

06/04/2025

GASTRO & HUB

Trabajo de Fin de Grado



Abel Moro Paje

DESARROLLO DE APLICACIONES MULTIPLATAFORMA 2024/25

CONTENTS

1. Resumen	2
2. Introducción	2
Módulos Implicados en el Desarrollo del Proyecto	2
Introducción a la Solución Objeto de Este Proyecto	3
3. Justificación del Proyecto y Objetivos	3
Estado del Arte.....	3
Objetivos del Proyecto.....	3
4. Análisis	4
Introducción	4
Diagrama de contexto.....	4
Módulos Funcionales	5
Requisitos funcionales	7
Requisitos no funcionales	15
Recursos y tecnologías necesarias.....	18
Planificación de realización del proyecto	23
5. Diseño	24
Arquitectura del sistema.....	24

Bienvenido a la documentación del proyecto "Gastro & Hub", una aplicación multiplataforma para la gestión hostelera desarrollada como Trabajo Final de Grado (TFG) del CFGS Desarrollo de Aplicaciones Multiplataforma (DAM).

1. RESUMEN

"Gastro & Hub" es una aplicación multiplataforma diseñada para revolucionar la gestión de bares y restaurantes, desde pequeños negocios locales hasta establecimientos medianos. En un sector donde la rapidez, la coordinación y el control de recursos son cruciales, este proyecto resuelve los problemas de comunicación lenta entre camareros y cocina, la falta de visibilidad sobre el inventario y la complejidad de gestionar pagos y finanzas de forma sencilla. La solución permite a los camareros registrar pedidos desde sus móviles en tiempo real, al dueño controlar existencias con análisis predictivo para prever compras, y a todo el equipo coordinarse con notificaciones instantáneas y un diseño claro de mesas.

Además, ofrece pagos digitales accesibles y un cierre diario que simplifica la contabilidad. Desarrollada como parte de la formación en Desarrollo de Aplicaciones Multiplataforma, "Gastro & Hub" destaca por ser una herramienta moderna, de muy bajo coste (solo requiere hosting de la base de datos), adaptable a cualquier negocio y accesible desde dispositivos cotidianos, mejorando la operación diaria y la experiencia del cliente en el entorno hostelero.

2. INTRODUCCIÓN

Este proyecto nace en el marco del Trabajo Final de Grado del CFGS en Desarrollo de Aplicaciones Multiplataforma (DAM), como una oportunidad para aplicar los conocimientos adquiridos en un caso real y relevante: la gestión tecnológica de bares y restaurantes, un sector clave en España que enfrenta retos como la digitalización accesible y la eficiencia operativa. "Gastro & Hub" es una propuesta ambiciosa que combina modernidad y practicidad para ofrecer una solución integral, adaptada a las demandas actuales del mercado hostelero.

MÓDULOS IMPLICADOS EN EL DESARROLLO DEL PROYECTO

- **Programación:** Desarrollo del backend con Java y Spring Boot, y del frontend con Dart y Flutter, aplicando lógica para coordinar funciones en tiempo real.
- **Bases de Datos:** Diseño y manejo de PostgreSQL para almacenar y organizar información de pedidos, existencias y empleados.
- **Sistemas Informáticos:** Uso de Docker para configurar y desplegar la aplicación en entornos locales o escalables.
- **Desarrollo de Interfaces:** Creación de pantallas intuitivas y accesibles en Flutter, adaptadas a móviles y ordenadores.

INTRODUCCIÓN A LA SOLUCIÓN OBJETO DE ESTE PROYECTO

"Gastro & Hub" actúa como un centro operativo digital para bares y restaurantes, conectando a camareros, cocineros y dueños en una plataforma única y moderna. Permite registrar pedidos al instante desde cualquier móvil, gestionar mesas y reservas con un diseño visual, controlar existencias de ingredientes y platos con análisis predictivo, aceptar pagos digitales directamente desde la app, y generar informes detallados para tomar decisiones informadas. Construida con herramientas actuales como Spring Boot, Flutter y PostgreSQL, y desplegada mediante Docker, esta solución es de muy bajo coste, adaptable a cualquier tipo de negocio y accesible desde dispositivos estándar, marcando una diferencia en el sector hostelero.

3. JUSTIFICACIÓN DEL PROYECTO Y OBJETIVOS

ESTADO DEL ARTE

El mercado de soluciones tecnológicas para la hostelería incluye sistemas como Toast POS, Square y OpenTable, que gestionan pedidos, reservas y pagos. Sin embargo, estas herramientas suelen ser caras, dependen de hardware especializado (como terminales POS propietarios) y están orientadas a grandes cadenas o mercados extranjeros, dejando a los pequeños bares y restaurantes de España con opciones limitadas. Estos negocios a menudo usan procesos manuales (libretas, tickets impresos) o software genérico que no ofrece coordinación instantánea, análisis predictivo de compras ni pagos digitales sencillos.

Según tendencias recientes, más del 60% de los pequeños negocios hosteleros en España no adopta sistemas integrados por su coste o complejidad (datos aproximados de 2024). "Gastro & Hub" llena este hueco con una solución moderna, de muy bajo coste (solo requiere hosting de la base de datos), accesible desde móviles estándar y adaptable a cualquier negocio. Su enfoque en análisis predictivo, pagos digitales y coordinación en tiempo real la posiciona como una alternativa innovadora frente a competidores tradicionales.

OBJETIVOS DEL PROYECTO

Desarrollar una aplicación multiplataforma para:

- Permitir a los camareros registrar pedidos y gestionar mesas desde sus móviles en tiempo real, asegurando un servicio rápido y sin errores.
- Ayudar al dueño a controlar el inventario de ingredientes y platos, con análisis predictivo que sugiera compras basadas en tendencias para evitar faltantes y desperdicios.
- Facilitar pagos digitales directamente desde la app y un cierre diario de caja claro, simplificando la gestión financiera del negocio.
- Generar informes detallados sobre pedidos, existencias y reservas, con predicciones que apoyen decisiones como ajustar menús o planificar personal.

