

Sistema Omnicanal y Plataforma Progresiva para la Gestión de Mesas, Órdenes y Pagos en Restaurantes

Alumnos: Natalia Anaya Ramírez, Diego Villagrán Salazar

Director: Guzmán Flores Jessie Paulina

Correo de contacto: nataliaaanayaa@gmail.com

Resumen

Desarrollo de una aplicación web progresiva (PWA) para restaurantes que implementa: (1) sistema de mesas virtuales mediante códigos QR únicos, (2) gestión de pedidos individuales y compartidos en tiempo real, (3) procesamiento de pagos flexibles (individual o por cuenta compartida), y (4) panel administrativo básico con métricas operacionales. El sistema centraliza la comunicación entre comensales, cocina y administración, proporcionando trazabilidad completa de transacciones y estados de pedidos mediante una base de datos unificada.

Palabras clave

Aplicaciones Progresivas; Restaurantes; Sistemas Distribuidos; Inteligencia de Negocios; Bases de Datos.

1. Introducción

En los restaurantes tradicionales, el proceso de división de cuentas al final de la visita genera inconvenientes operativos: identificación manual de consumos individuales, múltiples formas de pago fragmentadas y conciliación compleja de ingresos. Los sistemas actuales como Soft Restaurant, Upserve o TouchBistro carecen de funcionalidades de división automática de cuentas y selección individual de productos por parte del usuario.

La solución propuesta implementa cuatro módulos técnicos principales:

1. **Autenticación y mesas virtuales:** Códigos QR únicos de 6 dígitos para acceso controlado a mesas compartidas.
2. **Gestión de pedidos granular:** Cada usuario selecciona productos específicos de su consumo individual con trazabilidad completa.
3. **Sistema de pagos flexibles:** Liquidación individual o contribución proporcional a cuentas compartidas mediante múltiples métodos de pago.
4. **Sincronización en tiempo real:** WebSockets para actualizaciones instantáneas entre dispositivos conectados a la misma mesa.

A diferencia de aplicaciones de división de gastos (Splitwise, Tricount) que operan sobre montos fijos, este sistema integra directamente los productos del restaurante con la lógica de división, eliminando la desconexión entre consumo y liquidación.

No se encontraron trabajos terminales relacionados con un desarrollo de aplicaciones centralizado omnicanal para restaurantes que incluya divisiones de cuenta y pagos en una mesa virtual enfocado al sector restaurantero.

Para evaluar la posición competitiva de nuestra solución, se presenta a continuación un análisis comparativo con los principales sistemas de gestión de restaurantes del mercado:

Funcionalidad	Tricount/Splitwise [6][3]	TouchBistro [5]	Soft Restaurant [2]	Revel [1]	Upserve [7]	Toast [4]
División de cuentas	Excelente para gastos personales	Sí, split por mesa/asiento	Sí, gestión por mesa/comensal	Sí, división flexible	Sí, soporte completo	Sí, split avanzado
Auto-pedido cliente	No/limitado	Medio con integraciones	Sí, e-menu digital	Sí, pantallas cocina	Sí, apps integradas	Sí, pedidos directos
Comunicación integral	Limitada	Sí, flujo completo	Sí, orden → cocina → admin	Sí, pantallas+backend	Sí, canal operativo	Sí, sistema completo
Dashboard/Analítica	Muy limitada	Sí, reportes nube	Sí, analítica integrada	Sí, reportes empresariales	Sí, análisis avanzado	Sí, tiempo real
POS	No aplicable	Sí, especializado	Sí, módulo central	Sí, restaurante/comercio	Sí, enfoque restaurante	Sí, núcleo sistema
Inventarios	No/básico	Moderado integrado	Sí, control+alertas	Sí, manejo stock	Sí, seguimiento insumos	Sí, gestión completa
Reportes	Limitado gastos	Sí, ventas/uso	Sí, múltiples reportes	Sí, cadenas/sucursales	Sí, personal/ventas	Sí, métricas robustas
Menús/Cartas	No aplicable	Sí, gestión completa	Sí, digital+QR	Sí, cartas/modificadores	Sí, promociones	Sí, menús especiales
Gestión mesas	No aplicable	Sí, reservas/ asignación	Sí, estados/ocupación	Sí, mesas/reservas	Sí, mesas/turnos	Sí, gestión avanzada
Pagos tiempo real	Sí, gastos compartidos	Limitado, integración	Sí, procesamiento	Sí, múltiples métodos	Sí, tiempo real	Sí, completo
Solución Propuesta	División colaborativa avanzada + reportes métricas	Pedidos QR + gestión digital	Comunicación integrada + mesas inteligentes	Dashboard BI + pagos colaborativos	POS especializado + PWA omnicanal	Inventarios completos + sistema integral

Cuadro 1: Comparación de funcionalidades entre sistemas existentes y la solución propuesta.

