
This is the **published version** of the bachelor thesis:

Ye, Haoran. *Sistema de carta interactiva y gestión de pedidos adaptado a un restaurante de pueblo*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334452>

under the terms of the  license.

Sistema de carta interactiva y gestión de pedidos adaptado a un restaurante de pueblo

Haoran Ye

29 de junio de 2026

Resumen– Este trabajo presenta el diseño y desarrollo de un sistema híbrido de carta interactiva y gestión de pedidos adaptado a las necesidades reales de un restaurante familiar situado en un entorno rural. El proyecto busca reducir la duplicidad de tareas, los errores en la toma de comandas y los problemas de comunicación entre sala y cocina. La solución combina un backend para la gestión de datos y la lógica del sistema, una interfaz para dispositivos móviles, un panel central para la gestión del restaurante y sincronización en tiempo real entre dispositivos. Además, el sistema ha incorporado mejoras a partir del feedback proporcionado por la gerencia durante las pruebas en el entorno real.

Palabras clave– Gestión de pedidos, hostelería, React, Node.js, PostgreSQL, sincronización en tiempo real, sistema POS, restaurante.

Abstract– This project presents the design and development of a hybrid interactive menu and order management system adapted to the real needs of a family restaurant located in a rural environment. The project aims to reduce task duplication, errors in order taking, and communication problems between the dining area and the kitchen. The solution combines a backend for data management and system logic, an interface for mobile devices, a central panel for restaurant management, and real-time synchronization between devices. Furthermore, the system incorporates improvements based on the feedback provided by the management during testing in the real environment.

Keywords– Order management, hospitality, React, Node.js, PostgreSQL, real-time synchronization, POS system, restaurant.



1 INTRODUCCIÓN Y CONTEXTO

LA digitalización de la hostelería se ha extendido con rapidez en grandes ciudades mediante cartas digitales, códigos QR, sistemas de autoservicio y soluciones de punto de venta cada vez más completas. En estos contextos, el volumen de clientes y la rotación constante hacen que la rapidez del servicio sea un factor clave. Sin embargo, esta realidad no se traslada de forma directa a los restaurantes de pueblo, donde el valor principal del esta-

blecimiento no depende únicamente de la eficiencia, sino también de la cercanía, la confianza y el trato personal con clientes habituales. Por este motivo, implantar tecnología en este tipo de entorno exige un enfoque distinto: la herramienta no debe sustituir la relación humana, sino eliminar tareas repetitivas que dificultan el servicio en los momentos de mayor carga.

El proyecto se desarrolla a partir de una necesidad real detectada en un restaurante familiar situado en Tárrega (Lérida). La operativa diaria presenta unas características muy concretas. Por un lado, el equipo de cocina está formado por cocineros de origen chino que trabajan principalmente con códigos numéricos de platos. Por otro lado, el equipo de sala está formado por camareros hispanohablantes, mientras que los propietarios actúan como puente entre ambos grupos. Esta estructura genera una dependencia elevada de la comunicación oral y de la experiencia acumulada

-
- E-mail de contacto: yehaoran2003@gmail.com
 - Menció: Tecnològias de la Informació
 - Trabajo tutorizado por: Kevin Chow (Ingeniería de la Información y de las Comunicaciones)
 - Curso 2025/26

por el personal, especialmente cuando se reciben peticiones especiales como “sin verduras”, “sin picante” o modificaciones equivalentes.

La carta del restaurante incluye aproximadamente 150 platos, lo que obliga a memorizar una gran cantidad de códigos internos. De lunes a viernes la carga de trabajo puede ser asumible, pero durante fines de semana, celebraciones y comidas de grupos se produce un incremento notable de la demanda. En esos picos de actividad, el sistema manual actual provoca retrasos y errores: primero se anota la comanda en la mesa, después se transmite o se copia para cocina y finalmente se introduce de nuevo en el ordenador central para emitir el ticket o la factura. Esta triple introducción de datos consume tiempo, aumenta la probabilidad de equivocaciones y dificulta la rotación de mesas.

El objetivo de este Trabajo Final de Grado es diseñar e implementar una solución tecnológica ajustada a este contexto. No se pretende crear un sistema genérico de hostelería, sino una herramienta específica para un restaurante con restricciones reales de espacio, presupuesto, hardware, idioma y hábitos de trabajo. La contribución principal del proyecto reside precisamente en adaptar la ingeniería informática a un escenario concreto, manteniendo la identidad del restaurante y reduciendo las fricciones del servicio diario.

2 ESTADO DEL ARTE

En esta sección se presentan las principales soluciones tecnológicas relacionadas con la digitalización de la hostelería y su relación con el contexto del proyecto. En primer lugar, se describen los sistemas POS, utilizados habitualmente para centralizar la gestión de ventas y facturación. A continuación, se exponen las cartas digitales y los sistemas de comanda digital, orientados a mejorar la consulta de productos y la toma de pedidos. Finalmente, se analizan las limitaciones que estas herramientas presentan en el caso concreto del restaurante estudiado y se justifica la necesidad de un sistema adaptado.

2.1. Sistemas POS

Los sistemas POS, o puntos de venta, son una de las herramientas más habituales en la digitalización de restaurantes. Su función principal es centralizar operaciones como la gestión de productos, la apertura de mesas, el registro de ventas, el cobro y la emisión de tickets o facturas. En muchos establecimientos, este tipo de sistema actúa como el núcleo de la operativa diaria, ya que permite sustituir parte del trabajo manual por procesos informatizados.

Además de la facturación, algunos sistemas POS incorporan funciones adicionales, como control de stock, estadísticas de ventas, gestión de empleados o integración con dispositivos externos, como impresoras térmicas y cajones portamonedas. Por ello, se trata de soluciones amplias que buscan cubrir diferentes áreas de la gestión de un negocio de hostelería.

2.2. Cartas digitales y códigos QR

Otra solución extendida en los últimos años son las cartas digitales accesibles mediante códigos QR. Estas herramientas

permiten que el cliente consulte la oferta del restaurante desde su propio dispositivo, sin necesidad de utilizar una carta física. Su principal ventaja es la facilidad para actualizar productos, precios, imágenes o información adicional, como alérgenos o descripciones de los platos.

Este tipo de solución está especialmente orientada a la consulta autónoma por parte del cliente. En algunos casos, la carta digital se limita a mostrar información, mientras que en otros puede incorporar funciones de pedido directo. En ambos casos, su objetivo principal es mejorar el acceso a la información y reducir la dependencia de soportes físicos.