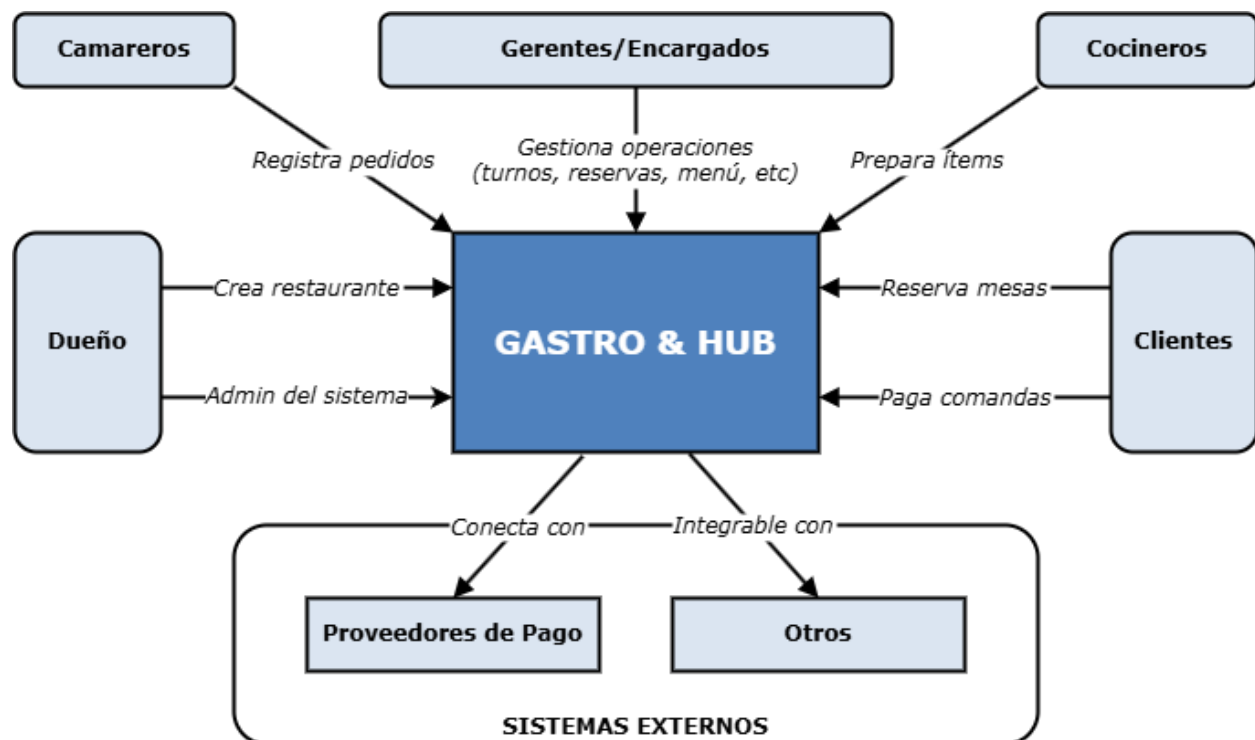
- Coordinar al equipo con notificaciones instantáneas sobre pedidos, existencias bajas o mesas reservadas, mejorando la comunicación interna.
- Permitir al dueño gestionar menús y ofrecer sugerencias de pedidos populares, adaptando la oferta a los clientes.
- Ofrecer una solución moderna, de muy bajo coste y escalable que funcione en cualquier móvil o tablet, adaptándose a bares y restaurantes de todos los tamaños.

4. ANÁLISIS

INTRODUCCIÓN

El apartado de análisis detalla las capacidades principales de 'Gastro & Hub', definiendo cómo la aplicación apoyará las operaciones diarias de bares y restaurantes. Aquí se describen los módulos funcionales que dan forma a la solución, desde el registro inicial del negocio y sus empleados hasta la gestión de pedidos y el análisis del rendimiento, estableciendo un marco claro para entender sus funciones y su interacción con los usuarios. Este análisis sienta las bases para especificar cómo la aplicación responderá a las necesidades del equipo hostelero, priorizando la simplicidad, la coordinación en tiempo real y un acceso controlado que comienza con el dueño como creador del restaurante.

DIAGRAMA DE CONTEXTO



MÓDULOS FUNCIONALES

1. REGISTRO DEL RESTAURANTE

Este módulo permite al dueño iniciar el uso de la aplicación registrándose y creando un restaurante único para su negocio. Tras completar el pago del servicio, que cubre el hosting de los datos, se genera un espacio exclusivo donde toda la operativa del establecimiento quedará organizada, sentando las bases para las demás funciones.

2. GESTIÓN DE USUARIOS Y ROLES

Una vez creado el restaurante, el dueño invita a los empleados (camareros, cocineros, gerentes) y les asigna roles que determinan lo que cada uno puede hacer dentro de la aplicación. Los gerentes tienen permisos adicionales, como organizar turnos o reservas, mientras que solo el dueño puede añadir nuevos usuarios o cambiar la configuración principal, asegurando un control claro y jerárquico sobre quién participa en la gestión.

3. ORGANIZACIÓN DE MESAS

Este módulo permite al dueño o gerente diseñar un plano visual del restaurante, colocando las mesas disponibles con sus números correspondientes. Este diseño sirve como base para asignar pedidos y reservas, proporcionando una vista clara del espacio físico que todos los empleados usarán en su día a día.

4. GESTIÓN DE RESERVAS

Con las mesas definidas, el equipo puede registrar reservas para los clientes, indicando fechas, horas y el número de personas. El módulo muestra qué mesas están disponibles, reservadas o ocupadas, permitiendo actualizar su estado en cualquier momento para planificar el servicio de manera eficiente.

5. GESTIÓN DE INVENTARIO

Este módulo ayuda al dueño a controlar las existencias de ingredientes, bebidas y platos preparados, registrando lo que hay disponible y alertando cuando algo está por agotarse. Incluye análisis predictivo basado en tendencias de consumo para sugerir compras, ayudando a evitar faltantes o desperdicios y optimizando los recursos del negocio.

