**Delivwich**

Integrantes

Jose otero

Juan David Serna

David Santiago Davila

# 

# 

# 2025

# 

# 

# 

# 

# Índice

[**Índice 2**](#_6mn3kk2nbloh)

[Contexto 3](#_q3jq0lewtt6q)

[Requisitos Funcionales 3](#_hqnjajb8nvkj)

[Requisitos No Funcionales 5](#_eloqnmetq1hq)

[Interesados 8](#_aja96zy6iao)

[Interesados VS Requerimientos No Funcionales 8](#_ap5056k82949)

[Los Atributos de Calidad Más Importantes 9](#_utg8tpusbr7q)

[Patrones Arquitectónicos a Utilizar 11](#_tuvuu6crvcku)

[Vistas 12](#_cqtunkv6pk54)

[Vista de Escenarios 12](#_egny3zevyap)

[Vista Lógica 15](#_togejriaazqx)

[Actividades 17](#_ncmqbtt4jx2c)

[Vista de Procesos 18](#_ociqw1ls3v79)

[Vista Física 20](#_wpf1m7slwqgz)

# 

# Contexto

Se requiere diseñar una solución tecnológica que permita a los usuarios realizar pedidos de productos alimenticios a través de canales digitales, específicamente mediante una aplicación móvil y una plataforma web. Esta funcionalidad complementará los métodos de pedido tradicionales y debe ofrecer una experiencia de usuario fluida, segura y personalizada.

El sistema debe permitir la selección de productos, aplicación de promociones, elección de modalidad de entrega (a domicilio o retiro en local), y múltiples opciones de pago. Además, debe integrarse con servicios internos como gestión de cocina, localización, despacho y usuarios. La solución debe ser escalable para soportar altos volúmenes de transacciones en horarios pico y debe incluir monitoreo, autenticación, y trazabilidad de eventos.

# Interesados

| **Nombre** | **Rol** | **Interés en el Sistema** |
| --- | --- | --- |
| Clientes | Usuario consumidor | Realizar pedidos fácilmente, recibir el producto a tiempo, acceder a promociones. |
| Repartidor | Usuario Repartidor | Recibir notificaciones de pedidos, Poder marcar el pedido como entregado, marcar alguna novedad con el pedido, poder ver la dirección del cliente |
| Dueños de Franquicia | Administrador local | Gestionar productos y precios, ver pedidos, aplicar promociones locales, obtener ganancias del local. |
| Dueños de la Marca | Administrador Corporativo | Gestionan productos y precios, aplican promociones nacionales, obtienen ganancias generales. |
| TI Corporativo | Equipo técnico central | Asegurar que el sistema sea seguro, escalable, disponible y fácil de mantener. |
| Marketing | Gestores de campañas | Crear y gestionar promociones a nivel nacional, para mejorar la experiencia del cliente. |
| Legal / Cumplimiento | Supervisión legal y regulatoria | Garantizar que el sistema cumpla con normas locales/internacionales (ej. pagos, datos personales). |

# Requisitos Funcionales

| ID | Nombre | Descripción | Interesados | Prioridad |
| --- | --- | --- | --- | --- |
| RF01 | Realizar pedido | El usuario puede crear un pedido seleccionando productos desde la app/web. | Cliente, Dueño de franquicia | Alta |
| RF02 | Asignar hora de recogida | El sistema calcula y muestra una hora estimada para recoger el pedido. | Cliente | Alta |
| RF03 | Instrucciones de geolocalización | Se integran servicios externos para dar indicaciones de llegada. | Cliente | Media |
| RF04 | Envío a domicilio | Si el local lo permite, se asigna un repartidor para entregar el pedido. | Cliente, Repartidor, Dueño de franquicia | Alta |
| RF05 | Aplicar promociones locales | Se aplican promociones definidas por cada franquicia. | Cliente, Dueño de franquicia | Media |
| RF06 | Aplicar promociones nacionales | Se aplican promociones definidas a nivel corporativo. | Cliente, Administrador corporativo | Alta |
| RF07 | Aceptar pagos en línea | El usuario puede pagar el pedido mediante plataformas de pago digital. | Cliente, Dueño de franquicia | Alta |
| RF08 | Aceptar pago contra entrega | El usuario puede pagar al recibir o retirar su pedido. | Cliente, Repartidor, Dueño de franquicia | Alta |
| RF09 | Soporte para dispositivos móviles | El sistema debe funcionar correctamente en dispositivos móviles. | Cliente, Equipo técnico | Alta |
| RF10 | Gestión de productos por local | Cada franquicia puede definir sus productos y precios. | Dueño de franquicia, Administrador | Media |
| RF11 | Visualizar estado del pedido | El cliente puede ver el estado de su pedido (preparando, en camino, etc.). | Cliente | Media |
| RF12 | Registro e inicio de sesión | Los usuarios deben poder registrarse e iniciar sesión de forma segura. | Cliente, Equipo técnico | Alta |