2. Objetivo

Desarrollar una aplicación progresiva (PWA) para la gestión integral de restaurantes que permita: crear mesas virtuales, gestionar pedidos, ofrecer esquemas flexibles de pago individual y compartido, y generar *dashboards* administrativos, integrando todos los canales en una base de datos centralizada.

3. Justificación

La problemática de división de cuentas al final de la estadía de un grupo de personas en un restaurante suele ser tediosa y propensa a errores, por lo cual la aplicación le facilita al cliente tener una cuenta de su consumo y le da el poder de elección si desea pagar la cuenta de otro comensal, aportar a la cuenta compartida y generar un cobro exacto de lo que haya decidido pagar, de tal forma que sea amigable y práctico para el usuario.

Por otro lado, la concentración de los datos y la trazabilidad de las transacciones genera un sistema que facilita la conciliación de los restaurantes y proporciona los recursos para generar dashboards administrativos o futuros análisis en busca de patrones o tendencias, así mismo evita el consumo de distintos proveedores, centralizando todo en un único sistema. Para los restaurantes es esencial tener control de su operación y, en adición, información que los ayude a generar nuevos modelos de negocio o tomar decisiones.

Existen plataformas que ya generan algunas de estas acciones, sin embargo ninguna incluye la omnicanalidad entre las áreas del comercio, división de cuentas, trazabilidad de pagos y estatus. Es un proyecto de complejidad alta que requiere conocimiento tanto en el área tecnológica como en las áreas financieras, fiscales, contables y de proveedores. Como analistas de datos utilizaremos materias vistas como Bases de Datos, Desarrollo Web, Análisis y Visualización de Datos.

Los productos serán:

1. Base de datos transaccional
2. Aplicación web progresiva con tres vistas: usuario, cocina y mesero.
3. Dashboard administrativo

4. Productos o Resultados esperados

El proyecto entregará una aplicación progresiva (PWA) integral que incluye interfaces diferenciadas para usuarios y administrativos encargados de inicializar las mesas virtuales, así como un módulo especializado de cocina con estados de órdenes organizados en un tablero Kanban seccionado por *status*. Adicionalmente, se desarrollará un panel administrativo completo para reportes y conciliación, junto con la implementación de esquemas flexibles de pagos individuales, colaborativos y compartidos.

El sistema incluirá código fuente completamente documentado, manual de usuario y documentación técnica detallada, además de una base de datos centralizada diseñada para futuros análisis e inteligencia de negocios. El sistema está diseñado para ofrecer un rendimiento óptimo con tiempos de respuesta rápidos y una experiencia de usuario fluida y eficiente.

La arquitectura tecnológica se fundamenta en un stack moderno que comprende React@18 + TypeScript + TailwindCSS@3 + Vite como bundler para el frontend, garantizando desarrollo rápido y optimización de producción. El backend utiliza Express.js@4 + TypeScript + Node.js para APIs RESTful con middleware de autenticación y validación, mientras que la base de datos emplea Supabase (PostgreSQL) con Row Level Security para control granular de acceso y Supabase Realtime para WebSockets.

Para optimización del rendimiento se implementará Redis como capa de cache para consultas frecuentes y gestión de sesiones, complementado con integración dual de Stripe y MercadoPago APIs para procesamiento seguro de transacciones. El sistema de autenticación utilizará Supabase Auth con soporte para múltiples proveedores (email, Google, Facebook), y la generación automática de códigos QR únicos se realizará mediante la biblioteca qrcode para acceso a mesas. El siguiente diagrama presenta la arquitectura completa del sistema, ilustrando la estructura de capas, los componentes tecnológicos principales y el flujo de datos que permitirá la integración efectiva de todos los módulos del sistema omnicanal:

