

# Spring Batch

Los procesos batch (o procesos por lotes) son programas que se lanzan de manera programada y que no requieren de intervención humana.

## Spring Batch Processing example



Suelen ser procesos pesados, por lo que normalmente se ejecutan en horarios con baja carga de trabajo.

## Ejemplo:

Este tipo de procesos podrían ser los destinados a la generación o tratamiento de ficheros de facturación o la generación masiva de documentos (ej: cartas de bienvenida a nuevos clientes).



# ¿Qué es Spring Batch?

Es un **framework** ligero enfocado específicamente en la creación de procesos batch o procesos por lotes



Marcar la pauta para el diseño de procesos.

Proporciona componentes que intentan dar soporte a las necesidades que surgen a la hora de crear estos programas.

Trazas

Transaccion  
alidad

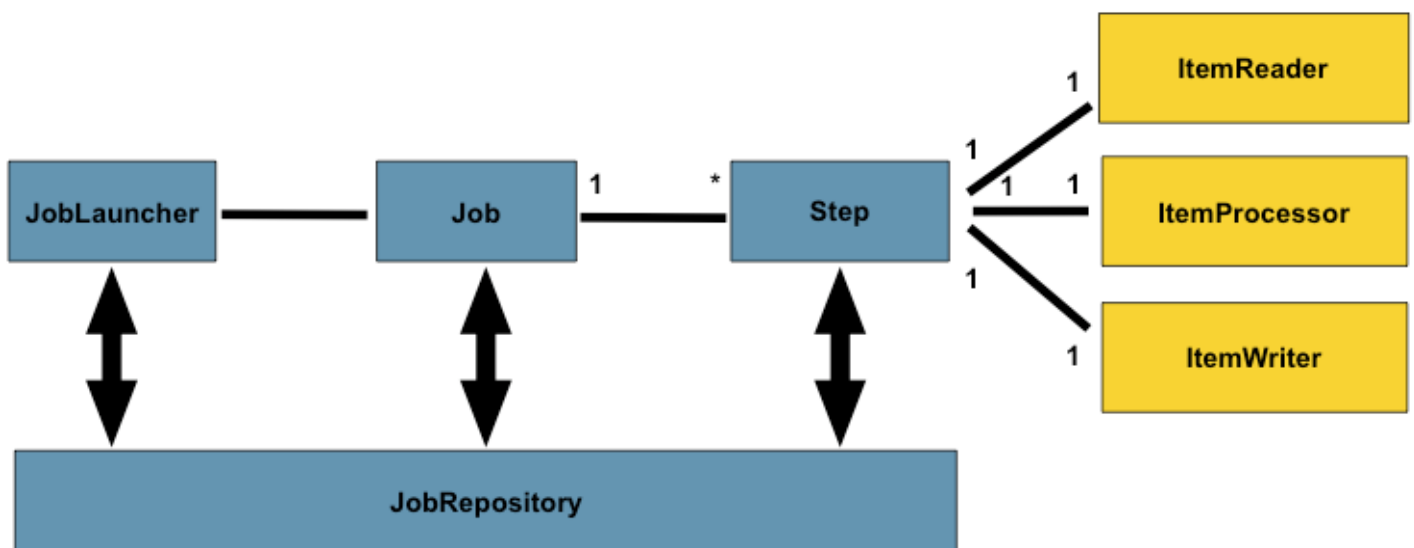
Estadísticas

Particiona  
miento

etc...

# Componentes principales

Spring Batch proporciona un diseño como el que se puede apreciar en la siguiente figura para construir los procesos.



