

CLICK MUNCH
INGENIERÍA DE SOFTWARE

PRESENTADO POR:

Michael Stiven Betancourt Gelves

Santiago Bejarano Ariza

Johan Sebastian Roa Rodriguez

PROFESOR

Oscar Eduardo Alvarez Rodriguez



Universidad Nacional de Colombia
Facultad de Ingeniería
2024

ÍNDICE

Levantamiento de requerimientos	3
1. Acceso a Menús Digitales	3
2. Reserva de Mesas	4
3. Realización de Pedidos	4
4. Sugerencia de Restaurantes por Cercanía	5
5. Seguimiento del Estado del Pedido	6
6. Valoraciones y Comentarios	7
7. Gestión de Menús para Restaurantes	7
8. Gestión de Pedidos en Tiempo Real	8
9. Estadísticas y Métricas de Ventas	9
Análisis de requerimientos	9
1. Acceso a Menús Digitales	9
2. Reserva de Mesas	10
3. Realización de Pedidos	11
4. Seguimiento del Estado del Pedido	12
5. Valoraciones y Comentarios	13
6. Gestión de Menús para Restaurantes	14
7. Gestión de Pedidos en Tiempo Real	15
8. Estadísticas y Métricas de Ventas	15
9. Sugerencia de Restaurantes por Cercanía	16
10. Tabla de estimaciones	17
Análisis Gestión del software	18
1. Tiempo	18
2. Costos	19
3. Alcance	20
Diseño y arquitectura	21
Arquitectura del sistema	21
Diseño de bases de datos	21

Levantamiento de requerimientos

1. Acceso a Menús Digitales

¿De dónde surge la idea?

La necesidad de digitalizar los menús fue identificada a través de entrevistas con clientes frecuentes de restaurantes, quienes mencionaron la incomodidad de los menús físicos y la falta de actualización en los precios. También se observó que muchos establecimientos utilizan imágenes de menús en redes sociales, lo que limita su accesibilidad.

¿Cómo nos pusimos de acuerdo para elegir la idea?

Se discutieron múltiples alternativas en reuniones del equipo. La decisión final se tomó mediante votación, priorizando soluciones con alto impacto y baja barrera de adopción.

¿Qué problemáticas busca resolver?

- Menús desactualizados generan confusión en los clientes.
- Restaurantes pequeños no tienen acceso a plataformas digitales robustas.
- La búsqueda de platillos específicos es engorrosa sin una base digital estructurada.

¿Qué expectativas tienen los usuarios?

- Acceso rápido a los menús desde cualquier lugar.

- Filtros por categoría, tipo de comida y precio.
- Información clara y actualizada sobre disponibilidad y precios.

¿Qué beneficios espera el equipo desarrollador?

- Aprender sobre el diseño de bases de datos relacionales y API RESTful.
- Implementar estrategias de optimización para mejorar el tiempo de carga en aplicaciones móviles.

2. Reserva de Mesas

¿De dónde surge la idea?

A partir de encuestas a clientes habituales y observación en restaurantes con alta demanda, se identificó que los tiempos de espera pueden afectar la experiencia del usuario y la eficiencia operativa del restaurante.

¿Cómo nos pusimos de acuerdo para elegir la idea?

Se realizó un análisis de impacto y viabilidad técnica. Aunque la idea fue propuesta desde el inicio, el equipo decidió incluirla como una funcionalidad secundaria en el MVP.

¿Qué problemáticas busca resolver?

- Clientes frustrados por largas esperas.
- Restaurantes sin un control claro sobre la ocupación de mesas.
- Falta de herramientas digitales en restaurantes pequeños para gestionar reservas.

¿Qué expectativas tienen los usuarios?

- Garantizar una mesa disponible al llegar al restaurante.
- Confirmaciones automáticas y recordatorios de reserva.
- Posibilidad de cancelar o modificar reservas de forma sencilla.

¿Qué beneficios espera el equipo desarrollador?

- Implementar modelos de gestión de disponibilidad en tiempo real.
- Integrar notificaciones automatizadas para mejorar la experiencia del usuario.

3. Realización de Pedidos

¿De dónde surge la idea?

Se identificó a través de observaciones en restaurantes y entrevistas con clientes. La

mayoría de los clientes prefieren pedir digitalmente para evitar errores en la comunicación con los meseros.

¿Cómo nos pusimos de acuerdo para elegir la idea?