2.3. Sistemas de comanda digital

Los sistemas de comanda digital permiten que el personal de sala registre los pedidos desde dispositivos electrónicos, como tablets o terminales móviles. Estos pedidos pueden enviarse posteriormente a cocina, a barra o al ordenador central del establecimiento. Su finalidad es reducir el uso de papel, evitar transcripciones repetidas y agilizar la comunicación interna durante el servicio.

Estas soluciones están directamente relacionadas con uno de los problemas principales del proyecto: la necesidad de evitar que una misma comanda tenga que introducirse varias veces en distintos puntos del proceso. Al digitalizar la toma de pedidos, la información puede compartirse entre diferentes áreas del restaurante y mantenerse de forma más ordenada.

2.4. Limitaciones en el contexto del proyecto

Aunque las soluciones anteriores ofrecen ventajas claras, su aplicación directa no siempre encaja con restaurantes pequeños o con una operativa muy particular. Un sistema POS genérico puede incluir muchas funciones útiles en términos generales, pero no necesariamente adaptadas al flujo de trabajo concreto del establecimiento. En el caso estudiado, el ordenador central se utiliza en un espacio reducido y con una interacción basada principalmente en teclado, por lo que una interfaz pensada para ratón o pantalla táctil no resulta suficiente.

Las cartas digitales mediante QR tampoco resuelven por sí solas el problema principal del restaurante. Aunque permiten consultar la carta, no eliminan la necesidad de que el camarero mantenga el trato directo con el cliente ni solucionan completamente la comunicación entre sala, cocina y caja. Además, una parte de la clientela habitual puede no sentirse cómoda utilizando el móvil para interactuar con el restaurante.

Por otro lado, los sistemas de comanda digital se acercan más a las necesidades del proyecto, pero muchas soluciones existentes están pensadas para flujos de trabajo más estándar. En este caso, era necesario contemplar una carta extensa basada en códigos numéricos, la existencia de una barrera idiomática entre sala y cocina, la necesidad de mantener el ordenador central como punto de control y la importancia de sincronizar la información entre dispositivos.

2.5. Adaptación al contexto del restaurante

A partir del análisis de estas soluciones, se observa que cada una de ellas resuelve una parte del problema. Los siste-

mas POS permiten centralizar la venta, el cobro y la gestión de productos; las cartas digitales facilitan la consulta de la oferta; y los sistemas de comanda electrónica ayudan a reducir el uso de papel y la repetición de tareas. Sin embargo, tratadas de forma aislada, estas soluciones no cubren completamente las necesidades del restaurante estudiado.

Por este motivo, el TFG se orienta al desarrollo de un sistema híbrido que combina conceptos de estas soluciones y los adapta a un contexto concreto. Esta integración busca mejorar la experiencia tanto del personal como del cliente.

3 MOTIVACIÓN

La motivación personal del trabajo nace de aplicar los conocimientos adquiridos durante el grado a un entorno cercano y verificable. El proyecto permite trabajar con tecnologías web modernas, bases de datos relacionales, comunicación en tiempo real, diseño de interfaces y validación con usuarios reales. Además, muestra que la labor de un ingeniero informático no consiste únicamente en programar una aplicación, sino en comprender las restricciones de uso, dialogar con los usuarios y ajustar el alcance cuando aparecen nuevas necesidades.

4 OBJETIVOS

El objetivo general del proyecto es implementar un sistema híbrido de carta interactiva y gestión de pedidos que optimice la operativa interna del restaurante, reduciendo los tiempos de espera y los errores en cocina sin sacrificar el trato personalizado ni la identidad rural del establecimiento.

Para alcanzar este objetivo general, el trabajo se estructura en una serie de objetivos específicos obligatorios. En primer lugar, desarrollar un sistema de gestión de comandas que permita registrar pedidos desde tablets o dispositivos móviles, almacenar la información de forma persistente y sincronizarla en tiempo real con el ordenador central. En segundo lugar, diseñar una base de datos robusta que almacene de forma estructurada la información relativa a productos, mesas, comandas, pagos y facturación. En tercer lugar, implementar un panel centralizado para el ordenador principal del restaurante, desde el cual se puedan gestionar mesas, pedidos, cobros y documentos asociados a la venta. En cuarto lugar, incorporar un sistema de control de acceso que diferencie las funciones disponibles según el tipo de usuario, evitando que acciones críticas como la modificación de productos, precios o datos de facturación puedan ser realizadas por cualquier empleado. En quinto lugar, reducir los errores derivados de la toma manual de comandas, las modificaciones especiales de los platos y la comunicación entre sala y cocina. Finalmente, adaptar el sistema a las necesidades reales del restaurante mediante mejoras detectadas durante las pruebas en el entorno de trabajo.

Además de estos objetivos principales, el proyecto contempla objetivos de valor añadido, como la analítica de ventas, la carta digital informativa para clientes, el filtrado de alérgenos y la adaptación progresiva a los requisitos de los sistemas informáticos de facturación. Estos objetivos se consideran secundarios respecto al núcleo operativo del sistema, por lo que su desarrollo queda condicionado a la

estabilidad y completitud de las funcionalidades principales.

5 ESTRUCTURA GENERAL DEL SISTEMA

Como se muestra en la Fig. 1, el sistema se ha organizado en tres bloques principales: una aplicación frontend para tablets y dispositivos móviles, una aplicación frontend para el ordenador central del restaurante y un backend común encargado de coordinar la lógica del sistema. Esta separación permite adaptar cada interfaz a un contexto de uso diferente, manteniendo la información centralizada en el servidor.

La aplicación *client* está orientada al personal de sala y permite tomar comandas desde tablets o móviles. La aplicación *client-central* está diseñada para el ordenador principal del restaurante, desde donde se gestionan mesas, pedidos, cobros y documentos asociados a la venta. Ambas aplicaciones frontend se han desarrollado con React [1] y se comunican con el backend mediante peticiones API REST para realizar operaciones del sistema, como consultar productos, crear comandas o actualizar datos.

Además de esta comunicación mediante API REST, el sistema incorpora Socket.IO [5] para la sincronización en tiempo real. Esta comunicación permite que los cambios realizados desde un dispositivo, como la modificación de una mesa o de una comanda, puedan reflejarse en el resto de interfaces sin necesidad de recargar manualmente la información.

El backend, ubicado en el bloque *server*, se ha desarrollado con Node.js y Express [2, 6]. Este bloque se encarga de gestionar la lógica principal de la aplicación, procesar las peticiones recibidas y comunicarse con la base de datos. Para ello utiliza Prisma [4] como capa intermedia de acceso a datos sobre PostgreSQL [3]. De esta forma, el sistema mantiene una estructura centralizada y persistente, en la que los datos de productos, mesas, comandas, pagos y documentos de venta quedan almacenados de forma coherente.