6. GESTIÓN DE MENÚS

Con el inventario establecido, este módulo permite al dueño crear y actualizar los menús del restaurante, usando los ingredientes y existencias disponibles para definir una carta fija y ofertas especiales (como menús del día o promociones). Ofrece la posibilidad de sugerir

pedidos populares a los camareros o clientes, adaptando la oferta según las preferencias y el consumo habitual, para mejorar la experiencia y las ventas.

7. REGISTRO DE PEDIDOS

Con los menús listos, los camareros pueden registrar los pedidos de los clientes desde sus móviles, seleccionando ítems del menú y asignándolos a una mesa específica del plano. Los pedidos se envían a la cocina en tiempo real, iniciando el proceso de preparación y coordinación entre sala y cocina.

8. PREPARACIÓN DE PEDIDOS

Este módulo permite a los cocineros recibir los pedidos y marcar cada plato o ítem individual dentro de una comanda como listo a medida que lo terminan. Una vez que todos los ítems están preparados, el cocinero indica que la comanda está completa, notificando al camarero que puede proceder con el servicio o el pago.

9. GESTIÓN DE NOTIFICACIONES

Este módulo proporciona alertas instantáneas al equipo sobre eventos importantes, como nuevos pedidos enviados a la cocina, ítems listos para servir, existencias bajas o mesas reservadas próximas a llegar. Mejora la comunicación interna, asegurando que todos estén informados y puedan actuar a tiempo.

10. GESTIÓN DE PAGOS

Una vez completados los pedidos, este módulo permite a los clientes pagar sus cuentas directamente desde la aplicación, usando métodos digitales como códigos QR, mientras el dueño puede registrar ingresos en efectivo si los hay. Cada pago se asocia a una comanda, preparando el terreno para el cierre financiero del día.

11. CIERRE DE CAJA

Al final del día, este módulo genera un resumen claro de todos los ingresos, desglosando los pagos digitales y en efectivo, para que el dueño pueda cerrar la caja de forma sencilla. Proporciona una visión general de las finanzas diarias, conectando los pedidos completados con el dinero recibido.

12. GESTIÓN DE INFORMES

Este módulo genera resúmenes detallados para el dueño sobre el rendimiento del negocio, incluyendo pedidos completados, consumo de inventario, reservas atendidas y tendencias de ventas. Con estas predicciones y datos, se pueden tomar decisiones informadas, como ajustar menús, planificar compras o redistribuir el personal.

REQUISITOS FUNCIONALES

1. REGISTRO DEL RESTAURANTE

RF-001: Registrar usuario

- El usuario inicia el uso de la aplicación creando una cuenta personal con su nombre, email y contraseña.
- Condición: Ninguna, es el primer paso para usar la aplicación.

RF-002: Crear un restaurante

- El dueño define un restaurante en la aplicación, dándole un nombre único que lo identifique como su negocio.
- Condición: El usuario debe estar registrado (RF-001).

RF-003: Pagar el servicio del restaurante

- El dueño realiza un pago para cubrir el almacenamiento de los datos del restaurante, activando su uso.
- Condición: El restaurante debe estar creado (RF-002).

RF-004: Activar el espacio del restaurante

- La aplicación genera un espacio único para el restaurante del dueño, donde se gestionarán todas sus operaciones.
- Condición: El pago del servicio debe estar confirmado (RF-003).

2. GESTIÓN DE USUARIOS Y ROLES

RF-005: Invitar a un empleado

- El dueño envía una invitación a una persona para que se una al equipo del restaurante, indicando su nombre y contacto.
- Condición: El restaurante debe estar activado (RF-004).

RF-006: Registrarse como empleado

- El empleado invitado crea su propia cuenta en la aplicación con un nombre de usuario y contraseña.
- Condición: El empleado debe haber recibido una invitación del dueño (RF-005).

RF-007: Asignar un rol a un empleado

El dueño define el rol del empleado (camarero, cocinero, gerente) para establecer qué puede hacer en la aplicación.

Condición: El empleado debe estar registrado (RF-006).

RF-008: Acceder al restaurante

- El empleado entra al espacio del restaurante en la aplicación usando sus credenciales, según su rol asignado.
- Condición: El empleado debe tener un rol definido (RF-007).

RF-009: Modificar datos de un empleado

- El dueño cambia el nombre, contacto o rol de un empleado para mantener actualizada la información del equipo.
- Condición: El empleado debe estar registrado y asignado al restaurante (RF-007).

RF-010: Eliminar a un empleado

- El dueño quita a un empleado del restaurante, bloqueando su acceso a la aplicación.
- Condición: El empleado debe estar registrado y asignado al restaurante (RF-007).

3. ORGANIZACIÓN DE MESAS

RF-011: Diseñar el plano de mesas

- El dueño o gerente crea un plano visual del restaurante, colocando mesas con un número único para identificarlas.
- Condición: El restaurante debe estar activado (RF-004) y el usuario debe ser dueño o gerente (RF-007).

RF-012: Añadir una mesa al plano

- El dueño o gerente agrega una nueva mesa al plano, asignándole un número que no se repita.
- Condición: El plano del restaurante debe estar creado (RF-011).

RF-013: Mover una mesa en el plano

- El dueño o gerente cambia la posición de una mesa en el plano para ajustarla al espacio real.
- Condición: La mesa debe estar en el plano (RF-012).

RF-014: Quitar una mesa del plano

- El dueño o gerente elimina una mesa del plano cuando ya no se usa en el restaurante.
- Condición: La mesa debe estar en el plano (RF-012) y no estar reservada ni ocupada.

RF-015: Consultar el plano de mesas

- Todos los empleados ven el plano del restaurante con las mesas numeradas y su estado (disponible, reservada, ocupada).
- Condición: El plano debe estar creado (RF-011).

4. GESTIÓN DE RESERVAS

RF-016: Hacer una reserva de mesa

- El gerente o camarero registra una reserva para un cliente, seleccionando una mesa, día, hora y número de personas.
- Condición: El plano de mesas debe estar creado (RF-011) y el usuario debe ser gerente o camarero (RF-007).