Se validó la importancia del módulo con dueños de restaurantes y se decidió priorizar dentro de las funcionalidades esenciales.

¿Qué problemáticas busca resolver?

- Errores en los pedidos por malentendidos.
- Falta de personal en ciertos horarios que ralentiza la toma de pedidos.
- Clientes que prefieren no interactuar con meseros.

¿Qué expectativas tienen los usuarios?

- Pedidos personalizados con opciones para agregar o quitar ingredientes.
- Recepción inmediata de confirmación y tiempo estimado de entrega.
- Facilidad de pago digital.

¿Qué beneficios espera el equipo desarrollador?

- Diseñar una arquitectura robusta para gestionar pedidos en tiempo real.
- Aprender sobre integración de pasarelas de pago.

4. Sugerencia de Restaurantes por Cercanía

¿De dónde surge la idea?

Se identificó al analizar aplicaciones actuales y notar que muchas sugerencias no son relevantes para la ubicación del usuario. Además, en entrevistas, los clientes indicaron que les gustaría encontrar opciones cercanas sin necesidad de buscar manualmente.

¿Cómo nos pusimos de acuerdo para elegir la idea?

Se prioriza tras evaluar su impacto en la experiencia del usuario y la viabilidad técnica de integrar geolocalización.

¿Qué problemáticas busca resolver?

- Usuarios reciben sugerencias irrelevantes o demasiado alejadas.
- Falta de visibilidad para restaurantes pequeños que no aparecen en búsquedas genéricas.

¿Qué expectativas tienen los usuarios?

- Recibir recomendaciones precisas basadas en ubicación.
- Ajustar manualmente la zona de búsqueda si desean explorar en otra área.
- Filtrar por tipo de cocina y precio.

¿Qué beneficios espera el equipo desarrollador?

- Implementar sistemas de geolocalización y mapas interactivos.
- Aprender sobre optimización de búsquedas basadas en distancia.

5. Seguimiento del Estado del Pedido

¿De dónde surge la idea?

Se identificó a través de encuestas con clientes de restaurantes y observación en locales con servicio a domicilio. Muchos clientes manifestaron su frustración por no saber en qué estado se encuentra su pedido ni cuánto tiempo falta para la entrega.

¿Cómo nos pusimos de acuerdo para elegir la idea?

Se discutió en equipo y se prioriza como una funcionalidad esencial, ya que mejora la experiencia del usuario y evita reclamos innecesarios.

¿Qué problemáticas busca resolver?

- Ansiedad y frustración de los clientes por no saber cuándo llegará su pedido.
- Altas tasas de llamadas a los restaurantes preguntando por el estado del pedido.
- Falta de transparencia en los tiempos de entrega.

¿Qué expectativas tienen los usuarios?

- Recibir actualizaciones en tiempo real sobre el estado del pedido.
- Notificaciones push o mensajes con cambios de estado.
- Información precisa sobre el tiempo estimado de entrega.

¿Qué beneficios espera el equipo desarrollador?

- Aprender a trabajar con WebSockets y sistemas en tiempo real.
- Mejorar habilidades en el desarrollo de interfaces dinámicas y notificaciones automáticas.

6. Valoraciones y Comentarios

¿De dónde surge la idea?

Se identificó en entrevistas con usuarios, quienes mencionaron que confían en opiniones de otros clientes para elegir restaurantes. Además, los dueños de restaurantes ven en este sistema una oportunidad para mejorar sus servicios con base en la retroalimentación.

¿Cómo nos pusimos de acuerdo para elegir la idea?

Se discutió su relevancia en la toma de decisiones del usuario y se determinó que, aunque no es esencial en un MVP, es un valor agregado importante.

¿Qué problemáticas busca resolver?

- Dificultad para conocer la calidad de un restaurante antes de visitarlo.
- Falta de un canal formal para que los clientes compartan su experiencia.
- Restaurantes que no reciben retroalimentación sobre su servicio.

¿Qué expectativas tienen los usuarios?

- Poder calificar y dejar comentarios sobre los restaurantes y platillos.
- Filtrar restaurantes por mejores calificaciones.
- Ver experiencias de otros usuarios antes de tomar una decisión.

¿Qué beneficios espera el equipo desarrollador?

- Diseñar y optimizar sistemas de interacción entre usuarios.
- Aprender sobre gestión de bases de datos para reseñas y calificaciones.

7. Gestión de Menús para Restaurantes

¿De dónde surge la idea?