# 

# Requisitos No Funcionales

| **ID** | **Nombre** | **Descripción** | **Interesados** | **Prioridad** | **Métrica** |
| --- | --- | --- | --- | --- | --- |
| RNF 01 | Alta Disponibilidad | El sistema debe estar disponible al menos 99.9% del tiempo para los usuarios finales. | Clientes, Dueños de franquicias, TI Corporativo | 1 | Uptime mensual ≥ 99.9% (medido por CloudWatch o Pingdom) |
| RNF 02 | Escalabilidad Horizontal | El sistema debe poder escalar automáticamente ante aumento de carga, especialmente en horas pico y campañas promocionales. | TI Corporativo | 1 | Tiempo de respuesta medio ≤ 1.5s con 10,000 usuarios concurrentes |
| RNF 03 | Tiempo de Respuesta | El sistema debe responder a las acciones del usuario en menos de 2 segundos en el 95% de los casos. | Clientes, Marketing | 1 | p95 de respuesta < 2s en pruebas de carga |
| RNF 04 | Soporte Multiplataforma | El sistema debe ser accesible desde navegadores móviles, tablets y computadoras. | Clientes, Marketing | 1 | Compatible con Chrome, Safari, Firefox, Edge en móvil y escritorio |
| RNF 05 | Seguridad en pagos | El sistema debe cumplir con el estándar PCI DSS para el manejo de pagos electrónicos. | Clientes, TI Corporativo | 1 | Certificación de cumplimiento PCI DSS o integración con proveedor compliant |
| RNF 06 | Privacidad de Datos | El sistema debe proteger los datos personales de los usuarios conforme al marco legal local e internacional. | Clientes, Legal, TI Corporativo | 1 | Cumplimiento con GDPR y ley de protección de datos local |
| RNF 07 | Portabilidad | La arquitectura debe permitir desplegar el sistema en múltiples regiones de AWS para expansión internacional. | TI Corporativo | 2 | Capacidad de deployment en 2+ regiones AWS sin cambios sustanciales |
| RNF 08 | Trazabilidad | El sistema debe permitir auditar las acciones clave de usuarios y operadores del sistema. | TI Corporativo, Legales | 2 | Logs auditables con retención de 90 días |
| RNF 09 | Tolerancia a fallos | El sistema debe recuperarse automáticamente ante fallas parciales sin interrumpir el servicio general. | Clientes, TI Corporativo | 1 | Failover automático probado en pruebas de resiliencia |
| RNF 10 | Integración con terceros | El sistema debe integrarse con servicios de geolocalización, tráfico, pasarelas de pago y gestión de entregas. | TI Corporativo | 1 | Integración activa con al menos 3 servicios externos |
| RNF 11 | Accesibilidad | El sistema debe cumplir con estándares mínimos de accesibilidad (WCAG 2.1 AA). | Clientes, TI Corporativo | 3 | Validación de accesibilidad automática con herramientas estándar |
| RNF 12 | Tiempo de carga inicial | La carga inicial de la aplicación en móvil debe completarse en menos de 3 segundos. | Clientes, Marketing | 2 | Lighthouse score > 80, carga inicial < 3s en red 4G simulada |

# Atributos de Calidad

|  | Clientes | Repartidor | Dueños Franquicia | Dueños Marca | TI Corporativo | Marketing | Legal / Cumpl. | Negociación (%) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Disponibilidad | 20 | 20 | 10 | 10 | 10 | 20 | 10 | 15,714% |
| Rendimiento | 15 | 15 | 10 | 10 | 10 | 15 | 0 | 10,714% |
| Usabilidad | 25 | 20 | 10 | 5 | 5 | 30 | 0 | 13,571% |
| Escalabilidad | 5 | 5 | 10 | 20 | 20 | 0 | 0 | 8,571% |
| Mantenibilidad | 0 | 0 | 0 | 10 | 15 | 0 | 0 | 3,571% |
| Seguridad | 10 | 10 | 10 | 15 | 15 | 10 | 50 | 15,714% |
| Compatibilidad | 10 | 10 | 20 | 5 | 5 | 5 | 0 | 7,857% |
| Portabilidad | 0 | 0 | 10 | 10 | 15 | 0 | 0 | 5% |
| Confiabilidad | 15 | 20 | 20 | 15 | 5 | 20 | 40 | 19,286% |
| TOTAL  (%) | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100% |

Ponderación

| **Atributo** | **Descripción** | **Impacto** | **Dificultad** | **Peso** | **Valor** |
| --- | --- | --- | --- | --- | --- |
| Escalabilidad | Capacidad del sistema para manejar desde miles hasta millones de usuarios, incluyendo eventos nacionales y locales simultáneamente |  |  |  |  |
| Disponibilidad | El sistema debe estar disponible 24/7 , especialmente durante horas pico,evitando interrupciones que afecten las ventas |  |  |  |  |
| Mantenibilidad | Necesario para adaptarse a promociones diarias,integración con nuevas franquicias y expansión internacional |  |  |  |  |
| Usabilidad | El sistema debe ser intuitivo y usable desde dispositivos móviles, con experiencia clara y fluida para el cliente |  |  |  |  |
| Integrabilidad | Se requiere conexión con múltiples APIs de geolocalización, sistemas de pago, logística y franquicias. |  |  |  |  |