RF-017: Confirmar la llegada de una reserva

- El gerente o camarero marca una mesa reservada como ocupada cuando el cliente llega al restaurante.
- Condición: La mesa debe estar reservada (RF-016).

RF-018: Cancelar una reserva

- El gerente o camarero marca una mesa reservada como disponible si el cliente no llega o cancela la reserva.
- Condición: La mesa debe estar reservada (RF-016).

RF-019: Ver el estado de las mesas

- El equipo consulta el plano del restaurante para ver qué mesas están disponibles, reservadas u ocupadas, con la hora de cada estado.
- Condición: El plano de mesas debe estar creado (RF-011).

5. GESTIÓN DE INVENTARIO

RF-020: Registrar un ingrediente en el inventario

- El dueño añade un nuevo ingrediente, como "carne" o "harina", indicando su nombre, unidad (kilos, litros) y cantidad inicial.
- Condición: El restaurante debe estar activado (RF-004).

RF-021: Registrar un producto preparado en el inventario

- El dueño añade un producto preparado, como "hamburguesa" o "refresco", indicando su nombre y cantidad inicial en unidades.
- Condición: El restaurante debe estar activado (RF-004).

RF-022: Añadir más cantidad a un ingrediente

- El dueño aumenta la cantidad de un ingrediente existente, como al recibir un pedido del proveedor.
- Condición: El ingrediente debe estar registrado (RF-020).

RF-023: Añadir más cantidad a un producto preparado

- El dueño aumenta la cantidad de un producto preparado existente, como al cocinar más unidades.
- Condición: El producto preparado debe estar registrado (RF-021).

RF-024: Modificar datos de un ingrediente

- El dueño cambia el nombre, unidad o cantidad de un ingrediente para corregirlo o actualizarlo.
- Condición: El ingrediente debe estar registrado (RF-020).

RF-025: Modificar datos de un producto preparado

- El dueño cambia el nombre o cantidad de un producto preparado para ajustarlo.
- Condición: El producto preparado debe estar registrado (RF-021).

RF-026: Eliminar un ingrediente del inventario

- El dueño quita un ingrediente del inventario cuando ya no se usa en el restaurante.
- Condición: El ingrediente debe estar registrado (RF-020) y no estar en el menú ni pedidos activos.

RF-027: Eliminar un producto preparado del inventario

- El dueño quita un producto preparado del inventario cuando ya no se usa.
- Condición: El producto preparado debe estar registrado (RF-021) y no estar en el menú ni pedidos activos.

RF-028: Establecer un nivel mínimo para un ingrediente

- El dueño define cuántas unidades de un ingrediente deben quedar antes de que se considere bajo.
- Condición: El ingrediente debe estar registrado (RF-020).

RF-029: Establecer un nivel mínimo para un producto preparado

- El dueño define cuántas unidades de un producto preparado deben quedar antes de que se considere bajo.
- Condición: El producto preparado debe estar registrado (RF-021).

RF-030: Consultar el inventario

- El dueño ve una lista de todos los ingredientes y productos preparados, mostrando su cantidad y si están bajos o agotados.
- Condición: Debe haber ingredientes o productos registrados (RF-020 o RF-021).

RF-031: Restar ingredientes al completar un pedido

- La aplicación reduce la cantidad de ingredientes usados en un pedido cuando este se completa, según los productos seleccionados.

- Condición: El pedido debe estar completado (RF-045).

RF-032: Restar productos preparados al completar un pedido

- La aplicación reduce la cantidad de productos preparados usados en un pedido cuando este se completa.
- Condición: El pedido debe estar completado (RF-045).

RF-033: Alertar sobre ingredientes bajos

- El dueño recibe una alerta cuando un ingrediente cae por debajo de su nivel mínimo, indicando que debe reponerlo.
- Condición: El nivel mínimo debe estar definido (RF-028).

RF-034: Alertar sobre productos preparados bajos

- El dueño recibe una alerta cuando un producto preparado cae por debajo de su nivel mínimo, indicando que debe prepararlo.
- Condición: El nivel mínimo debe estar definido (RF-029).

6. GESTIÓN DE MENÚS

RF-035: Crear una categoría de menú

- El dueño añade una categoría, como "bebidas" o "postres", para organizar los productos que se ofrecen a los clientes.
- Condición: El restaurante debe estar activado (RF-004).

RF-036: Añadir un producto al menú

- El dueño incluye un producto del inventario en el menú, como un plato o bebida, asignándole un precio y una categoría.
- Condición: El producto debe estar en el inventario (RF-020 o RF-021) y la categoría creada (RF-035).

RF-037: Modificar un producto del menú

- El dueño cambia el nombre, precio o categoría de un producto en el menú para ajustarlo.
- Condición: El producto debe estar en el menú (RF-036).

RF-038: Eliminar un producto del menú

- El dueño quita un producto del menú cuando ya no se ofrece a los clientes.
- Condición: El producto debe estar en el menú (RF-036) y no en pedidos activos.

RF-039: Consultar el menú para pedidos

- Los camareros ven una lista organizada por categorías con todos los productos del menú al registrar pedidos.

- Condición: Debe haber productos en el menú (RF-036).

RF-040: Crear una oferta especial

- El dueño define una oferta especial, como un menú del día, combinando productos del inventario con un precio único.
- Condición: Debe haber productos en el inventario (RF-020 o RF-021).

RF-041: Generar un documento del menú

- El dueño crea un documento con todas las categorías y productos del menú para compartirlo con los clientes.
- Condición: Debe haber productos en el menú (RF-036).

RF-042: Generar un código visual del menú

- El dueño crea un código visual que lleva al documento del menú, para que los clientes lo vean fácilmente.
- Condición: El documento del menú debe estar generado (RF-041).

RF-043: Ver el documento y código del menú

- Todos los empleados consultan el documento del menú y su código visual en la aplicación para compartirlo con los clientes.
- Condición: El código visual debe estar creado (RF-042).