6 METODOLOGÍA

El desarrollo del proyecto se ha organizado mediante una metodología ágil e incremental basada en funcionalidades. En lugar de construir el sistema por capas completamente separadas, cada funcionalidad se ha trabajado de forma completa.

El proceso seguido en cada iteración se ha dividido en cuatro fases principales. En primer lugar, se ha realizado el planteamiento de la funcionalidad, definiendo qué problema debía resolver y, cuando era necesario, elaborando bocetos o prototipos para visualizar su comportamiento. En segundo lugar, se ha llevado a cabo la implementación técnica de la funcionalidad, integrando tanto la parte visual como la lógica necesaria para conectarla con el servidor y la base de datos. En tercer lugar, se han realizado pruebas manuales para comprobar que el comportamiento era correcto y que la comunicación entre los distintos componentes funcionaba adecuadamente. Finalmente, se han aplicado reajustes sobre la funcionalidad en función de los errores detectados o de las necesidades observadas.

Esta metodología ha permitido mantener el sistema en un estado funcional durante la mayor parte del desarrollo y evi-

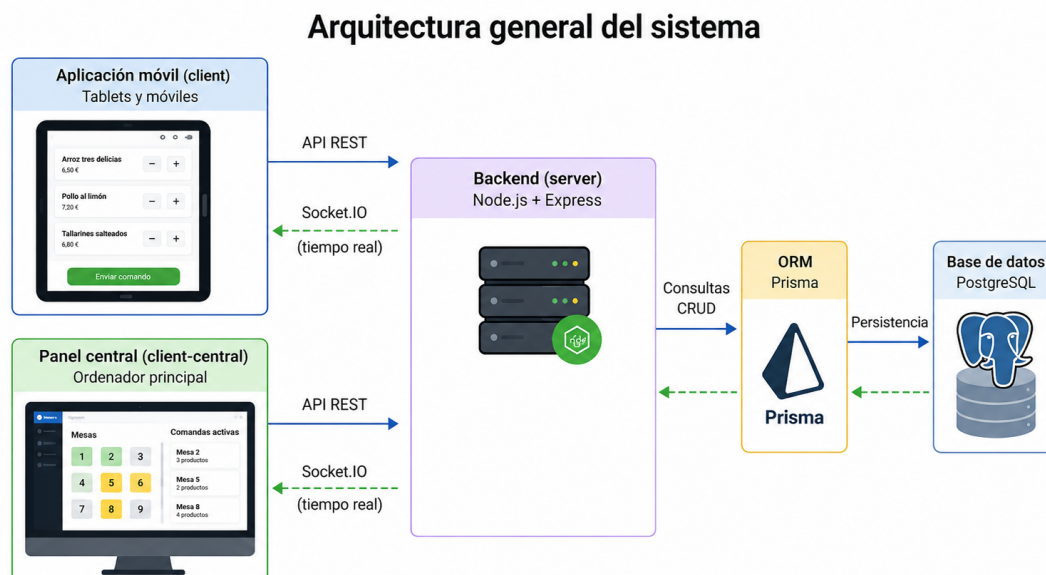


Fig. 1: Esquema de la estructura general del sistema.

tar que los errores se acumularan hasta fases finales del proyecto. Además, el proceso incluyó validaciones puntuales con la gerencia para comprobar la adecuación del sistema al funcionamiento del restaurante.

7 DESARROLLO DEL SISTEMA

En esta sección se describe el desarrollo de los principales bloques que componen el sistema. Cada subsección presenta una parte concreta del proyecto, explicando su función dentro de la solución, las decisiones tomadas durante su implementación y las dificultades o reajustes que aparecieron durante el proceso.

7.1. Interfaz para tablets y móviles

La interfaz para tablets y dispositivos móviles está orientada al personal de sala y tiene como finalidad principal facilitar la toma de comandos directamente desde la mesa. Esta parte del sistema sustituye el proceso tradicional basado en papel por una herramienta digital que permite consultar la carta, seleccionar productos y asociarlos a una mesa concreta.

El diseño de esta interfaz se planteó a partir de las acciones que debía realizar el camarero durante la toma de una comanda: seleccionar una mesa, consultar los productos disponibles, añadir platos al pedido, modificar cantidades y revisar la comanda antes de enviarla. Por este motivo, la prioridad no fue únicamente la adaptación visual a dispositivos móviles, sino asegurar que el flujo de trabajo básico del personal de sala pudiera realizarse de forma clara y sin pasos innecesarios.

Antes de iniciar la implementación, se elaboraron bocetos y prototipos de la interfaz con Figma [7]. Esta fase permitió representar visualmente las pantallas principales

y comprobar si la distribución propuesta respondía al flujo real de trabajo del personal de sala. El uso de prototipos fue útil para ordenar las funcionalidades antes de trasladarlas al código. Al tratarse de una interfaz destinada a tablets y móviles, era importante evitar pantallas sobrecargadas y priorizar las acciones más frecuentes durante el servicio. Los prototipos permitieron comparar distintas formas de distribuir la información y decidir qué elementos debían aparecer en cada momento para que el camarero pudiera trabajar de forma rápida y clara. Los prototipos iniciales se muestran en la Fig. 2.

A partir de este planteamiento, la primera parte implementada fue la consulta de la carta desde la propia aplicación. Para ello, los productos se obtienen desde el backend en lugar de estar definidos de forma fija en la interfaz. Esta decisión permite que la carta se mantenga centralizada y que cualquier cambio en los productos pueda reflejarse en los dispositivos sin modificar manualmente el frontend. Sobre esta base se añadieron las operaciones necesarias para que el camarero pudiera construir una comanda: seleccionar productos, añadirlos al pedido, revisar las líneas introducidas y asociar la comanda a una mesa.

La implementación de esta interfaz también sirvió para validar la comunicación entre la aplicación móvil, el servidor y la base de datos. Al trabajar con datos reales obtenidos desde el backend, fue posible comprobar que las operaciones realizadas en la interfaz tenían efecto sobre el sistema y no quedaban limitadas a una representación visual aislada. De este modo, la aplicación móvil pasó a formar parte del flujo completo de trabajo, conectando la toma de pedidos en sala con el resto del sistema.

