

Spring Batch es un **framework de procesamiento por lotes** que te ayuda a manejar grandes volúmenes de datos de forma eficiente y automatizada. Es perfecto para tareas repetitivas como **importar/exportar datos, migraciones, generación de reportes y más.**

Spring Batch se basa en **Jobs y Steps**:

- **Job:** Un proceso completo (Ejemplo: Procesar un archivo CSV y guardarlo en la base de datos).
- **Step:** Cada etapa dentro del Job (Ejemplo: Leer → Procesar → Escribir).

Cada Step puede dividirse en **chunks** (bloques de datos) para hacer el procesamiento más rápido y eficiente.

SB usa además los siguientes componentes clave:

1. **JobRepository:** Guarda el estado de los Jobs (para poder reiniciarlos si fallan).
2. **JobLauncher:** Inicia la ejecución de los Jobs.
3. **ItemReader:** Lee los datos desde una fuente (CSV, BD, JSON, XML, etc.).
4. **ItemProcessor:** Procesa los datos (Ejemplo: transformar nombres a mayúsculas).
5. **ItemWriter:** Guarda los datos procesados en un destino (BD, archivo, API, etc.).

Ventajas:

- Ligero y fácil de usar: tiene una API simple e intuitiva que facilita el desarrollo de aplicaciones por lotes. Además, es liviano y no necesita muchos recursos para funcionar.
- Diseño modular: esto permite usar solo los componentes o herramientas que se necesitan, en definitiva, personalizarlo.
- Gestión de transacciones: proporciona soporte integrado para la gestión de transacciones, asegurando la coherencia y la integridad de los datos.
- Programación y ejecución de Jobs: ofrece un marco de programación y ejecución que permite ejecutar Jobs por lotes de forma programada o manual.

Desventajas:

- Escalabilidad limitada: Spring Batch está diseñado para el procesamiento por lotes en una sola máquina y es posible que no se escale bien para grandes conjuntos de datos.
- Soporte limitado para procesamiento distribuido: no proporciona soporte integrado para procesamiento distribuido, lo que puede ser una limitación para algunos casos de uso.
- Soporte limitado para lenguajes que no son Java: está basado en el lenguaje de programación Java lo que significa que puede no ser adecuado para proyectos que requieran soporte para otros lenguajes de programación.

Hay una serie de principios a la hora de definir un proceso batch:

- **Simplificar** todo lo posible la **lógica**: de forma que quede fragmentada en procesos muy pequeños de lectura, procesamiento y escritura.
- Utilizar los **mínimos recursos posibles**: ya que se va a procesar un enorme volumen de datos.
- Revisar y **optimizar** sentencias **sql**: es esencial que las consultas estén optimizadas ya que se verá tanto en el redimiendo de la base de datos como en los tiempos de ejecución del job.
- Utilizar comprobaciones **checksum**: esto es muy útil a la hora de generar ficheros de gran tamaño, incluir en el pie del fichero un contador con el numero de registros e información del procesamiento realizado que puede ayudar a comprobar la integridad del fichero.
- Utilizar pruebas de **stress** con datos lo más realistas posibles.