Informe 3: Prototipado y Validación del Sistema Los Naranjos

Restaurante Los Naranjos

Cristobal Aguila

Angelo Millán

Oscar Muñoz

Profesor Álvaro Herrera

Ingeniería De Software

Junio, 2025

# **1. Introducción**

## **1.1. Contexto del Proyecto**

El restaurante "Los Naranjos", ubicado en la comuna de La Florida, ha experimentado un crecimiento significativo que, paradójicamente, ha generado desafíos operacionales críticos. El aumento del 400% en reclamos relacionados con deficiencias en la atención, demoras en la entrega y errores en cobros ha posicionado la mejora de la experiencia del cliente como el objetivo estratégico principal para el año 2025. La creación de la Unidad de Sistemas & Procesos refleja el compromiso de la organización con la modernización tecnológica como respuesta a estos desafíos.

## **1.2. Evolución del Proyecto de Modernización**

En los informes anteriores, se establecieron los fundamentos del proyecto de modernización del sistema de gestión del restaurante. El **Informe 1** presentó el análisis del problema y la conceptualización inicial del sistema de software, donde se identificaron las causas raíz de los problemas operacionales y los requisitos derivados de éstos, desde la gestión manual de reservas hasta la falta de estandarización en los procesos de cocina. El **Informe 2** se centró en el diseño y la arquitectura del software, aplicando el modelo de vistas 4+1 y estándares de calidad. Para ello, se elaboraron y explicaron diversos diagramas UML para representar visualmente la estructura, el comportamiento y el despliegue del futuro sistema.

Como resultado de esta fase de análisis y diseño, se desarrolló un **prototipo inicial** que materializó las primeras conceptualizaciones del sistema, proporcionando una base tangible para la validación de supuestos y la identificación de mejoras.

## **1.3. Alcance de este informe: De la Conceptualización a la Validación**

El presente informe marca una transición crítica en el proyecto: el paso del diseño conceptual a la **validación empírica** del sistema propuesto. Esta fase se centra en cuatro pilares fundamentales que sustentan el desarrollo de software moderno:

### **1.3.1. Metodología de Prototipado y MVP**

Se explorará el marco teórico y práctico del desarrollo de prototipos y Productos Mínimos Viables (MVP), analizando cómo estas metodologías permiten una validación temprana de hipótesis de negocio y reducen los riesgos asociados al desarrollo de software.

### **1.3.2. Estrategia de Validación y Planes de Prueba**

Se presentará una metodología integral de pruebas diseñada específicamente para validar prototipos y MVPs en el contexto de sistemas de gestión restaurantera. Esta estrategia incluye pruebas de usabilidad, rendimiento, funcionalidad y aceptación del usuario, considerando las particularidades operacionales del negocio. Para ello el lector requerirá examinar los anexos de este informe.

### **1.3.3. Criterios de Aceptación y Métricas de Éxito**

Se definirán criterios objetivos y medibles que determinen el éxito del prototipo y las iteraciones subsecuentes. Estos criterios estarán alineados con el objetivo estratégico de mejorar la experiencia del cliente y abordarán tanto aspectos técnicos como de negocio.

### **1.3.4. Plan de Versionamiento y Evolución Continua**

Se establecerá una estrategia de versionamiento que permita la evolución controlada del sistema, garantizando la trazabilidad de cambios y la gestión eficiente de releases en un entorno operacional crítico como el de un restaurante.

## **1.4. Entregables del Informe**

El presente documento incluirá la **presentación del prototipo desarrollado**, detallando sus funcionalidades implementadas, las decisiones de diseño adoptadas y las limitaciones identificadas durante su desarrollo. Además, se presentará el **nuevo MVP** que surge como resultado directo de los procesos de testing y validación realizados, incorporando las mejoras y ajustes identificados a través de la retroalimentación de usuarios reales y pruebas técnicas.

Esta evolución del prototipo inicial al MVP refinado representa no solo un avance técnico, sino una validación del enfoque metodológico adoptado, demostrando cómo la iteración basada en evidencia empírica conduce a soluciones más robustas y alineadas con las necesidades reales del negocio.

## **1.5. Impacto Esperado**

La implementación de esta metodología de prototipado y validación posiciona al proyecto Los Naranjos como un caso de estudio en la aplicación de principios de ingeniería de software moderna a la resolución de problemas operacionales reales. El enfoque adoptado no solo busca resolver los desafíos inmediatos del restaurante, sino establecer un marco metodológico replicable para futuros proyectos de modernización en el sector gastronómico.

# **2. Marco Teórico: Prototipo y MVP**

**2.1. Fundamentos y Estrategia de Prototipado**

El prototipado es un pilar fundamental en la metodología de este proyecto, ya que transforma las ideas abstractas en representaciones tangibles con las que los interesados pueden interactuar. El objetivo principal de esta fase es mantener los costos bajos y la resolución simple, permitiendo que el equipo falle rápido y barato. Esta interacción directa con una versión preliminar del sistema es clave para generar empatía, resolver diferencias de opinión entre los stakeholders y eliminar ambigüedades sobre los requerimientos. Al permitir que el personal del restaurante experimentara con los flujos de trabajo propuestos, se iniciaron conversaciones cruciales que permitieron comprender a fondo los problemas operativos, como la gestión manual de reservas y las comandas ilegibles, asegurando que las soluciones propuestas fueran realmente efectivas.

En el contexto de "Los Naranjos", el prototipado sirvió para múltiples propósitos. Se utilizaron **prototipos de validación** para probar funcionalidades específicas y refinar las soluciones a los cuellos de botella identificados en el servicio. Además, funcionaron como **prototipos de inspiración**, mostrando a la gerencia y al personal una visión clara de cómo la tecnología podría mejorar radicalmente su día a día. Siguiendo este enfoque, el proyecto avanzó desde la creación de varios prototipos de baja fidelidad (véase imágenes de este informe), que permitieron explorar diversas alternativas de forma ágil, hasta consolidar los aprendizajes en un único prototipo de alta fidelidad. Este último sirvió como la versión final de validación antes de evolucionar hacia la construcción del Producto Mínimo Viable (MVP).

**2.2. Metodología MVP (Minimum Viable Product)**

La metodología del Producto Mínimo Viable (MVP) se centra en lanzar una versión inicial de un producto con solo las características indispensables para ser funcional y aportar valor. Su propósito es validar una idea de negocio a través del ciclo **Construir-Medir-Aprender**: se **construye** una solución básica, se **mide** su impacto con usuarios reales y se **aprende** de sus comentarios para guiar las futuras mejoras. Este enfoque reduce los riesgos de desarrollo y asegura que el producto evolucione basándose en datos concretos en lugar de suposiciones.

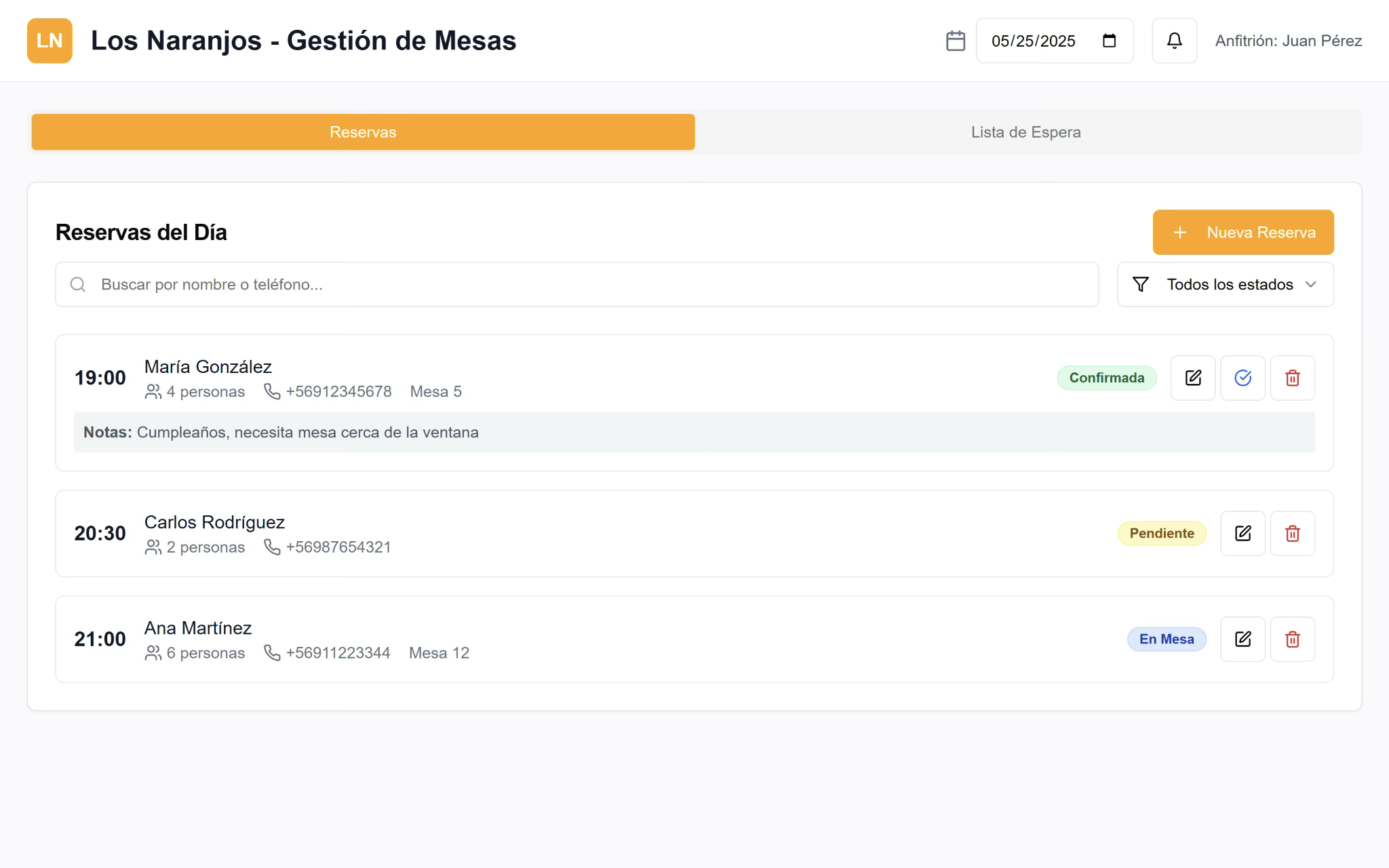
En el caso de "Los Naranjos", el MVP fue diseñado para solucionar los problemas más críticos que afectan la experiencia del cliente, como la desorganización en la recepción y los errores en las comandas y cobros. Esto fue construido a partir del prototipo de alta fidelidad y de los resultados de las pruebas obtenidas. Es por esto que desde un principio el alcance se limitó a tres módulos esenciales: Gestión de Reservas y Lista de Espera, Gestión de Mesas y Portal de Administración. De esta forma, se busca validar la hipótesis central (que la digitalización del servicio mejora la eficiencia y la satisfacción) antes de invertir en funcionalidades secundarias como reservas web o estrategias de marketing, las cuales están fuera del alcance del MVP. En este informe se detalla el error crítico de diseño y cómo el MVP llega a solventarlo.

