# 1. Descripción General

Esta propuesta presenta una arquitectura de datos mínima para una aplicación de delivery que contemple escalabilidad, accesibilidad y gobierno de datos. Se cubren cinco capas esenciales: fuentes, almacenamiento, procesamiento, acceso y seguridad.

# 2. Fuentes de Datos

- Usuarios: información personal, preferencias, ubicaciones, historial de pedidos.  
- Pedidos: productos solicitados, estados del pedido, tiempos de entrega, pagos.  
- Restaurantes: menús, horarios, calificaciones, ubicación.  
- App móvil y web: interacciones en tiempo real.

# 3. Almacenamiento

- Base de datos SQL (PostgreSQL): almacenamiento estructurado para usuarios, pedidos y datos transaccionales.  
- Base de datos NoSQL (MongoDB): para menús dinámicos, configuraciones por restaurante, geolocalización en tiempo real.  
- Data Lake (Amazon S3): almacenamiento de grandes volúmenes de logs, historiales de pedidos, análisis de comportamiento.

# 4. Procesamiento

- Procesamiento Batch: ETL nocturno con Apache Airflow para reportes de negocio.  
- Procesamiento en Tiempo Real: Apache Kafka y Spark Streaming para eventos de pedido en curso, alertas y monitoreo.  
- Limpieza y Transformación: validación de datos con Apache Beam.

# 5. Acceso y Visualización

- APIs RESTful: acceso a datos en tiempo real desde app móvil/web.  
- Dashboards (Power BI): visualización de KPIs para el negocio.  
- Panel administrativo: gestión de restaurantes, pedidos y usuarios.

# 6. Seguridad y Gobernanza

**Seguridad**: Se debe implementar **cifrado de datos** en tránsito (TLS/SSL) y en reposo. El **control de acceso** debe ser estricto, con políticas de "mínimo privilegio", y se debe usar **anonimización** y **seudonimización** para datos sensibles como la información de los usuarios.

**Gobernanza**: Se deben definir políticas claras para el **manejo de datos**, la **calidad de los datos** y su **ciclo de vida**. La gobernanza asegura que los datos sean confiables y se usen de manera responsable. Es necesario establecer un **catálogo de datos** para que los analistas sepan qué datos están disponibles y cómo acceder a ellos.

**Escalabilidad**: El diseño de la arquitectura debe ser **escalable horizontalmente**. Esto significa que se pueden agregar más servidores o recursos para manejar el aumento de usuarios y datos sin comprometer el rendimiento. Esto es particularmente importante en bases de datos y sistemas de procesamiento en tiempo real.

# 7. Diagrama

Seguridad y Gobernanza

Acceso y Visualización

Procesamiento

Almacenamiento

Fuentes de datos