A través de entrevistas con dueños de restaurantes, se descubrió que muchos no tienen control digital sobre su menú y dependen de cambios manuales en pizarras o imágenes en redes sociales.

¿Cómo nos pusimos de acuerdo para elegir la idea?

Se definió como una funcionalidad imprescindible, ya que sin ella los restaurantes no podrían actualizar sus ofertas en la plataforma.

¿Qué problemáticas busca resolver?

- Menús desactualizados que generan confusión en los clientes.
- Falta de herramientas digitales para administrar el menú de manera eficiente.
- Dificultad para incluir promociones o cambios en los platillos rápidamente.

¿Qué expectativas tienen los usuarios?

- Poder modificar sus menús en tiempo real.
- Subir imágenes y descripciones de los platillos fácilmente.
- Gestionar categorías y precios de manera eficiente.

¿Qué beneficios espera el equipo desarrollador?

- Aprender a desarrollar CRUD eficientes y con autenticación segura.
- Optimizar el flujo de trabajo de los administradores de restaurantes.

8. Gestión de Pedidos en Tiempo Real

¿De dónde surge la idea?

Durante la fase de entrevistas, los restaurantes mencionaron que una de sus principales dificultades es la falta de un sistema eficiente para organizar los pedidos en curso.

Actualmente, muchos negocios dependen de notas escritas o llamadas telefónicas, lo que genera errores y retrasos.

¿Cómo nos pusimos de acuerdo para elegir la idea?

Se identificó como una funcionalidad **should-have**, ya que mejora la operatividad del restaurante pero no es imprescindible en el MVP.

¿Qué problemáticas busca resolver?

- Pedidos desorganizados que generan errores en la entrega.
- Falta de visibilidad del estado de los pedidos dentro del restaurante.
- Clientes que reciben comida incorrecta por confusión en la cocina.

¿Qué expectativas tienen los usuarios?

- Ver una lista clara de todos los pedidos activos.
- Poder actualizar el estado del pedido con un solo clic.
- Sincronización en tiempo real entre el restaurante y el cliente.

¿Qué beneficios espera el equipo desarrollador?

- Aprender a trabajar con WebSockets para gestionar estados dinámicos.
- Diseñar dashboards administrativos eficientes y optimizados.

9. Estadísticas y Métricas de Ventas

¿De dónde surge la idea?

Se identificó como una necesidad de los restaurantes que desean conocer cuáles son sus platillos más vendidos, en qué horarios tienen más demanda y qué estrategias de venta funcionan mejor.

¿Cómo nos pusimos de acuerdo para elegir la idea?

Se consideró una funcionalidad secundaria, pero con alto valor para los administradores, por lo que se incluyó como **could-have**.

¿Qué problemáticas busca resolver?

- Falta de información sobre tendencias de ventas.
- Dificultad para analizar el rendimiento financiero del restaurante.
- Falta de datos para tomar decisiones estratégicas.

¿Qué expectativas tienen los usuarios?

- Ver reportes de ventas en tiempo real.
- Conocer los platillos más populares.
- Identificar las horas de mayor demanda.

¿Qué beneficios espera el equipo desarrollador?

- Aprender a trabajar con gráficos dinámicos y generación de reportes en tiempo real.
- Diseñar sistemas de análisis de datos optimizados.

Análisis de requerimientos

1. Acceso a Menús Digitales

Descripción del Trabajo:

Implementar el acceso a menús digitales requiere diseñar una base de datos robusta donde se almacenen los platillos, categorías y sus detalles (nombre, descripción, precio e imágenes). Se debe desarrollar una API que permita a los usuarios consultar menús filtrados por categorías y características específicas. Además, se creará una interfaz visual

optimizada, con diseño responsivo y soporte para conexiones lentas, asegurando una experiencia fluida en cualquier dispositivo.

Clasificación: Must-Have

Días Estimados: 13 días

Costo Estimado: \$5,200,000 COP

Análisis Detallado

Complejidad Técnica: Media. La implementación de un sistema CRUD con filtrado de menús por categorías no representa un gran desafío, pero se deben optimizar las consultas para carga rápida y garantizar una UX fluida en dispositivos móviles.

Recursos Disponibles: Se requieren al menos un desarrollador backend para la API y base de datos, y un frontend para la visualización optimizada.

Impacto en la Experiencia del Usuario: Alto. Un menú accesible y bien estructurado mejora la percepción de la aplicación y reduce la fricción en la toma de decisiones del cliente.