**3. Presentación del Prototipo Inicial**

**3.1. Funcionalidades implementadas**

El prototipo de alta fidelidad se enfocó en validar el flujo administrativo y de atención al cliente desde la entrada hasta el pago, consolidando las funcionalidades en tres módulos principales. Deliberadamente, se excluyó la complejidad de la comunicación con cocina (comandas) para centrarse primero en resolver los problemas de organización en la recepción, la asignación de mesas y la correcta facturación.

#### **3.1.1. Módulo de Gestión de Reservas y Lista de Espera**



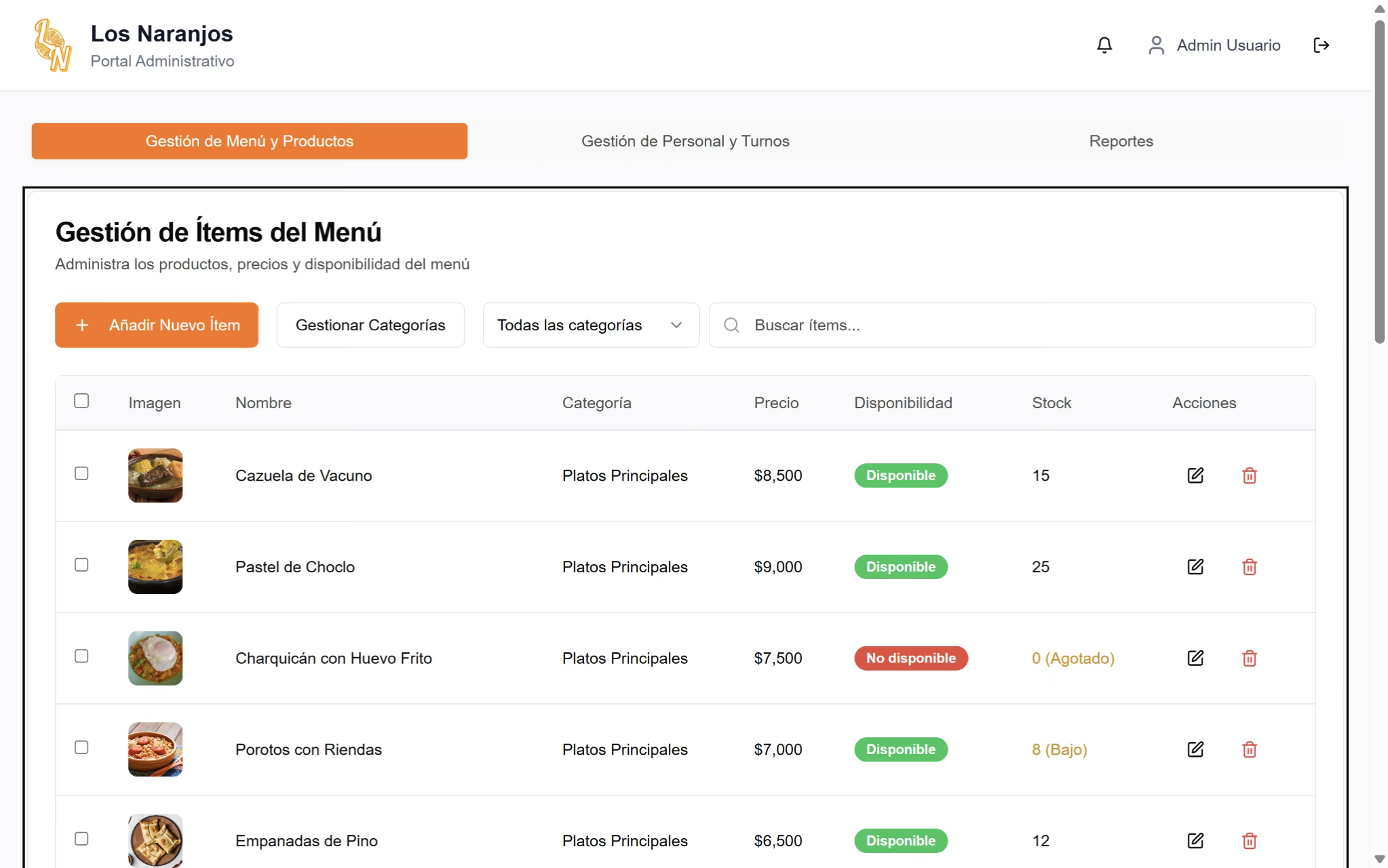
Este módulo se diseñó para el rol de Anfitrión y atacaba directamente la desorganización de las reservas en papel. El prototipo permitía registrar digitalmente las reservas que llegaban por teléfono, capturando los datos del cliente. A su llegada, el personal podía buscar la reserva por nombre o teléfono para confirmarla. Adicionalmente, se implementó una lista de espera digital para gestionar el flujo de clientes sin reserva cuando el local estaba lleno, permitiendo registrarlos y asignarles una mesa por orden de llegada una vez que esta se desocupara.

#### **3.1.2. Módulo de Gestión de Mesas**



Esta fue la pieza central del prototipo, fusionando la operación del salón con la facturación. El actor principal de este módulo es el mozo, el cual tiene la responsabilidad de manejar adecuadamente el punto de entrada del sistema y de este modo conectarse directamente con el flujo de facturación. Para validar el ciclo de facturación, el prototipo permitía que, una vez ocupada una mesa, se le pudieran añadir ítems de consumo de forma simulada. Al solicitar la cuenta, el sistema calculaba automáticamente el total, eliminando los errores de cálculo manual. El prototipo permitía registrar el pago con distintos medios (tarjeta, efectivo, cheque), simular la impresión de un recibo y finalmente, interactuar con los otros módulos, actualizándolos adecuadamente.

#### **3.1.3. Módulo de Portal Administrativo**



Este módulo estaba restringido al rol de Gerente y se enfocaba en dos problemas críticos. Primero, permitía cargar y gestionar los precios de todos los productos del menú, solucionando la causa raíz de la ambigüedad de los precios de éstos. Segundo, ofrecía una función básica de administración de usuarios para crear cuentas para el personal y asignarles roles y turnos, abordando la problemática de la poca claridad de los roles del personal. Tercero, mostraba al usuario del sistema un reporte completo de las operaciones del negocio.

### **3.2. Limitaciones Identificadas**

### Aunque el prototipo de alta fidelidad fue exitoso para validar los flujos de recepción y administración, las pruebas con usuarios revelaron limitaciones críticas, principalmente en el módulo de "Gestión de Mesas" que integraba el Punto de Venta (POS). Estas limitaciones hicieron evidente la necesidad de reestructurar la arquitectura del sistema para el MVP.

### La principal limitación fue el **conflicto de roles y la sobrecarga funcional** en una sola interfaz. Al fusionar la gestión del estado de las mesas con el proceso de facturación, se forzaba a personal con distintas responsabilidades, como el Anfitrión y el Cajero, a interactuar con una misma pantalla. Las pruebas de usabilidad demostraron que esto generaba un flujo de trabajo confuso y poco eficiente, yendo en contra del objetivo estratégico de tener roles claros y definidos para el personal. Un Anfitrión necesita una vista rápida y limpia para asignar mesas, mientras que un Cajero requiere una interfaz detallada para procesar pagos y emitir boletas, necesidades que resultaron incompatibles en un único módulo.

### La segunda y más crítica limitación fue la **ausencia total del flujo de comunicación con la cocina**. El prototipo, al centrarse solo en la facturación, no ofrecía ninguna solución para el problema de fondo que causa las demoras y errores: las comandas escritas a mano y sin un formato estándar. Durante la validación, el personal de prueba (como el Mozo Experimentado) señaló que, sin un sistema para enviar pedidos digitalmente a una pantalla en cocina (KDS), el mayor cuello de botella del servicio permanecería intacto, impidiendo cumplir con el objetivo de "Mejorar la experiencia de los clientes".

