INGESTAS

Ingestas

Las ingestas y el procesamiento van de la mano.
Este es el primer paso del **BIG DATA**.

Ingestas

Las ingestas, como ya hemos comentado, pueden venir de diferentes fuentes: *Comercio electrónico, Aplicaciones, Sensores, Hogar conectado, Geolocalización, Redes Sociales, Dispositivos portátiles, Sistemas de vigilancia, Ventas, Transacciones y pagos, Industria y Coche conectado.*

Ingestas

Se pueden emplear diferentes estrategias atendiendo a la velocidad y al volumen requerido, puediendo ser las siguientes:

- Procesamiento por **Lotes** (Batch Processing).
- Procesamiento en **Tiempo Real** (Real Time Processing).
- Procesamiento **Casi en Tiempo Real** (Near Real Time Precessing).

Procesamiento por Lotes

Consiste en recopilar y procesar grandes conjuntos de datos en intervalos específicos.
Los datos se acumulan durante un periodo y se procesan juntos como un "*lote*".

Procesamiento por Lotes

Ventajas	Desventajas
• Eficiencia en el procesamiento de grandes volúmenes de datos.	• Latencia alta: No apto para necesidades inmediatas.
Costos más bajos al poder programar tareas en horarios no pico.	Falta de actualidad: Los datos procesados pueden no reflejar el estado más reciente.
• Simplicidad en la implantación y mantenimiento.	

Procesamiento por Lotes

Herramientas para el procesamiento por lotes

Se programa en Python y las puedes utilizar en las principales plataformas (AWS tiene un servicio propio que gestiona Airflow).
Apache Airflow es de código abierto.

No es Big Data propiamente dicho, es un CRON con esteroides.

Procesamiento en Tiempo Real

Tipo	Tiempo	Críticidad	Ejemplo
Hard Real-Time	Inmediato, estrictamente dentro del límite.	Fallo crítico si no se cumple.	**Control de vuelo, sistemas médicos.**
Soft Real-Time	Rápido, pero puede tolerar algunos retrasos.	No catastrófico, afecta el rendimiento.	**Streaming de video, videojuegos online.**
Near Real-Time	Segundos o minutos de retraso.	No crítico, mantiene actualización rápida.	**Monitoreo de redes, paneles de control.**

Procesamiento en Tiempo Real

El Big Data, puede estar en lo que se denomina Streaming, entre el “Soft” y el “Near” Real Time. El “Soft” es rápido pero puede tolerar algún retraso, por ejemplo, reproduciendo vídeo, en videojuegos, en una llamada de teléfono, esta empieza a empeorar cuando el retraso supera los 500ms

Procesamiento en Tiempo Real

Ventajas	Desventajas
• Procesa los datos inmediatamente (o casi) con un pequeño retardo hasta su presentación.	• **Complejidad técnica**: Requiere infraestructura robusta y escalable.
	Costos más altos: Mayor consumo de recursos y necesidades de mantenimiento.