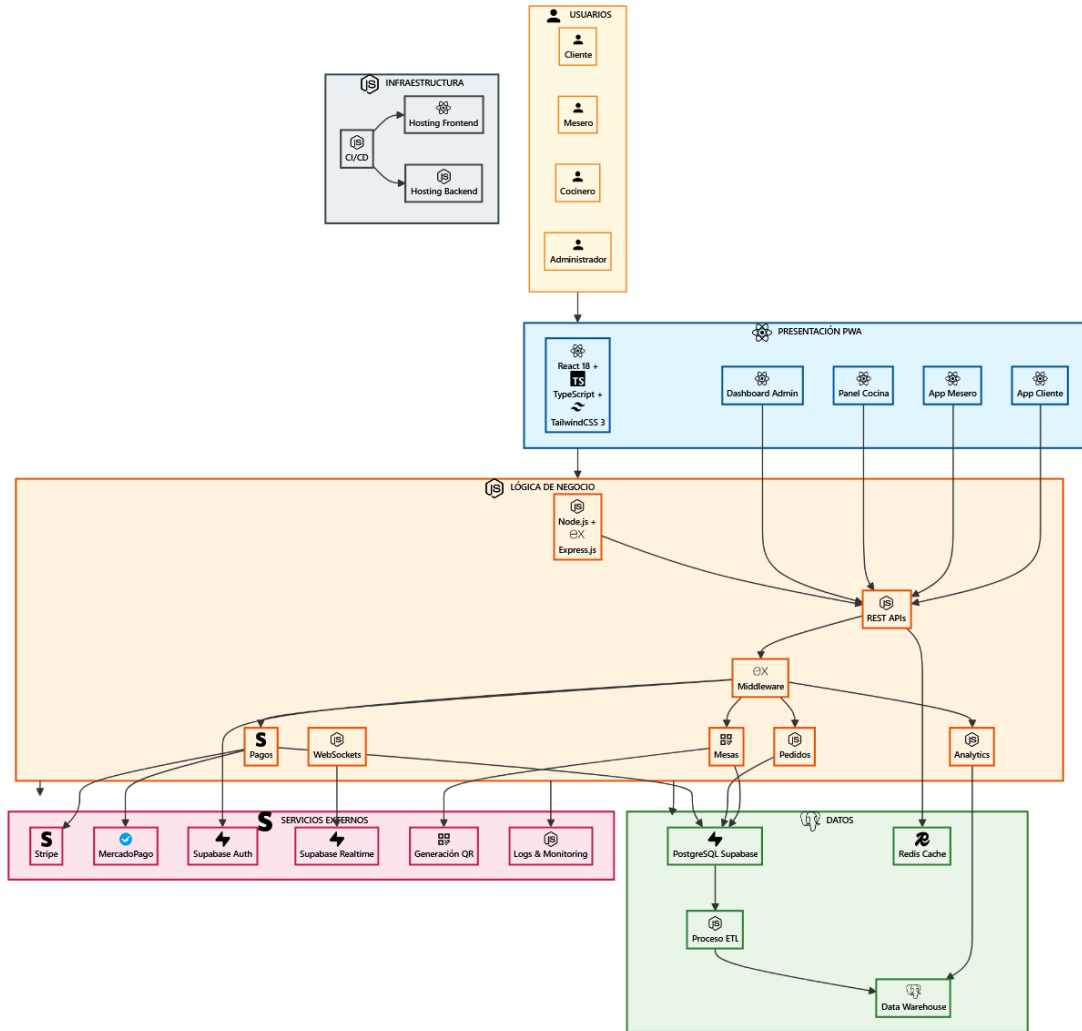


Figura 1: Arquitectura completa del sistema.

4.1. Definición de Módulos del Sistema

El sistema está estructurado en cinco capas que trabajan de manera coordinada para ofrecer una experiencia omnicanal completa:

Usuarios del Sistema

- **Cliente:** Escanea el código QR de la mesa para acceder al menú digital, selecciona productos y decide cómo dividir la cuenta al momento del pago.
- **Mesero:** Administra las mesas del restaurante, puede agregar productos a las órdenes existentes y monitorea el progreso de los pedidos desde la cocina.
- **Cocinero:** Visualiza las órdenes entrantes en un tablero organizado, actualiza el estatus de preparación y coordina los tiempos de entrega.

- **Administrador:** Configura el menú del restaurante, revisa reportes de ventas y accede a métricas detalladas del negocio.

Aplicaciones Web Progresivas

- **Stack Tecnológico:** React 18 con TypeScript para mayor robustez del código y TailwindCSS 3 para un diseño consistente y responsivo.
- **Aplicación Cliente:** Permite navegar el menú, agregar productos al carrito personal y procesar pagos de forma individual o colaborativa.
- **Panel Mesero:** Facilita la gestión de múltiples mesas simultáneamente y la comunicación directa con el área de cocina.
- **Interfaz Cocina:** Organiza las órdenes en columnas según su estatus (recibido, preparando, listo) similar a un tablero Kanban.
- **Dashboard Administrativo:** Concentra reportes de ventas, configuración de productos y análisis de tendencias del restaurante.