# 

# 

# Los Atributos de Calidad Más Importantes

| Atributo | Descripción | Justificación |
| --- | --- | --- |
| Escalabilidad | Capacidad del sistema para crecer de forma eficiente en respuesta a un aumento de usuarios, locales o volumen de pedidos. | El sistema debe atender a miles de usuarios simultáneamente, especialmente durante promociones nacionales o en la expansión internacional. Es clave para garantizar el rendimiento y experiencia del usuario. |
| Disponibilidad | El sistema debe estar accesible en todo momento sin interrupción | La venta depende directamente de la disponibilidad del sistema. Cualquier falla afecta ingresos y reputación. Imprescindible para operaciones continuas 24/7. |
| Mantenibilidad | Facilidad con la que el sistema puede ser actualizado, corregido o extendido. | Las promociones diarias, personalización por local y la futura internacionalización exigen un sistema flexible y fácilmente modificable sin afectar el servicio. |
| Usabilidad | Facilidad de uso e interacción desde diversos dispositivos (principalmente móviles). | El sistema debe conectarse con servicios externos como mapas, tráfico, pasarelas de pago y sistemas de logística, por lo que una arquitectura integrable es fundamental. |
| Integrabilidad | Capacidad del sistema para integrarse con APIs de terceros y otros servicios internos. | Los usuarios deben poder ordenar rápidamente desde sus celulares, con una experiencia intuitiva y sin fricciones que favorezca la conversión. |

Tácticas para atacar los atributos

| Atributo de Calidad | Tácticas Específicas |
| --- | --- |
| Escalabilidad | - Arquitectura basada en microservicios  - Autoescalado horizontal en cloud (EKS, ECS, GKE)  - Uso de colas (ej. RabbitMQ, Kafka) para desacoplar procesos  - Cache distribuido (Redis, Memcached) |
| Disponibilidad | - Balanceadores de carga (ALB, NGINX)  - Múltiples instancias por servicio con failover  - Health checks + restart automático  - Réplicas de base de datos y backups frecuentes |
| Mantenibilidad | - Separación por dominios lógicos (modularidad)  - API bien documentadas (OpenAPI / Swagger)  - CI/CD con pruebas automáticas y linting  - Logs centralizados y trazabilidad con observabilidad (Grafana, ELK) |
| Usabilidad | - Interfaces responsive (web/mobile)  - Flujos claros y consistentes (UX first)  - Validaciones en tiempo real y feedback inmediato  - Accesibilidad (a11y), lenguaje claro y botones visibles |
| Integrabilidad | - Uso de APIs RESTful bien definidas  - Documentación pública para terceros  - Adaptadores para servicios externos (Geo, pagos)  - Uso de formatos estándar (JSON, OAuth2, webhooks) |

# 

# Patrones Arquitectónicos a Utilizar

| Patrón Arquitectónico | Rol en el sistema | Justificación |
| --- | --- | --- |
| Microservices Architecture | Patrón base principal | Permite escalar servicios por separado, delegar funciones a franquicias si es necesario, facilitar CI/CD y expansión internacional. Encaja con arquitectura AWS distribuida.  Ideal para servicios como: usuarios, pedidos, pagos, promociones, geolocalización, notificaciones. |
| Event-driven Architecture | Mecanismo de comunicación entre servicios | Alta eficiencia y resiliencia. Encaja perfectamente con eventos como creación de pedidos, cambios de estado, envío de mensajes, pagos. AWS proporciona herramientas nativas: EventBridge, SQS, SNS. |
| Layered Architecture | Patrón interno de cada microservicio | Proporciona orden lógico y separación de responsabilidades dentro de cada microservicio. Facilita pruebas, mantenimiento y comprensión del código. |
| Client-Server Architecture | Modelo de interacción cliente/API | Aplicaciones móviles y web actúan como clientes. Se comunican con los servidores a través de HTTP/HTTPS y API Gateway. Patrón clásico compatible con los otros. |

Esta combinación es perfectamente compatible con AWS, permite escalar por franquicia, soporta expansión internacional, y garantiza alta disponibilidad, mantenibilidad y rendimiento, cumpliendo con los atributos de calidad priorizados.

