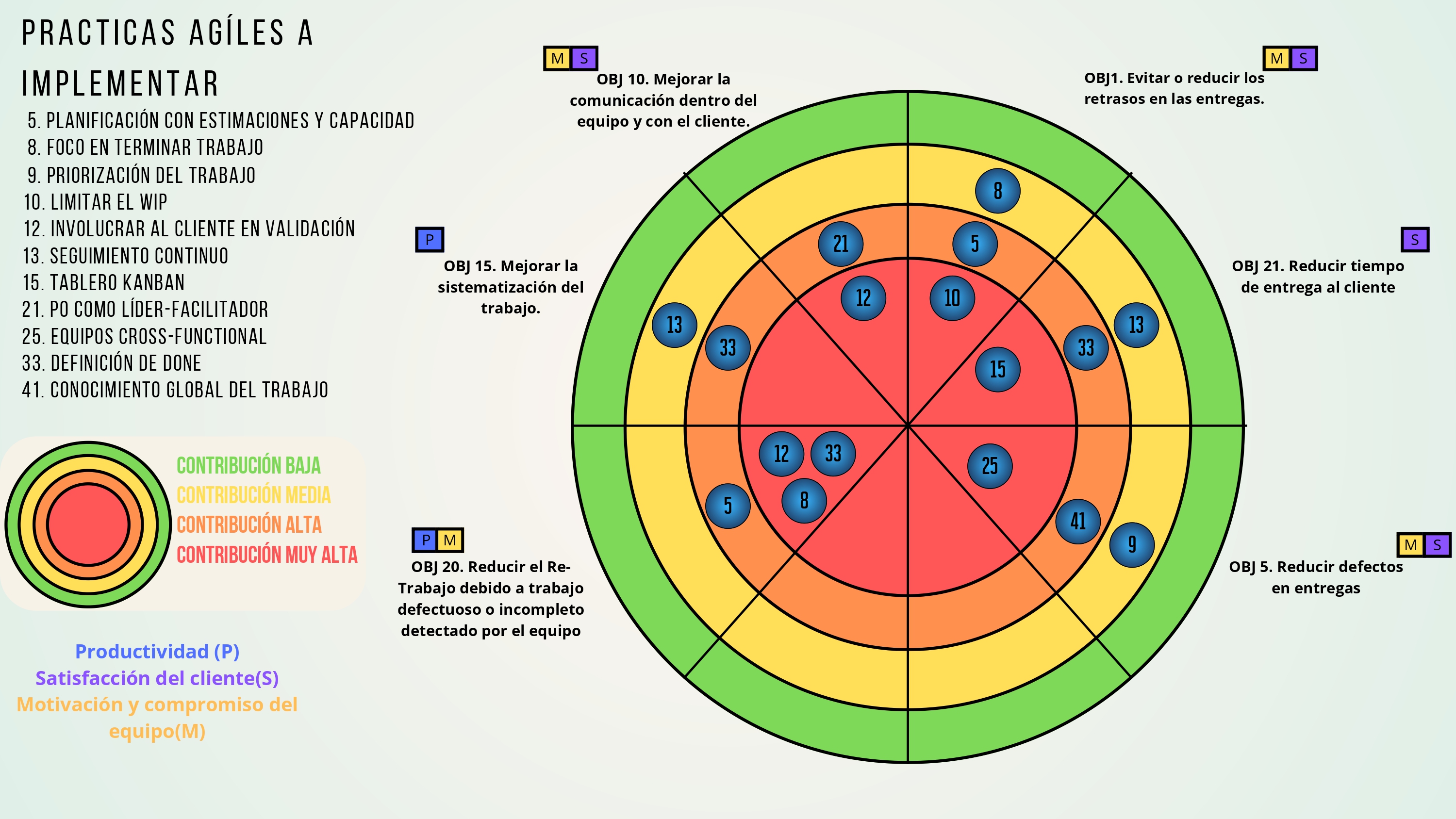
La labor dentro de un Sprint ciertos días puede resultar completamente caótica, aunque ello nunca nos priva de nuestra máxima fundamental: el compromiso con el orden. Prueba de ello es la siguiente tabla, en la que se muestra como es un día típico dentro del desarrollo de nuestra solución:

|  |  |
| --- | --- |
| **Actividad** | **Detalles** |
| Daily Scrum Meeting | Diariamente, al iniciar la jornada de trabajo, todo el equipo se reúne para informar al resto de los integrantes de la misma los avances, problemas y tareas por hacer durante el día. Se hacen transparentes los contratiempos. Mejora la comunicación. El equipo avanza.  La reunión no dura más de 15 minutos. |
| Desarrollo y Programación | Una vez terminada la reunión inicial es que los desarrolladores comienzan sus labores. |
| Revisión del Código | Cada intertanto, los desarrolladores van haciendo revisión del código de otros colaboradores, todo esto en busca de la optimización del tiempo, y con ello también mejorando la calidad del código. |
| Pruebas | El equipo de pruebas puede hacer los testeos necesarios a los desarrollos que ,ya cumpliendo con todos los parámetros solicitados por parte del PO y el Cliente, ya son candidatos a ser aceptados. |
| Retroalimentación | Al finalizar la jornada se busca que los colaboradores puedan actualizar los tableros Kanban, con el propósito de informar al resto del equipo los avances: lo hecho, lo por hacer, lo que se está haciendo. |

A continuación se muestra un ejemplo de Tablero Kanban asociado a una jornada típica. El ejemplo se ubica temporalmente en los inicios del primer Sprint:



También se añade el Menú Ágil:

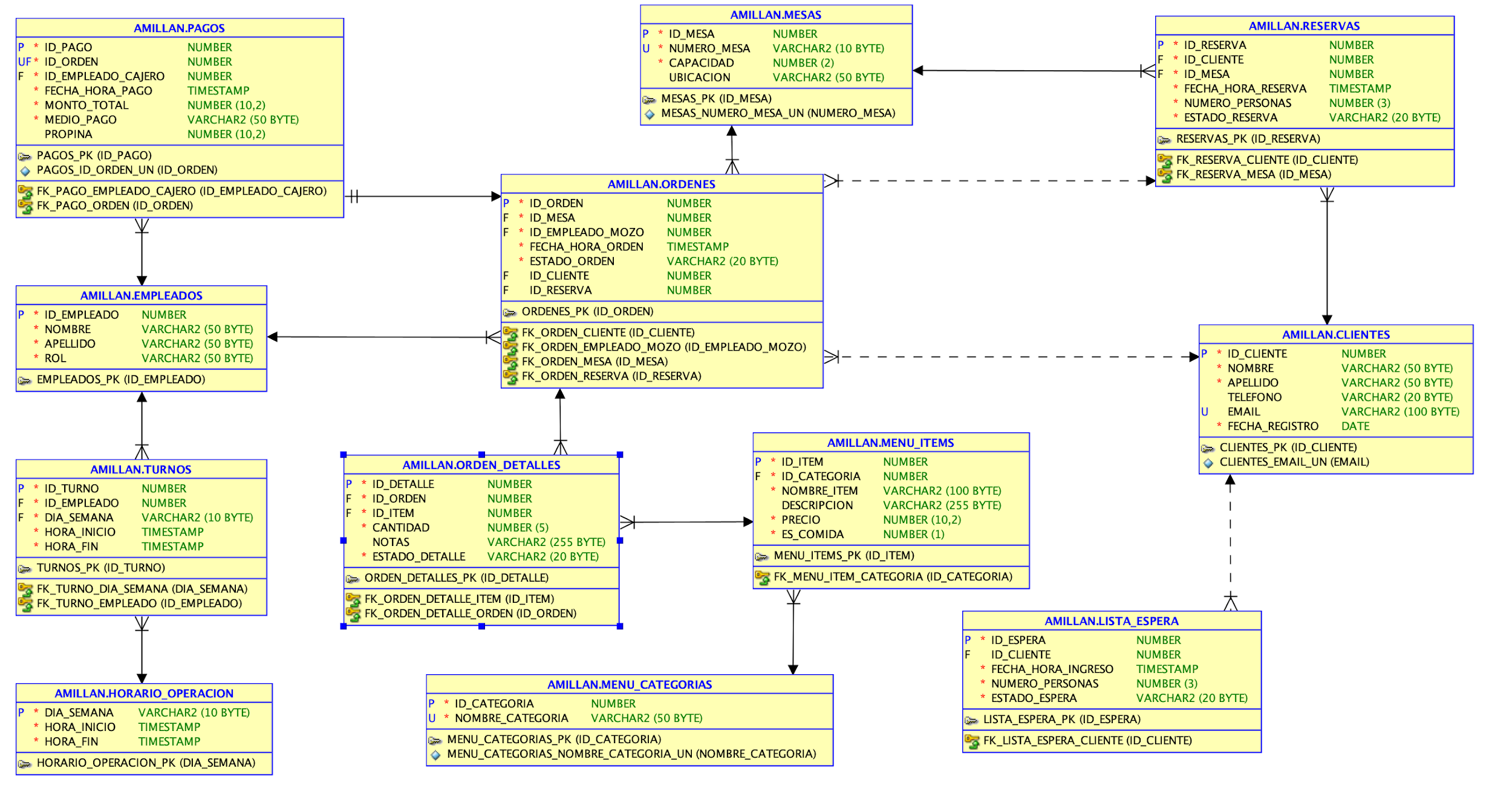


**8. Modelo Entidad Relación (MER)**

El Modelo Entidad - Relación nos da una idea respecto al funcionamiento interno del

sistema a implementar.