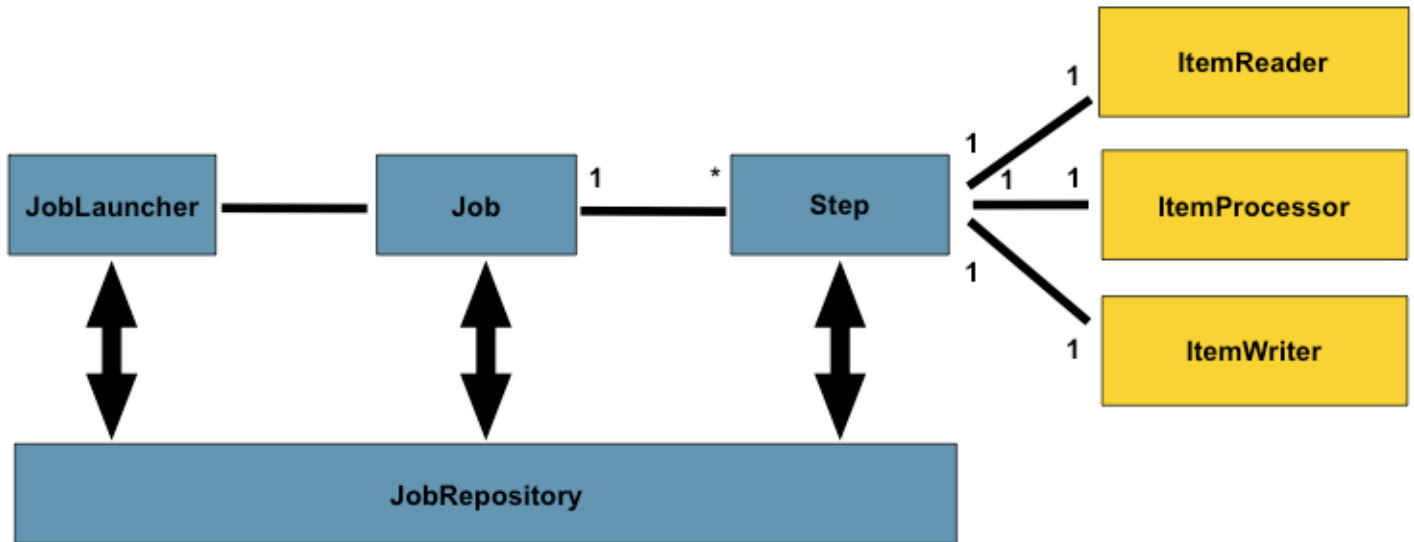
## JobRepository

Es el componente encargado de la persistencia de metadatos relativos a los procesos tales como procesos en curso o estados de las ejecuciones.

## JobLauncher

Es el componente encargado de lanzar los procesos suministrando los parámetros de entrada deseados.

# Componentes principales



## Job

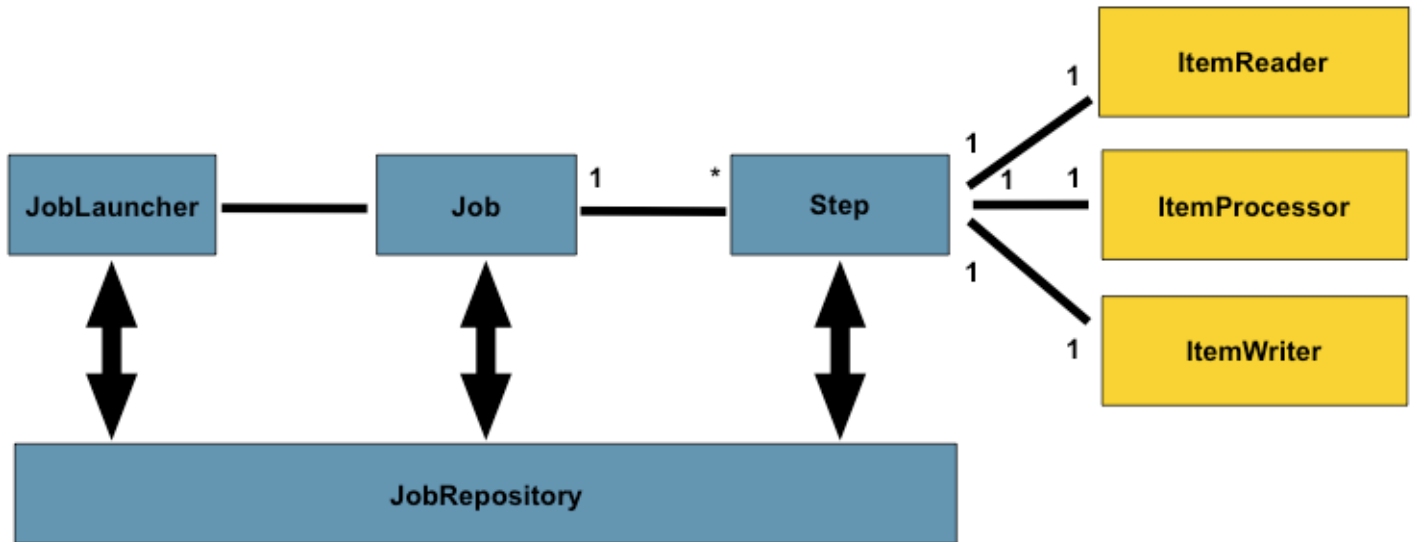
El Job es la representación del proceso. Un proceso, a su vez, es un contenedor de pasos (steps).

## Step

Un step (paso) es un elemento independiente dentro de un Job (un proceso) que representa una de las fases de las que está compuesto dicho proceso. Un proceso (Job) debe tener, al menos, un step.

Aunque no es obligatorio, un step puede estar compuesto de tres elementos: reader, writer y processor.

# Componentes principales



## ItemReader

Elemento responsable de leer datos de una fuente de datos (BBDD, fichero, cola de mensajes, etc...)