Estrategia

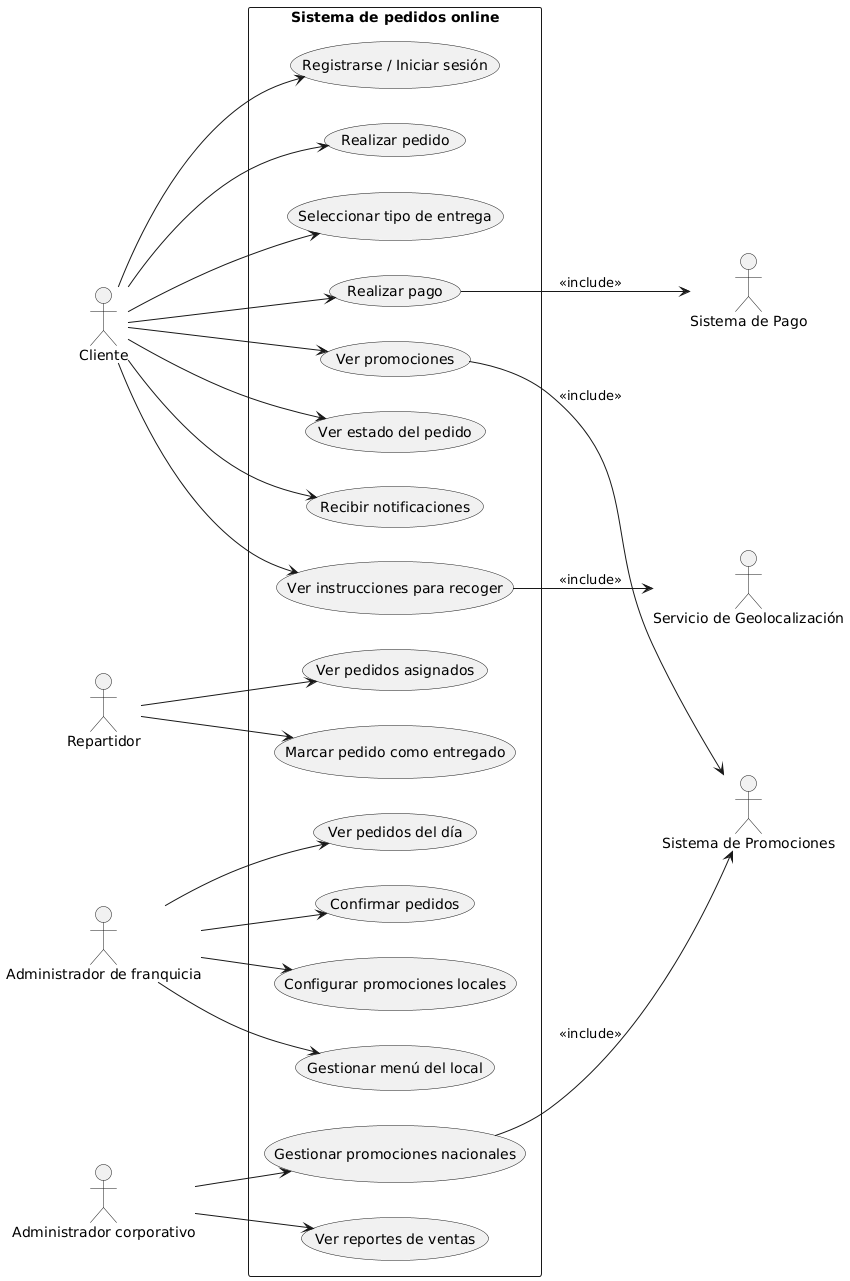
| **Atributo de Calidad** | **Tácticas Aplicadas** | **Justificación** |
| --- | --- | --- |
| Disponibilidad | Redundancia, Failover, Health Checks | Minimiza el impacto de fallos. Alta disponibilidad con balanceadores y multizonas. |
| Escalabilidad | Desacoplamiento, Auto Scaling, Contenedores | Permite crecer horizontalmente sin afectar otros servicios. |
| Modificabilidad | Separación de responsabilidades, inyección de dependencias | Cambios localizados. Cada microservicio evoluciona sin afectar a otros. |
| Rendimiento | Caching, asincronismo, compresión de payloads | Mejora el tiempo de respuesta y reduce la carga. |
| Integrabilidad | Adaptadores, API Gateway, colas de eventos | Facilita integración con mapas, pagos y notificaciones externas. |