Relación con los Objetivos del Proyecto: Fundamental, ya que sin menús digitales, la aplicación no tiene propósito para los consumidores.

Justificación del Tiempo

- **3 días** para diseñar una base de datos eficiente que estructure categorías y relaciones entre menús y restaurantes.
- **5 días** para desarrollar el backend con endpoints que gestionen menús, filtros y validaciones.
- **5 días** para el desarrollo del frontend, asegurando una interfaz clara y optimizada para móviles.

2. Reserva de Mesas

Descripción del Trabajo:

La funcionalidad de reservas implica diseñar un sistema que permita a los usuarios seleccionar una fecha, hora y número de personas, verificando la disponibilidad en tiempo real. Para ello, se desarrollará un backend que gestione la lógica de reservas y una interfaz

intuitiva para los usuarios. Además, se configurarán notificaciones automáticas que confirmen las reservas realizadas y envíen recordatorios antes de la fecha.

Clasificación: Should-Have

Días Estimados: 8 días

Costo Estimado: \$3,200,000 COP

Análisis Detallado

Complejidad Técnica: Media. Se requiere manejar disponibilidad en tiempo real, prevenir sobreasignaciones y notificar a los usuarios.

Recursos Disponibles: Un backend para manejar disponibilidad y lógica de negocio, y un frontend con una interfaz intuitiva.

Impacto en la Experiencia del Usuario: Medio-Alto. Aunque no es esencial, mejora la organización del usuario y la planificación de los restaurantes.

Relación con los Objetivos del Proyecto: Aporta valor adicional a la plataforma, diferenciándose de opciones más básicas sin reserva integrada.

Justificación del Tiempo

- **3 días** para el backend con lógica de disponibilidad.
- **3 días** para el frontend con formularios interactivos.
- **2 días** para pruebas y validaciones.

3. Realización de Pedidos

Descripción del Trabajo:

La realización de pedidos será uno de los pilares del sistema. Requiere desarrollar un backend que gestione las solicitudes de los usuarios, validando la disponibilidad de los platillos y calculando el precio total. En el frontend, se diseñará un flujo que permita a los usuarios personalizar sus pedidos y seleccionar entre distintas modalidades (en restaurante, para recoger o a domicilio). Además, se integrarán resúmenes claros y confirmaciones de pedido.

Clasificación: Must-Have

Días Estimados: 13 días

Costo Estimado: \$5,200,000 COP

Análisis Detallado

Complejidad Técnica: Alta. Se requiere una arquitectura robusta para validar disponibilidad de platillos, gestionar pedidos en tiempo real y sincronizar estados de pago.

Recursos Disponibles: Se necesita integración con una base de datos eficiente y un sistema de gestión de pedidos en tiempo real.

Impacto en la Experiencia del Usuario: Crítico. Sin esta funcionalidad, la aplicación pierde su razón de ser.

Relación con los Objetivos del Proyecto: Directamente alineado con el propósito principal de la plataforma.

Justificación del Tiempo

- **5 días** para la lógica de backend con validaciones y cálculos de precios.
- **5 días** para el frontend, incluyendo personalización de pedidos.
- **3 días** para pruebas funcionales y ajustes finales.

4. Seguimiento del Estado del Pedido

Descripción del Trabajo:

Para el seguimiento del estado del pedido, se implementará un sistema en tiempo real basado en WebSockets, que permitirá a los usuarios visualizar el progreso de su pedido desde la confirmación hasta la entrega. Se desarrollará un backend que gestione estos estados y un frontend que los muestre claramente, junto con notificaciones push que informen sobre cada cambio.

Clasificación: Must-Have

Días Estimados: 8 días

Costo Estimado: \$3,200,000 COP

Análisis Detallado

- **Complejidad Técnica:** Media-Alta. La actualización en tiempo real con WebSockets puede ser compleja, especialmente con múltiples pedidos simultáneos.
- **Recursos Disponibles:** Se requiere un desarrollador backend con experiencia en WebSockets y un frontend con manejo de estados en tiempo real.
- **Impacto en la Experiencia del Usuario:** Alto. Mejora la percepción de confiabilidad del sistema.
- **Relación con los Objetivos del Proyecto:** Clave para mantener informados a los clientes y garantizar confianza en la plataforma.

Justificación del Tiempo

- **3 días** para la implementación de WebSockets en el backend.
- **3 días** para el frontend con visualización del estado del pedido.
- **2 días** para pruebas de sincronización y ajustes.