### A raíz de estas limitaciones, se tomó la decisión estratégica de dividir el módulo "Gestión de Mesas" para el MVP. Se determinó que era indispensable crear dos módulos separados y especializados:

### **Módulo de Comandas y Cocina**: Dedicado exclusivamente a la toma de pedidos digital por parte del mozo y su visualización en el KDS de cocina y bar, para garantizar rapidez y precisión en la preparación.

### **Módulo de Punto de Venta (POS) y Caja**: Un módulo enfocado únicamente en la gestión de precios, generación de cuentas, registro de pagos e impresión de boletas, asegurando un proceso de cobro eficiente y sin errores.

**4. Estrategia de Validación del Prototipo**

# Para el proyecto "Los Naranjos", la **calidad** se define como la capacidad del sistema para satisfacer de manera medible las necesidades del restaurante, asegurando que cada función aporte valor al negocio y contribuya directamente al objetivo estratégico de "Mejorar la experiencia de los clientes". La calidad no se limita a la ausencia de errores, sino que se mide por el cumplimiento de los criterios de aceptación funcionales y de rendimiento establecidos para cada módulo, como la precisión en los cobros y la agilidad en la toma de comandas.

# Para alcanzar este estándar de calidad, el proyecto adoptó un enfoque ágil. En este marco, el **rol del tester** trasciende la detección de fallos en una fase final; en cambio, el equipo de pruebas se integró desde el inicio como un colaborador activo en el ciclo de desarrollo. Trabajando estrechamente con desarrolladores y stakeholders (como el Gerente, Anfitrión y Mozos), su labor se centró en prevenir defectos y asegurar la usabilidad y el valor de cada entrega. Esta colaboración se organizó siguiendo el modelo de los **Cuadrantes del Testing Ágil**, que combina pruebas automatizadas orientadas a la tecnología con pruebas manuales enfocadas en el negocio. Este enfoque garantiza una cobertura completa, validando tanto la robustez técnica del código como la correcta funcionalidad desde la perspectiva del usuario final.

# La aplicación de esta estrategia comenzó con la validación del prototipo de alta fidelidad (*véase anexo A*) . Se ejecutó un plan de pruebas inicial que, aunque básico, fue fundamental para confirmar las fallas de usabilidad y los conflictos de roles en el módulo consolidado de "Gestión de Mesas", como se detalló en la sección anterior. Los aprendizajes obtenidos en esta fase fueron cruciales y llevaron a la redefinición de la arquitectura del sistema. Consecuentemente, para la versión MVP se elaboró un plan de pruebas (*véase anexo B*) de la misma forma en la cual se validó el prototipo.

**5. Criterios de Aceptación y Evaluación**

A partir de todo lo aprendido identificamos los siguientes criterios de aceptación que definen las condiciones específicas que el sistema debe cumplir para ser aceptado por la gerencia. Sirven como la base para las pruebas de validación para versiones futuras del sistema y garantizan que el producto final resuelva los problemas de negocio identificados y cumpla con los objetivos estratégicos del restaurante "Los Naranjos".

#### **5.1.1. Módulo 1: Salón y Recepción**

Este módulo es responsable de gestionar el flujo de clientes desde su llegada hasta que son sentados en una mesa.

* **Épica 1: Gestión Visual de Mesas**
  + **CA.1.1.1:** El Anfitrión debe poder ver una representación gráfica del plano del restaurante con el estado de cada mesa actualizado por color (disponible, ocupada, reservada, esperando limpieza).
  + **CA.1.1.2:** Al seleccionar una mesa, el sistema debe mostrar su número y capacidad.
  + **CA.1.1.3:** Cuando un Mozo abre una cuenta, el estado de la mesa debe cambiar automáticamente a ocupada.
  + **CA.1.1.4:** Cuando el Cajero cierra la cuenta, el estado de la mesa debe cambiar automáticamente a esperando limpieza.
  + **CA.1.1.5:** La actualización del estado de una mesa debe reflejarse en todos los dispositivos en menos de 2 segundos.
* **Épica 2: Gestión de Reservas**
  + **CA.1.2.1:** El Anfitrión debe poder registrar nuevas reservas con nombre, teléfono, cantidad de personas, fecha y hora.
  + **CA.1.2.2:** El sistema debe permitir buscar una reserva por nombre o número de teléfono del cliente.
  + **CA.1.2.3:** El Anfitrión debe poder asignar una reserva encontrada a una mesa que se encuentre disponible.
  + **CA.1.2.4:** El sistema debe guardar los datos de las reservas de forma segura para prevenir la pérdida de información.
* **Épica 3: Gestión de Lista de Espera**
  + **CA.1.3.1:** El Anfitrión debe poder agregar clientes sin reserva a una lista de espera digital cuando el local esté lleno.
  + **CA.1.3.2:** La lista de espera debe mostrar a los clientes ordenados por hora de llegada.
  + **CA.1.3.3:** El sistema debe permitir asignar a un cliente de la lista a una mesa que quede disponible.

#### **5.1.2. Módulo 2: Comandas y Cocina**

Este módulo es el corazón operativo, diseñado para eliminar los errores y demoras en la toma y preparación de pedidos.

* **Épica 1: Toma de Pedidos Digital**
  + **CA.2.1.1:** El Mozo debe poder tomar pedidos desde un dispositivo móvil seleccionando productos de una carta digital organizada por categorías.
  + **CA.2.1.2:** Debe ser posible añadir notas o modificaciones específicas a un producto del pedido (ej: "sin cebolla").
  + **CA.2.1.3:** Al confirmar la comanda, el sistema debe enviar los productos de comida a la pantalla de Cocina y los bebestibles a la del Bar.
  + **CA.2.1.4:** El envío de la comanda a cocina/bar no debe tener una latencia mayor a 2 segundos.
* **Épica 2: Visualización de Comandas en Cocina y Bar (KDS)**
  + **CA.2.2.1:** El personal de cocina y bar debe recibir las comandas en una pantalla en un formato digital y estandarizado.
  + **CA.2.2.2:** Los pedidos deben mostrarse en orden cronológico, indicando el tiempo de espera de cada uno.
  + **CA.2.2.3:** El personal debe poder marcar un pedido como "En Preparación" para informar visualmente su estado.
* **Épica 3: Notificación de Pedido Listo**
  + **CA.2.3.1:** Al marcar un pedido como "Listo para Retirar" en el KDS, el sistema debe enviar una notificación automática al dispositivo del Mozo que generó la comanda.
  + **CA.2.3.2:** La notificación debe llegar al Mozo en menos de 5 segundos.

#### **5.1.3. Módulo 3: Punto de Venta (POS) y Caja**

Este módulo centraliza la gestión financiera para asegurar la precisión en los cobros y agilizar los pagos.

* **Épica 1: Generación y Cobro de Cuentas**
  + **CA.3.1.1:** El sistema debe calcular el total de la cuenta de una mesa de forma automática, basándose en los productos consumidos.
  + **CA.3.1.2:** Debe permitir el registro del pago seleccionando múltiples medios: tarjeta, efectivo o cheque.
  + **CA.3.1.3:** El sistema debe garantizar un 100% de exactitud en los cálculos de la cuenta.
  + **CA.3.1.4:** La generación de la cuenta no debe superar los 3 segundos.
* **Épica 2: Cierre de Caja**
  + **CA.3.2.1:** Al finalizar la jornada, un usuario autorizado debe poder generar un reporte de cierre de caja.
  + **CA.3.2.2:** El reporte debe mostrar un desglose de los ingresos totales por cada medio de pago registrado.

#### **5.1.4. Módulo 4: Administración y Reportes**

Este módulo ofrece las herramientas de control y gestión para la toma de decisiones estratégicas.

* **Épica 1: Gestión de Roles y Usuarios**
  + **CA.4.1.1:** El Gerente debe poder crear cuentas de usuario para los empleados y asignarles un rol predefinido (Anfitrión, Mozo, Cajero, etc.).
  + **CA.4.1.2:** El sistema debe restringir el acceso a las funciones según el rol del usuario (ej: un mozo no puede modificar precios).
* **Épica 2: Reportería de Desempeño y Ventas**
  + **CA.4.2.1:** El Gerente debe poder visualizar un dashboard con indicadores clave del día (ventas totales, clientes atendidos, etc.).
  + **CA.4.2.2:** El sistema debe permitir generar reportes de los productos más y menos vendidos en un rango de fechas.
  + **CA.4.2.3:** Debe ser posible exportar los datos de los reportes a formatos como CSV o PDF.
  + **CA.4.2.4:** La carga de reportes complejos no debe tardar más de 15 segundos.

**6. Generación del MVP**

**6.1. Análisis de feedback y mejoras identificadas**

La fase de validación del prototipo de alta fidelidad fue un punto de inflexión en el proyecto. Las sesiones de testing con usuarios finales, incluyendo al Anfitrión, Mozo, Cajero y Gerente, generaron un feedback invaluable que permitió identificar brechas críticas y refinar el alcance del Producto Mínimo Viable (MVP).