Gráficamente, este Modelo puede representarse de la siguiente manera:



Se pasa a explicar el funcionamiento de las entidades identificadas:

**CLIENTES**: Almacena la información de los clientes.

**MESAS**: Almacena la información de las mesas del restaurante.

**RESERVAS**: Registra las reservas realizadas por los clientes.

**LISTA\_ESPERA**: Almacena la información de los clientes en lista de espera.

**EMPLEADOS**: Almacena la información del personal del restaurante, incluyendo su rol.

**MENU\_CATEGORIAS**: Almacena las diferentes categorías del menú *(ej. Entradas, Platos Principales, Postres, Bebidas).*

**MENU\_ITEMS**: Almacena los detalles de cada ítem del menú, incluyendo si es comida o bebida.

**ORDENES**: Representa la orden general de un cliente en una mesa, atendida por un mozo. Para permitir la flexibilidad necesaria en el registro de la actividad del restaurante, la entidad ORDENES ha sido (re)diseñada para relacionarse **opcionalmente** con CLIENTES y RESERVAS. Esto significa que una orden puede ser creada para clientes que no tienen una reserva previa (clientes de "walk-in") y para los cuales su id\_cliente no es registrado, o para clientes que sí tienen una reserva. La clave foránea id\_cliente en ORDENES y la clave foránea id\_reserva en ORDENES son **nulleables**, permitiendo que una orden exista sin una vinculación obligatoria a una reserva o a un cliente registrado en el sistema. Sin embargo, siempre estará vinculada a una mesa y a un empleado (mozo) que la atiende.

**ORDEN\_DETALLES**: Almacena los ítems específicos que se incluyen en cada orden *(la comanda).*

**PAGOS:** Registra la información de los pagos realizados por los clientes.

**HORARIO\_OPERACION**: Define el horario de atención del restaurante por día de la semana.

**TURNOS**: Almacena la información de los turnos asignados a cada empleado.

Consideraciones adicionales:

**Carta Digital**: Para la carta digital, la información se tomaría directamente de la tabla MENU\_ITEMS.

**Estados**: Se han incluido campos de estado en varias tablas *(reservas, lista\_espera, ordenes)* para rastrear el progreso de cada proceso.

**Roles de Empleados**: La tabla EMPLEADOS permite identificar el rol de cada empleado, lo cual es importante para asignar responsabilidades *(mozo para tomar pedidos, cajero para procesar pagos, anfitrión para reservas y lista de espera).*

**Trazabilidad**: Las tablas ÓRDENES y PAGOS están relacionadas para poder rastrear qué pago corresponde a qué orden.

**Turnos**: Se ha agregado la tabla TURNOS para gestionar los horarios de trabajo de los empleados, vinculándola con la tabla EMPLEADOS a través de la columna `empleado\_id`.

Dentro de la estructura fue necesario crear tablas auxiliares para una mejor operación de la base.

¿Y qué pasa si una mesa disponible con capacidad de solo 2 comensales quiere ser reservada para 4 personas? Esta validación es perfectamente posible hacerla desde la lógica de la base de datos, añadiendo restricciones, pero se consideró para esta vez que dicha lógica sea trabajo de la aplicación misma, delegando a la base de datos su trabajo principal: almacenar y leer datos.

**9. Diagramas UML**

9.1 Diagrama de Casos de Uso

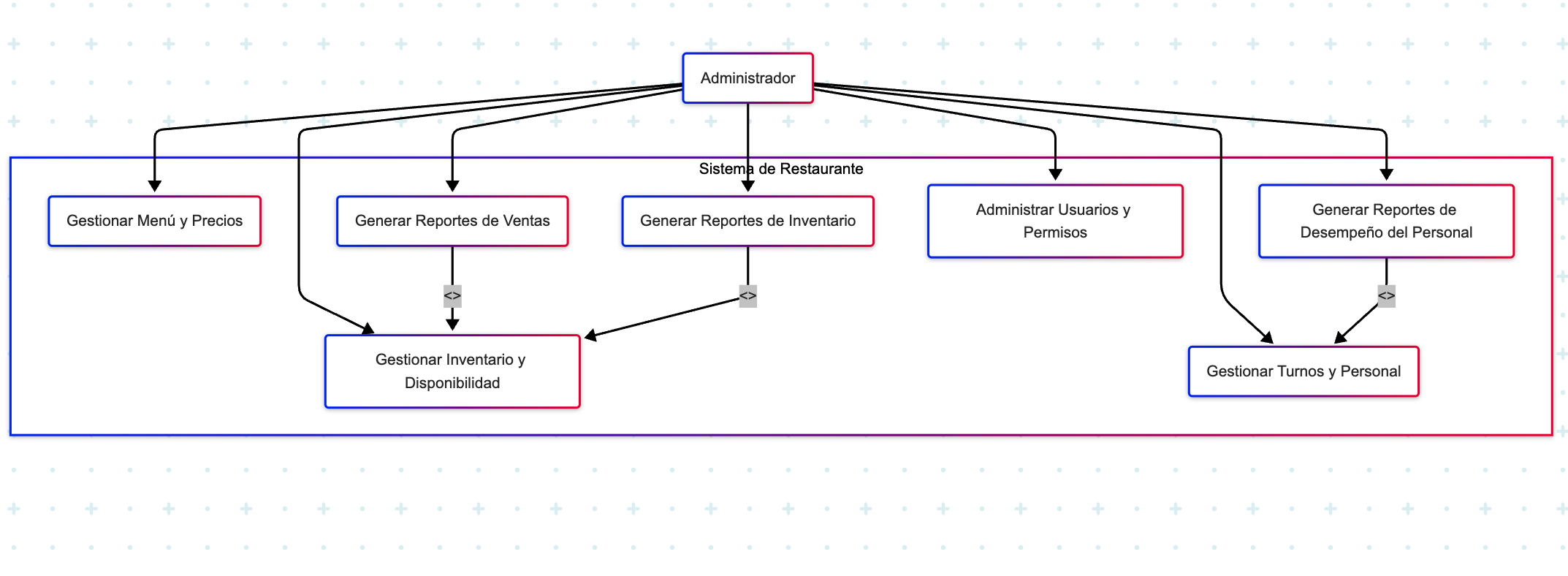
Los siguientes diagramas UML de Casos de Uso ilustran las funcionalidades clave que el sistema de gestión del restaurante 'Los Naranjos' ofrecerá a sus usuarios (actores) y cómo interactuarán con ellas. Cada caso de uso representa un servicio o una tarea que el sistema realiza para un actor específico. Para una comprensión integral del sistema, se presentan dos diagramas: uno enfocado en las **Operaciones del Cliente y del Personal Operativo**, y otro dedicado a la **Gestión Interna y Reportes** por parte del administrador.

Diagrama de Casos de Uso: operaciones del cliente y del personal operativo.



Este diagrama detalla las interacciones directas con el cliente y las tareas principales realizadas por el personal operativo del restaurante. Se identifican los siguientes actores y sus interacciones principales:

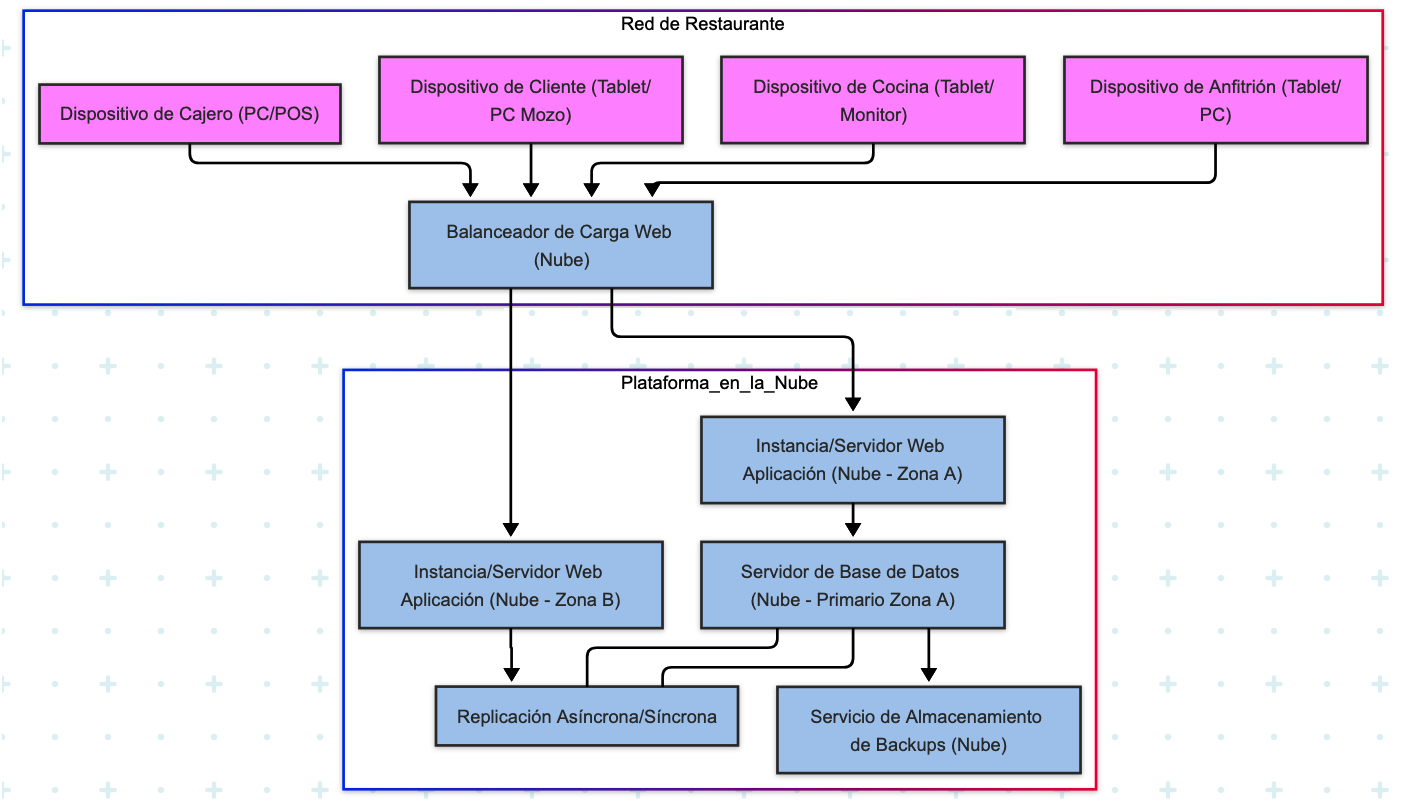
* **Cliente (o comensales):** Interactúa con las funcionalidades de Gestionar Lista de Espera y Gestionar Reservas.
* **Anfitrión:** Es responsable de Gestionar Lista de Espera y Gestionar Reservas.
* **Mozo (o barman):** Realiza Tomar Comanda y Generar Detalle de Venta/Factura.
* **Cocinero:** Recibe y procesa las Comandas a través del sistema.
* **Cajero:** Es el encargado de Generar Detalle de Venta/Factura. Los casos de uso como Tomar Comanda y Generar Detalle de Venta/Factura incluyen la interacción con el caso de uso Gestionar Inventario y Disponibilidad, lo que asegura que la disponibilidad de los productos se actualice en tiempo real y que los precios sean correctos.