5. Valoraciones y Comentarios

Descripción del Trabajo:

El módulo de valoraciones permitirá a los usuarios calificar y comentar sobre los restaurantes y platillos. Se diseñará una base de datos para almacenar las valoraciones y comentarios, un backend que facilite la creación y consulta de estas opiniones, y un frontend para que los usuarios puedan interactuar con este módulo.

Clasificación: Could-Have

Días Estimados: 5 días

Costo Estimado: \$2,000,000 COP

Análisis Detallado

Complejidad Técnica: Baja. Es un sistema de gestión de reseñas que se puede modelar con CRUD estándar.

Recursos disponibles: Un backend básico para almacenar comentarios y un frontend para mostrar y gestionar reseñas.

Impacto en la Experiencia del Usuario: Medio. No es imprescindible, pero mejora la confianza en los restaurantes y ayuda en la toma de decisiones.

Relación con los Objetivos del Proyecto: Complementa la experiencia, pero no es fundamental para la funcionalidad inicial.

Justificación del Tiempo

- **2 días** para backend con almacenamiento de valoraciones.
- **2 días** para el frontend con interfaz de comentarios.
- **1 día** para pruebas e integración.

6. Gestión de Menús para Restaurantes

Descripción del Trabajo:

Los administradores de restaurantes necesitarán una interfaz para crear, editar o eliminar platillos, subir imágenes y actualizar precios. Se diseñará un backend con operaciones CRUD y validaciones para garantizar datos completos, así como un frontend que facilite la administración visual de los menús.

Clasificación: Must-Have

Días Estimados: 13 días

Costo Estimado: \$5,200,000 COP

Análisis Detallado

Complejidad Técnica: Media. Se requiere autenticación, validaciones de entrada y una interfaz que permita ediciones rápidas.

Recursos Disponibles: Se necesita un backend con autenticación y permisos de usuario y un frontend con una UI clara.

Impacto en la Experiencia del Usuario: Crítico para restaurantes. Sin esto, los administradores no podrán mantener su oferta actualizada.

Relación con los Objetivos del Proyecto: Fundamental para la operatividad del sistema.

Justificación del Tiempo

- **5 días** para el backend con operaciones CRUD.
- **5 días** para el frontend con herramientas de gestión visual.
- **3 días** para pruebas funcionales e integración.

7. Gestión de Pedidos en Tiempo Real

Descripción del Trabajo:

Esta funcionalidad permitirá a los administradores monitorear y actualizar el estado de los pedidos en tiempo real. Se desarrollará un dashboard interactivo que muestre todos los pedidos activos y permita actualizarlos fácilmente. El backend integrará WebSockets para sincronización en tiempo real, lo que permitirá una operación eficiente y precisa.

Clasificación: Should-Have

Días Estimados: 8 días

Costo Estimado: \$3,200,000 COP

Análisis Detallado

Complejidad Técnica: Media-Alta. Se necesita sincronización en tiempo real entre múltiples clientes y cocinas.

Recursos Disponibles: Se requieren conocimientos en WebSockets y diseño de interfaces administrativas intuitivas.

Impacto en la Experiencia del Usuario: Alto para restaurantes, ya que optimiza la organización y reduce errores.

Relación con los Objetivos del Proyecto: No es esencial para el MVP, pero mejora la eficiencia operativa.

Justificación del Tiempo

- **3 días** para backend con WebSockets.
- **3 días** para el frontend con dashboard interactivo.
- **2 días** para pruebas de sincronización.

8. Estadísticas y Métricas de Ventas

Descripción del Trabajo:

Los restaurantes tendrán acceso a un módulo que muestre estadísticas clave, como platillos más vendidos, horarios de mayor demanda y ventas totales. Se diseñará un backend que

recopile y procese estos datos, junto con un frontend que los presenta mediante gráficos interactivos y tablas claras.

Clasificación: Could-Have

Días Estimados: 5 días

Costo Estimado: \$2,000,000 COP

Análisis Detallado

Complejidad Técnica: Media. Se necesita procesamiento de datos históricos y gráficos dinámicos.

Recursos Disponibles: Se requiere integración con librerías de visualización de datos.

Impacto en la Experiencia del Usuario: Medio-Alto para administradores. No es crítico para los clientes.

Relación con los Objetivos del Proyecto: Aporta valor estratégico, pero puede agregarse en una fase posterior.