Como resultado, la comanda introducida desde sala queda integrada en el flujo general del sistema, evitando que el mismo pedido tenga que introducirse de nuevo en el ordenador central. Además, facilita el trabajo del personal con

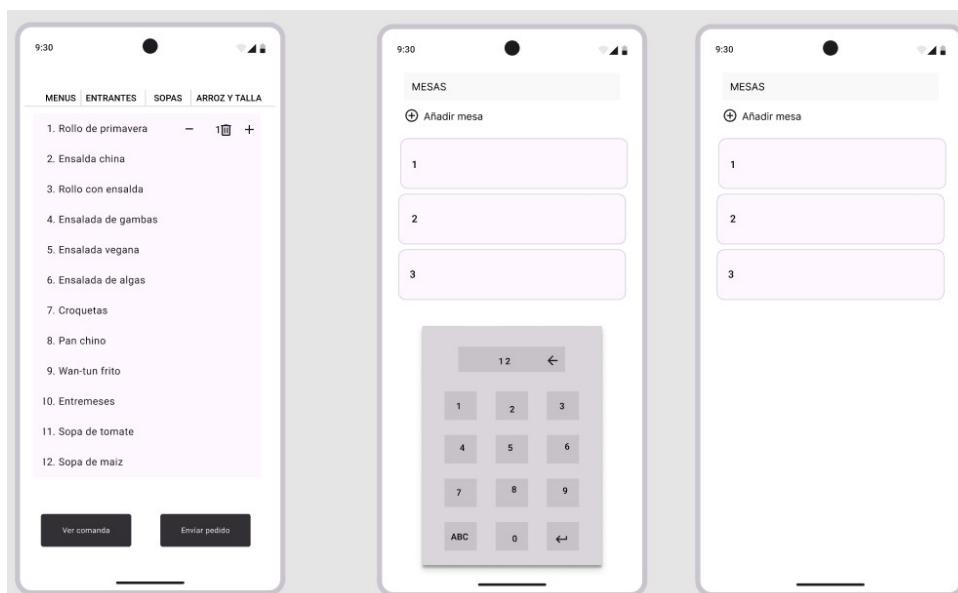


Fig. 2: Bocetos y prototipos iniciales de la interfaz para tablets y dispositivos móviles.

menos experiencia, porque permite consultar los productos desde la aplicación y no depender únicamente de la memorización de los códigos de la carta.

7.2. Interfaz para el ordenador central

La interfaz para el ordenador central se diseñó como el punto principal de gestión operativa del restaurante. Desde esta pantalla se concentran las funciones necesarias para controlar el servicio diario, como la apertura y consulta de mesas, la revisión de comandas, la modificación de líneas de pedido, el acceso al cobro y la preparación de los documentos asociados a la venta.

El diseño de esta interfaz se planteó a partir de las condiciones reales del puesto de trabajo. El ordenador central se encuentra en un espacio reducido y su uso durante el servicio requiere rapidez, por lo que se priorizó una interacción basada en teclado. Este enfoque permite realizar acciones mediante códigos, comandos y atajos, reduciendo la dependencia del ratón y facilitando un flujo de trabajo más ágil en momentos de alta actividad.

Para facilitar la adaptación del personal, la organización visual y funcional del panel tomó como referencia el programa utilizado previamente en el restaurante. El objetivo fue mantener una lógica de uso reconocible, evitando que el cambio de sistema obligara a aprender desde cero una forma completamente distinta de trabajar. Sobre esta base se incorporaron las funciones propias del nuevo sistema, adaptadas a la gestión de mesas, comandas y cobros.

Dentro del panel central se implementaron operaciones orientadas a cubrir situaciones habituales del servicio. Entre ellas se incluye la posibilidad de abrir una mesa, consultar los productos introducidos, modificar cantidades, eliminar líneas de comanda y acceder al proceso de cobro.

El resultado es una interfaz enfocada a la rapidez operativa y al control centralizado del restaurante. Su diseño busca que las tareas principales puedan realizarse con el menor número posible de interrupciones, manteniendo una forma de trabajo cercana a la que el personal ya conoce.

7.3. Backend y persistencia de datos

El backend se desarrolló como el núcleo encargado de gestionar la lógica principal del sistema y de actuar como intermediario entre las interfaces de usuario y la base de datos. Su función consiste en recibir las operaciones realizadas desde las aplicaciones, procesarlas de forma controlada y devolver la información necesaria para mantener el funcionamiento del sistema.

La persistencia de datos se implementó mediante una base de datos relacional gestionada con Prisma. Esta herramienta permite definir el modelo de datos de forma estructurada y facilita la comunicación entre el servidor y la base de datos. A través de este modelo se representan las entidades principales del restaurante, como productos, mesas, comandas, líneas de pedido, pagos y documentos asociados a la venta.

Uno de los aspectos más importantes fue asegurar que la información no dependiera del estado local del navegador. En las primeras pruebas, algunos datos podían mantenerse temporalmente en el propio dispositivo, lo que resultaba insuficiente para un sistema utilizado desde varios puntos del restaurante. Por este motivo, se trasladó la información relevante a la base de datos, garantizando que el estado de mesas, comandas y productos quedara almacenado de forma persistente.

Durante el desarrollo se detectó que el modelo inicial estaba demasiado orientado a la venta y no representaba correctamente la operativa espacial del restaurante. La ausencia de una entidad específica para las mesas dificultaba asociar pedidos a una ubicación concreta y compartir ese estado entre dispositivos. Para corregir esta limitación, se incorporó la mesa como entidad propia dentro del modelo de datos, permitiendo vincular cada comanda a una mesa y reflejar mejor el funcionamiento real del establecimiento. La distribución final de la base de datos se muestra en el Apéndice mediante un esquema UML. Este diagrama permite visualizar las entidades principales del sistema y las relaciones entre ellas, facilitando la comprensión de cómo se organiza la información relativa a mesas, comandas, pro-

ductos, pagos y documentos de venta.

Esta estructura permitió construir una base más sólida para el resto del sistema. Al centralizar la información en el backend y almacenarla de forma persistente, se mejora la coherencia de los datos, se reducen errores derivados de información desactualizada y se facilita la ampliación futura de nuevas funcionalidades.

7.4. Sincronización en tiempo real

La sincronización en tiempo real se incorporó para mantener coherente la información del sistema cuando varios dispositivos trabajan de forma simultánea. En el contexto del restaurante, una misma mesa o comanda puede ser consultada o modificada desde distintos puntos, por lo que resulta necesario que los cambios se reflejen de manera inmediata y no dependan de recargas manuales.

Para resolver esta necesidad se utilizó Socket.IO, que permite establecer una comunicación bidireccional entre el servidor y los clientes conectados. Cuando se produce una acción relevante, como la creación de una comanda, la modificación de un pedido o el cambio de estado de una mesa, el servidor puede emitir un evento para notificar al resto de dispositivos. De esta manera, las interfaces actualizan la información visible sin que el usuario tenga que solicitarla manualmente.