Diagrama de Casos de Uso: Gestión Interna y Reportes.

Este segundo diagrama se centra en las funcionalidades administrativas, cruciales para la operación y mejora continua del restaurante. El principal actor en este ámbito es el **Administrador**, quien interactúa con los siguientes casos de uso:

* **Gestionar Menú y Precios:** Permite al administrador controlar y actualizar la oferta de productos y sus costos.
* **Gestionar Inventario y Disponibilidad:** Facilita el seguimiento del stock y la gestión de la disponibilidad de ítems.
* **Gestionar Turnos y Personal:** Apoya la organización y asignación de horarios de trabajo para los empleados.
* **Generar Reportes de Ventas:** Proporciona análisis sobre el rendimiento comercial del restaurante.
* **Generar Reportes de Inventario:** Ofrece una visión del estado actual del almacén.
* **Generar Reportes de Desempeño del Personal:** Permite evaluar la eficiencia del equipo.
* **Administrar Usuarios y Permisos:** Para la gestión de acceso y roles dentro del sistema. Estos casos de uso son vitales para la toma de decisiones estratégicas y la optimización de los recursos internos del restaurante.

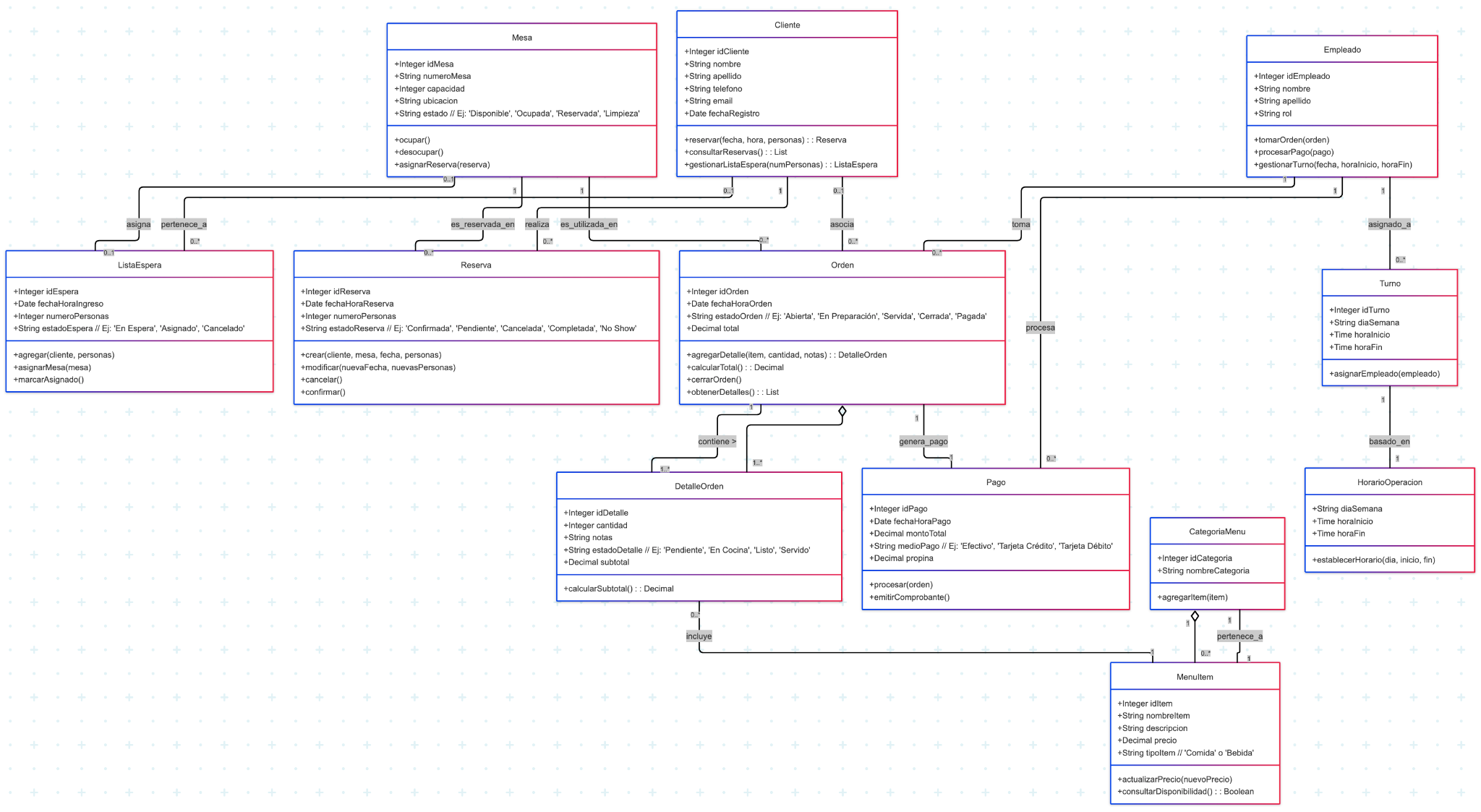
9.2 Diagrama de Despliegue



El Diagrama de Despliegue para el sistema del Restaurante 'Los Naranjos' ilustra una arquitectura basada en la **Plataforma en la Nube**, diseñada para garantizar **alta disponibilidad, escalabilidad y resiliencia**. Esta arquitectura se compone de los siguientes nodos y componentes:

1. **Dispositivos de Usuario Final:** Representan los puntos de acceso del personal del restaurante al sistema. Estos incluyen:  
   * **Dispositivo de Cliente (Tablet/PC Mozo):** Utilizados por los mozos para tomar comandas y gestionar órdenes en las mesas.
   * **Dispositivo de Cocina (Tablet/Monitor):** Para la visualización de comandas y gestión de la preparación de platillos.
   * **Dispositivo de Anfitrión (Tablet/PC):** Para la gestión de reservas y lista de espera.
   * **Dispositivo de Cajero (PC/POS):** Para procesar pagos y emitir boletas/facturas. Estos dispositivos se conectan a la aplicación web a través de la red del restaurante y la infraestructura de la nube.
2. **Balanceador de Carga Web (Nube):** Es el punto de entrada principal para todas las solicitudes entrantes desde los dispositivos de usuario. Su función es distribuir el tráfico de manera eficiente entre las diferentes instancias de la aplicación web. Esto es vital para:  
   * **Escalabilidad Horizontal:** Permite añadir o quitar instancias de servidor de aplicaciones según la demanda.
   * **Alta Disponibilidad:** Si una instancia de la aplicación web falla, el balanceador de carga redirige automáticamente las solicitudes a las instancias saludables, evitando interrupciones en el servicio.
3. **Servidor Web de Aplicación (Nube - Instancias Replicadas):** Representa el entorno donde reside la lógica de negocio del sistema (front-end y API REST/back-end). Se despliega en **múltiples instancias replicadas** (ej. en diferentes Zonas de Disponibilidad de la nube). Esta redundancia es clave para la tolerancia a fallos; si una instancia deja de funcionar, las otras continúan sirviendo las solicitudes, minimizando el tiempo de inactividad.
4. **Servidor de Base de Datos (Nube - Primario y Réplica):** La persistencia de los datos del restaurante se gestiona a través de una base de datos desplegada en la nube (ej. Oracle Cloud Database, RDS). Se configura con:  
   * **BD Oracle Primaria:** La instancia principal que maneja las operaciones de lectura y escritura.
   * **BD Oracle Réplica:** Una copia idéntica de la base de datos primaria, mantenida en sincronía (usualmente en una zona de disponibilidad diferente) mediante **replicación asíncrona/síncrona**. Esta réplica es crucial para la recuperación ante desastres (DRP), permitiendo una rápida conmutación por error (failover) en caso de que la instancia primaria falle, asegurando la continuidad del negocio y minimizando la pérdida de datos.
5. **Servicio de Almacenamiento de Backups (Nube):** Se utiliza un servicio de almacenamiento dedicado en la nube (ej. S3 en AWS, Blob Storage en Azure) para realizar **copias de seguridad automatizadas y periódicas** de la base de datos. Este nodo garantiza la recuperación total del sistema y los datos en caso de una corrupción de datos, un fallo masivo o un desastre, siendo un componente fundamental del Plan de Recuperación ante Desastres.