Justificación del Tiempo

- **2 días** para backend con lógica de recopilación de datos.
- **2 días** para frontend con gráficos interactivos.
- **1 día** para pruebas y validaciones.

9. Sugerencia de Restaurantes por Cercanía

Descripción del Trabajo:

Este módulo utilizará la ubicación del usuario para recomendar restaurantes cercanos. Se implementará un backend con lógica de geolocalización y cálculos de proximidad, junto con un frontend que muestre estas recomendaciones en un formato visual atractivo. También se ofrecerá la opción de ajustar manualmente la ubicación.

Clasificación: Must-Have

Días Estimados: 8 días

Costo Estimado: \$3,200,000 COP

Análisis Detallado

Complejidad Técnica: Media. Se requiere integración con servicios de geolocalización y un algoritmo de priorización.

Recursos Disponibles: Se necesita acceso a APIs de mapas y experiencia en filtrado de datos basado en distancia.

Impacto en la Experiencia del Usuario: Alto. Sin esto, los usuarios podrían recibir sugerencias irrelevantes.

Relación con los Objetivos del Proyecto: Crítico para asegurar la relevancia del contenido para cada usuario.

Justificación del Tiempo

- **3 días** para la integración de servicios de geolocalización en el backend.
- **3 días** para desarrollar el frontend con visualización de sugerencias.
- **2 días** para pruebas y validación con distintos escenarios de ubicación.

10. Tabla de estimaciones

Funcionalidad	Clasificación MoSCoW	Días	Costo estimado	Complejidad Técnica	Impacto en la experiencia de Usuario
Acceso a Menús digitales	Must-have	13	\$5,200,000	Media	Alto
Reserva de mesas	Should-have	8	\$3,200,000	Media	Medio-Alto
Realización de Pedidos	Must-have	13	\$5,200,000	Alta	Crítico
Seguimiento del Estado del pedido	Must-have	8	\$3,200,000	Media-Alta	Alto
Valoraciones y comentarios	Could-have	5	\$2,000,000	Baja	Medio
Gestión de Menús para	Must-have	13	\$5,200,000	Media	Crítico

restaurantes					
Gestión de pedidos en tiempo real	Should-have	8	\$3,200,000	Media-Alta	Alto
Estadísticas y Métricas de ventas	Could-have	5	\$2,000,000	Media	Medio-Alto
Sugerencias de restaurantes por cercanía	Must-have	8	\$3,200,000	Media	Alto

Análisis Gestión del software

1. Tiempo

Estimación de Desarrollo por Funcionalidad

Funcionalidad	Diseño	Desarrollo	Pruebas	Total
Acceso a menús digitales	3	5	5	13
Reserva de mesas	3	3	2	8
Realización de pedidos	5	5	3	13
Seguimiento de Estado del estado del pedido	3	3	2	8
Valoraciones y Comentarios	2	2	1	5
Gestión de menús para restaurantes	5	5	3	13
Gestión de pedidos en tiempo real	3	3	2	8
Estadísticas y métricas de	2	2	1	5

ventas				
Sugerencia de restaurantes por cercanía	3	3	2	8

Para un total de 81 días.

2. Costos

Costos Salariales

Categoría	Cantidad	Salario Mensual (COP)	Fuente
Desarrollador Junior	1	\$5.000.000	https://talently.tech/herramientas/colombia/salario?utm_source=chatgpt.com
Desarrollador Senior	1	\$11.500.000	https://talently.tech/herramientas/colombia/salario?utm_source=chatgpt.com
Tester QA	1	\$2.900.000	https://www.cooltesters.com/blog/salarios-en-software-testing-2022-colombia?utm_source=chatgpt.com
Diseñador UX/UI	1	\$4.000.000	https://co.talent.com/salario/?job=ux+ui&utm_source=chatgpt.com

Costo total mensual de salarios estimado: ~\$23.400.000 COP.

Infraestructura y Herramientas

Elemento	Costo Mensual (COP)	Referencia
Servicios en la Nube	\$645.000	Estimación basada en proveedores como AWS y Azure.
Base de datos PostgreSQL	\$215.000	aws.amazon.com
APIs Externas	\$860.000	Estimación basada en

		proveedores comunes de APIs.
Herramientas de desarrollo	\$430.000	Estimación basada en herramientas como GitHub, Postman, Figma, etc.