#### **6.1.1. Síntesis de Hallazgos del Testing**

El análisis del feedback se puede resumir en los siguientes puntos clave:

1. **Conflicto de Roles en el Módulo de "Gestión de Mesas"**: El personal de salón (Anfitrión y Mozo) confirmó que la interfaz unificada era problemática. Mientras que el Anfitrión necesitaba una vista simple y ágil del plano del restaurante para asignar mesas, el Mozo y el Cajero requerían una interfaz más detallada para gestionar el consumo y los pagos. Esta sobrecarga funcional creaba confusión y ralentizaba el servicio en horas pico, contradiciendo el objetivo de agilizar la operación.
2. **Brecha Crítica: Ausencia de Comunicación con Cocina**: El feedback más contundente provino del personal de operaciones. Señalaron que, aunque el prototipo resolvía los errores de cobro, dejaba intacta la causa raíz de las mayores demoras y reclamos: el sistema de comandas manuales. Se concluyó que sin un módulo que digitalizara la toma de pedidos y los enviara a una pantalla en cocina (KDS), el objetivo de "Mejorar la experiencia de los clientes" sería inalcanzable.
3. **Necesidad de un Punto de Venta (POS) Especializado**: El personal de caja valoró positivamente la automatización del cálculo de cuentas, ya que eliminaba los "cobros que no corresponden". Sin embargo, indicaron que el proceso de pago, al estar integrado en una vista de gestión general, no era lo suficientemente rápido ni robusto para un entorno de alto volumen, y carecía de funciones esenciales como un reporte de cierre de caja detallado.

#### **6.1.2. Priorización de Mejoras y Decisiones para el MVP**

Basado en estos hallazgos, se tomaron las siguientes decisiones estratégicas para la transición del prototipo al MVP:

* **Creación del Módulo de Comandas y Cocina**: Se definió como la máxima prioridad desarrollar un módulo completamente nuevo, inexistente en el prototipo, dedicado a la toma de pedidos digital y a la gestión del KDS. Este módulo se volvió el pilar del MVP, ya que atacaba directamente la principal causa de demoras y errores en el servicio.
* **División del Módulo Central**: Para resolver el conflicto de roles, se decidió desmantelar el módulo "Gestión de Mesas" del prototipo. Sus funcionalidades se dividieron en dos nuevos módulos para el MVP:
  + **Módulo 1: Salón y Recepción**: Enfocado exclusivamente en las tareas del Anfitrión (plano de mesas, reservas y lista de espera).
  + **Módulo 3: Punto de Venta (POS) y Caja**: Un módulo especializado para el Cajero, con todas las funcionalidades de facturación, registro de pagos y cierre de caja.

**6.2. Evolución del prototipo al MVP**

Las consideraciones anteriormente mencionadas encapsulan una serie de elementos fundamentales que deben ser considerados si es que se quiere mejorar por sobre la versión del prototipo de alta fidelidad creado en un comienzo. En el desarrollo del MVP se necesitó reformular el diseño del sistema de manera que se ejecuten con mayor eficiencia los procesos críticos del negocio, y se ahondó en funcionalidades de los módulos a partir del feedback recibido. De esta forma, podemos listar los siguientes avances realizados:

**6.2.1 Módulo Salón y Recepción (anteriormente Gestión de Reservas)**

Dentro del contexto del prototipo, este módulo cumplía la función de mostrar y ejecutar acciones relacionadas con las reservas de los clientes del negocio y la lista de espera. El MVP evolucionó con lo propuesto en un principio, y se incorporó un mapa del restaurante de modo que el Anfitrión tenga información más detallada del estado del negocio en todo momento. Además, se describe la lista de espera con más detalle, discriminando entre clientes que esperan a ser atendidos y clientes ya notificados, lo que genera una plusvalía en el trabajo del Anfitrión. Adicionalmente, se agrega un apartado de estadísticas de este módulo.

**6.2.2 Módulo Comandas y Cocina**

La creación de este módulo responde directamente a la principal limitación identificada durante la validación del prototipo: la ausencia de un sistema para gestionar la comunicación entre el personal de servicio y la cocina. Para abordar esta brecha, se implementa una solución basada en un Kitchen Display System (KDS) o Sistema de Visualización en Cocina. Este sistema está diseñado para sincronizar de manera eficiente el flujo de trabajo entre los mozos y el personal de cocina, eliminando las fricciones y ambigüedades causadas por las comandas manuales. El objetivo es fluidificar la comunicación y reducir drásticamente los tiempos de preparación y los errores en los pedidos.

El módulo opera como un sistema dual y cohesivo, donde las interacciones están estrictamente definidas por los roles y permisos de cada usuario. Los mozos utilizan la interfaz desde sus dispositivos móviles para registrar y enviar los pedidos, mientras que el personal de cocina interactúa con las pantallas del KDS para visualizar, gestionar y confirmar la preparación de cada plato.

**6.2.3 Módulo de Punto de Venta (POS) y Caja**

Este módulo se establece en el MVP como una interfaz especializada y separada para el Cajero, con el fin de gestionar de manera transparente y eficiente la facturación y las transacciones financieras diarias del negocio. Su diseño se enfoca en la precisión y la agilidad para resolver directamente los "cobros que no corresponden". El módulo se estructura en las siguientes vistas funcionales:

* **Vista de Mesas y Cuentas:** Funciona como el panel de control operativo del Cajero, ofreciendo una visualización en tiempo real del estado de las mesas y sus cuentas activas. Esta interfaz permite consultar el detalle del consumo por mesa, aplicar descuentos autorizados y gestionar la división de cuentas, una función esencial para grupos. Crucialmente, incluye la capacidad de agregar productos manualmente, sirviendo como un plan de contingencia inmediato ante cualquier incidencia en el módulo de comandas.
* **Vista de Catálogo:** Proporciona una herramienta para la gestión dinámica de la oferta. Permite al Cajero consultar la disponibilidad de los productos y, como una función de flexibilidad altamente solicitada, le otorga permiso para editar precios de manera rápida según las necesidades del momento. Esta vista complementa la gestión principal del catálogo que realiza el Gerente.
* **Vista de Procesamiento de Pagos:** Es el núcleo transaccional del sistema. Aquí, el Cajero finaliza la venta visualizando el resumen de la cuenta y registrando el pago según el medio elegido por el cliente (tarjeta, efectivo o cheque). Para mantener una clara separación de roles, aunque el Mozo puede solicitar la generación de la cuenta desde su dispositivo, es el Cajero quien utiliza esta vista especializada para procesar el pago. Se incluye también un historial de las transacciones realizadas durante la sesión para una rápida consulta.
* **Vista de Cierre de Caja y Reportes:** Esta sección provee las herramientas de control financiero. Permite al Cajero realizar el cierre de caja al final de la jornada, generando un reporte consolidado con el resumen de ventas y el desglose por método de pago. El control de efectivo es de vital importancia para garantizar la integridad financiera, minimizar descuadres y prevenir pérdidas en un entorno de alto flujo de transacciones. El sistema también mantiene un historial de sesiones anteriores para fines de auditoría y análisis comparativo.

**6.2.4 Módulo Portal Administrativo**

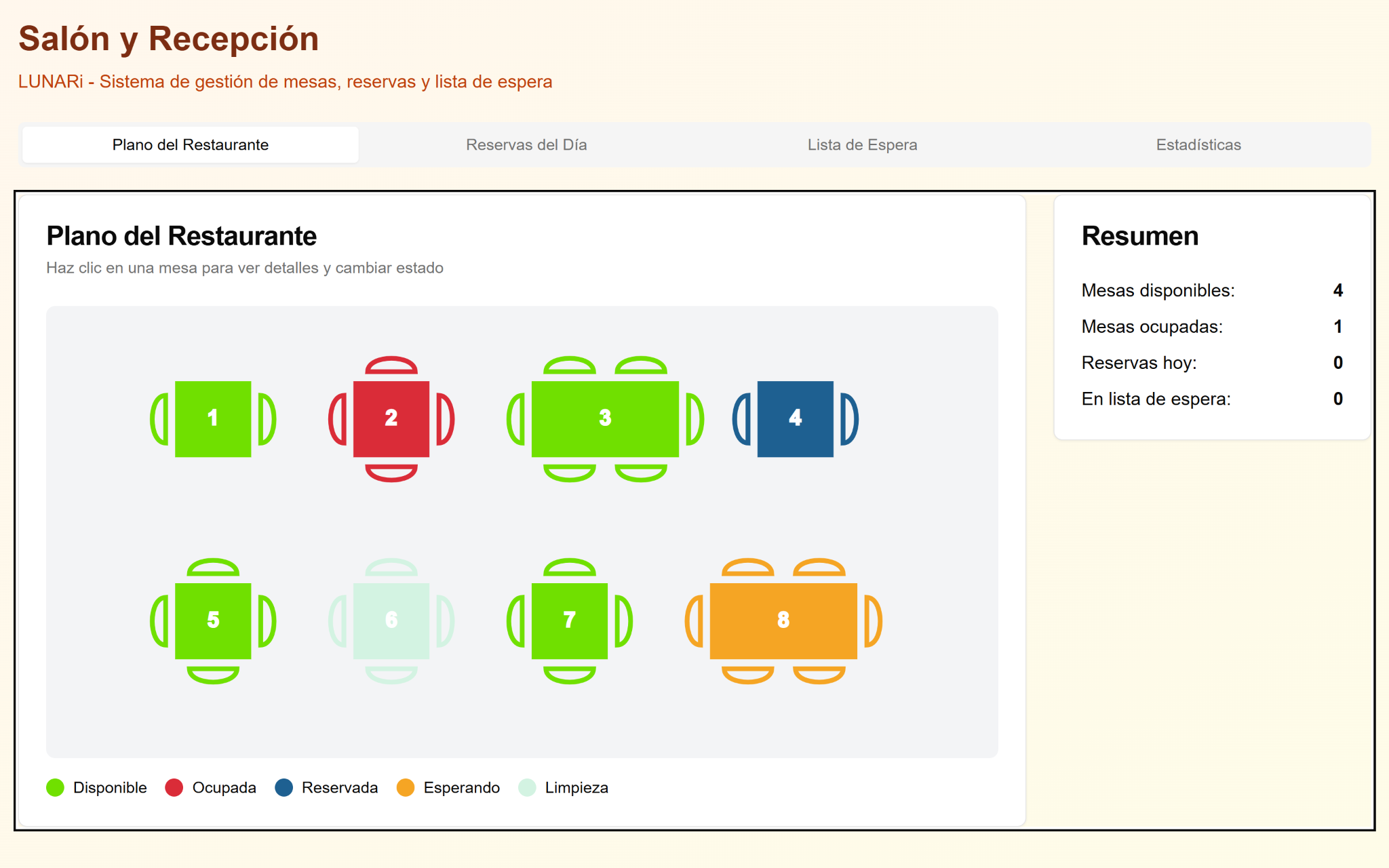
El Módulo de Administración y Reportes, que evolucionó desde el "Portal Administrativo" del prototipo, fue objeto de un refinamiento centrado principalmente en la interfaz de usuario (UI). En respuesta a la retroalimentación obtenida durante las pruebas de validación, se optimizó la presentación visual para garantizar una mayor claridad y una interpretación más intuitiva de la información por parte de la gerencia.

