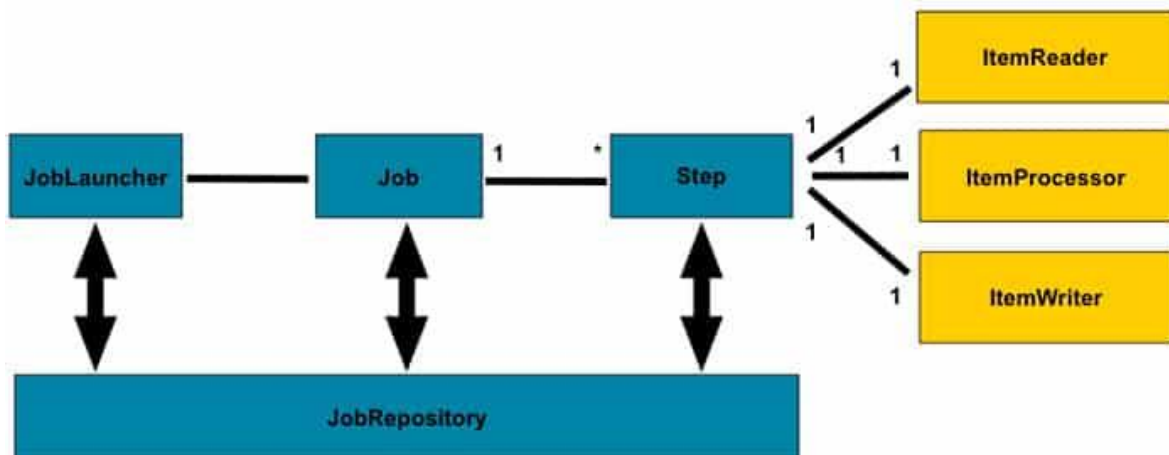


Spring Batch

Spring Batch es un framework de Spring para procesar grandes cantidades de datos de forma eficiente, robusta y escalable, en procesos por lotes. Se utiliza comúnmente para tareas como:

- Importar o exportar datos de bases de datos.
- Procesar archivos CSV, XML, JSON.
- Generar reportes, facturas, resúmenes.
- Automatizar tareas nocturnas o programadas.

Arquitectura de Spring Batch



JobLauncher

- Es el punto de entrada para ejecutar un Job.
- Se invoca normalmente desde un controlador o una clase `@Component` para lanzar el proceso Batch.
- Usa JobRepository para guardar y consultar ejecuciones anteriores.

Job

- Representa todo el proceso batch.
- Un Job está compuesto por uno o más Steps.
- Tiene lógica para manejar decisiones, flujos condicionales y reinicios.

Step

- Es una unidad de trabajo dentro de un Job
- Cada Step contiene:
 - *ItemReader*: Lee los datos.

- *ItemProcessor*: Transforma o valida los datos.
- *ItemWriter*: Escribe los datos en la salida (DB, archivo, etc).

ItemReader

- Lee datos de una fuente(base de datos, archivo, etc).

ItemProcessor

- Procesa, valida, transforma los datos.
- Opcional: Puedes dejarlo vacío si no necesitas procesar.

ItemWriter

- Escribe los datos ya procesados en la salida deseada.

JobRepository

- Guarda el estado de ejecución del Job: cuándo empezó, si falló, reinicios, etc.
- Permite reiniciar procesos fallidos desde el punto en que quedaron.

Resumen gráfico del flujo

JobLauncher → **Job** → **Step** → *ItemReader* → *ItemProcessor* → *ItemWriter*

↘ Estado en **JobRepository** ✓

Spring Batch te permite dividir un proceso en pasos organizados, leer datos, procesarlos y escribirlos, todo con control de fallos, reinicios y seguimiento. Es ideal para procesamiento masivo **estructurado** y **automatizado**.