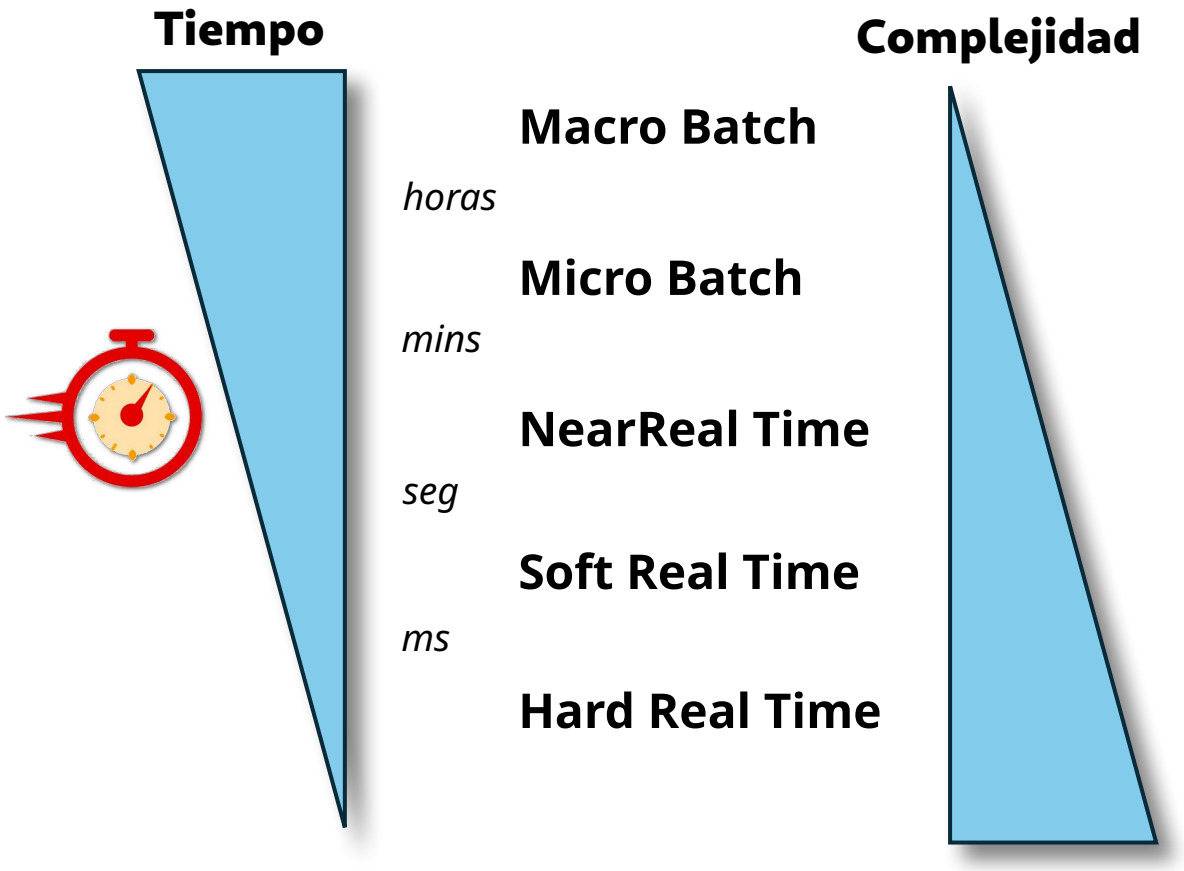
Procesamiento en Tiempo Real

Otra de las desventajas es que hay que montar otra arquitectura, entre las más famosas están la arquitectura Lambda y Kappa.
Mientras que el procesamiento por lotes es más sencillo y económico.

Procesamiento en Tiempo Real

Si tu la forma de trabajar de la organización en mediante “microbatchs”, o sea, procesos que se lanzan cada 15 minutos. Montar una infraestructura de Real Time resultaría muy caro y complejo.

Latencia vs Complejidad



Patrones de Captura

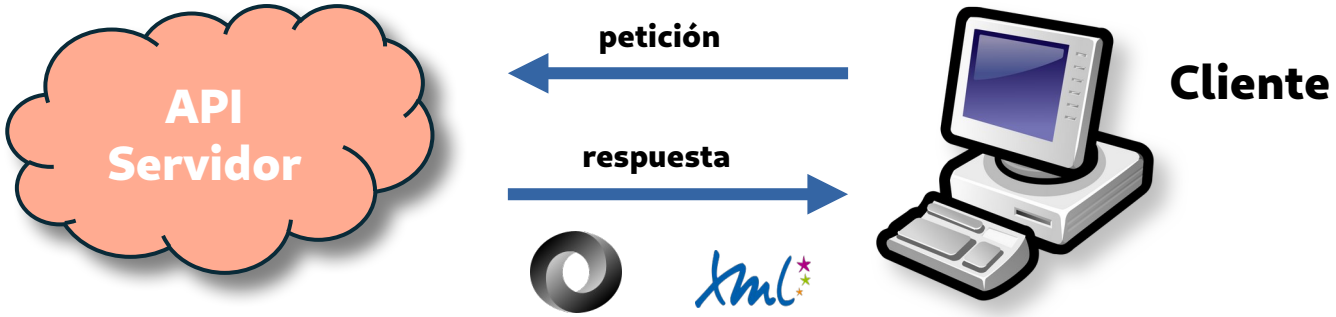
De donde capturamos la información

Patrones de Captura

Petición-Respuesta:
Este es el patrón más usuado.
¿Cómo funciona?
Se realiza la conexión a una base de datos y obtienes los resultados en formato JSON o en XML, esto resulta ser lo más sencillo, ¿pero para grandes cantidades de datos…?
Necesitamos algo mejor.

Patrones de Captura

Petición-Respuesta:
Comunicación síncrona donde un cliente envía una solicitud y espera una respuesta del servidor.