Un cambio estructural significativo en su diseño fue la separación de las funciones en dos vistas especializadas y distintas: una para la **Gestión de Personal**, enfocada en la administración de roles y permisos, y otra para la **Reportería y Análisis**, dedicada a los indicadores de rendimiento y ventas. Esta división asegura una navegación más lógica y facilita a la gerencia un acceso más ordenado y directo a las herramientas de control estratégico.

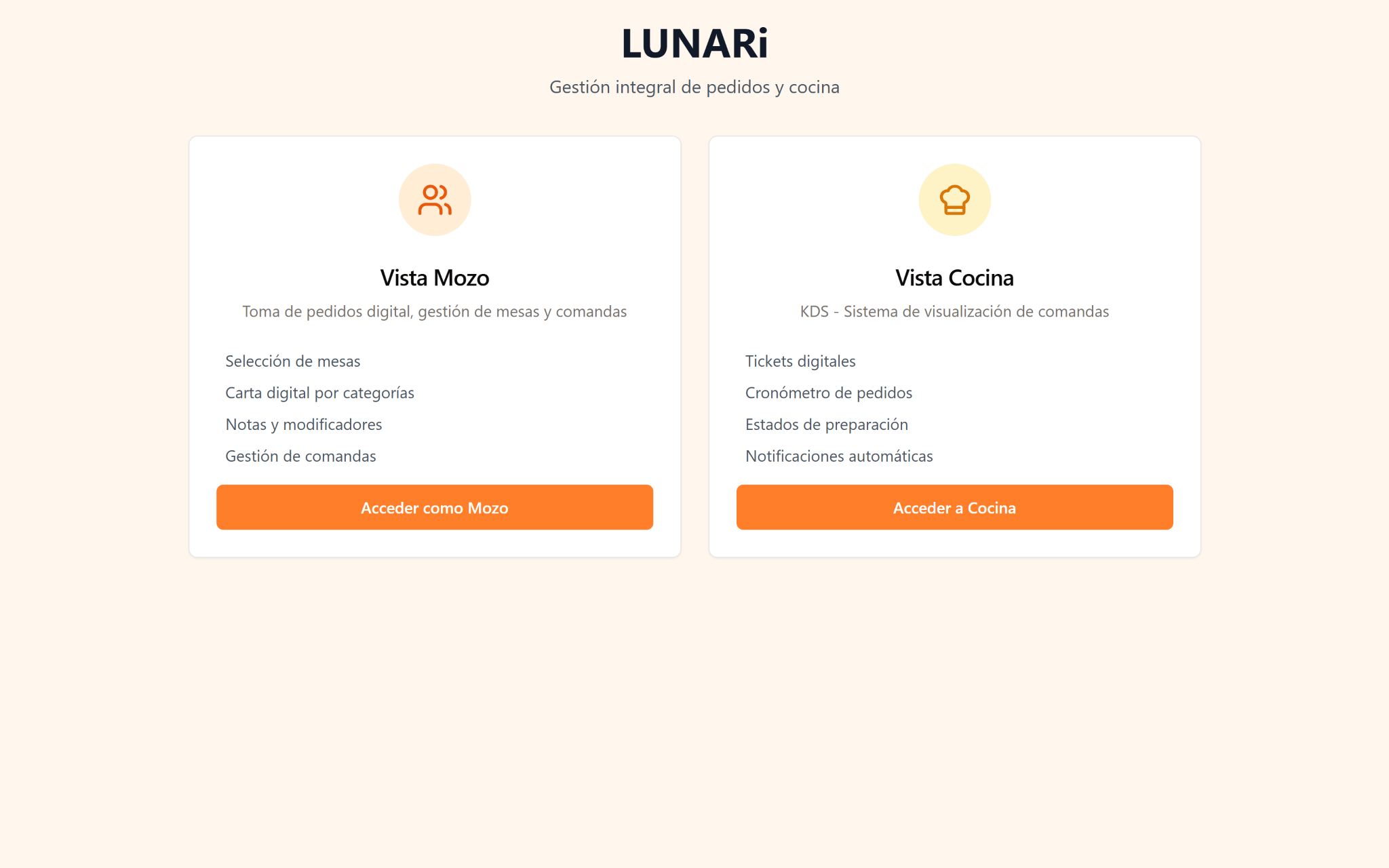
# **6.3. Presentación del MVP**

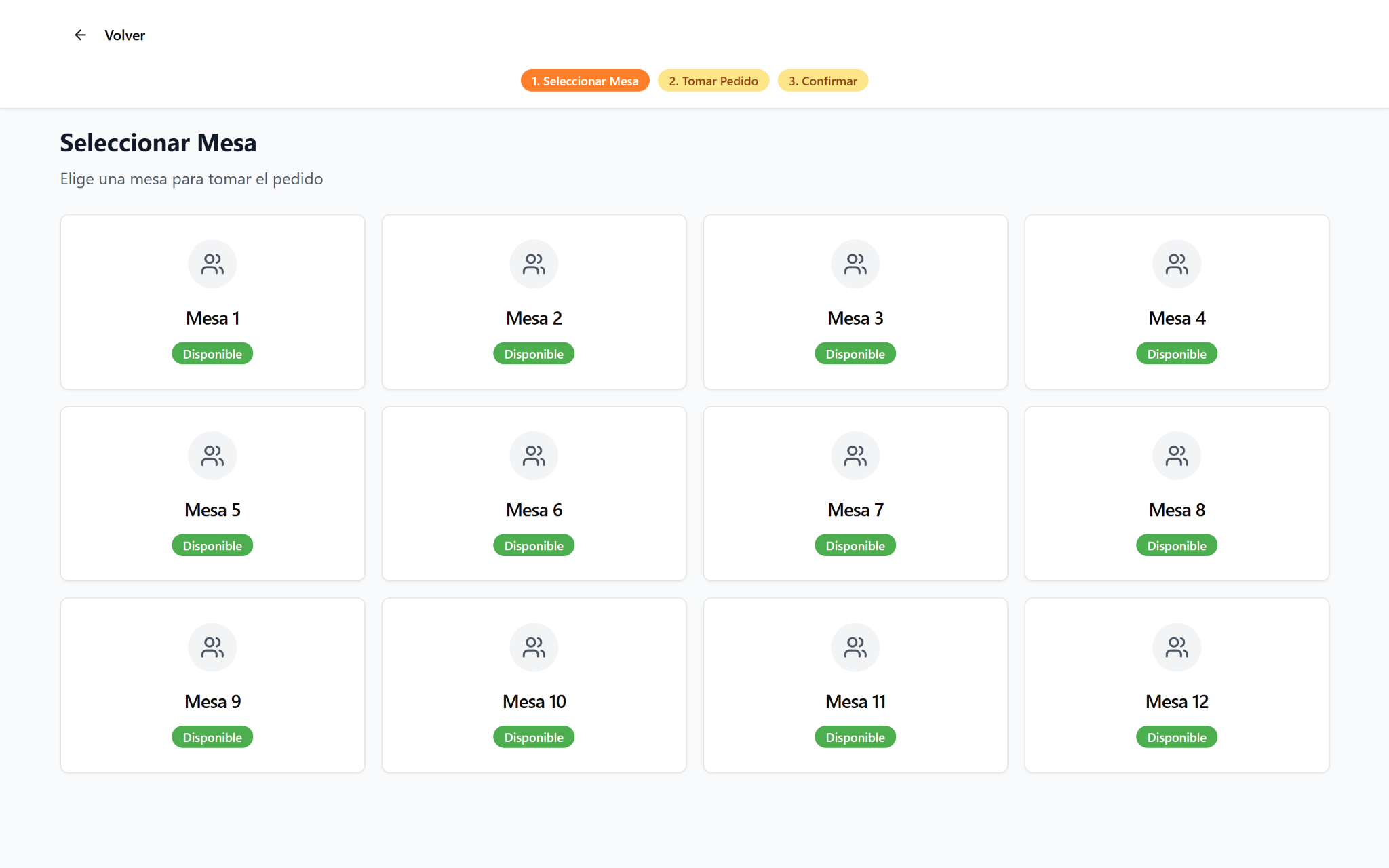
A continuación se presentan las vistas principales del MVP creado, para mayor detalle en el funcionamiento de estos módulos véase los