Esta funcionalidad fue especialmente importante en la gestión de mesas. El estado de una mesa no podía depender únicamente del dispositivo desde el que se había abierto o modificado, ya que eso generaba inconsistencias entre la información mostrada en distintas pantallas. Con la sincronización en tiempo real, las mesas y comandas pasan a formar parte de un estado compartido, accesible desde todos los dispositivos conectados al sistema.

7.5. Refactorización del código

Durante las primeras fases del proyecto, el código se desarrolló de forma sencilla para validar rápidamente las funcionalidades principales. Esta forma de trabajo permitió avanzar con rapidez, pero a medida que el sistema fue creciendo algunos archivos empezaron a concentrar demasiadas responsabilidades. En el backend se acumulaban tareas como la definición de rutas, la lógica de negocio, la comunicación con la base de datos, la configuración de Socket.IO y la inicialización del servidor. De forma similar, en la interfaz del ordenador central se concentraban en las pantallas principales funciones relacionadas con la gestión de mesas, la visualización de comandas, el teclado numérico, los paneles de productos y las operaciones de pedido.

Esta acumulación de código dificultaba el mantenimiento y hacía más complejo localizar errores o añadir nuevas funcionalidades. Para solucionar este problema, se realizó una refactorización general del código con el objetivo de separar responsabilidades y mejorar la organización interna del proyecto.

En el backend, la nueva estructura divide el código en varios módulos. Las rutas se encargan de definir los endpoints de la API y recibir las peticiones del frontend. Los servicios contienen la lógica de negocio y las operaciones relacionadas con la base de datos. Por último, las funciones auxiliares

agrupan operaciones reutilizables que no pertenecen directamente a una ruta concreta.

Con esta reorganización, el archivo principal del servidor queda centrado en tareas de inicialización, como configurar Express, activar la comunicación en tiempo real y montar las rutas principales. Esta separación hace que el código sea más legible, facilita la ampliación del sistema y reduce el riesgo de introducir errores al modificar una funcionalidad concreta.

En la interfaz del ordenador central, la refactorización se realizó separando el código en componentes reutilizables y módulos con responsabilidades más concretas. Los componentes agrupan partes visuales y funcionales de la interfaz, como el panel de categorías, el panel de productos, la comanda, el teclado o los elementos relacionados con el pedido. Las páginas se encargan de estructurar las vistas principales, mientras que los servicios concentran la comunicación con el backend. Además, cada componente dispone de su propio archivo de estilos, lo que permite modificar la apariencia de una parte concreta de la interfaz sin tener que revisar una hoja de estilos general.

Esta organización facilita trabajar sobre zonas concretas del panel central. Por ejemplo, si se quiere ajustar la visualización de productos, el cambio puede realizarse directamente en el componente y en el archivo de estilos correspondiente, sin modificar la pantalla principal ni afectar al resto de elementos de la interfaz.

En conjunto, la refactorización del backend y del frontend mejora la mantenibilidad del proyecto. La separación por responsabilidades hace que el sistema sea más fácil de entender, probar y ampliar, algo especialmente importante en un proyecto que ha ido creciendo progresivamente a partir de nuevas necesidades detectadas durante el desarrollo.

8 VALIDACIÓN Y MEJORAS DERIVADAS DEL ENTORNO REAL

Una parte importante del proyecto fue la validación preliminar del sistema en el propio restaurante. Esta prueba permitió comprobar el funcionamiento de la aplicación en un contexto real de uso y no únicamente en un entorno de desarrollo. El objetivo era observar si las funcionalidades implementadas se ajustaban a la forma de trabajo del establecimiento y detectar posibles necesidades no contempladas durante las fases iniciales.

Durante esta validación se recogió feedback de la gerencia del restaurante. A partir de estas observaciones se identificaron necesidades operativas que permitieron ajustar el sistema a situaciones habituales del servicio. Estas mejoras no surgieron como funciones aisladas, sino como respuestas a problemas reales que debían poder resolverse de forma rápida y controlada durante la jornada de trabajo.

8.1. Control de acceso y funciones de administración

Una de las necesidades más importantes detectadas fue el control de acceso por tipo de usuario. En el uso diario del sistema no todos los empleados deben tener los mismos permisos, ya que las responsabilidades de un camarero y las de un administrador son diferentes. Por este motivo, se in-

corporó una separación entre perfiles de trabajo, limitando las acciones disponibles para el personal de sala y reservando las funciones de administración a un perfil con permisos superiores.

La incorporación del control de acceso permitió añadir funcionalidades que no estaban previstas inicialmente con el mismo nivel de prioridad. Entre ellas se encuentran la gestión de productos, la modificación de precios y el acceso a datos relacionados con la facturación. Estas acciones se consideran sensibles porque afectan directamente a la información del negocio, por lo que deben estar protegidas y disponibles únicamente para usuarios autorizados.

8.2. Gestión de documentos de venta

Otra mejora derivada del feedback fue la ampliación del proceso de venta para contemplar distintos tipos de documentos. Además del ticket simplificado, el sistema incorpora la posibilidad de trabajar con factura completa y albarán. Esta funcionalidad permite adaptar el cobro a diferentes situaciones reales del restaurante y responde a una necesidad expresada por la gerencia durante la validación.

La gestión de estos documentos también refuerza la separación entre operaciones básicas y operaciones administrativas. Mientras que el personal de sala puede realizar acciones relacionadas con la toma de comandas y el servicio, las opciones vinculadas a datos fiscales o documentos de venta quedan asociadas a perfiles con mayor nivel de permisos.

8.3. Corrección de errores en comandas

También se incorporó la gestión de errores en comandas. En un restaurante es habitual que se produzcan equivocaciones durante el servicio, como asignar un pedido a una mesa incorrecta. Para evitar tener que cancelar una comanda y volver a introducir todos los productos, se incorporó la posibilidad de traspasar una comanda entre mesas. Esta función permite corregir errores de forma más eficiente y reduce la pérdida de tiempo durante momentos de alta actividad.

Esta mejora resulta importante porque el sistema debe adaptarse a la realidad del servicio, donde los errores operativos pueden ocurrir incluso con herramientas digitales. Permitir correcciones controladas hace que la aplicación sea más útil para el personal y evita que pequeñas equivocaciones interrumpen el ritmo de trabajo.

8.4. Adaptación a una interfaz táctil

Otra mejora planteada a partir de la validación en el entorno real fue la adaptación del panel central a una posible pantalla táctil. En el software utilizado anteriormente por el restaurante, la interacción estaba pensada principalmente para teclado, ya que el establecimiento no disponía de una pantalla táctil y la adquisición de nuevo hardware suponía un coste adicional. Por este motivo, la implementación inicial del proyecto también priorizó el uso mediante teclado, códigos numéricos y botones de operación rápida.