Total mensual en infraestructura y herramientas por mes: 2.150.000

Para un total de costos mensuales de 25.550.000 y como el proyecto tiene una duración de 81 días, osea tres meses el costo final del proyecto es de 76.650.000 (COP)

3. Alcance

Funcionalidades Incluidas en el MVP

Para el MVP (Producto Mínimo Viable), priorizamos las funcionalidades clave para que los usuarios puedan buscar restaurantes, realizar pedidos y hacer reservas.

Incluimos:

1. Acceso a Menús Digitales
 - Búsqueda y filtros por ubicación, tipo de comida, palabras clave.
 - Optimización para conexiones lentas.
2. Reserva de Mesas
 - Registro de reservas con confirmaciones automáticas.
 - Notificaciones recordatorias para los usuarios.
3. Realización de Pedidos
 - Pedido en restaurante, para recoger y a domicilio.
 - Personalización de platillos y validación de disponibilidad.
4. Seguimiento del Estado del Pedido
 - Notificaciones en tiempo real sobre el estado del pedido.
 - Historial de pedidos.

Funcionalidades Post-MVP

Las siguientes funcionalidades se implementarán en versiones futuras:

5. Sugerencia de Restaurantes por Cercanía
 - Recomendaciones basadas en la ubicación del usuario.
 - Filtros según tipo de comida y valoraciones.
6. Valoraciones y Comentarios

- Sistema de clasificaciones y reseñas de usuarios.
- 7. Gestión de Menús para Restaurantes
 - Interfaz para que los restaurantes editen y actualicen sus menús.
- 8. Gestión de Pedidos en Tiempo Real
 - Tablero de control para visualizar pedidos en vivo.
- 9. Estadísticas y Métricas de Ventas
 - Reportes de ventas, horarios de mayor demanda.

Diseño y arquitectura

Arquitectura del sistema

En cuanto a la arquitectura del sistema elegimos un modelo vista controlador (MVC), esto debido a que esta nos permite optimizar los recursos y nos deja tener escalabilidad sobretodo a largo plazo, también el MVC nos facilita la organización del código y poder separar mejor las responsabilidades en cuanto al modelo de la base de datos, el frontend que es la interfaz del usuario en react y la lógica del negocio y el controlador en springboot. También la escogimos por el hecho de que miramos este proyecto a largo plazo entonces el sistema debe ser capaz de poder crecer sin una pérdida de rendimiento teniendo escalabilidad fácil debido a que cada componente es independiente por lo que por ejemplo el front se puede cambiar sin alterar la base de datos, se pueden agregar funciones nuevas sin modificar o alterar todo el sistema y además que springboot deja que el backend pueda escalar si es necesario en el futuro con microservicios sin afectar todo el funcionamiento de la página.

Diseño de bases de datos

En cuanto a las decisiones tomadas para el diseño del esquema de la base de datos tenemos esto:

Normalización

En cuanto a la normalización tratamos de eliminar redundancias dividiendo las entidades en tablas relacionadas, además de esto creamos relaciones entre entidades.

Índices

Para los índices se establecieron correctamente entre las claves primarias y foráneas para poder optimizar las consultas y mantenibilidad, también el email de los usuarios es único para evitar los duplicados en los datos.

Claves primarias y foráneas

En cuanto a este punto cada tabla tiene su clave primaria serial y las relaciones están puestas con claves foráneas para que se pueda asegurar la integridad de los datos.

Restricciones

Para las restricciones establecimos valores no nulos en campos clave que lo necesitaban y pusimos un check en algunos campos importantes para que no puedan introducir valores inválidos.

Escalabilidad

Este apartado de la escalabilidad se evidencia en que por ejemplo hay una relación entre pedidos y detalles_pedido que hace que se puedan poner múltiples platillos en un pedido además que al tener los menús y platillos separados ayuda a que sea más fácil las actualizaciones en las respectivas categorías.

Para concluir este apartado de el diseño de la bases de datos hay que decir que escogimos una base de datos relacional debido a varios factores como que necesitábamos tener relaciones bien definidas entre entidades ya que en un restaurante hay muchas relaciones entre datos, necesitábamos usar la normalización y el uso de claves primarias y foráneas para evitar que haya datos huérfanos o inconsistentes, también es necesario que haya eficiencia y que se puedan realizar consultas complejas y por último también era necesario tener una escalabilidad y estabilidad en la base para manejar una gran cantidad de datos y conexiones simultaneas sin perder estabilidad.