7. REGISTRO DE PEDIDOS

RF-044: Registrar un nuevo pedido

- El camarero crea un pedido para una mesa, seleccionando productos del menú y enviándolo a la cocina en tiempo real. Una comanda es el pedido completo de una mesa, formado por uno o más ítems del menú.
- Condición: El plano de mesas debe estar creado (RF-011) y el menú debe tener productos (RF-036).

RF-045: Añadir una nota a un pedido

- El camarero incluye una nota en un pedido, como "sin sal", para indicar preferencias del cliente antes de prepararlo.
- Condición: El pedido debe estar registrado (RF-044) y no en preparación.

RF-046: Marcar un pedido como urgente

- El camarero señala un pedido como prioritario para que la cocina lo prepare antes que otros.
- Condición: El pedido debe estar registrado (RF-044) y no haber más de cinco pedidos urgentes.

RF-047: Mostrar el tiempo estimado de un pedido

- El camarero ve cuánto tiempo se espera que tarde la cocina en preparar un pedido, según los productos seleccionados.
- Condición: El pedido debe estar registrado (RF-044).

RF-048: Consultar los pedidos activos

- Los camareros y cocineros ven una lista de pedidos actuales, filtrada por estado, mesa o fecha, para seguir su progreso.
- Condición: Debe haber pedidos registrados (RF-044).

RF-049: Cancelar un pedido

- El camarero elimina un pedido antes o durante su preparación, indicando un motivo como "error del cliente".
- Condición: El pedido debe estar registrado (RF-044) y requiere confirmación.

8. PREPARACIÓN DE PEDIDOS

RF-050: Empezar la preparación de un pedido

- El cocinero marca un pedido como "en preparación" cuando comienza a trabajar en él, indicando que ya no se puede cambiar.
- Condición: El pedido debe estar registrado (RF-044) y pendiente.

RF-051: Marcar un ítem como listo

- El cocinero indica que un ítem dentro de un pedido, como un plato o bebida, está preparado y listo para servir.
- Condición: El pedido debe estar en preparación (RF-050).

RF-052: Completar un pedido

- El cocinero marca un pedido como "completado" cuando todos sus ítems están listos, avisando al camarero.
- Condición: Todos los ítems del pedido deben estar marcados como listos (RF-051).

RF-053: Rechazar un pedido

- El cocinero rechaza un pedido si no puede prepararlo, indicando un motivo como "sin ingredientes".
- Condición: El pedido debe estar registrado (RF-044) y pendiente o en preparación, con confirmación.

9. GESTIÓN DE NOTIFICACIONES

RF-054: Alertar a la cocina de un nuevo pedido

- La cocina recibe una alerta inmediata cuando el camarero registra un pedido, mostrando los detalles para prepararlo.
- Condición: El pedido debe estar registrado (RF-044).

RF-055: Alertar al camarero de un ítem listo

- El camarero recibe una alerta cuando un ítem de un pedido está preparado, para organizar el servicio.
- Condición: El ítem debe estar marcado como listo (RF-051).

RF-056: Alertar al camarero de un pedido completado

- El camarero recibe una alerta cuando un pedido está listo para servirse o pagarse.
- Condición: El pedido debe estar completado (RF-052).

RF-057: Alertar al camarero de un pedido rechazado

- El camarero recibe una alerta cuando la cocina rechaza un pedido, mostrando el motivo para actuar.
- Condición: El pedido debe estar rechazado (RF-053).

RF-058: Alertar al dueño de existencias bajas

- El dueño recibe una alerta cuando un ingrediente o producto preparado cae por debajo de su nivel mínimo.
- Condición: El nivel mínimo debe estar definido (RF-028 o RF-029).

RF-059: Alertar al equipo de una reserva próxima

- El equipo recibe una alerta cuando una reserva está a punto de llegar, indicando la mesa y la hora.
- Condición: La reserva debe estar registrada (RF-016).

10.GESTIÓN DE PAGOS

RF-060: Pagar un pedido digitalmente

- El cliente paga una comanda completa usando un método digital, como un código QR mostrado por el camarero.
- Condición: El pedido debe estar completado (RF-052).

RF-061: Registrar un pago en efectivo

- El camarero o dueño registra un pago en efectivo para una comanda, indicando la cantidad recibida.
- Condición: El pedido debe estar completado (RF-052).

RF-062: Confirmar el pago de un pedido

- La aplicación marca una comanda como "pagada" tras recibir el pago, ya sea digital o en efectivo.
- Condición: El pago debe estar registrado (RF-060 o RF-061).

11. CIERRE DE CAJA

RF-063: Generar el resumen diario de caja

- El dueño ve un resumen de todos los pagos recibidos en el día, dividido entre digitales y en efectivo.
- Condición: Debe haber comandas pagadas (RF-062).

RF-064: Ajustar el resumen diario

- El dueño modifica el resumen diario, añadiendo o corrigiendo ingresos como propinas no registradas.
- Condición: El resumen diario debe estar generado (RF-063).

12. GESTIÓN DE INFORMES

RF-065: Consultar un informe de pedidos

- El dueño ve un resumen de los pedidos del día, incluyendo completados, rechazados y pendientes.
- Condición: Debe haber pedidos registrados (RF-044).

RF-066: Consultar un informe de inventario

- El dueño ve un resumen del consumo de ingredientes y productos preparados, mostrando uso y existencias.
- Condición: Debe haber productos en el inventario (RF-020 o RF-021).

RF-067: Consultar un informe de reservas

- El dueño ve un resumen de reservas atendidas y pendientes para planificar el uso de mesas.
- Condición: Debe haber reservas registradas (RF-016).

RF-068: Consultar un informe de tendencias

- El dueño ve un análisis de ventas y consumo con predicciones sobre compras y ajustes al menú.
- Condición: Debe haber pedidos completados (RF-052).

REQUISITOS NO FUNCIONALES

PRESTACIONES

RNF-001: Soportar ejecución en múltiples plataformas

- La aplicación debe ejecutarse sin errores en sistemas operativos móviles y de escritorio, con un diseño responsive que se adapte a pantallas de 320x480 píxeles hasta 1920x1080 píxeles.
- Criterio: Probado en entornos móviles y de escritorio con diferentes resoluciones, asegurando funcionalidad completa.