Lógica de Negocio

- **Backend Principal:** Node.js con Express.js maneja todas las operaciones del servidor y expone APIs REST para cada funcionalidad.
- **Endpoints API:** Rutas específicas para login, gestión de mesas, procesamiento de pedidos, manejo de pagos y generación de reportes.
- **Middleware:** Valida datos de entrada, verifica permisos de usuario y maneja errores de manera centralizada.
- **Comunicación Tiempo Real:** WebSockets mantienen sincronizados todos los dispositivos conectados a una misma mesa.
- **Gestión de Mesas:** Genera códigos QR únicos de 6 dígitos y controla el acceso de usuarios a cada mesa virtual.
- **Procesamiento de Pedidos:** Registra cada producto seleccionado por usuario individual y mantiene el historial completo de la orden.
- **Sistema de Pagos:** Integra múltiples pasarelas para procesar transacciones seguras y dividir cuentas automáticamente.
- **Módulo de Reportes:** Procesa datos transaccionales para generar métricas operativas y análisis de ventas.

Almacenamiento de Datos

- **Base de Datos Principal:** PostgreSQL a través de Supabase con políticas de seguridad a nivel de fila para proteger información sensible.
- **Cache de Sesiones:** Redis almacena temporalmente datos de sesiones activas y consultas frecuentes para mejorar el rendimiento.
- **Almacén Analítico:** Data warehouse separado para consultas históricas y análisis de inteligencia de negocios.
- **Procesos ETL:** Scripts automatizados que extraen, transforman y cargan datos desde la base transaccional hacia el almacén analítico.

Integraciones Externas

- **Stripe:** Procesa pagos con tarjetas internacionales y maneja suscripciones recurrentes si el restaurante lo requiere.
- **MercadoPago:** Acepta métodos de pago locales como transferencias bancarias, efectivo en tiendas y billeteras digitales.
- **Autenticación Supabase:** Permite login con email/contraseña o cuentas sociales como Google y Facebook.
- **Supabase Realtime:** Proporciona la infraestructura de WebSockets para actualizaciones instantáneas entre dispositivos.
- **Generador QR:** Biblioteca JavaScript que crea códigos QR únicos para cada mesa del restaurante.
- **Monitoreo:** Herramientas de logging y alertas para detectar errores y monitorear el rendimiento del sistema.

Infraestructura de Despliegue

- **Frontend:** Aplicaciones React desplegadas en CDN para carga rápida desde cualquier ubicación geográfica.
- **Backend:** Servidores Node.js con escalamiento automático según la demanda de usuarios concurrentes.
- **Automatización:** Pipeline CI/CD que ejecuta pruebas automáticas y despliega nuevas versiones sin interrumpir el servicio.

5. Metodología

El desarrollo del sistema omnicanal para restaurantes seguirá una metodología ágil basada en Scrum, organizando el trabajo en sprints de duración fija. Esta aproximación permite entregar funcionalidades incrementales que pueden ser probadas y validadas de manera continua.

La metodología seleccionada se fundamenta en los siguientes principios:

- **Desarrollo iterativo:** Cada sprint de 2-3 semanas produce un entregable funcional que agrega valor al sistema.
- **Validación temprana:** Los prototipos y funcionalidades se prueban desde las primeras iteraciones para detectar problemas de usabilidad.
- **Flexibilidad de requisitos:** La arquitectura modular permite ajustar funcionalidades según feedback de usuarios y restricciones técnicas.
- **Integración continua:** Cada módulo desarrollado se integra inmediatamente con los componentes existentes.

El proyecto se estructura en 12 sprints que abarcan desde el análisis inicial hasta las pruebas finales del sistema. Cada sprint tiene objetivos específicos y entregables medibles:

1. **Analítica de negocio y Arquitectura**
2. **Sprint 2: Diseño de la base de datos**
3. **Sprint 3: Desarrollo de la base de datos**
4. **Sprint 4: Diseño de vistas y mockups**
5. **Sprint 5: Diseño y arquitectura de backend**
6. **Sprint 6: Definición de KPI y vistas de Dashboard**
7. **Sprint 7: Implementación del dashboard**
8. **Sprint 8: Desarrollo del PAW mesero**
9. **Sprint 9: Desarrollo del PAW comensal**
10. **Sprint 10: Desarrollo del PAW cocina**
11. **Sprint 11: Integración de la API de pagos**
12. **Sprint 12: Pruebas**

Se generará un tablero Kanban en GitHub, el cual tendrá los estatus: inicializado, en progreso, en revisión y finalizado. Cabe aclarar que ninguno de estos estatus es final, ya que puede volver a cualquier otro estado excepto inicializado. Se pueden agregar nuevas tareas dentro del sprint si es necesario.