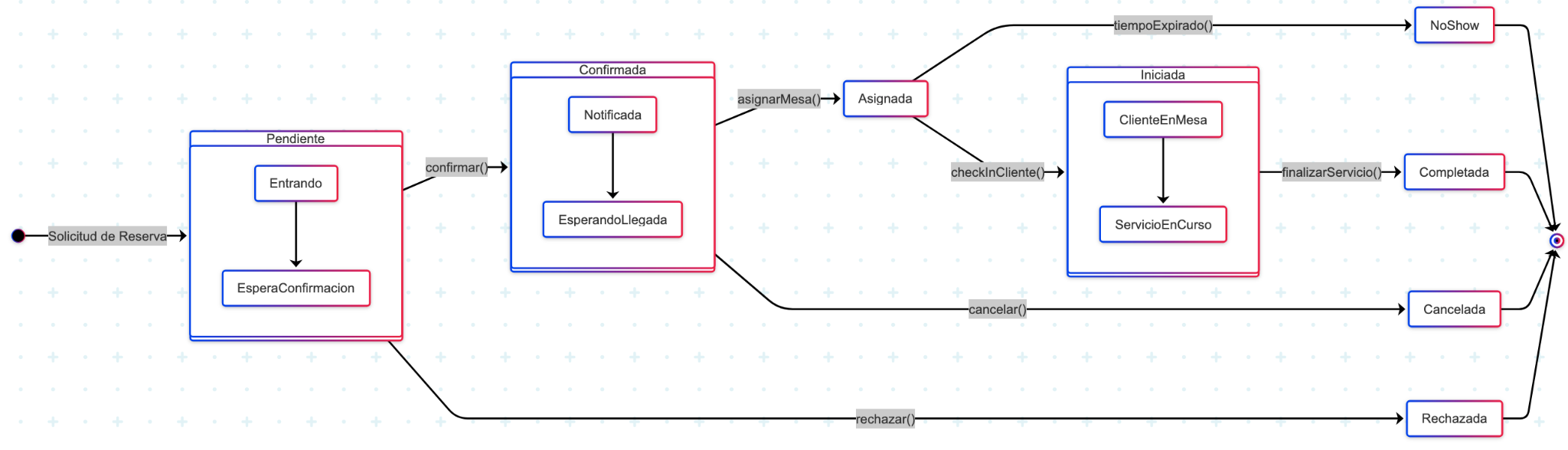
9.3 Diagrama de Clases



El Diagrama de Clases modela las entidades clave del negocio del restaurante y sus interacciones en el contexto del software. Se deriva directamente del Modelo Entidad-Relación (MER) previamente definido, transformando las entidades de datos en clases con comportamientos (o métodos) asociados. Las clases más relevantes y sus relaciones son:

1. **Cliente:** Representa a los usuarios del restaurante. Puede realizar múltiples Reservas y también asociar sus Ordenes. Se incluye la capacidad de reservar, consultarReservas y gestionarListaEspera.
2. **Empleado:** Modela a los diferentes tipos de personal (mozo, cajero, etc.). Un Empleado puede tomarOrdenes y procesarPagos. También se le puede asignar a un Turno.
3. **Mesa:** Representa las mesas físicas del restaurante. Una Mesa puede ser\_reservada\_en múltiples Reservas y ser\_utilizada\_en múltiples Ordenes. Incluye métodos para cambiar su estado (ocupar/desocupar).
4. **Reserva:** Encapsula la información de una reserva específica para una Mesa por un Cliente. Incluye métodos para crear, modificar y cancelar la reserva, y para cambiar su estado.
5. **ListaEspera:** Gestiona a los clientes sin reserva. Permite agregar clientes y asignarMesa cuando esté disponible.
6. **Orden:** La clase central que representa una comanda completa de un cliente en una Mesa tomada por un Mozo. Puede contener uno o más DetalleOrden (relación de composición) y genera\_pago. Incluye métodos para agregarDetalle, calcularTotal y cerrarOrden.
7. **DetalleOrden:** Representa un ítem específico dentro de una Orden. Incluye un MenuItem y registra la cantidad y notas. Calcula su subtotal.
8. **MenuItem:** Define los elementos del menú del restaurante. Pertenece\_a una CategoriaMenu. Incluye atributos como precio y métodos para actualizarPrecio.
9. **CategoriaMenu:** Agrupa los MenuItems por tipo (ej. "Entradas", "Platos Fuertes", "Bebidas").
10. **Pago:** Registra los detalles de una transacción de pago asociada a una Orden. Permite procesar el pago y emitirComprobante.
11. **Turno:** Gestiona los turnos laborales de los Empleados, basado\_en un HorarioOperacion.
12. **HorarioOperacion:** Define los horarios de apertura y cierre del restaurante para cada diaSemana.

9.4 Diagrama de Estados

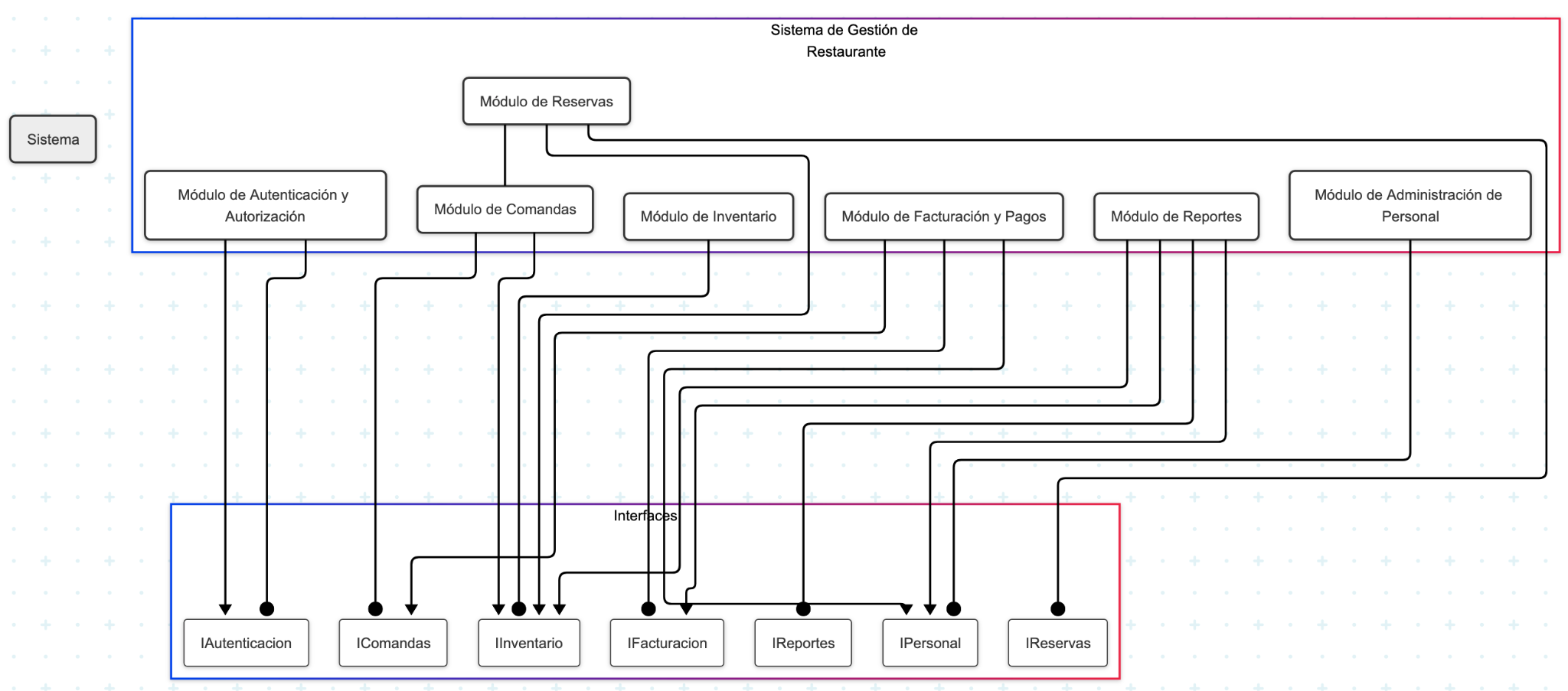


El Diagrama de Estados para la entidad Reserva detalla las diferentes fases por las que atraviesa una reserva desde su origen hasta su finalización. Este modelo es crucial para gestionar de manera efectiva la disponibilidad de mesas y la experiencia del cliente. Los estados y sus transiciones principales son:

* **Inicial ([\*])**: Punto de partida del ciclo de vida de una reserva, indicando que una Solicitud de Reserva ha sido creada.
* **Pendiente**: Este es el estado inicial de una reserva después de ser solicitada. Internamente, puede tener sub-estados como:
  + **Entrando**: La reserva acaba de ser registrada.
  + **EsperaConfirmacion**: La reserva está a la espera de la validación y confirmación por parte del personal del restaurante.
  + Desde Pendiente, la reserva puede transicionar a Confirmada (si es aceptada) o Rechazada (si no se puede atender).
* **Confirmada**: La reserva ha sido aceptada por el restaurante. Esto implica que la mesa está provisionalmente asignada a la reserva para la fecha y hora indicadas. Puede tener sub-estados que indican el seguimiento:
  + **Notificada**: Se ha enviado una confirmación al cliente.
  + **EsperandoLlegada**: Se aguarda la presencia del cliente en la fecha y hora de la reserva.
  + Desde Confirmada, puede pasar a Asignada (cuando se le asigna la mesa específica) o ser Cancelada por el cliente o el restaurante.
* **Asignada**: Una mesa física ha sido formalmente vinculada a la reserva confirmada. Desde este estado, la reserva puede transicionar a Iniciada (si el cliente llega y ocupa la mesa) o a NoShow (si el cliente no se presenta en el tiempo establecido).
* **Iniciada**: El cliente ha llegado al restaurante y ocupado la mesa reservada, y el servicio está en curso. Se pueden considerar sub-estados como:
  + **ClienteEnMesa**: El cliente ha sido recibido y está sentado.
  + **ServicioEnCurso**: El mozo está tomando la comanda o el cliente ya está siendo atendido.
  + Desde Iniciada, la reserva transiciona a Completada una vez que el servicio finaliza.
* **Completada**: La reserva ha finalizado exitosamente, el cliente ha sido atendido y el servicio ha concluido.
* **Cancelada**: La reserva ha sido anulada por el cliente o por el restaurante antes de ser iniciada. Es un estado final.
* **Rechazada**: La solicitud de reserva inicial no pudo ser aceptada por el restaurante debido a falta de disponibilidad u otras razones. Es un estado final.
* **NoShow**: El cliente no se presentó a su reserva dentro del tiempo de gracia. Es un estado final.
* **Final ([\*])**: El punto final del ciclo de vida de la reserva, una vez que ha sido completada, cancelada, rechazada o marcada como NoShow.

