



**ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE**



**Integrantes:** Gabriel Maldonado, Daniel Moncayo y Jhair Zambrano

**Curso:** GR2SW

**Fecha:** 6 de diciembre de 2025

**Examen 01**

**Aplicación**

Uber Eats

**Descripción**

Plataforma de “delivery” bajo demanda que conecta a usuarios con restaurantes locales y repartidores independientes. Permite explorar menús, realizar pedidos pagar digital y físicamente, así como rastrear la entrega en tiempo real mediante GPS.

**Requisitos Funcionales**

1. Como **usuario**, quiero buscar restaurantes por tipo de comida para encontrar rápidamente lo que se me antoja comer.
2. Como **usuario** que espera su comida, quiero ver la ubicación del repartidor en un mapa en tiempo real, para saber exactamente cuándo salir a recibir el pedido.
3. Como **gerente** de restaurante, quiero marcar platos como "agotados" en tiempo real, para evitar recibir pedidos que no puedo preparar y tener que cancelarlos.
4. Como **repartidor**, quiero recibir una notificación con la ganancia estimada y la distancia antes de aceptar un pedido, para decidir si me conviene realizar el viaje.
5. Como **usuario** nuevo, quiero registrarme utilizando mi cuenta de Google o Facebook, para acceder a la aplicación rápidamente sin tener que recordar una nueva contraseña.
6. Como **usuario** con restricciones alimenticias, quiero poder añadir notas específicas a cada plato (ej. "sin cebolla", "alergia al maní"), para asegurar que la comida sea segura para mi consumo.
7. Como **usuario** al momento de pagar, quiero gestionar múltiples tarjetas de crédito o débito, para elegir con cuál pagar según mi disponibilidad de saldo.
8. Como **usuario** que finalizó un pedido, quiero calificar al restaurante y al repartidor por separado, para premiar el buen servicio o reportar problemas de calidad.
9. Como **usuario** ahorrador, quiero poder ingresar códigos promocionales en el carrito de compras, para obtener descuentos en el total de mi pedido.
10. Como **usuario** con un problema, quiero acceder a un centro de ayuda automatizado sobre mi pedido reciente (ej. "faltaron ítems"), para solicitar un reembolso sin necesidad de llamar por teléfono.

**Diagrama de arquitectura**



# ESCUELA POLITÉCNICA NACIONAL

## FACULTAD DE INGENIERÍA DE SISTEMAS

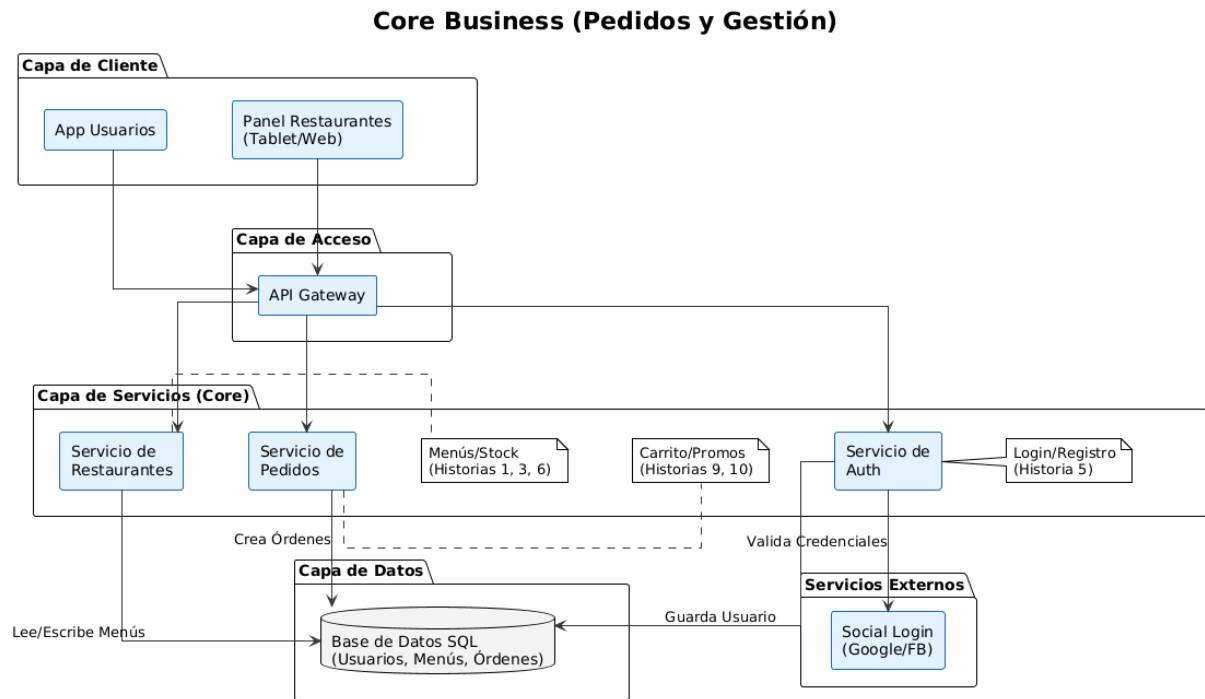
### CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE



Debido a la complejidad de la plataforma Uber Eats, se ha dividido la arquitectura en dos vistas lógicas para facilitar su comprensión.

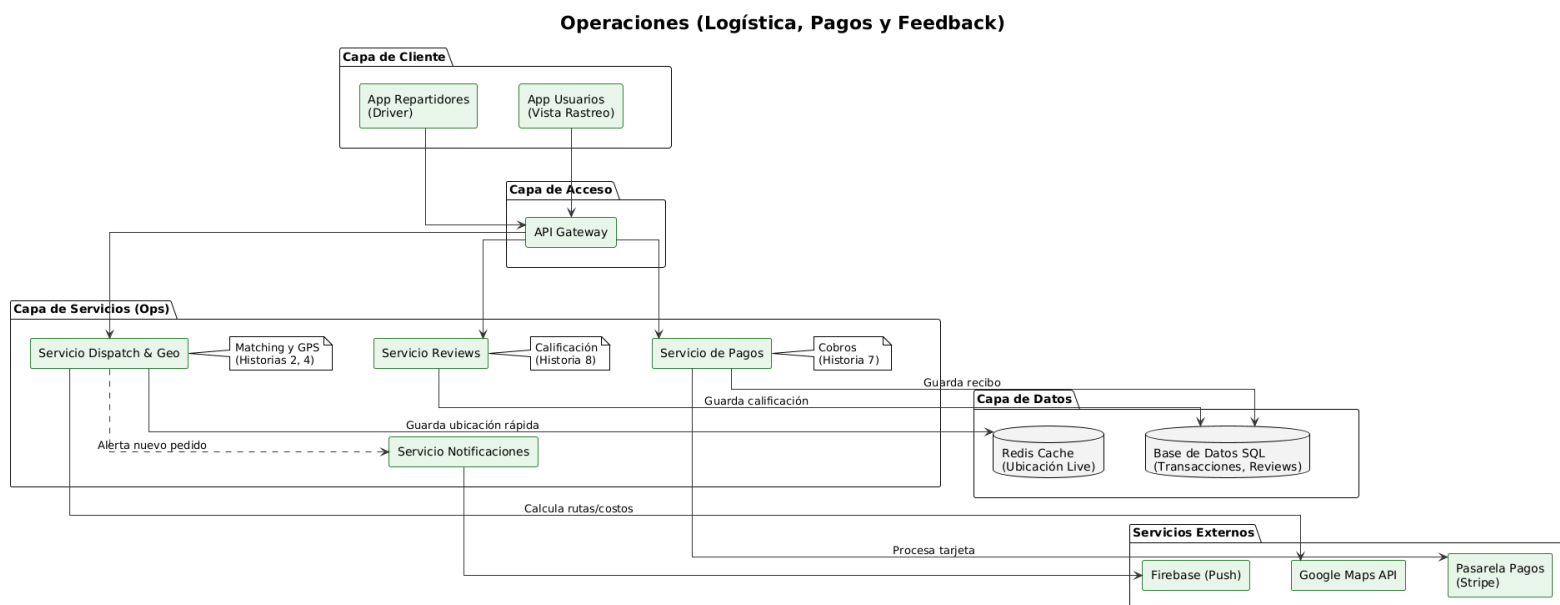
## 1. Diagrama del Negocio Principal

Enfocada en la gestión de contenido (restaurantes) y la captura de pedidos.



## 2. Diagrama de Operaciones (Logística y Ejecución)

Enfocada en la logística de reparto, geolocalización en tiempo real y transacciones financieras.