Sin embargo, durante la validación con la gerencia se planteó que, si la interfaz táctil se implementaba de forma adecuada y aportaba una mejora real al servicio, podría valorarse la compra de una pantalla táctil para el ordenador

central. Esto convirtió la adaptación táctil en una mejora relevante, ya que permitiría reducir la dependencia de los códigos de platos y facilitar el uso del sistema por parte de empleados con menos experiencia.

La distribución de la pantalla se diseñó dividiendo el espacio en tres zonas principales, siguiendo una proporción aproximada de 40–20–40 (mostrada en la Fig. 3). La primera zona, situada a la izquierda, ocupa alrededor del 40 % de la pantalla y está destinada a mostrar la comanda activa, incluyendo los platos añadidos, cantidades, precios y total acumulado. Esta zona permite al usuario revisar rápidamente el contenido del pedido antes de enviarlo o cobrarlo.

La zona central ocupa aproximadamente el 20 % de la pantalla y se reserva para la selección táctil de categorías y productos. Su objetivo es permitir que el usuario pueda localizar un plato a partir de su categoría sin necesidad de recordar directamente su código numérico. De esta forma, el sistema mantiene la compatibilidad con el uso tradicional basado en códigos, pero añade una alternativa más visual y accesible.

La última zona, situada a la derecha y con un ancho aproximado del 40 %, agrupa el campo de entrada, el teclado numérico y las operaciones principales de la comanda y del pago. En esta parte se concentran acciones como introducir códigos, modificar líneas, enviar la comanda, aplicar descuentos o acceder al cobro. Esta organización busca que las operaciones más frecuentes estén disponibles en una misma área y puedan ejecutarse de forma rápida durante el servicio.

Esta mejora refleja la evolución del proyecto a partir del feedback recibido. Aunque la pantalla táctil no formaba parte de la planificación inicial, su posible incorporación permite ampliar la usabilidad del sistema y preparar la aplicación para un entorno de trabajo más flexible, combinando el uso por teclado con una interacción táctil más intuitiva.

9 SEGUIMIENTO DE LA PLANIFICACIÓN Y ALCANCE FINAL

La planificación inicial del proyecto se organizó en varias fases consecutivas: configuración del entorno, diseño de la base de datos, desarrollo del sistema, validación en el restaurante, incorporación de funcionalidades de valor añadido y redacción final del informe. Esta planificación sirvió como guía para ordenar el trabajo y establecer una previsión temporal de las tareas principales. El diagrama de Gantt correspondiente a esta planificación inicial se incluye en el Apéndice.

Durante el desarrollo, parte de esta planificación se cumplió según lo previsto. Se configuró el entorno de trabajo, se diseñó una base de datos relacional, se desarrolló una aplicación para la toma de comandas, se creó un panel central para la gestión del restaurante y se incorporó comunicación en tiempo real entre dispositivos. También se realizó una validación preliminar en el restaurante, lo que permitió comprobar el funcionamiento del sistema en un contexto real.

Sin embargo, la planificación inicial también sufrió cambios importantes. Al tratarse de mi primer proyecto completo de estas características, la planificación fue demasiado estricta y poco realista en algunos puntos. No se dejó suficiente margen para errores de desarrollo, reajustes técnicos,

ID	PLATO	CANT.	PRECIO	TOTAL
1	Rollo de primavera	1	2.05 €	2.05 €
2	Ensalada china	1	4.70 €	4.70 €
3	Rollo de primavera con ensalada	1	5.15 €	5.15 €
12	Sopa agrigicante	8	3.95 €	31.60 €
37	Verduras variadas saiteadas	1	5.55 €	5.55 €
50	Langostinos crujientes	1	8.60 €	8.60 €
91	Plato con bambú y setas chinas	1	9.45 €	9.45 €

Total: 67.10 €

Fig. 3: Distribución táctil del panel central para la gestión de comandas y operaciones de venta.

cambios de alcance o necesidades detectadas durante las pruebas. Esta limitación se hizo evidente cuando la validación en el entorno real y el feedback de la gerencia mostraron que algunas funcionalidades no previstas con suficiente prioridad eran necesarias para que el sistema resultara útil en el día a día.

Entre estas funcionalidades destacan varias mejoras operativas surgidas durante la validación, relacionadas con permisos, administración, documentos de venta y corrección de incidencias en comandas. Como consecuencia, el alcance final del proyecto se centró en consolidar el núcleo operativo del sistema: gestión de mesas, toma de comandas, sincronización entre dispositivos, persistencia de datos, panel central, control de acceso y funciones básicas de venta y administración.

Esta decisión implicó posponer funcionalidades inicialmente planteadas como valor añadido, como la analítica de ventas, la carta digital informativa para clientes, el filtrado de alérgenos y la adaptación completa a los requisitos de VERI*FACTU [8, 9]. Estas tareas quedan como líneas de continuación del proyecto, ya que no eran imprescindibles para validar el funcionamiento principal del sistema, pero sí pueden aportar valor en futuras versiones.

En conjunto, esta evolución de la planificación supone una de las lecciones más importantes del proyecto. La planificación inicial fue útil como punto de partida, pero el desarrollo demostró la necesidad de reservar tiempo para imprevistos, pruebas, correcciones y cambios derivados del uso real. Esta experiencia permitió comprender que una planificación eficaz no debe ser únicamente una lista cerrada de tareas, sino una herramienta flexible capaz de adaptarse a la realidad técnica y funcional del proyecto.

10 CONCLUSIONES Y TRABAJO FUTURO

El proyecto ha permitido desarrollar una solución funcional de carta interactiva y gestión de pedidos adaptada a la operativa de un restaurante familiar. La solución final conecta la toma de comandas desde dispositivos móviles con un panel central de gestión, un backend común, una base de datos persistente y comunicación en tiempo real entre dispositivos. De esta forma, se ha conseguido construir un sistema que responde al problema inicial de reducir la duplici-

dad de tareas, mejorar la comunicación interna y disminuir los errores derivados del uso de comandas manuales.

Uno de los resultados más importantes ha sido la consolidación de una estructura técnica capaz de representar mejor el funcionamiento real del restaurante. La incorporación de las mesas como entidad propia en la base de datos, la persistencia de comandas y productos, y la sincronización mediante Socket.IO han permitido que la información no dependa de un único dispositivo ni de estados temporales en el navegador. Esto proporciona una base más sólida para el uso simultáneo del sistema desde sala y desde el ordenador central.