Este diagrama de estados asegura que el sistema maneje el ciclo de vida de las reservas de manera consistente y controlada, reflejando las políticas de negocio del restaurante.

9.5 Diagrama de Componentes

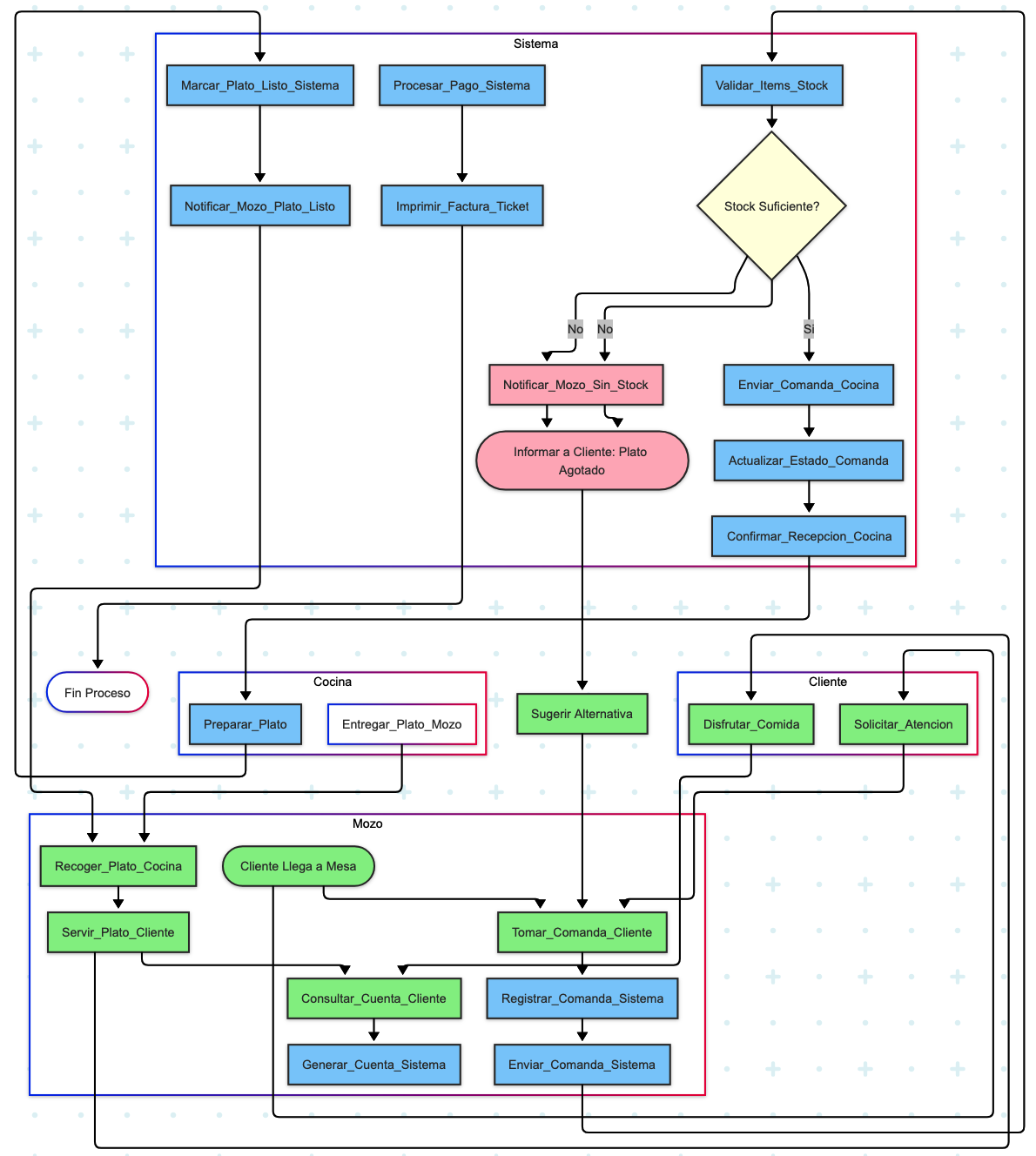


El diagrama de componentes del sistema de gestión del Restaurante 'Los Naranjos' detalla la arquitectura modular del software, dividiéndolo en subsistemas lógicos que interactúan entre sí. Esta estructura fomenta la independencia de los módulos y la reutilización de funcionalidades. Los componentes clave y sus interacciones son:

1. **Sistema de Gestión de Restaurante (Contenedor General):** Representa el sistema completo, englobando todos los módulos funcionales.
2. **Módulo de Reservas:** Responsable de toda la lógica relacionada con la creación, modificación, consulta y cancelación de reservas.
   * **Interfaz Provista:** IReservas (ej. crearReserva(), consultarDisponibilidadMesas()).
   * **Dependencia:** Requiere el Módulo de Inventario para consultar la disponibilidad de mesas.
3. **Módulo de Comandas:** Gestiona el ciclo de vida de los pedidos (comandas), desde la toma hasta el envío a cocina y el seguimiento.
   * **Interfaz Provista:** IComandas (ej. tomarComanda(), actualizarEstadoItem()).
   * **Dependencia:** Requiere el Módulo de Inventario para verificar la disponibilidad de productos y actualizar el stock.
4. **Módulo de Inventario:** Encargado de la gestión de stock de productos, menú y disponibilidad de mesas. Es un componente central que provee servicios a otros módulos.
   * **Interfaz Provista:** IInventario (ej. consultarStock(), actualizarStock(), consultarDisponibilidadMesa()).
5. **Módulo de Facturación y Pagos:** Procesa los pagos de las órdenes, genera facturas y gestiona los diferentes métodos de pago.
   * **Interfaz Provista:** IFacturacion (ej. procesarPago(), generarFactura()).
   * **Dependencia:** Requiere del Módulo de Comandas para obtener los detalles de la orden a facturar, y del Módulo de Administración de Personal para obtener datos del empleado que procesa el pago.
6. **Módulo de Reportes:** Genera informes estadísticos y operativos clave para la toma de decisiones (ventas, inventario, desempeño).
   * **Interfaz Provista:** IReportes (ej. generarReporteVentas(), generarReporteInventario()).
   * **Dependencia:** Requiere datos de los módulos de Facturación y Pagos, Inventario y Administración de Personal.
7. **Módulo de Administración de Personal:** Gestiona la información de los empleados, sus roles, permisos y turnos.
   * **Interfaz Provista:** IPersonal (ej. consultarEmpleado(), asignarTurno()).
8. **Módulo de Autenticación y Autorización:** Componente transversal que gestiona la seguridad de acceso al sistema, validando usuarios y asignando permisos basados en sus roles.
   * **Interfaz Provista:** IAutenticacion (ej. login(), verificarPermisos()).
   * **Dependencia:** Prácticamente todos los demás módulos y el sistema en general requieren de sus servicios para controlar el acceso.

Esta arquitectura de componentes favorece un desarrollo distribuido, donde cada equipo puede trabajar en un módulo de forma independiente, y la integración se realiza a través de las interfaces bien definidas. Esto conduce a un sistema más escalable, fácil de mantener y con alta cohesión interna y bajo acoplamiento entre sus partes."

9.6 Diagrama de Actividad



El Diagrama de Actividad ilustra el flujo completo del proceso de toma y gestión de una comanda dentro del sistema de gestión del Restaurante 'Los Naranjos', destacando las interacciones entre los actores clave: **Cliente**, **Mozo**, **Sistema** y **Cocina**.