# Vistas

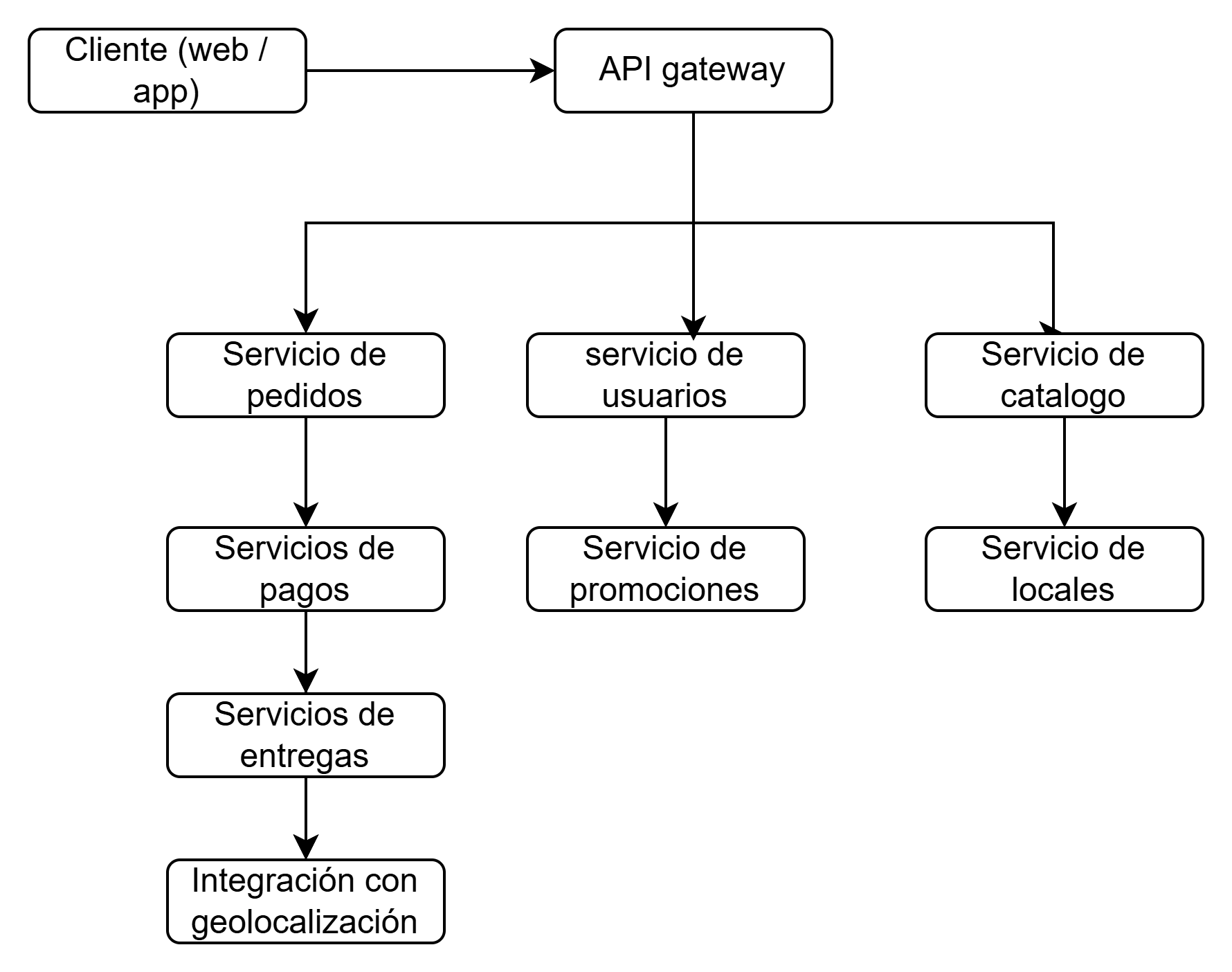
## Vista de Escenarios

Se representan los principales escenarios funcionales del sistema de pedidos de sándwiches en línea, desde el punto de vista de los usuarios (actores) y sus interacciones con el sistema. Se modelan los comportamientos esperados del sistema sin entrar en detalles de implementación.

| **Elemento** | **Descripción** |
| --- | --- |
| Actor | Representa a los usuarios o sistemas externos que interactúan con el sistema. |
| Usecase | Define una funcionalidad o comportamiento esperado del sistema. |
| Package | Agrupa los casos de uso dentro del sistema para representar límites del sistema. |
| --> | Indica una relación entre actor y caso de uso (el actor ejecuta esa funcionalidad). |
| <<include>> | Representa un caso de uso incluido en otro, es decir, un comportamiento reutilizable o dependiente. |