**6.3.1. Módulo Salón y Recepción**



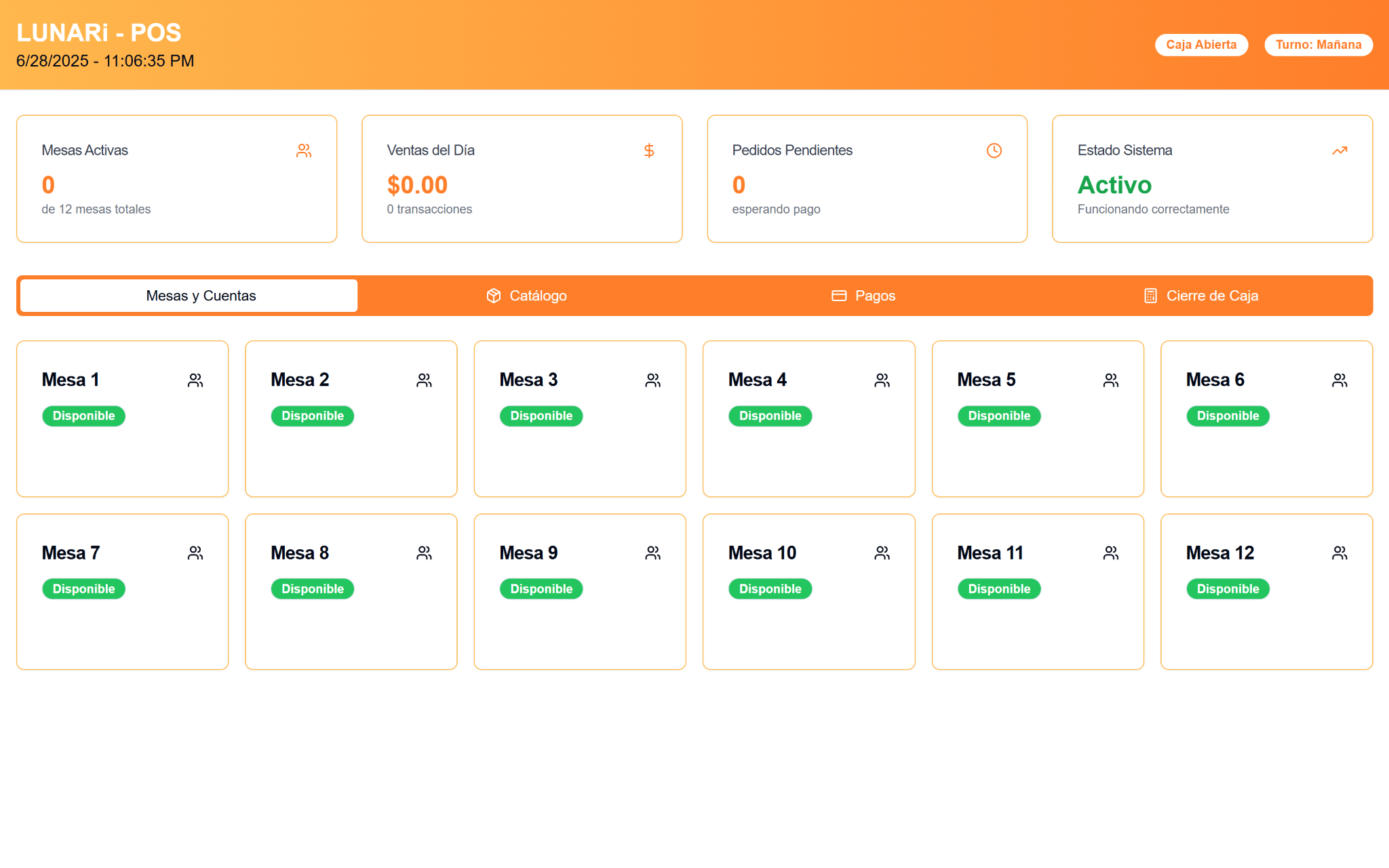
**6.3.2 Módulo Comandas y Cocina**

****

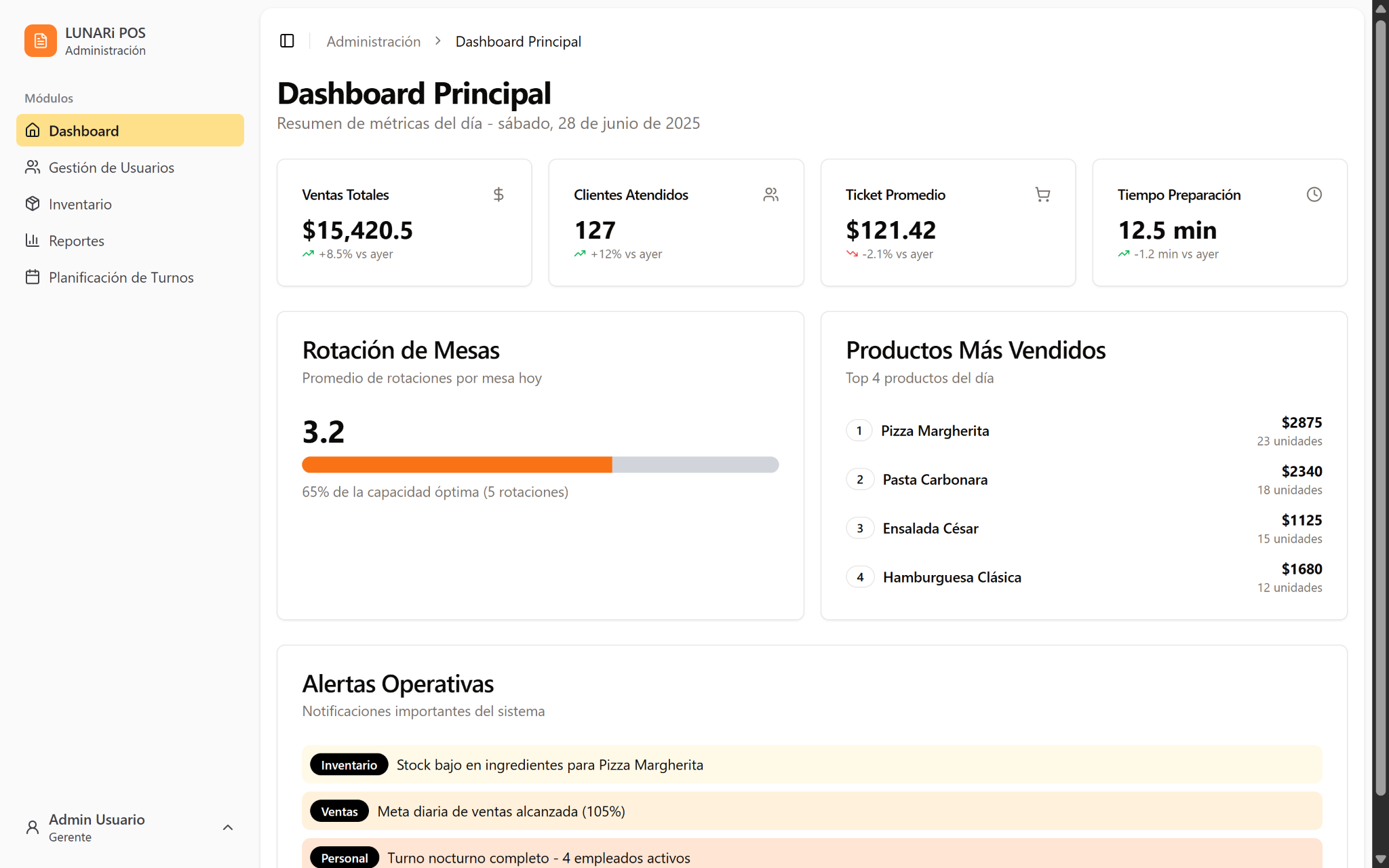
****

****

**6.3.3 Módulo de Punto de Venta y Caja**

****

**6.3.4 Módulo Portal Administrativo**

****

**7. Gestión de Versiones**

**7.1. Estrategia de Versionamiento**

Para gestionar la evolución del sistema de forma ordenada y predecible más allá del lanzamiento del MVP, se establece una estrategia formal de versionamiento y gestión de cambios. Esta estrategia asegura la estabilidad del producto en producción mientras se desarrollan nuevas funcionalidades. Para estos propósitos hemos decidido usar Git como software de control de versiones y GitHub para alojar el proyecto.

#### **Metodología Adoptada: Versionamiento Semántico (SemVer)**

El proyecto adoptará el estándar de Versionamiento Semántico (SemVer), que utiliza un formato de tres números: MAYOR.MENOR.PARCHE.

* **MAYOR (X.y.z):** Se incrementará para cambios incompatibles con versiones anteriores o rediseños arquitectónicos significativos. Por ejemplo, la futura implementación de un módulo completo de "Gestión de Inventario" podría justificar un salto de la versión 1.x.x a la 2.0.0.
* **MENOR (x.Y.z):** Se incrementará al añadir nuevas funcionalidades que son compatibles con versiones anteriores. La mayoría de las características planeadas para futuros sprints, como las reservas a través de una página web o la división de cuentas, corresponderían a una versión menor (ej: 1.1.0, 1.2.0).
* **PARCHE (x.y.Z):** Se incrementará para correcciones de errores retrocompatibles. La solución de un defecto encontrado en producción, como un error de cálculo en el cierre de caja, se liberaría como una versión de parche (ej: 1.0.1).

El lanzamiento inicial del Producto Mínimo Viable (MVP) será designado como la versión **1.0.0**.

#### **Política de Branches y Releases**

Para la gestión del código fuente se utilizará un modelo de ramificación (branching) basado en GitFlow, el cual se alinea con los distintos ambientes de prueba definidos en el plan de pruebas.