El proceso se inicia cuando el **Cliente Llega a Mesa** y procede a **Solicitar Atención** al **Mozo**. A partir de ahí, el flujo es el siguiente:

* **Toma de Comanda (Mozo):** El **Mozo** procede a **Tomar Comanda del Cliente**, registrando los ítems solicitados.
* **Registro en Sistema (Mozo -> Sistema):** Una vez tomada, el **Mozo** la **Registra Comanda en Sistema**, que a su vez se encarga de **Enviar Comanda a Sistema** (paso de transición).
* **Validación de Stock (Sistema):** El **Sistema** automáticamente **Valida Ítems y Stock** de los productos solicitados.
* **Decisión de Stock (Sistema):** Se llega a una **Decisión** ({Stock Suficiente?}).
  + **Si (Sí hay Stock):** El **Sistema** procede a **Enviar Comanda a Cocina**, **Actualizar Estado Comanda** a 'En Preparación' y **Confirmar Recepción Cocina**.
  + **No (No hay Stock):** El **Sistema** **Notifica Mozo Sin Stock**. Luego, el **Mozo** debe **Informar a Cliente: Plato Agotado** y **Sugerir Alternativa**. Una vez el cliente elige, el proceso regresa a **Tomar Comanda del Cliente** para registrar la nueva selección.
* **Preparación del Plato (Cocina):** Una vez la comanda es recibida en **Cocina**, el personal procede a **Preparar Plato**.
* **Plato Listo y Notificación (Cocina -> Sistema -> Mozo):** Al finalizar, la **Cocina** **Marca Plato Listo en Sistema**. El **Sistema** entonces **Notifica Mozo Plato Listo**.
* **Servicio al Cliente (Mozo -> Cliente):** El **Mozo** **Recoge Plato Cocina** y lo **Sirve Plato Cliente**.
* **Disfrute y Solicitud de Cuenta (Cliente):** El **Cliente** procede a **Disfrutar Comida**. Al finalizar, el **Cliente** puede **Consultar Cuenta Cliente**.
* **Generación y Procesamiento de Cuenta (Mozo -> Sistema):** El **Mozo** procesa esta solicitud en el sistema para **Generar Cuenta Sistema**. El **Sistema** luego **Procesa Pago Sistema** y **Imprime Factura/Ticket**.
* **Fin del Proceso:** La impresión de la factura/ticket marca el **Fin del Proceso** de la comanda.

**10. Definición de los estándares de calidad**

La calidad del software a construir es algo que debe estar presente desde el principio del ciclo de desarrollo de cada uno de los proyectos, pues estos son las piedras angulares de los buenos cimientos que garantizan una buena durabilidad y buena performance del software a lo largo de todo su ciclo de vida. Esta calidad no debe ser vista únicamente como el fin de entregar un producto de calidad, sino que también permite entregar la confianza necesaria a la compañía que lo construyó, siendo siempre un producto bien hecho la mejor publicidad que una empresa puede tener.

La calidad también se expresa en un sistema que está documentado.

A continuación, se exponen estos estándares de calidad aplicados al caso Los Naranjos. Para el sistema construido se priorizaron los siguientes:

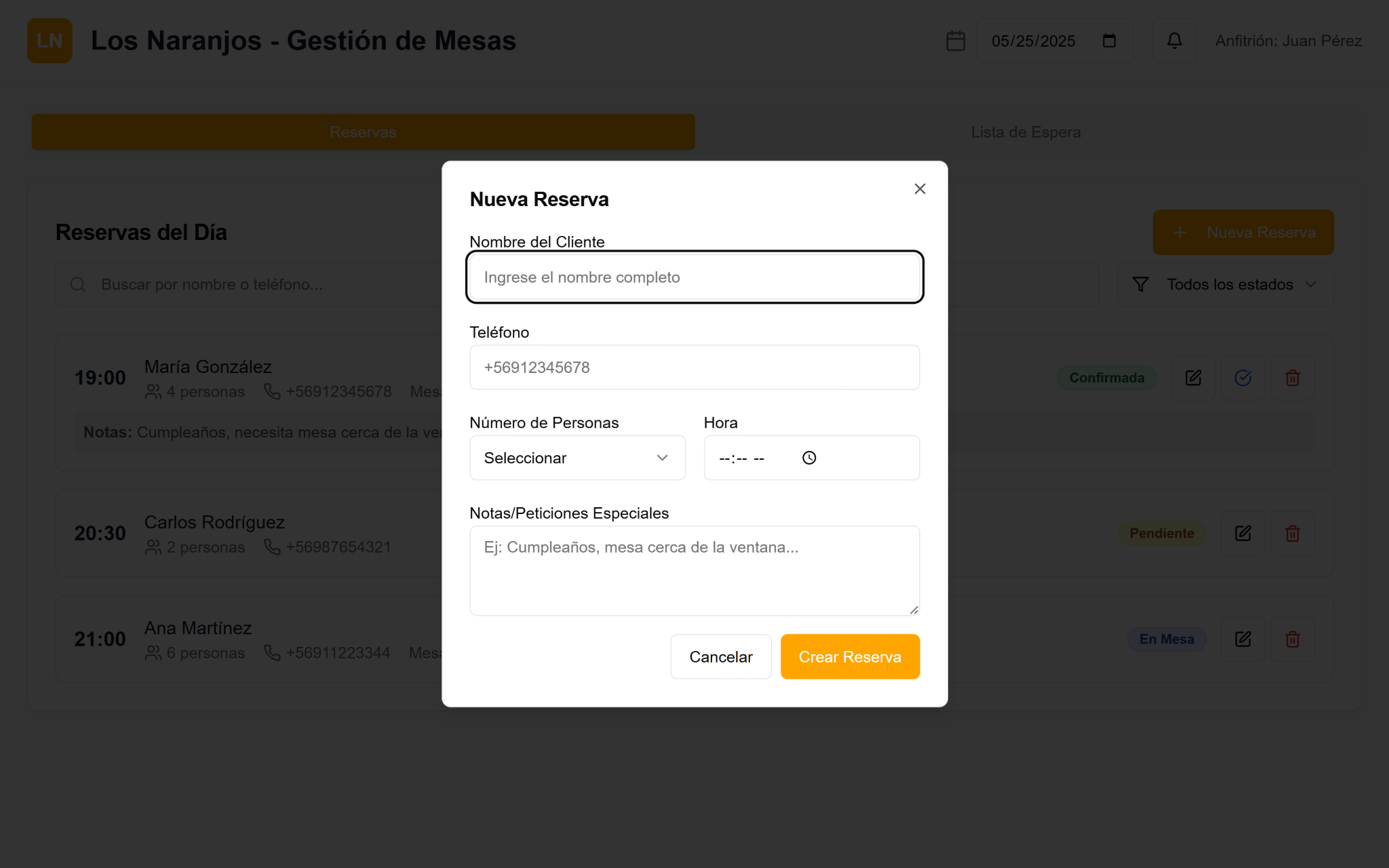
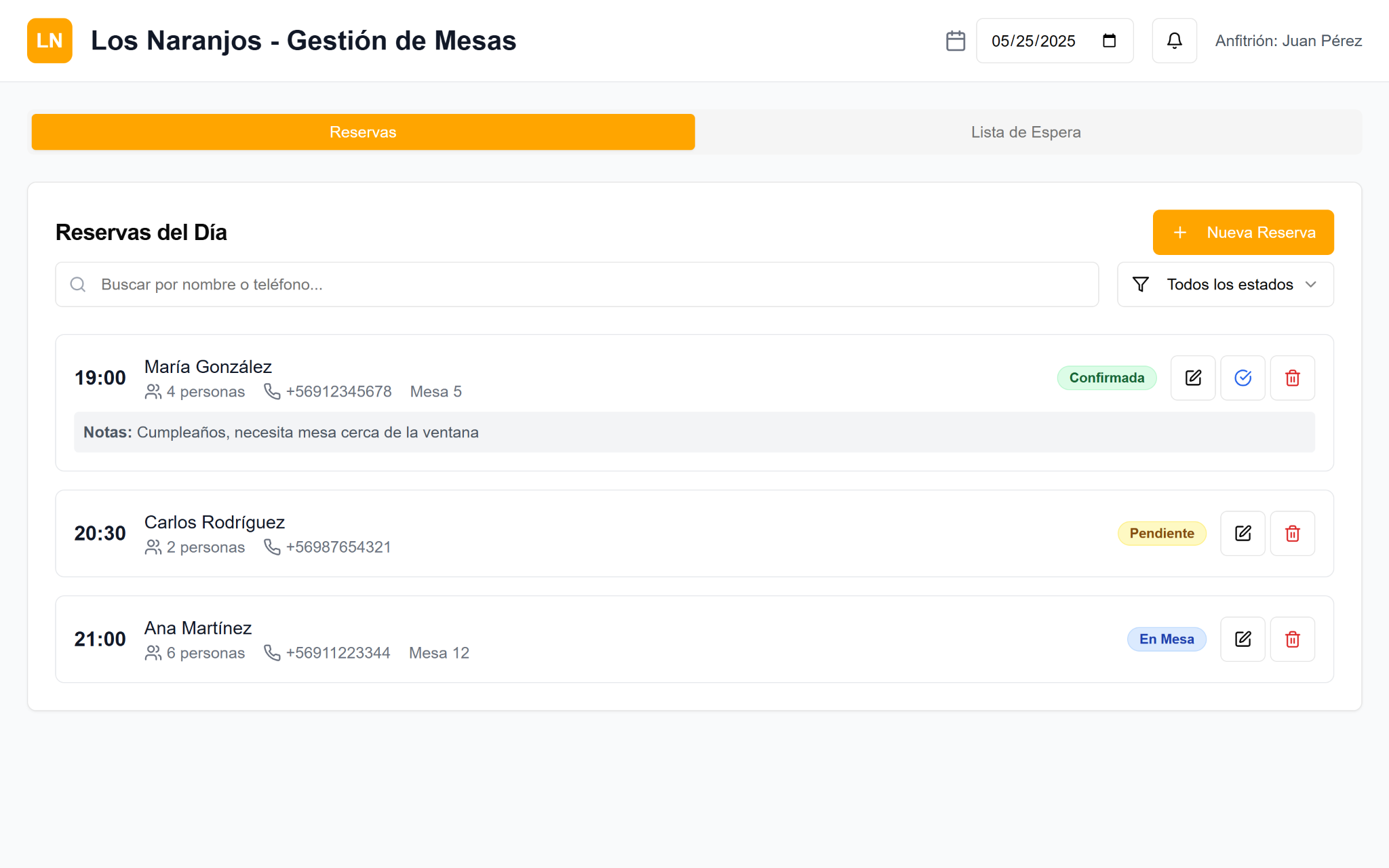
* **Modularidad:** El sistema se descompone en componentes lógicos y funcionales (como se observa en el Diagrama de Componentes). Esto permite que cada módulo sea desarrollado, probado y mantenido de forma independiente, reduciendo la complejidad general.
* **Cohesión (Alta Cohesión):** Los elementos dentro de cada módulo o clase están fuertemente relacionados entre sí y trabajan para un propósito único bien definido. Por ejemplo, el Módulo de Reservas encapsula toda la lógica de reservas, lo que facilita su comprensión y modificación.
* **Acoplamiento (Bajo Acoplamiento):** Se reducen las dependencias entre los diferentes módulos y las clases. Los componentes interactúan a través de interfaces definidas, con ello bajando el impacto de los cambios en un componente sobre los otros, y facilitando la reutilización.
* **Abstracción:** Se ocultan los detalles de implementación, exponiendo solo la información y funciones esenciales. Esto hace más simple el uso de los componentes y mejora la legibilidad del código.
* **Encapsulación:** Se agrupan los datos (atributos) y las operaciones (métodos) que actúan sobre esos datos dentro de una misma clase, protegiendo la integridad de la información y controlando el acceso externo a ella.
* **Reusabilidad:** El diseño se enfoca en crear componentes que sean genéricos y bien definidos, y así puedan ser utilizados en diferentes partes del sistema o incluso en futuros proyectos, lo que optimiza el tiempo y los recursos de desarrollo.
* **Tolerancia a Fallos y Resiliencia:** El diseño incorpora mecanismos para manejar errores y recuperarse de fallos de manera elegante (ej. balanceadores de carga, replicación de bases de datos, manejo de excepciones), como se evidencia en el Diagrama de Despliegue.
* **Escalabilidad:** La arquitectura y el diseño se conciben para permitir que el sistema pueda crecer y manejar un mayor volumen de usuarios, datos o transacciones sin una degradación significativa del rendimiento, y esto es posible apreciarlo, por ejemplo, en el diseño modular y uso de servicios en la nube.
* **Mantenibilidad:** El diseño se considera claro y consistente, lo que facilita futuras modificaciones, correcciones de errores y la evolución del sistema a lo largo del tiempo.
* **Testabilidad:** El diseño modular y de bajo acoplamiento facilita la creación de pruebas unitarias, de integración y de sistema, asegurando que cada componente funcione correctamente de forma aislada y en conjunto.