## Vista Lógica

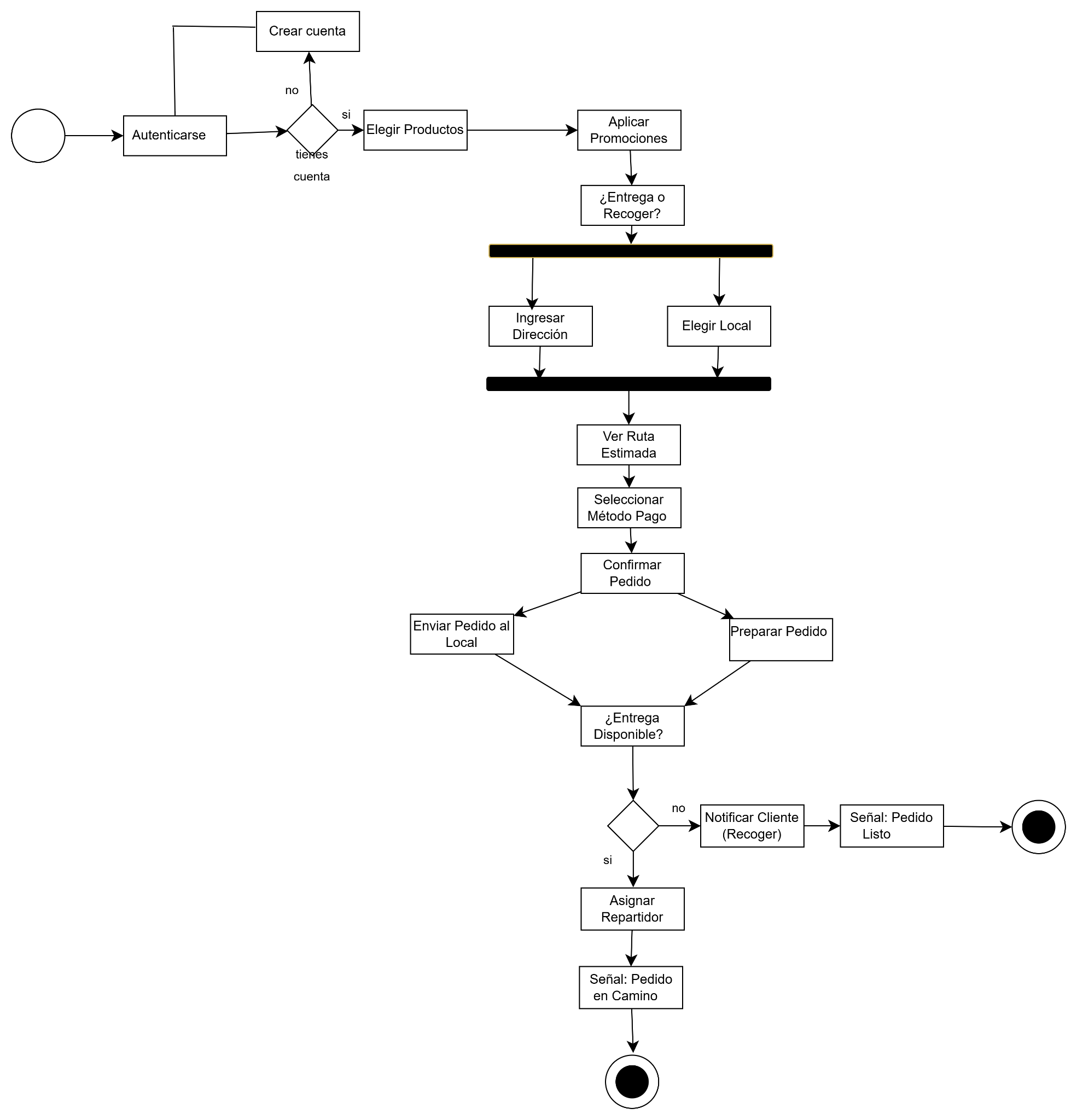


Esta es la Vista Lógica del Sistema, que representa cómo se comunican los diferentes servicios y componentes funcionales dentro de la arquitectura, sin tener en cuenta la infraestructura física (como servidores o redes).

Lo que estás viendo es el flujo lógico de información y responsabilidades entre componentes clave desde que un usuario accede al sistema hasta que se gestionan pedidos, pagos, entregas y consultas. Muestra cómo están conectados entre sí y qué rol cumple cada uno para lograr la funcionalidad esperada del sistema.

| **Elemento** | **Significado / Papel en la Vista** |
| --- | --- |
| Cliente (Web/App) | Representa al usuario final que interactúa con el sistema desde una app móvil o interfaz web. |
| API Gateway | Punto de entrada para todas las solicitudes del cliente. Encaminador principal hacia los servicios. |
| Servicio de Pedidos | Encargado de la lógica de creación, gestión y seguimiento de pedidos. Coordina pagos, entregas, promos. |
| Servicio de Usuarios | Gestiona el registro, autenticación y manejo de la información del usuario. |
| Servicio de Catálogo | Proporciona la información de los productos disponibles para el usuario. |
| Servicio de Pagos | Responsable de procesar pagos y comunicarse con la pasarela externa de pagos. |
| Pasarela de Pago Externa | Sistema externo que efectúa el cobro real, ya sea tarjeta de crédito, débito u otros métodos. |
| Servicio de Entregas | Organiza la logística del envío del pedido, coordinando con servicios de localización y almacenes. |
| Servicio de Locales | Proporciona datos sobre almacenes o puntos físicos disponibles para despacho o retiro. |
| Servicio de Geolocalización | Obtiene y procesa datos de ubicación para rutas de entrega. |
| API de Mapas/Tráfico | Plataforma externa que ofrece mapas, tráfico y coordenadas en tiempo real. |
| Servicio de Promociones | Aplica descuentos, promociones y beneficios a pedidos de clientes elegibles. |

## Actividades



Esta vista representa el flujo lógico del sistema desde la interacción del cliente hasta la preparación y entrega del pedido. Se enfoca en mostrar cómo el cliente, el sistema backend y el local colaboran a través de una secuencia de acciones que reflejan la funcionalidad principal de la plataforma: realizar un pedido y entregarlo o prepararlo para recogida.