## Modelo de datos

El sistema requiere un modelo de base de datos relacional para la persistencia de la información transaccional, complementado con un almacenamiento en caché para datos volátiles (ubicación). Las entidades principales identificadas son:

- **Usuario (User):** Almacena la identidad del cliente.
  - Datos clave: ID, Nombre, Email, Token de autenticación social (Historia 5), Preferencias dietéticas (Historia 6), Ubicaciones guardadas.
- **Restaurante (Restaurant):** Contiene la información de los comercios afiliados.
  - Datos clave: ID, Nombre, Tipo de comida (Categoría para búsqueda - Historia 1), Ubicación GPS, Estado operativo (Abierto/Cerrado).
- **Plato/Item (Menu\_Item):** Representa los productos vendibles.
  - Datos clave: Nombre, Precio, Foto, Estado de disponibilidad (Disponible/Agotado - Historia 3).
- **Pedido (Order):** La entidad central que vincula usuarios, restaurantes y repartidores.
  - Datos clave: Estado (Preparando/En camino/Entregado), Código promocional aplicado (Historia 9), Costo total, Timestamp de creación.
- **Detalle de Pedido (Order\_Details):** Relaciona los platos con un pedido específico.
  - Datos clave: ID Pedido, ID Plato, Cantidad, Notas de preparación (ej: "Sin cebolla" - Historia 6).
- **Repartidor (Driver):** Perfil de los socios conductores.
  - Datos clave: ID, Vehículo, Licencia, Saldo acumulado/Ganancias (Historia 4), Calificación promedio.
- **Ubicación en Tiempo Real (Live\_Location - Redis):** Datos efímeros de alta velocidad.
  - Datos clave: ID Repartidor, Latitud, Longitud, Timestamp (Actualizado cada 3-5s para Historia 2).
- **Calificación (Review):** Feedback post-servicio.
  - Datos clave: Puntaje (1-5 estrellas), Comentario, Tipo de objetivo (Restaurante o Repartidor - Historia 8).
- **Método de Pago (Payment\_Method):**
  - Datos clave: Token de tarjeta encriptado, Tipo (Crédito/Débito/Paypal - Historia 7).

## Reglas de Negocio

A partir del análisis del modelo de negocio de Uber Eats (plataforma multilateral) y la observación del funcionamiento de la aplicación, se identificaron las siguientes reglas de negocio que gobiernan su operación:



**ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE**



- El modelo de ingresos se basa en comisiones múltiples. La plataforma debe ser capaz de calcular y separar automáticamente los ingresos en cada transacción: una comisión cobrada al restaurante por el uso del servicio, una tarifa de envío cobrada al usuario y la ganancia neta para el repartidor.
- La asignación de pedidos está restringida por geolocalización (Radio de Cobertura). Para garantizar la frescura de los alimentos y tiempos de entrega razonables (Value Proposition), los usuarios solo pueden visualizar y pedir de restaurantes que se encuentren dentro de un radio geográfico calculado dinámicamente respecto a su ubicación de entrega.
- Validación obligatoria de socios repartidores (Trust & Safety). Antes de poder recibir pedidos ("Go Online"), todo repartidor debe haber pasado un proceso de verificación de documentos (licencia, antecedentes, seguro). El sistema debe bloquear el acceso si la documentación expira.
- Política de cancelación dependiente del estado del pedido. Para proteger económicamente a los restaurantes, el usuario puede cancelar sin costo solo si el pedido está en estado "Pendiente". Una vez que el restaurante marca el pedido como "En Preparación", cualquier cancelación debe cobrar el costo total o parcial al usuario.
- El sistema de calificación debe ser bilateral y ciego. Tanto el usuario como el repartidor deben poder calificarse mutuamente al finalizar la entrega para mantener la calidad de la comunidad. Las calificaciones son anónimas en el momento de visualizarlas para evitar represalias inmediatas.
- Privacidad de contacto mediante enmascaramiento (Data Privacy). La comunicación telefónica entre el repartidor y el usuario debe realizarse a través de un sistema de enmascaramiento de números (proxy) o VoIP dentro de la app, protegiendo los números telefónicos reales de ambas partes.

## Referencias

- [1]. Uber Engineering, "Uber Engineering Blog," Uber, 2024. [En línea]. Disponible en: <https://www.uber.com/en-US/blog/engineering/>
- [2]. Google Cloud, "Google Maps Platform Documentation," 2024. [En línea]. Disponible en: <https://developers.google.com/maps/documentation#maps-documentation>
- [3]. Redis, "Redis for Real-Time Geospatial Data," Redis.io, 2023. [En línea]. Disponible en: <https://redis.io/docs/latest/develop/data-types/geospatial/>
- [4]. Stripe, "Payment Processing for Marketplaces," Stripe Docs, 2024. [En línea]. Disponible en: <https://docs.stripe.com/connect>