* **Rama main**: Contendrá únicamente el código de producción, estable y etiquetado con la versión correspondiente a cada lanzamiento (ej: v1.0.0).
* **Rama develop**: Será la rama principal de integración donde convergen todas las nuevas funcionalidades ya desarrolladas. Corresponde al ambiente de desarrollo y pruebas continuas.
* **Ramas feature/\***: Cada nueva historia de usuario o funcionalidad se desarrollará en una rama de característica propia, partiendo de develop. Esto permite el trabajo en paralelo y aísla los cambios hasta que estén listos.
* **Ramas release/\***: Cuando develop alcanza un estado maduro para un nuevo lanzamiento, se crea una rama de release. En esta rama se realizan las pruebas finales, correcciones de última hora y se prepara el despliegue, correspondiendo a los ambientes de Staging y Pre-Producción.
* **Ramas hotfix/\***: Se crearán a partir de main para solucionar errores críticos que surjan en el ambiente de producción, y luego se fusionarán tanto en main como en develop para asegurar la consistencia del código.

#### **Gestión de Cambios**

Todo cambio en el sistema, ya sea una nueva funcionalidad o la corrección de un error, deberá estar asociado a un requerimiento documentado (como una historia de usuario) o un informe de defecto. Los cambios deben seguir estrictamente la política de branches y superar todas las pruebas funcionales y de regresión definidas en el plan de pruebas antes de ser integrados a la rama develop. Cualquier modificación que altere significativamente el alcance de una futura versión deberá ser aprobada por los stakeholders principales, como el Gerente General, para garantizar su alineación con los objetivos del negocio.

### **7.2 Plan de Releases Futuros**

La evolución del sistema más allá de la versión 1.0.0 (MVP) se guiará por un roadmap de desarrollo iterativo. El plan general es agrupar las funcionalidades futuras en releases posteriores (ej: v1.1, v1.2), priorizándolas según su impacto estratégico en el negocio y la retroalimentación obtenida del uso del MVP.

Ya se ha identificado un backlog de historias de usuario de alto valor que no fueron incluidas en el MVP. Estas se agrupan en las siguientes áreas temáticas:

* **Experiencia del Cliente y Marketing:** Incluye funcionalidades como un portal de reservas web para clientes, la visualización de la carta mediante códigos QR y la implementación de programas de fidelización.
* **Operaciones y Pagos Avanzados:** Contempla mejoras para agilizar el servicio, como la capacidad de procesar pagos con tarjeta directamente en la mesa y herramientas avanzadas para dividir la cuenta en grupos grandes.
* **Administración y Analítica Avanzada:** Se planea incorporar un módulo para la planificación de turnos del personal, una mejora futura considerada crítica para la operación, junto con reportes más sofisticados para analizar el rendimiento del negocio en detalle.

Cada una de estas funcionalidades futuras ya cuenta con sus propios criterios de aceptación predefinidos, que guiarán su desarrollo y validación. La decisión sobre qué grupo de mejoras se implementará en la siguiente versión se tomará en base a las necesidades operativas y estratégicas que la gerencia de "Los Naranjos" identifique tras la puesta en marcha del sistema inicial.

**8. Conclusiones**

El desarrollo del sistema para el restaurante "Los Naranjos" ha validado exitosamente el enfoque metodológico propuesto, logrando entregar un Producto Mínimo Viable que responde directamente a las necesidades críticas del negocio. El proceso no estuvo exento de aprendizajes; de hecho, el mayor valor se obtuvo al **fallar rápido y a bajo costo** con el prototipo inicial. La decisión de consolidar funcionalidades en solo tres módulos se demostró errónea durante las pruebas, revelando que sin una separación clara de roles y un sistema dedicado a la comunicación con la cocina, los problemas de fondo persistirían. Este hallazgo fue fundamental y permitió redefinir la arquitectura hacia la estructura de cuatro módulos del MVP, que se demostró mucho más robusta y alineada con la operación real del restaurante.

El éxito de esta versión final se midió rigurosamente contra los **criterios de aceptación** definidos para cada módulo, los cuales actuaron como un contrato entre el sistema y los objetivos del negocio. Cumplir con estos criterios garantiza que el producto no solo funciona técnicamente, sino que aporta un valor tangible a la operación diaria. Finalmente, la implementación de una estricta **estrategia de versionamiento semántico** y una política de gestión de cambios asegura que el sistema tenga una hoja de ruta clara para su evolución futura, permitiendo incorporar nuevas funcionalidades de manera ordenada y manteniendo la estabilidad de la plataforma a largo plazo.

Entender el **MVP** y tener **criterios de aceptación claros** es clave para su futuro mantenimiento y adición o modificación de funciones. Esto nos permite evolucionar nuestra idea de forma controlada, añadiendo valor progresivamente y ajustando el rumbo cuando sea necesario. Así, evitamos sorpresas costosas y nos aseguramos de que el producto final sea realmente lo que la gente necesita y quiere. En resumen, el enfoque MVP nos permite construir, aprender y mejorar continuamente, haciendo que el camino desde la idea hasta el éxito sea mucho más eficiente y centrado en el usuario.

**9. ANEXOS**

**9.1. Anexo A: Plan de Pruebas Prototipo**

### **Proyecto: Evaluación del Prototipo - Sistema Los Naranjos**

### **Plan de Pruebas**

**Preparado por:** Gemini

#### **Introducción**

El presente plan de prueba está diseñado para evaluar el prototipo funcional del Sistema de Gestión para el Restaurante "Los Naranjos". El objetivo principal no es solo la identificación de errores de software (bugs), sino **validar la viabilidad de la arquitectura propuesta y exponer de manera objetiva sus limitaciones críticas**, incluyendo la sobrecarga funcional, los conflictos de roles y las ausencias de flujos operativos indispensables. Los resultados de este plan servirán como base para la toma de decisiones sobre el rediseño del sistema.

#### **Recursos**



| **Tester** | **% Participación** |
| --- | --- |
| Analista QA Senior | 100% |
| Gerente de Restaurante (para UAT) | 25% |

#### **Alcance**

Las pruebas se centrarán exclusivamente en los tres módulos definidos en el prototipo:

1. **Módulo de Gestión de Reservas y Lista de Espera:** Pruebas sobre la creación y búsqueda de reservas, y la gestión de la lista de espera digital.
2. **Módulo de Gestión de Mesas:** Pruebas sobre la asignación de mesas, adición simulada de ítems de consumo, cálculo de cuenta y registro de pago.
3. **Módulo de Portal Administrativo:** Pruebas sobre la gestión de precios, creación de usuarios/roles y visualización de reportes.

#### **Fuera del Alcance**

* **Flujo de Comunicación con Cocina:** La comunicación de comandas a la cocina (KDS) no es parte del prototipo y está **explícitamente fuera del alcance** de las pruebas funcionales. Se documentará esta ausencia como un resultado clave del plan.
* **Pruebas de Rendimiento a gran escala:** Se realizarán pruebas de carga ligera, pero no pruebas de estrés masivas.
* **Integraciones con Hardware de Terceros:** No se probará la integración real con terminales de pago (POS) o impresoras fiscales.

### **Pruebas Específicas para Evidenciar Fallos Críticos**

Esta sección es el núcleo del plan y está diseñada para demostrar las debilidades arquitectónicas del prototipo.

##### **1. Evidencia de la Sobrecarga Funcional y Conflicto de Roles (Módulo de Gestión de Mesas)**

El objetivo es demostrar que el rol de "Mozo" es inoperable en un escenario realista debido a la concentración de responsabilidades de caja.

* **Caso de Prueba CT-01 (Prueba de Conflicto de Roles):**
  + **Pasos:**
    1. Con el rol de **Gerente**, crear un usuario con el rol de "Cajero" en el Portal Administrativo.
    2. Iniciar sesión como "Cajero". Intentar acceder a funciones de cobro o cierre de mesa.
    3. Cerrar sesión e iniciar sesión como "Mozo". Acceder a las mismas funciones de cobro.
  + **Resultado Esperado (Hipótesis):** El usuario "Cajero" no tendrá ninguna funcionalidad útil, mientras que el "Mozo" podrá realizar todas las acciones de caja.
  + **Evidencia:** Se documentará la imposibilidad de que el rol "Cajero" opere, demostrando que la asignación de roles en el módulo administrativo está desconectada de la funcionalidad real del sistema.
* **Caso de Prueba CT-02 (Prueba de Carga Operativa):**
  + **Pasos:**
    1. Simular 3 mesas ocupadas y asignadas a un único usuario "Mozo".
    2. En la Mesa 1, añadir ítems.
    3. Simultáneamente, el "cliente" de la Mesa 2 solicita la cuenta y pagar con tarjeta.
    4. Mientras el Mozo está en el proceso de cobro de la Mesa 2, el "cliente" de la Mesa 3 lo llama para hacer un nuevo pedido.
  + **Resultado Esperado (Hipótesis):** Se producirá un cuello de botella evidente. El Mozo no podrá atender a la Mesa 3 hasta finalizar el largo proceso de cobro de la Mesa 2, generando una mala experiencia de servicio.
  + **Evidencia:** Se medirá el tiempo de espera del cliente de la Mesa 3. Se registrará cualitativamente la imposibilidad del Mozo para gestionar tareas de servicio y de caja al mismo tiempo.

##### **2. Evidencia de la Ausencia del Flujo de Comunicación con Cocina**

El objetivo es demostrar que el sistema, en su estado actual, no resuelve el problema de la comunicación de comandas y, de hecho, podría empeorarlo al crear un doble trabajo.