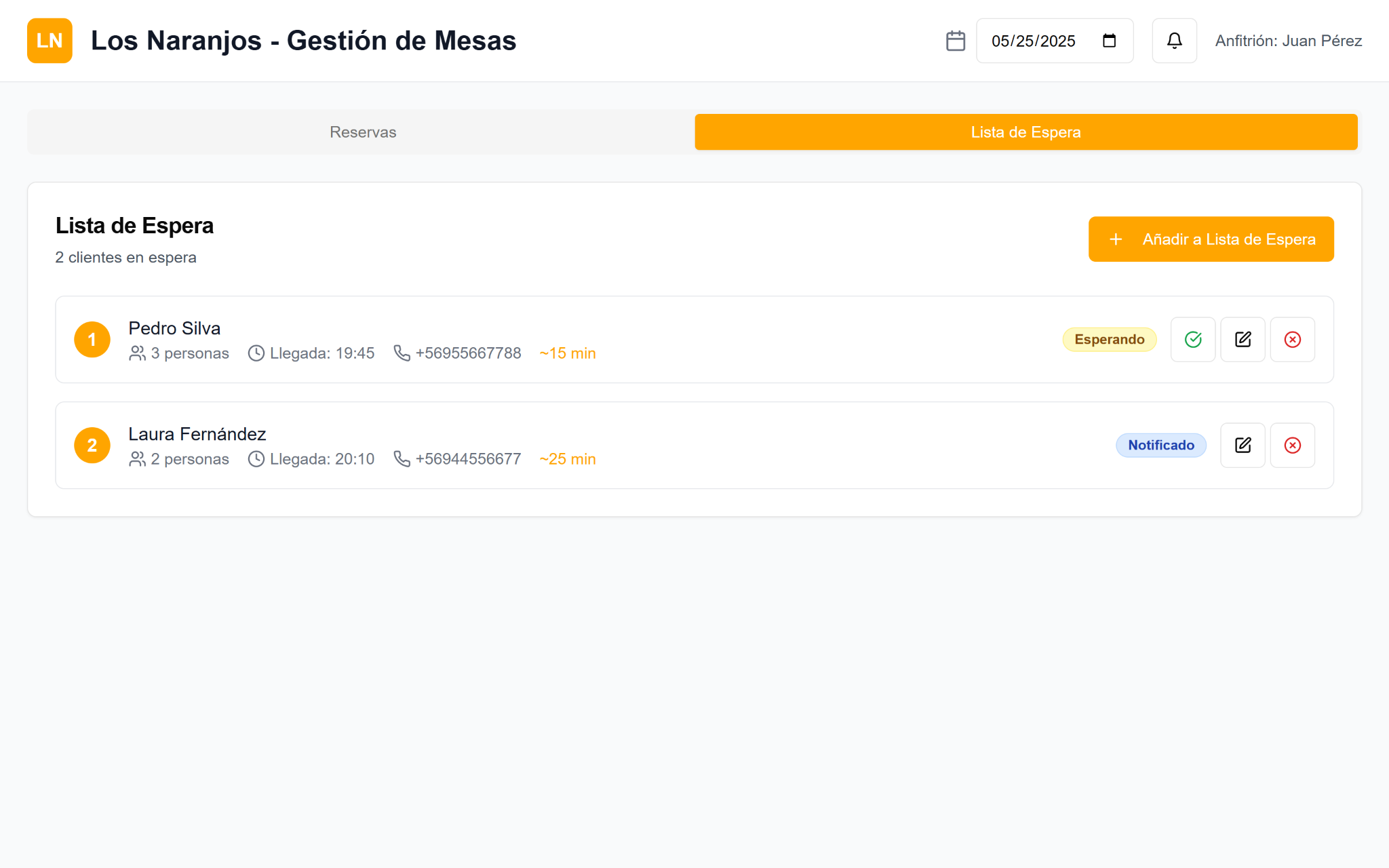
Es importante resaltar que este informe incluye descripciones de cada uno de los diagramas incluidos, lo que consideramos apropiado en pro de una buena documentación de cada uno de los componentes del sistema. Aquello es un paso hacia la mejora en la calidad que se exige para un sistema de este peso.

Todo lo anterior no hace más que reafirmar el compromiso con la calidad y las buenas prácticas.