*Línea de nado (Lane): Cliente*

* Inicio: Punto donde comienza la interacción del usuario con el sistema.
* Autenticarse: Verifica su identidad (probablemente con Cognito).
* Elegir Productos: Selecciona ítems desde el catálogo.
* Aplicar Promociones: Verifica y aplica descuentos disponibles.
* ¿Entrega o Recoger?: Define el flujo que seguirá (entrega a domicilio o recogida en tienda).
* Ingresar Dirección / Elegir Local: Dependiendo de la opción, selecciona dónde se enviará el pedido o en qué local lo recogerá.
* Ver Ruta Estimada: Muestra tiempo y distancia (requiere integración con servicio de geolocalización).
* Seleccionar Método de Pago: Elige cómo pagar (conexión a pasarela de pagos).
* Confirmar Pedido: Envia el pedido al sistema.

*Línea de nado: Sistema*

* Enviar Pedido al Local: Redirige la solicitud al local más cercano o seleccionado.
* Sincronización: Etapa abstracta que representa la coordinación entre componentes del sistema (APIs, microservicios, base de datos).

*Línea de nado: Local*

* Preparar Pedido: El personal del local inicia la preparación del pedido.
* ¿Entrega Disponible?: Verifica si hay repartidor.
  + Sí: Asignar Repartidor: El pedido será entregado (usa microservicio de entregas).
  + No: Notificar Cliente (Recoger): Indica al cliente que debe ir al local.
* Señal: Pedido en Camino / Pedido Listo: Comunicación de estado al cliente (puede implicar eventos SNS/SQS).