La implementación de las aplicaciones web progresivas para los diferentes roles (comensal, cocina, mesero) y el desarrollo del dashboard administrativo requieren una arquitectura robusta y escalable que soporte las funcionalidades descritas.

Para la ejecución exitosa del proyecto, se ha desarrollado un cronograma detallado que abarca el período de febrero a diciembre de 2026. Este cronograma está dividido en dos diagramas de Gantt que muestran la distribución de actividades y responsabilidades entre los miembros del equipo de desarrollo. La planificación temporal considera las dependencias entre tareas, los recursos disponibles y los hitos críticos del proyecto, asegurando una progresión ordenada desde la fase de análisis hasta la implementación completa del sistema omnicanal para restaurantes.

		Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Sprint 1	Analítica de negocio y Arquitectura											
Sprint 2	Diseño de la base de datos											
Sprint 3	Desarrollo de la base de datos											
Sprint 4	Diseño de Vistas y Mockups											
	Reporte técnico TT1											
Sprint 6	Definición de KPI y vistas de Dashboard											
Sprint 7	Implementación del Dashboard											
Sprint 11	Integración de la API de pagos											
Sprint 12	Pruebas											
	Reporte técnico TT2											

Figura 2: Cronograma de actividades - Natalia (Febrero - Diciembre 2026).

		Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Sprint 1	Analítica de negocio y Arquitectura											
Sprint 2	Diseño de base de datos											
Sprint 4	Diseño de Vistas y Mockups											
Sprint 5	Diseño y arquitectura de backend											
Sprint 8	Desarrollo PWA mesero											
	Reporte técnico TT1											
Sprint 9	Desarrollo PWA comensal											
Sprint 10	Desarrollo PWA cocina											
Sprint 11	Integración de la API de pagos											
Sprint 12	Pruebas											
	Reporte técnico TT2											

Figura 3: Cronograma de actividades - Diego (Febrero - Diciembre 2026).

6. Referencias

- [1] Revel Systems. *Revel Systems POS*. Software de gestión y punto de venta. 2024. URL: <https://revelsystems.com/>.
- [2] Soft Restaurant. *Soft Restaurant*. Software de gestión de restaurantes. 2024. URL: <https://softrestaurant.com/>.
- [3] Splitwise Inc. *Splitwise - Split bills and expenses*. Aplicación móvil para división de gastos. 2024. URL: <https://www.splitwise.com/> (visitado 15-01-2024).
- [4] Toast, Inc. *Toast POS*. Software de punto de venta para restaurantes. 2024. URL: <https://pos.toasttab.com>.
- [5] TouchBistro Inc. *TouchBistro*. Software de punto de venta para restaurantes. 2024. URL: <https://www.touchbistro.com/>.
- [6] Tricount SPRL. *Tricount - Shared expenses made simple*. Aplicación móvil para gestión de gastos compartidos. 2024. URL: <https://www.tricount.com/> (visitado 15-01-2024).
- [7] Upserve. *Upserve POS*. Software de gestión de restaurantes. 2024. URL: <https://upserve.com/>.

7. Alumnos y Directores

Anaya Ramirez Natalia

Alumna de la Licenciatura en Ciencia de Datos, ESCOM/IPN.

Boleta: 2023630868

Tel: 56 41 11 74 06

Email: nataliaanayaa@gmail.com

Firma: 


Villagrán Salazar Diego

Alumno de la Licenciatura en Ciencia de Datos, ESCOM/IPN.

Boleta: 2023630590

Tel: 5572071183

Email: dvillagrans@alumno.ipn.mx

Firma: 

Guzmán Flores Jessie Paulina

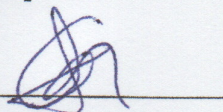
Doctora en Ecoeducación, Maestra en Desarrollo de Competencias Docentes por el Instituto Universitario Puebla y Licenciada en Turismo por EST-IPN.

Docente del nivel superior en ESCOM desde septiembre 2011.

Áreas de interés: Educación, formulación de proyectos, sostenibilidad, gobierno de TI, gestión empresarial, turismo.

Tel: 55 55 07 46 47

Email: jguzmanf@ipn.mx

Firma: 

CARÁCTER: Confidencial

FUNDAMENTO LEGAL: Art. 11 Fracc. V y Arts. 108, 113 y 117 de la Ley Federal de Transparencia y Acceso a la Información Pública.

PARTES CONFIDENCIALES: Número de boleta y teléfono.