La validación en el entorno real también ha sido clave para orientar el proyecto hacia necesidades concretas del establecimiento. A partir del feedback recibido se incorporaron mejoras como el control de acceso por tipo de usuario, las funciones de administración, la gestión de documentos de venta, la corrección de errores en comandas y la adaptación del panel central a una interfaz táctil. Estas funciones muestran que el sistema no se limita a digitalizar una carta, sino que evoluciona hacia una herramienta de apoyo a la gestión diaria del restaurante.

A nivel personal y técnico, el proyecto ha supuesto un aprendizaje importante sobre el desarrollo de software aplicado a un cliente real. Durante el proceso se hizo evidente la necesidad de planificar con margen suficiente para errores, pruebas y cambios de alcance. También se comprobó la importancia de refactorizar el código cuando el sistema empieza a crecer, ya que una organización interna clara facilita el mantenimiento, la corrección de errores y la incorporación de nuevas funcionalidades.

Como trabajo futuro se plantea completar las funcionalidades de valor añadido que no se han priorizado en esta versión, como la analítica de ventas, la carta digital informativa para clientes y el filtrado de alérgenos. También queda pendiente profundizar en la adaptación completa a los requisitos de VERI*FACTU, ampliar la integración con hardware físico como impresoras térmicas y cajón portamonedas, y validar la interfaz táctil con una pantalla física en el entorno real. Estas líneas permitirían convertir el prototipo funcional desarrollado en una solución más completa y preparada para un uso continuado en el restaurante.

USO DE LA IA GENERATIVA

Durante el desarrollo del proyecto se han utilizado herramientas de inteligencia artificial generativa como apoyo auxiliar. Su uso se ha centrado principalmente en la comprensión de documentación técnica, la traducción de conceptos, la mejora de la redacción académica y la corrección lingüística de los informes.

El uso de estas herramientas no ha sustituido el trabajo de análisis, diseño ni programación. Cada sugerencia generada por IA ha sido revisada críticamente antes de incorporarse al proyecto. La responsabilidad sobre las decisiones técnicas, el contenido del informe, la validación del sistema y la autoría final corresponde íntegramente al estudiante.

AGRADECIMIENTOS

Quiero expresar mi agradecimiento a mi tutor, Kevin Chow, por su orientación, seguimiento y por el material facilitado durante el desarrollo de este Trabajo de Fin de Grado. También agradezco especialmente su paciencia, comprensión y flexibilidad ante las dificultades de compaginar el proyecto con una elevada carga de trabajo.

Mis agradecimientos se dirigen también a la gerencia del restaurante por permitirme realizar pruebas en su propio entorno de trabajo y utilizar el ordenador central del establecimiento para validar el funcionamiento del sistema en condiciones reales.

REFERENCIAS

- [1] React, “React Reference,” react.dev. [En línea]. Disponible en: <https://react.dev/reference/react>. [Consultado: 02-mar-2026].
- [2] Node.js, “Node.js API Documentation,” nodejs.org. [En línea]. Disponible en: <https://nodejs.org/docs/latest/api/>. [Consultado: 02-mar-2026].
- [3] The PostgreSQL Global Development Group, “PostgreSQL: The world’s most advanced open source database,” postgresql.org. [En línea]. Disponible en: <https://www.postgresql.org/>. [Consultado: 05-mar-2026].
- [4] Prisma, “Prisma ORM Documentation,” prisma.io. [En línea]. Disponible en: <https://www.prisma.io/docs/orm>. [Consultado: 02-mar-2026].
- [5] Socket.IO, “Socket.IO,” socket.io. [En línea]. Disponible en: <https://socket.io/>. [Consultado: 22-may-2026].
- [6] Express, “Express - Node.js web application framework,” expressjs.com. [En línea]. Disponible en: <https://expressjs.com/>. [Consultado: 24-jun-2026].
- [7] Figma, “Figma: The collaborative interface design tool,” 2026. [En línea]. Disponible en: <https://www.figma.com>. [Accedido: 10-abr-2026].
- [8] Jefatura del Estado, “Ley 11/2021, de 9 de julio, de medidas de prevención y lucha contra el fraude fiscal, de transposición de la Directiva (UE) 2016/1164, del Consejo,” *Boletín Oficial del Estado*, n.º 164, pp. 82212–82283, 10-jul-2021.
- [9] Ministerio de Hacienda, “Real Decreto 1007/2023, de 5 de diciembre, por el que se aprueba el Reglamento que establece los requisitos de los sistemas informáticos de facturación,” *Boletín Oficial del Estado*, n.º 291, pp. 161400–161434, 6-dic-2023.

APÉNDICE

A. Diagrama de Gantt inicial

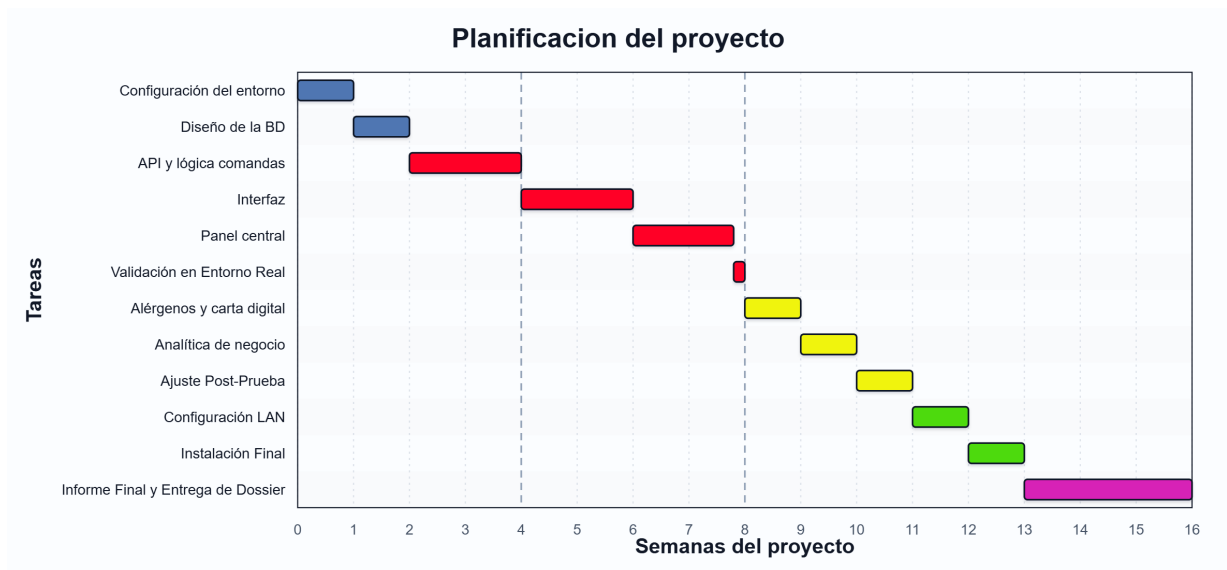


Fig. 4: Diagrama de Gantt correspondiente a la planificación inicial del proyecto.