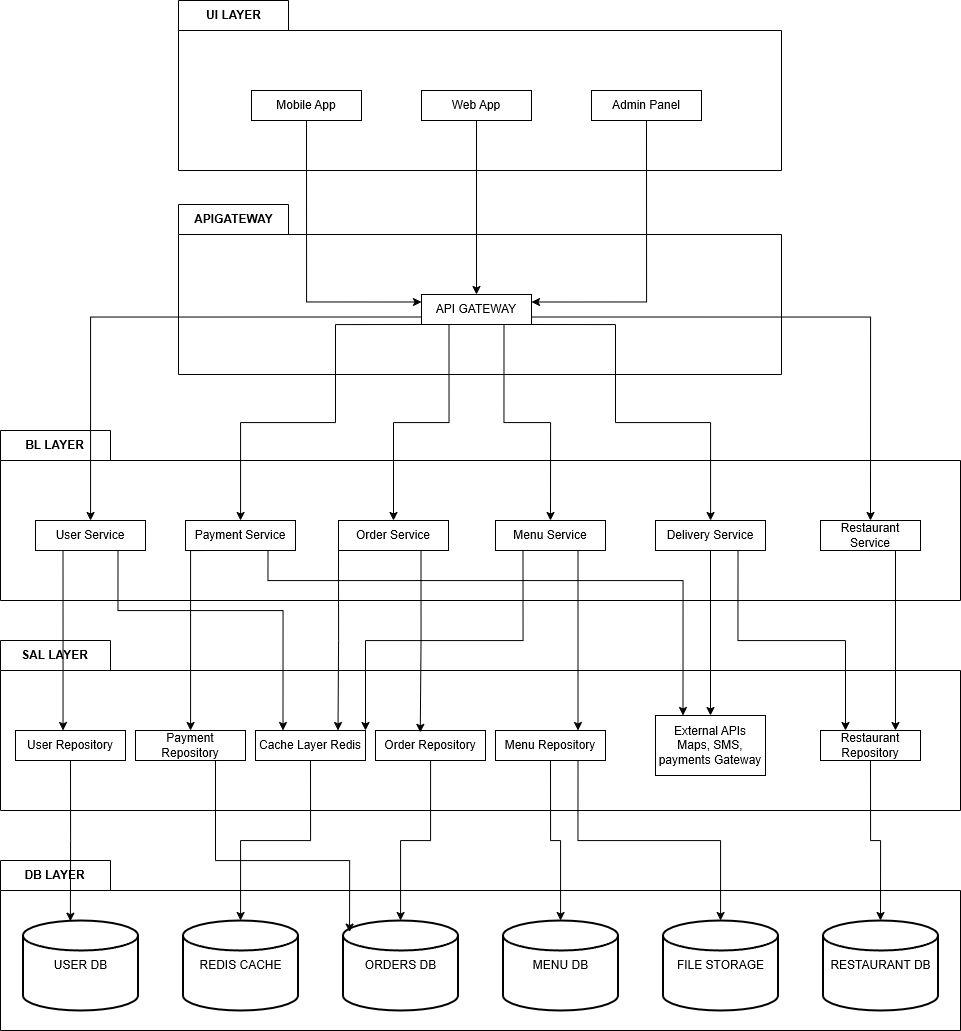
## 

## 

## 

## 

## Vista de implementación

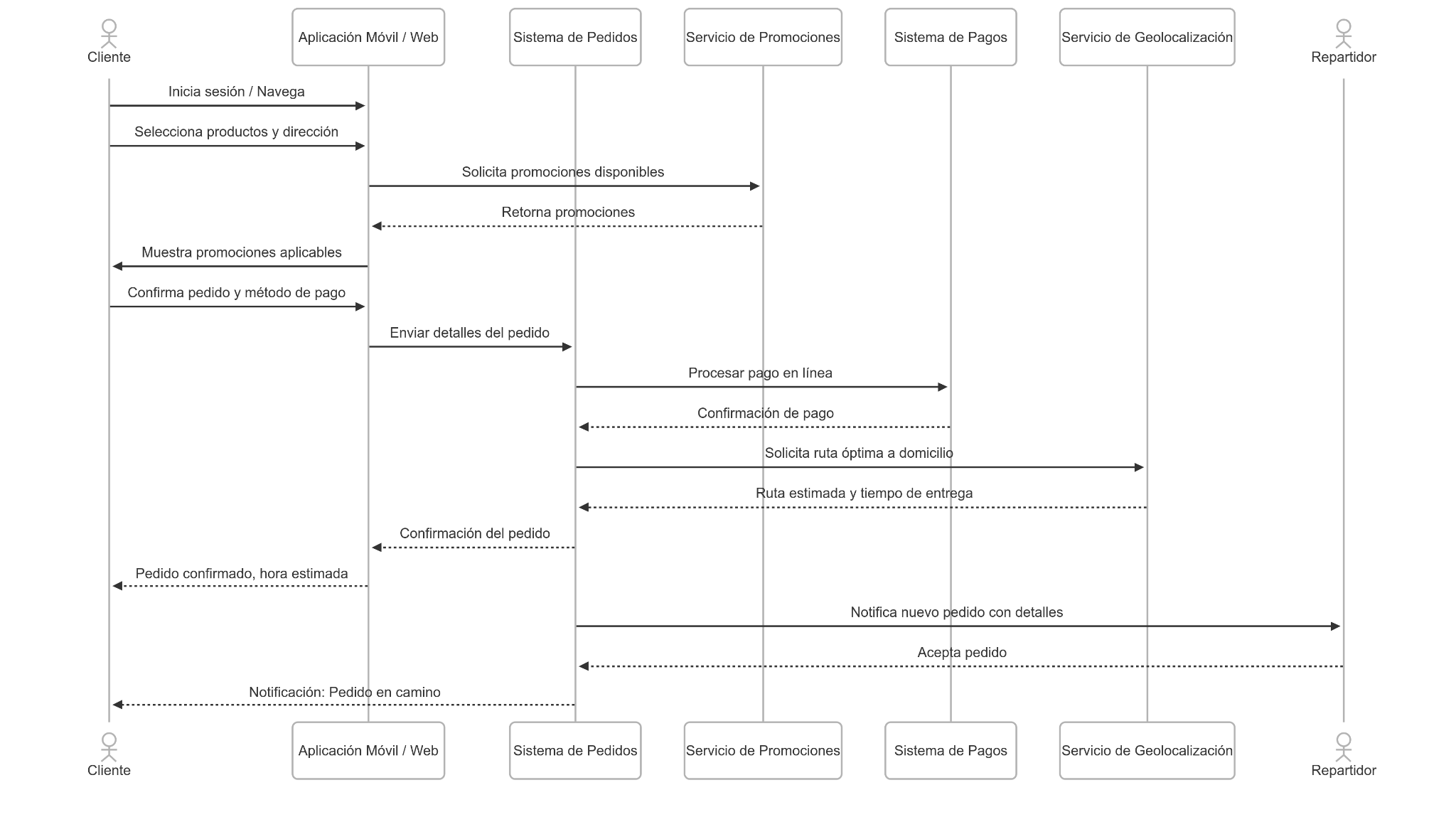


## 

## 

## Vista de Procesos

Escenario seleccionado: Realizar pedido para entrega a domicilio con pago en línea



Se presenta el flujo de una de las funcionalidades más importantes que se pueda presentar en el sistema, que es la realización de un pedido con pago en línea por parte de un cliente.

Flujo:

1. El cliente selecciona sus productos y dirección.
2. El sistema consulta las promociones vigentes.
3. Al confirmar el pedido, se envían los datos al backend, que realiza el cobro con el sistema de pagos.
4. Se consulta la ruta más óptima mediante un servicio de geolocalización.
5. Se confirma el pedido al cliente y se notifica al repartidor.
6. El repartidor acepta el pedido, y se notifica al cliente que su sándwich está en camino.

| Elemento | Descripción |
| --- | --- |
| Actor | Usuario final que realiza el pedido desde un dispositivo móvil o navegador y repartidor que recibe notificaciones de pedidos |
| Participante | Son las herramientas o servicios que utiliza el cliente para realizar el pedido y que el repartidor usa para recibir el pedido |

## Vista Física ( POR CORREGIR)