RNF-002: Garantizar tiempos de respuesta rápidos

- Las operaciones críticas (registro de comandas, actualización de estados, consultas de inventario) deben tener un tiempo de respuesta promedio inferior a 2000 ms bajo carga de 100 usuarios concurrentes y 10,000 registros.
- Criterio: Verificado en pruebas de rendimiento con carga simulada, manteniendo latencia máxima de 2000 ms.

RNF-003: Gestionar alta concurrencia de usuarios

- La aplicación debe soportar hasta 100 usuarios simultáneos ejecutando operaciones críticas con un tiempo de respuesta máximo de 2500 ms.
- Criterio: Evaluado en pruebas con 100 usuarios activos, asegurando estabilidad y respuesta dentro del límite.

RNF-004: Procesar notificaciones en tiempo real

- Las alertas (nuevos pedidos, ítems listos, existencias bajas) deben entregarse a los usuarios en menos de 1000 ms tras el evento que las genera.
- Criterio: Medido en pruebas de eventos simulados, confirmando entrega en menos de 1000 ms.

SEGURIDAD**RNF-005:** Almacenar contraseñas con cifrado seguro

- Las contraseñas deben almacenarse usando un algoritmo de hash con sal (mínimo 12 rondas), garantizando que no sean recuperables en texto claro.
- Criterio: Comprobado que las contraseñas no se puedan descifrar ni exponer en el sistema.

RNF-006: Gestionar sesiones con autenticación segura

- Las sesiones deben usar tokens seguros con expiración de 24 horas, renovables manualmente, para proteger el acceso de los empleados.
- Criterio: Validado en pruebas de autenticación, asegurando que los tokens expiren tras 24 horas.

RNF-007: Proteger contra accesos no autorizados

- La aplicación debe bloquear el acceso tras 3 intentos fallidos consecutivos, con un tiempo de bloqueo mínimo de 300 segundos.
- Criterio: Confirmado en pruebas de intentos fallidos, verificando activación del bloqueo.

RNF-008: Asegurar la integridad de los pagos digitales

- Las transacciones digitales deben procesarse con cifrado de extremo a extremo, garantizando que los datos no sean interceptados ni alterados.
- Criterio: Evaluado en pruebas de pago simulado, asegurando integridad de los datos.

MANTENIBILIDAD**RNF-009:** Registrar eventos del sistema

- La aplicación debe guardar un historial en formato estructurado de errores y acciones clave (login, cambios, pagos), con rotación diaria y retención mínima de 7 días.
- Criterio: Revisado que el historial sea completo, legible y accesible por al menos 7 días.

RNF-010: Facilitar actualizaciones continuas

- La aplicación debe soportar actualizaciones sin interrupción del servicio, con un tiempo de despliegue máximo de 300 segundos.
- Criterio: Probado aplicando cambios en un entorno activo, manteniendo disponibilidad.

USABILIDAD**RNF-011:** Garantizar navegación intuitiva según ISO 9241-11

- Las acciones principales (registrar pedido, consultar inventario, gestionar reservas) deben ser accesibles en un máximo de 3 interacciones desde la pantalla inicial.
- Criterio: Validado con usuarios completando tareas en menos de 60 segundos sin asistencia.

RNF-012: Proporcionar retroalimentación inmediata según ISO 9241-11

- Cada acción (enviar pedido, completar comanda) debe mostrar una confirmación visual o sonora en menos de 500 ms tras ejecutarse.
- Criterio: Medido en pruebas, asegurando respuesta en menos de 500 ms.

ACCESIBILIDAD**RNF-013:** Cumplir con contraste mínimo según WCAG 2.1 nivel A

- Los textos y elementos interactivos deben tener un contraste mínimo de 4.5:1 contra el fondo para garantizar legibilidad.
- Criterio: Verificado en pruebas de contraste, cumpliendo el estándar WCAG 2.1 nivel A.

RNF-014: Soportar navegación por teclado según WCAG 2.1

- Todas las funciones deben ser operables con teclado (Tab, Enter, flechas), con foco visible en elementos activos.
- Criterio: Evaluado en pruebas manuales, asegurando navegación completa con teclado.

RNF-015: Incluir etiquetas descriptivas según WCAG 2.1

- Los campos y botones deben tener etiquetas legibles por lectores de pantalla, con descripciones claras de su propósito.
- Criterio: Validado en pruebas con lectores de pantalla, confirmando compatibilidad.

ESCALABILIDAD**RNF-016:** Adaptarse a diferentes tamaños de negocio

- La aplicación debe soportar desde 5 mesas y 2 empleados hasta 50 mesas y 20 empleados sin degradación del rendimiento.
- Criterio: Probado en configuraciones pequeñas y medianas, manteniendo tiempos de respuesta estables.

RNF-017: Soportar integración con sistemas externos

- La aplicación debe permitir conexiones futuras con sistemas externos mediante una interfaz estándar, procesando hasta 1000 solicitudes por minuto.
- Criterio: Evaluado con una simulación de integración, asegurando capacidad de respuesta.

ROBUSTEZ**RNF-018:** Mantener estabilidad bajo fallos parciales

- La aplicación debe seguir funcionando si falla una operación (ej. pago digital), permitiendo uso continuo de otras funciones.
- Criterio: Probado simulando fallos, verificando que el sistema no se detenga completamente.

RECURSOS Y TECNOLOGÍAS NECESARIAS

A continuación, se detallan las herramientas y tecnologías necesarias para el desarrollo, implementación y mantenimiento de "Gastro & Hub", justificadas según los requisitos funcionales (RF) y no funcionales (RNF) definidos.

DESARROLLO FRONTEND**Flutter**

Descripción: Framework de Google para aplicaciones multiplataforma con una sola base de código, usando Dart.

Justificación: Cumple RNF-001, RF-044, RF-039, RF-015, RF-060, RF-048.