**11. Mockups**

11.1 Vista Gestión de Reservas y Lista de Espera



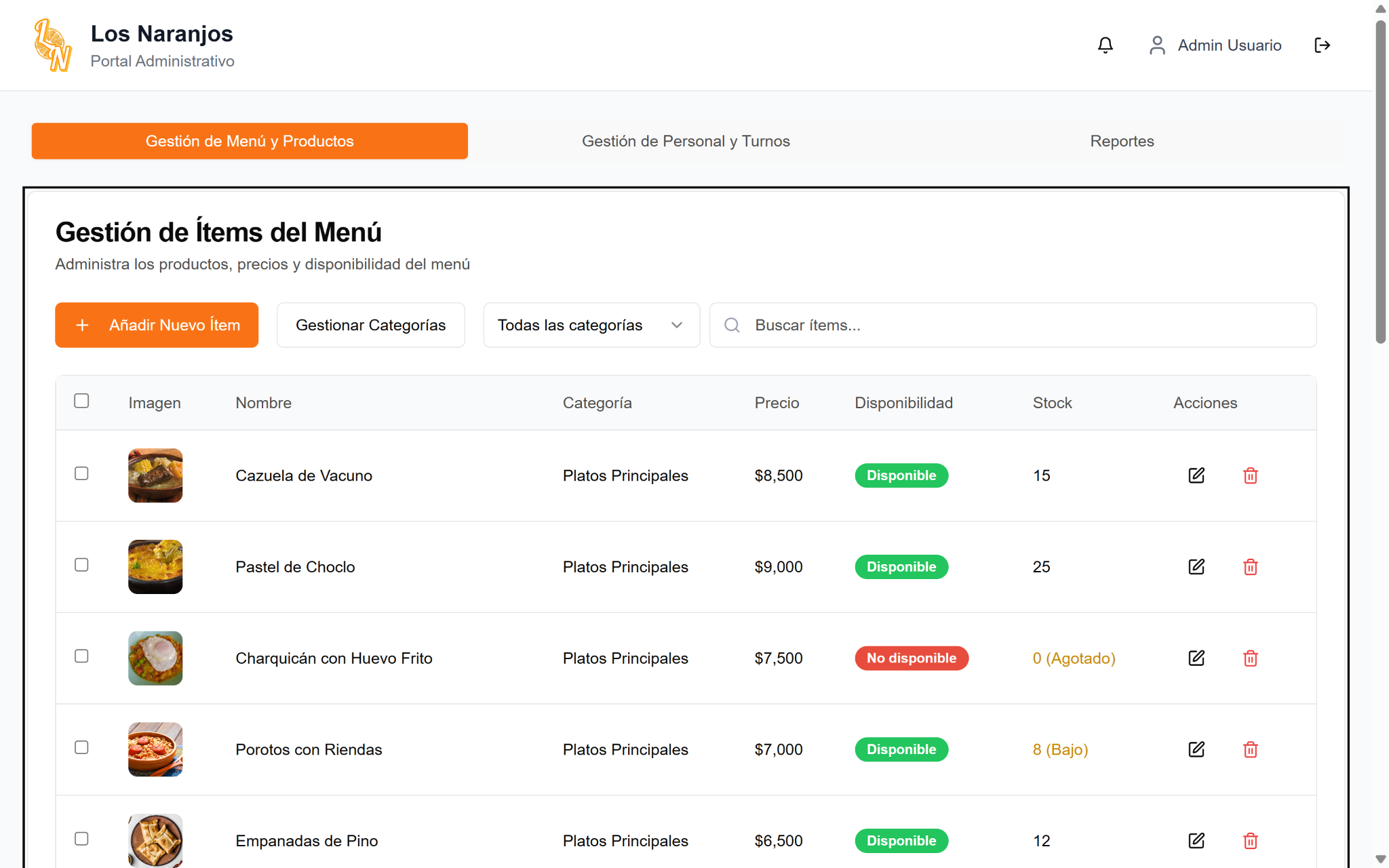


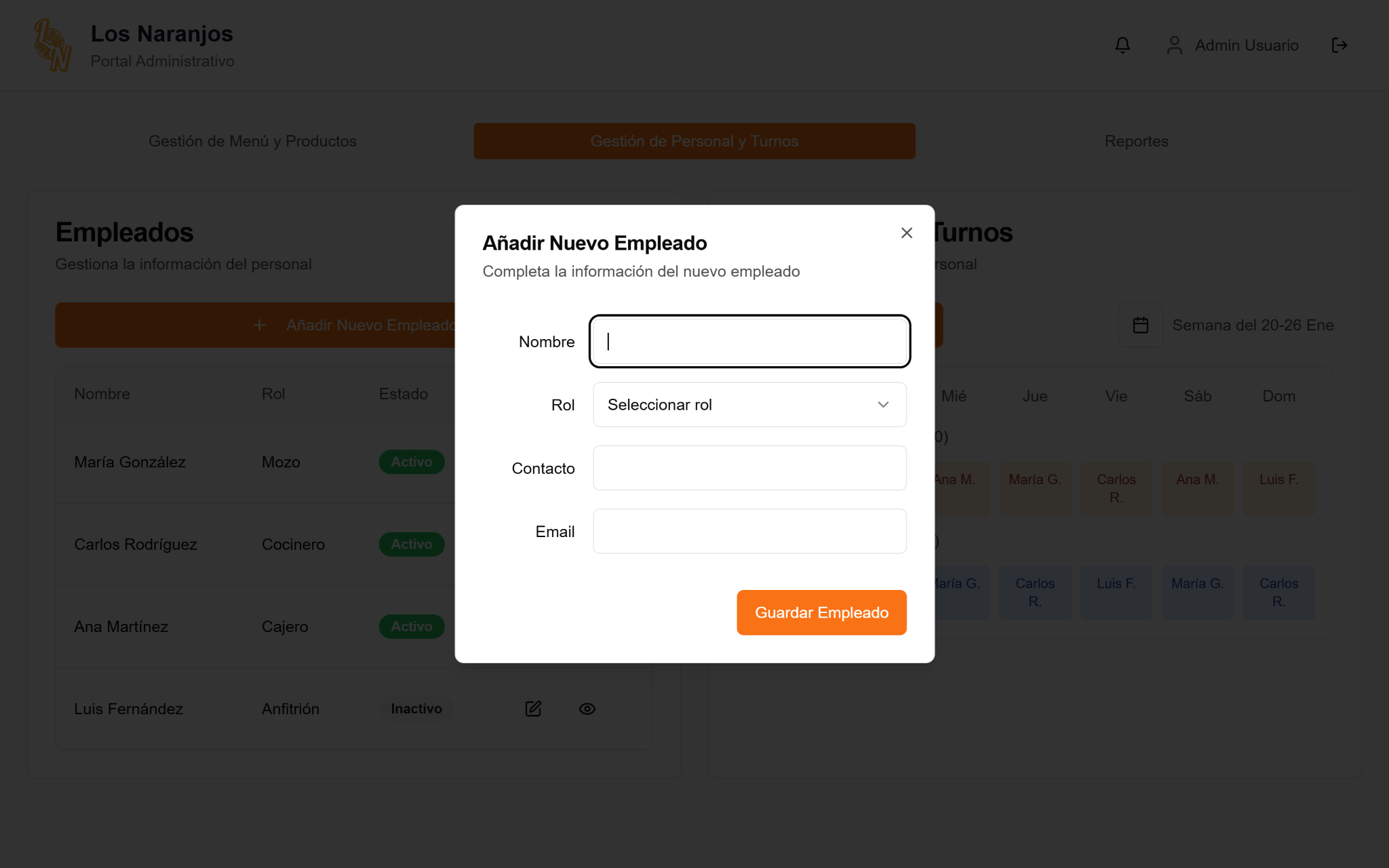
11.2 Vista Gestión de Mesas

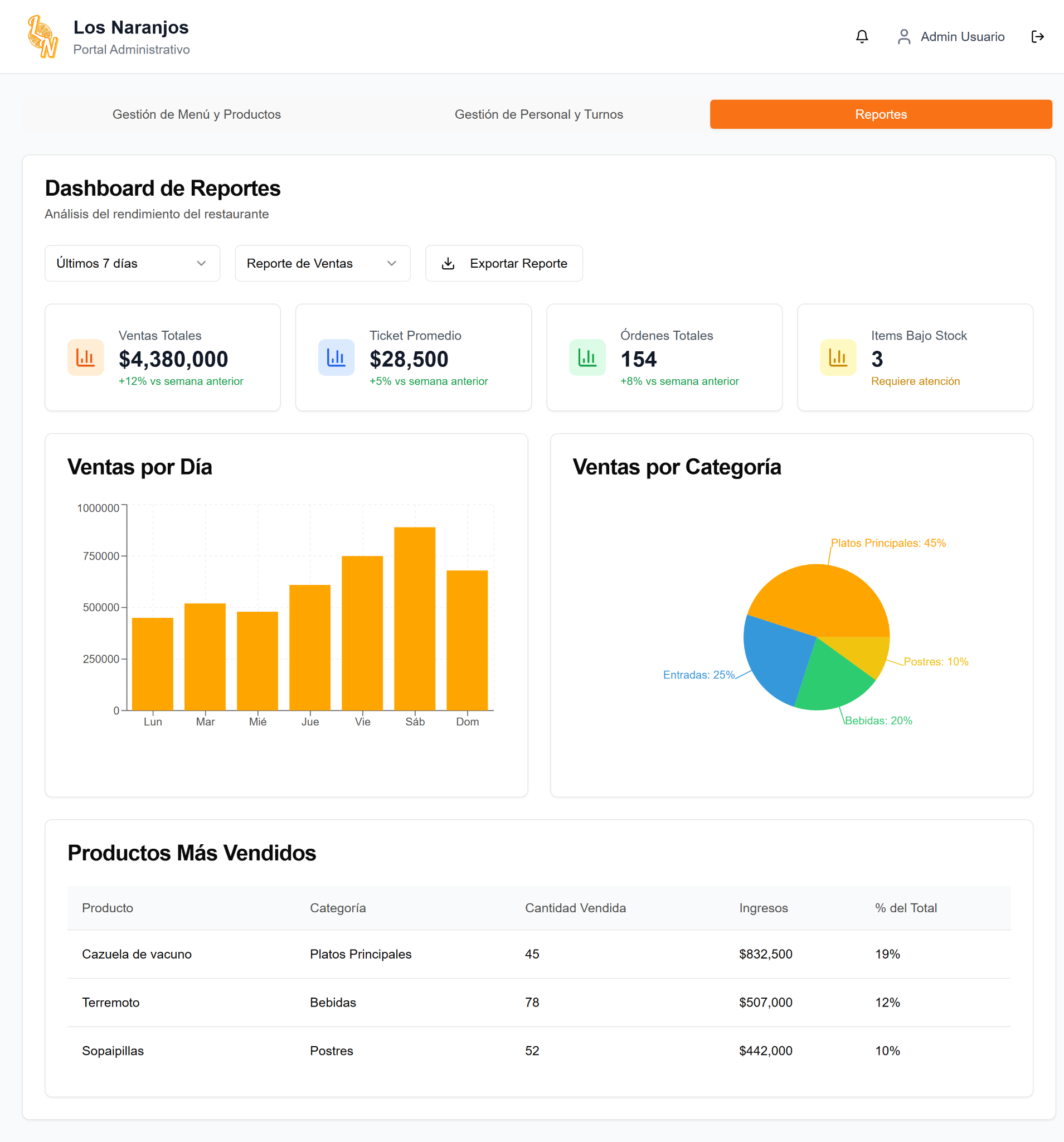




11.2 Vista Portal Administrativo







**12. Conclusiones**

La implementación de un nuevo sistema para el Restaurante Los Naranjos supone un reto en muchos sentidos y puntos de vista. No es solamente la búsqueda de la mejora en cuanto a los procesos internos de un local de comida, sino que también implica la reestructuración completa de la forma de trabajar de los empleados, por lo tanto ellos también deben ser partícipes de las soluciones, pues son ellos quienes usarán las soluciones propuestas.

El apego a los distintos métodos de representación de la estructura interna del proyecto nos lleva a la mejora del producto mismo, con lo cual podemos asegurar la calidad interna (la que vemos quienes desarrollamos el producto), y la calidad de cara a nuestro cliente Los Naranjos, permitiendo una positiva adopción del sistema.

La implementación de un sistema ágil y digitalizado para el restaurante "Los Naranjos" permitirá una mejora significativa en la experiencia del cliente, optimizando el tiempo de atención, reduciendo errores humanos y mejorando la eficiencia operativa. La metodología ágil facilita una integración continua y adaptativa, permitiendo que el sistema evolucione conforme a las necesidades cambiantes del negocio.

Cabe destacar que los puntos de mejora indicados por el profesor respecto a la Entrega 1 de este informe fueron incorporados a esta nueva versión, dando así cumplimiento a lo solicitado.