B. Diagrama de Gantt final

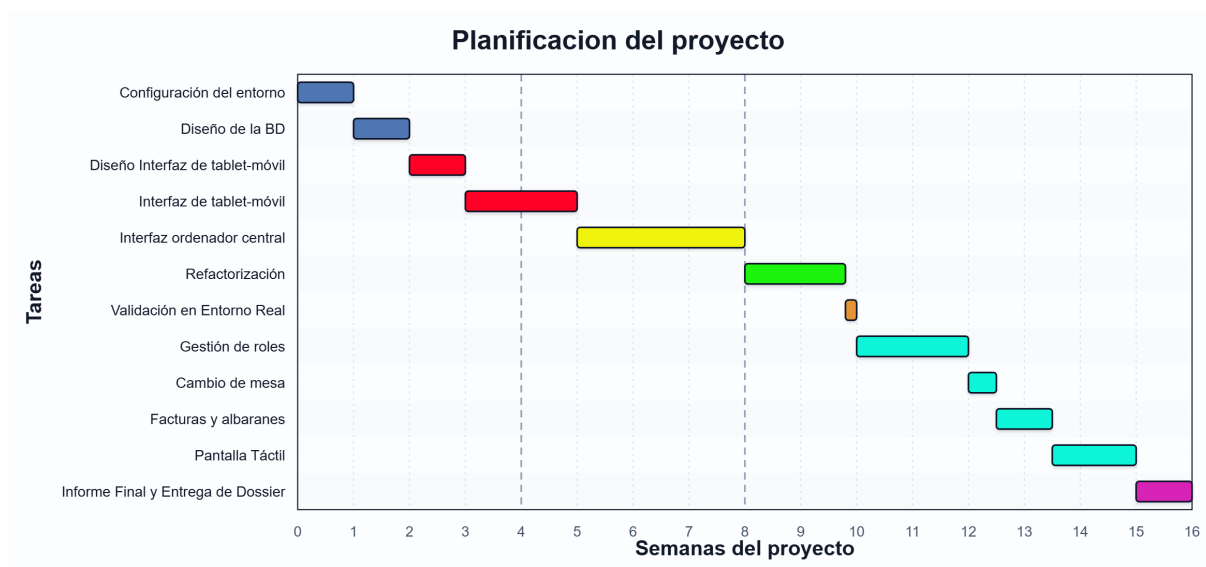


Fig. 5: Diagrama de Gantt correspondiente al desarrollo final del proyecto.

C. Esquema lógico de la base de datos

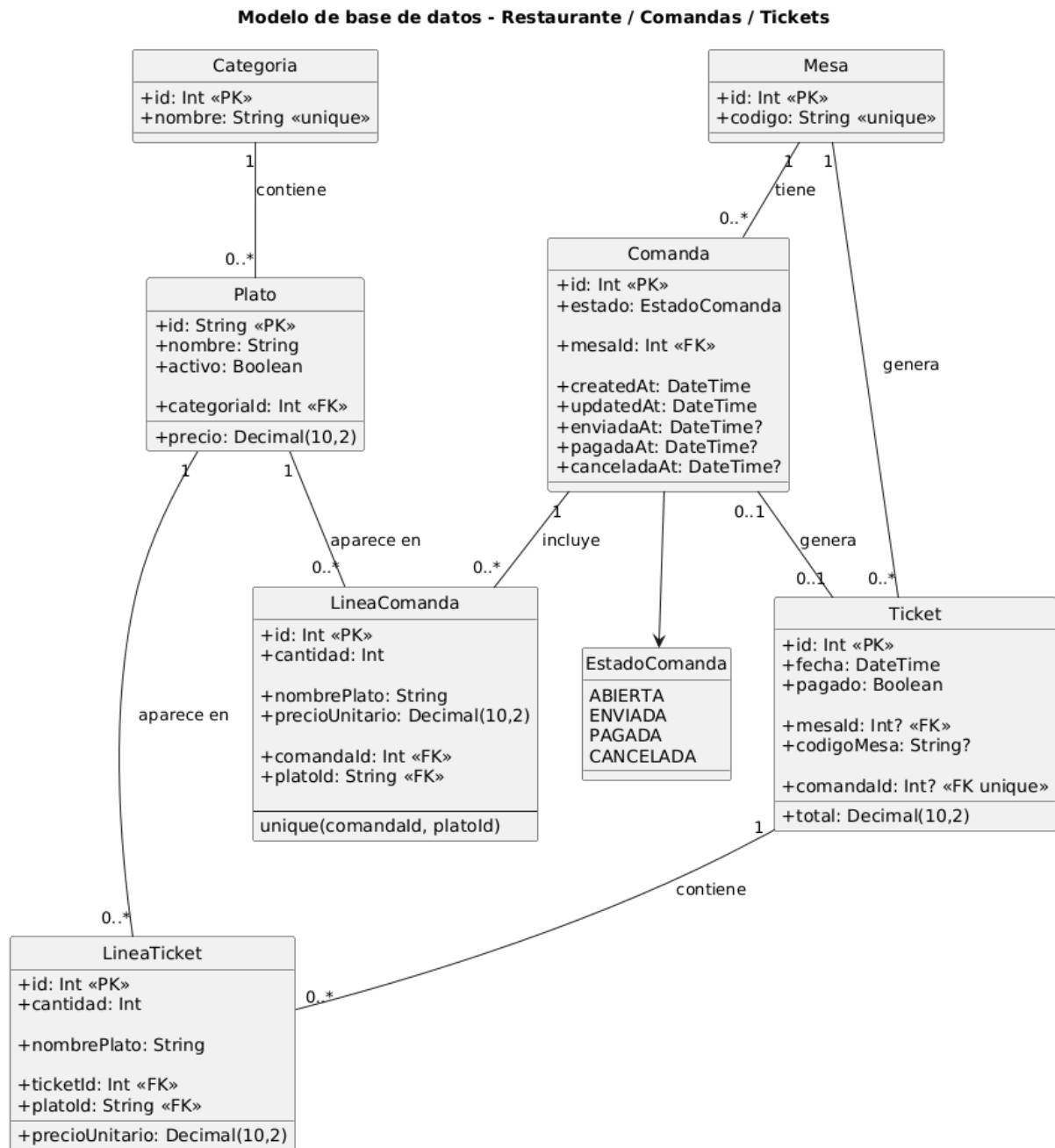


Fig. 6: Esquema UML de la base de datos del sistema.