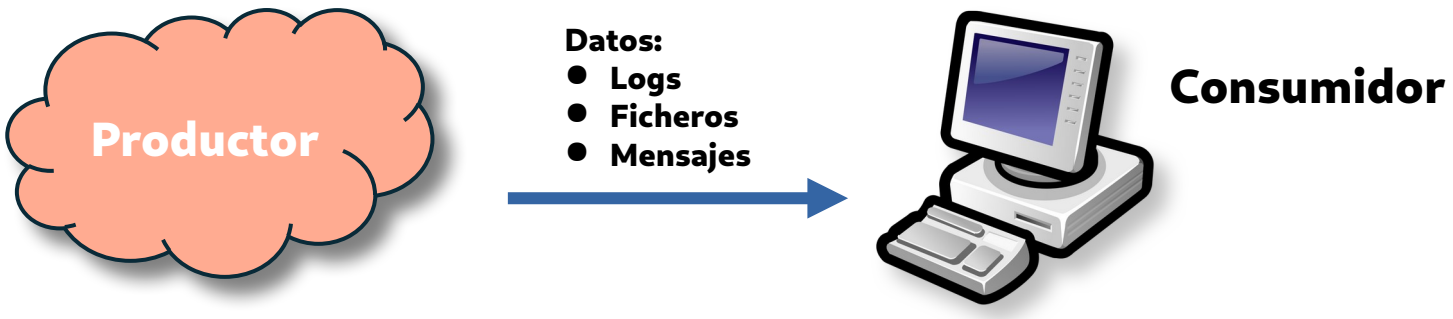
Patrones de Captura

Petición-Respuesta:

Ventajas	Desventajas
• Simplicidad en la implementación y comprensión.	• No escalable para grandes volúmenes de datos.
• Facilidad para manejar errores y excepciones.	• Puede generar latencia y cuellos de botella en sistemas con alta concurrencia.

Patrones de Captura

One way:
Comunicación asíncrona donde un cliente envía un mensaje sin esperar una respuesta.



Patrones de Captura

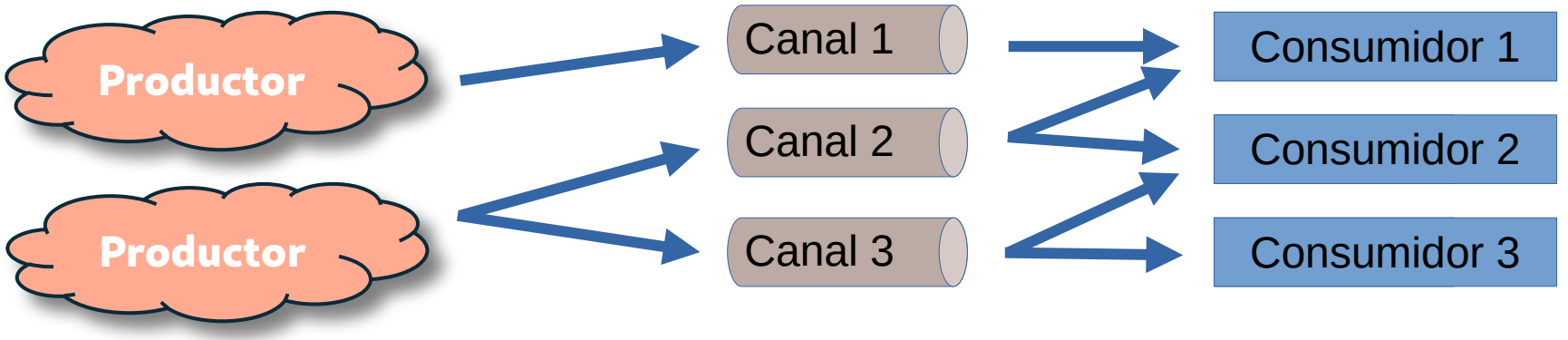
One way:

Ventajas	Desventajas
• Mayor rendimiento y escalabilidad.	• Dificultades para garantizar la entrega y el procesamiento.
• Desacoplamiento entre productor y consumidor.	• Menor control sobre el flujo de datos y manejo de errores.

Patrones de Captura

Publicador-Suscriptor:

Los productores publican mensajes en canales o temas, y los consumidores se suscriben para recibirlos.



Patrones de Captura

Publicador-Suscriptor:

- Basado en sistemas de mensajería como **Apache Kafka**.
- Soporta procesamiento tanto **batch** como en **tiempo real**.

Patrones de Captura

Publicador-Suscriptor:

Las colas de mensajes esto ya es Real Time.
Cada generador está generando datos (eventos).
La herramienta que se utiliza es Apache Kafka.
Es el servicio que se utiliza para streaming, no solo para Big Data, también para comunicación entre procesos.

Patrones de Captura

Publicador-Suscriptor:

Apache Kafka es más complejo de configurar y mantener, sobre todo por la consistencia que se debe mantener entre todos los canales.
Desde el punto de vista del productor es más sencillo, este manda un mensaje al canal, y es el consumidor quien procesa estos mensajes.

Patrones de Captura

Publicador-suscriptor:

Ventajas	Desventajas
• Alta escalabilidad y tolerancia a fallos.	• Complejidad en la configuración y mantenimiento.
• Desacoplamiento temporal entre productores y consumidores.	• Requiere gestionar la consistencia y el orden de los mensajes.

Patrones de Captura

Stream Processing:

Ha salido hace relativamente poco el procesamiento continuo de datos con Apache Flink como herramienta destacada (Real Time). Este recibe un conjunto de datos y los procesa en Tiempo Real