Uso: Interfaz para camareros, cocineros, gerentes y dueño en todos los módulos funcionales.

Dart

Descripción: Lenguaje optimizado para UI, usado por Flutter.

Justificación: Implementa RF-047, RF-042, RF-045, RF-039, RNF-011, RNF-012.

Uso: Lógica y validaciones en la interfaz del cliente.

DESARROLLO BACKEND

Spring Boot

Descripción: Framework de Java para aplicaciones backend robustas con configuración mínima.

Justificación: Soporta RF-054, RF-009, RF-063, RF-068, RF-060, RNF-009, RNF-006, RNF-008.

Uso: API REST, WebSocket y cálculos predictivos para todos los módulos funcionales.

Java

Descripción: Lenguaje robusto y portátil.

Justificación: Base para RF-065, RF-041, RF-031, RF-068, RNF-002.

Uso: Desarrollo de reglas de negocio, predicción y procesamiento de datos.

BASE DE DATOS

PostgreSQL

Descripción: Base de datos relacional de código abierto con soporte JSON.

Justificación: Almacena datos de RF-001, RF-020, RF-035, RF-007, RF-063, RF-068, RNF-007, RNF-005.

Uso: Persistencia de toda la información del sistema.

COMUNICACIONES

WebSocket (Spring WebSocket)

Descripción: Protocolo y librería para comunicación bidireccional en tiempo real.

Justificación: Cumple RF-054, RF-055, RF-057, RF-059, RNF-004.

Uso: Notificaciones entre camareros, cocineros y dueño.

REST (Spring REST)

Descripción: Arquitectura para APIs basadas en HTTP, implementada en Spring Boot.

Justificación: Soporta RF-041, RF-048, RF-060, RF-066, RNF-002.

Uso: Endpoints de consulta y gestión.

Google API (Google Sign-In)

Descripción: API para autenticación de usuarios mediante cuentas de Google.

Justificación: Permite RF-001, RF-008, RNF-005, RNF-006.

Uso: Autenticación simplificada para usuarios y empleados mediante cuentas de Google.

GESTIÓN DE PAGOS

Stripe API

Descripción: API para procesamiento de pagos digitales con soporte para transacciones seguras y códigos QR.

Justificación: Implementa RF-060, RNF-008, RNF-002.

Uso: Integración de pagos digitales desde la aplicación.

GENERACIÓN DE DOCUMENTOS Y CÓDIGOS

iTextPDF

Descripción: Librería Java para generar PDFs dinámicos.

Justificación: Implementa RF-041, RNF-013.

Uso: Creación del documento del menú compartible.

ZXing

Descripción: Librería para generar y leer códigos QR.

Justificación: Soporta RF-042, RNF-011.

Uso: Generación de códigos QR descargables.

PRUEBAS

JUnit

Descripción: Framework de pruebas unitarias para Java.

Justificación: Verifica RF-047, RF-052, RF-031, RF-068.

Uso: Pruebas unitarias del backend.

Flutter Test

Descripción: Framework de pruebas integrado en Flutter.

Justificación: Valida RF-048, RF-039, RF-060, RNF-011.

Uso: Pruebas unitarias y de interfaz en frontend.

JMeter

Descripción: Herramienta de Apache para pruebas de carga y rendimiento.

Justificación: Valida RNF-002, RNF-003, RNF-017.

Uso: Pruebas de estrés y rendimiento del sistema.

Testcontainers

Descripción: Librería para pruebas de integración con contenedores Docker.

Justificación: Verifica RF-044, RF-054, RF-060, RF-009.

Uso: Pruebas automatizadas de flujos completos.

CONTROL DE VERSIONES Y COLABORACIÓN**Git**

Descripción: Sistema de control de versiones distribuido.

Justificación: Soporta RNF-010.

Uso: Repositorio del proyecto.

GitHub

Descripción: Plataforma para alojar repositorios Git.

Justificación: Alineado con RNF-010, RNF-017.

Uso: Almacenamiento y gestión del proyecto.

GESTIÓN DE PROYECTO**ProjectLibre**

Descripción: Software de gestión de proyectos de código abierto con soporte para Gantt charts, planificación Agile y DevOps.

Justificación: Cubre RF-001 a RF-068, RNF-016.

Uso: Planificación, seguimiento y simulación de sprints y entregas del proyecto.

Excel

Descripción: Hoja de cálculo para análisis y seguimiento de métricas de proyecto.

Justificación: Permite RF-065, RF-066, RF-068.

Uso: Gestión de métricas como sprint burnout, planificación de sprints y análisis de datos.

ENTORNO DE DESARROLLO**Visual Studio Code (VS Code)**

Descripción: Editor ligero y personalizable para múltiples lenguajes.

Justificación: Soporta RF-044, RF-054, RF-068, RF-060.

Uso: Editor principal para frontend y backend.

IntelliJ IDEA Ultimate

Descripción: IDE avanzado para Java y Spring Boot.

Justificación: Desarrollo para RF-054, RF-063, RF-068.

Uso: Desarrollo y depuración del backend.

Android Studio

Descripción: IDE para desarrollo Android y Flutter.

Justificación: Pruebas de RF-044, RF-060, RF-039, RNF-001.

Uso: Desarrollo y pruebas del frontend en móviles.

DESPLIEGUE

Docker

Descripción: Plataforma para contenedores que empaqueta aplicaciones.

Justificación: Soporta RNF-002, RNF-010, RNF-003.

Uso: Empaquetado del backend y base de datos.

Docker Compose

Descripción: Herramienta para definir y ejecutar aplicaciones multicontenedor.

Justificación: Orquesta RF-054, RF-060, RNF-017.

Uso: Configuración de despliegue local.

Servicios en la Nube (AWS, GCP, Azure)

Descripción: Plataformas para hospedaje en producción.

Justificación: Escalado para RNF-016, RNF-017, RNF-003.

Uso: Despliegue en producción opcional.