* **Caso de Prueba CT-03 (Prueba de Flujo End-to-End):**
  + **Pasos:**
    1. Un **Mozo** toma el pedido de una mesa usando el prototipo, registrando los ítems para facturación.
    2. Se inicia un cronómetro.
    3. El tester debe describir qué acción manual y fuera del sistema debe realizar el Mozo para que la cocina se entere del pedido (ej: escribir en un papel, gritar el pedido).
    4. Se documenta este paso como "GAP OPERACIONAL CRÍTICO".
    5. Se detiene el cronómetro cuando el plato estaría "listo".
  + **Resultado Esperado (Hipótesis):** El prototipo no aporta ninguna solución al problema de las comandas a cocina, que sigue siendo 100% manual y propenso a errores.
  + **Evidencia:** Se documentará el flujo de trabajo real, demostrando que el mozo debe registrar el pedido en el sistema Y ADEMÁS comunicarlo manualmente, duplicando su esfuerzo.

#### **Pruebas de Regresión**

Dado que es un prototipo, la regresión se centrará en el único flujo de valor consolidado. Se ejecutará después de cualquier ajuste.

* **RF-P-01 (Regresión-Prototipo):**
  1. El **Gerente** puede iniciar sesión y verificar/modificar el precio de un producto.
  2. El **Anfitrión** puede crear una reserva y luego asignarla a una mesa.
  3. El **Mozo** puede añadir ese producto (con el precio verificado) a la cuenta de la mesa.
  4. El **Mozo** puede generar la cuenta y el total debe ser correcto.
  5. El **Mozo** puede registrar el pago, cerrando el ciclo.
  6. El reporte del **Gerente** refleja la venta correctamente.

#### **Suposiciones**

* El prototipo está desplegado en un entorno de pruebas estable.
* El equipo de desarrollo es consciente de que el objetivo de las pruebas es la validación arquitectónica, no solo la corrección de errores menores.

#### **Riesgos**

| **No** | **Riesgos** | **Probabilidad (1-5)** | **Impacto (1-5)** | **Severidad (Prob\*Impct)** | **Plan de Mitigación** |
| --- | --- | --- | --- | --- | --- |
| 1 | Los resultados confirman que la arquitectura del prototipo es inviable. | 4 | 5 | 20 | **(Este es un resultado esperado)** Presentar los hallazgos de forma clara y objetiva al equipo para justificar un rediseño de la arquitectura de los módulos antes de continuar con el desarrollo. |
| 2 | El prototipo es demasiado inestable y presenta constantes bloqueos. | 3 | 4 | 12 | Implementar un "smoke test" diario para validar la estabilidad básica antes de ejecutar los casos de prueba del día. |
| 3 | El equipo interpreta los reportes de fallos como simples bugs y no como problemas de diseño conceptual. | 3 | 5 | 15 | Realizar una sesión de presentación de resultados con el equipo de desarrollo, explicando el "porqué" de cada prueba y cómo los resultados demuestran los fallos arquitectónicos. |

**9.2. Anexo B: Plan de Pruebas Mejorado**

### **Proyecto: Sistema de Gestión Restaurante Los Naranjos**

### **Plan de Pruebas**

**Preparado por:** Oscar Muñoz

#### **Introducción**

El plan de prueba está diseñado como una línea de base para identificar lo que se considera dentro y fuera del alcance de las pruebas para el nuevo Sistema de Gestión del Restaurante "Los Naranjos". El objetivo es validar que el software cumple con los requerimientos funcionales y no funcionales, asegurando que su implementación contribuya directamente al objetivo estratégico de “Mejorar la experiencia de los clientes”.

#### **Recursos**

| **Tester** | **% Participación** |
| --- | --- |
| Jefe de Pruebas QA | 50 % |
| Analista de Pruebas QA | 100% |

#### **Alcance**

Las pruebas incluyen todas las funcionalidades nuevas definidas para el MVP, las funcionalidades de alto riesgo de la suite de regresión, pruebas de aceptación de usuario (UAT) y pruebas de rendimiento. El alcance cubre los cuatro módulos principales del sistema:

1. **Módulo de Salón y Recepción:** Gestión del plano de mesas, reservas y lista de espera.
2. **Módulo de Comandas y Cocina:** Flujo de toma de pedidos, envío a cocina/bar y notificaciones.
3. **Módulo de Punto de Venta (POS) y Caja:** Generación de cuentas, procesamiento de pagos y gestión de precios.
4. **Módulo de Administración y Reportes:** Gestión de roles, Gestión de inventario, Gestión de usuarios y reportería básica.

##### **Pruebas de Regresión**

Las pruebas de regresión son un componente crítico de este plan y se ejecutarán tras cada despliegue de una nueva versión del software o corrección de errores. El objetivo es **asegurar que las nuevas funcionalidades o cambios no han introducido defectos en las funcionalidades existentes y que el flujo principal de operación sigue siendo estable.**

El conjunto de pruebas de regresión se enfocará en el **flujo crítico de la experiencia del cliente de principio a fin**:

* **RF-01 (Regresión-Flujo):**
  1. Un **Anfitrión** crea una reserva para un cliente en el sistema.
  2. El cliente llega y el **Anfitrión** encuentra la reserva y le asigna una mesa, cambiando su estado a Ocupada.
  3. Un **Mozo** toma un pedido (que incluye comida y bebestibles) desde un dispositivo móvil para esa mesa.
  4. Se verifica que la comanda llega instantáneamente y de forma correcta a las pantallas de **Cocina** y **Bar**.
  5. El personal de **Cocina/Bar** marca el pedido como Listo.
  6. El **Mozo** recibe la notificación y entrega el pedido.
  7. El cliente pide la cuenta. El **Cajero** la genera y verifica que el cálculo es automático y preciso.
  8. El **Cajero** registra el pago con uno de los medios aceptados (tarjeta).
  9. Se verifica que la mesa cambia su estado a Esperando Limpieza, quedando disponible para el siguiente ciclo.

Las pruebas de regresión manual de baja prioridad (ej. modificación de datos de un usuario) se ejecutarán si el tiempo lo permite.

#### **Fuera del Alcance**

* Pruebas sobre el hardware específico (durabilidad de las tablets, impresoras térmicas, etc.). Se asume que los dispositivos son funcionales.
* Pruebas de los sistemas operativos base de los dispositivos (se probará la aplicación, no el SO).
* Pruebas de seguridad exhaustivas de la red Wi-Fi del restaurante.

#### **Pruebas de Rendimiento**

Se realizarán pruebas de rendimiento para validar que el sistema mantiene tiempos de respuesta aceptables durante los horarios de máxima demanda (ej. sábado entre las 20:00 y 22:00 horas), simulando múltiples mozos tomando pedidos y varias mesas siendo gestionadas simultáneamente.

#### **Pruebas de Aceptación (UAT)**

Las pruebas de aceptación serán realizadas y coordinadas con la gerencia de "Los Naranjos". Se escogerán usuarios finales clave para validar que el sistema es útil, intuitivo y resuelve los problemas operativos del día a día.

* **Usuarios seleccionados:** Se escogerá al menos un empleado por cada rol principal: Anfitrión, Mozo, Cajero y el Gerente.
* **Objetivo:** Validar que el flujo de trabajo en el sistema es más eficiente que el proceso manual anterior.

#### **Infraestructura**

El laboratorio de pruebas debe tener instalados los siguientes componentes para simular el ambiente productivo:

* 2 Dispositivos móviles (Tablet/Smartphone) para el rol de Mozo.
* 1 Tablet o PC para el rol de Anfitrión en la recepción.
* 2 Pantallas (pueden ser tablets) para simular el KDS de Cocina y Bar.
* 1 PC de escritorio con una impresora térmica para el rol de Cajero.
* Una red Wi-Fi aislada y estable que interconecte todos los dispositivos.

#### **Suposiciones**

* La infraestructura de red (Wi-Fi) en el restaurante es estable y tiene buena cobertura en todo el local.
* El personal clave seleccionado para las UAT tendrá disponibilidad en los horarios acordados.
* El menú del restaurante, con sus categorías y precios correctos, ha sido cargado en el sistema antes del inicio de las pruebas.

#### **Riesgos**

| **No** | **Riesgos** | **Probabilidad (1-5)** | **Impacto (1-5)** | **Severidad (Prob\*Impct)** | **Plan de Mitigación** |
| --- | --- | --- | --- | --- | --- |
| 1 | Retrasos en la implementación de las funcionalidades del MVP. | 3 | 5 | 15 | Realizar seguimiento semanal del avance del desarrollo y re-planificar el ciclo de pruebas si es necesario. Priorizar las pruebas de los módulos ya entregados. |
| 2 | Resistencia al cambio por parte del personal del restaurante. | 3 | 4 | 12 | Involucrar a los usuarios finales desde etapas tempranas (UAT). Realizar jornadas de capacitación enfocadas en los beneficios directos para su trabajo. |
| 3 | Conectividad Wi-Fi deficiente en algunas zonas del restaurante (ej. terraza). | 2 | 5 | 10 | Realizar un mapa de calor de la cobertura Wi-Fi antes del despliegue. Asegurar que el plan de contingencia del sistema (modo offline) es robusto y ha sido probado. |
| 4 | Los usuarios seleccionados para UAT no están disponibles en las fechas planificadas. | 2 | 4 | 8 | Coordinar con la gerencia la selección de los usuarios y un plan de respaldo con al menos dos semanas de antelación. Ofrecer flexibilidad en los horarios de prueba. |

**9.3. Anexo C: Demostración funcional del prototipo y MVP**

El presente anexo contiene material audiovisual que complementa la descripción técnica del prototipo presentado en las secciones 3 y 6.3 del informe, mostrando su funcionamiento en condiciones reales de operación.

Todos los videos están contenidos y propiamente etiquetados en el README del siguiente repositorio: [link a repositorio](https://github.com/Svkkit/IngSoftware-AOC)