HARDWARE

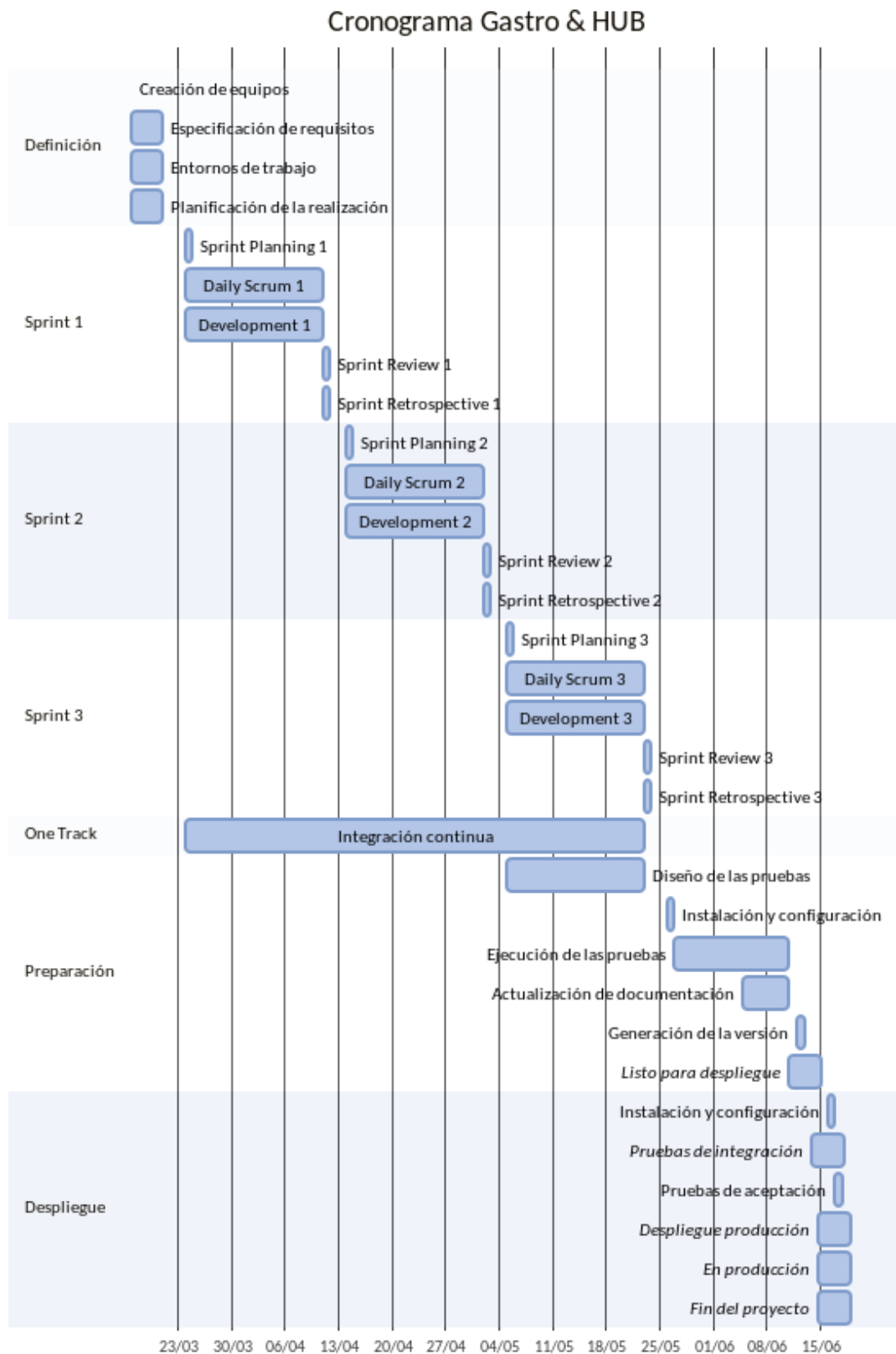
PC con Windows

Descripción: Equipo de desarrollo con mínimo 8GB RAM y procesador de 2GHz o superior.

Justificación: Compatible con RNF-001, RNF-002.

Uso: Desarrollo, pruebas y despliegue local.

PLANIFICACIÓN DE REALIZACIÓN DEL PROYECTO



5. DISEÑO

ARQUITECTURA DEL SISTEMA

La arquitectura de "Gastro & Hub" sigue un modelo cliente-servidor con tres capas (presentación, lógica de negocio, datos), especificando los entornos de desarrollo/pruebas y producción.

ENTORNO DE DESARROLLO Y PRUEBAS:

- **Hardware:** PC con Windows, mínimo 8GB RAM y procesador 2GHz, soporta herramientas de desarrollo (RNF-001).
- **Capa de presentación:** Flutter y Dart ejecutados en emuladores de Android Studio sobre Windows.
- **Capa de lógica de negocio:** Spring Boot y Java en un contenedor Docker, corriendo en un servidor local sobre WSL.
- **Capa de datos:** PostgreSQL en un contenedor Docker, corriendo en el mismo servidor local sobre WSL.
- **Comunicaciones externas:** Stripe API y Google API en modo sandbox, integradas desde el backend.
- **Despliegue:** Docker y Docker Compose orquestan los contenedores en el servidor local.

ENTORNO DE PRODUCCIÓN:

- **Hardware:** Instancias Linux en servicios en la nube (AWS, GCP o Azure), escalables para 100 usuarios simultáneos (RNF-003, RNF-016).
- **Capa de presentación:** Flutter y Dart ejecutados en sistemas Android/iOS reales, para uso operativo (RF-044, RF-060, RNF-001).
- **Capa de lógica de negocio:** Spring Boot y Java en un contenedor Docker, corriendo en un servidor cloud sobre Linux, gestiona operaciones en vivo.
- **Capa de datos:** PostgreSQL en alta disponibilidad en un contenedor Docker, corriendo en el mismo servidor cloud sobre Linux.
- **Comunicaciones externas:** Stripe API y Google API en modo live, integradas desde el backend.
- **Despliegue:** Docker y Docker Compose orquestan los contenedores en la nube.

DIAGRAMAS DE ARQUITECTURA SOFTWARE