## ItemProcessor


Elemento responsable tratar la información obtenida por el reader. No es obligatorio su uso.

## ItemWriter


Elemento responsable guardar la información leída por el reader o tratada por el processor. Si hay un reader debe haber un writer.

# ¿Cómo funciona Spring Batch?

**Chunk-Oriented** es la técnica que utiliza Spring Batch para la ejecución de las fases de un proceso



El **reader** lee una porción de datos de la fuente de datos y los convierte en un «chunk» (entidad que representa esa porción de información leída).



Si existe un processor, ese chunk pasa al processor para que lo trate.

Todo esto se realiza dentro de un límite transaccional o, lo que es lo mismo, leemos y tratamos tantos chunks como queramos antes de que sean persistidos por el writer.

## Ejemplo:

Se tiene un fichero en texto plano, donde debemos tratar cada línea para luego persistir cierta información en una base de datos.



Se configura el step con un intervalo de commit igual a 10, lo que haría sería leer una línea del fichero, luego tratarla, leer otra línea del fichero, volver a tratarla, así hasta 10 veces.

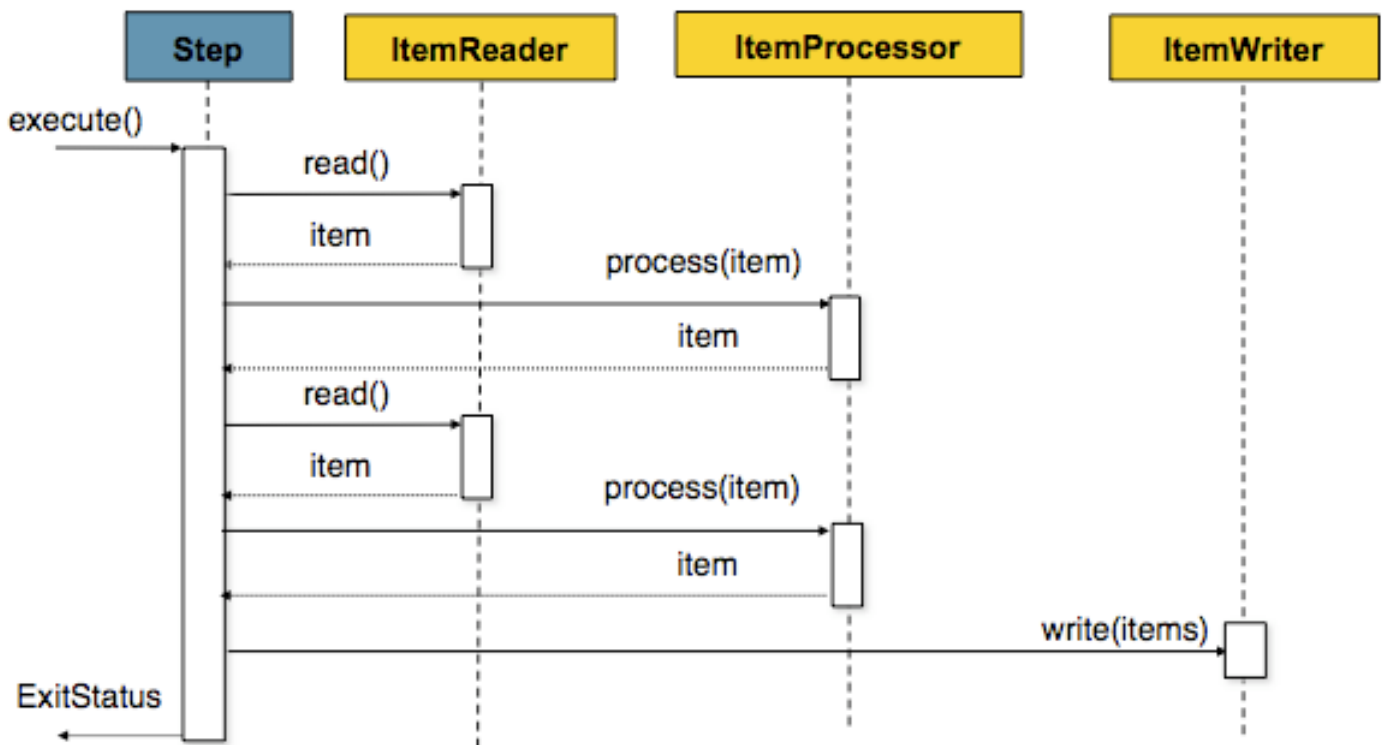


Una vez que ya hemos leído y tratado 10 líneas, el writer recibe esa información (los 10 chunks) y los persiste en base de datos.



Este proceso se repetiría hasta terminar con todas las líneas el fichero.

## Ejemplo:



Para ver un ejemplo práctico, visitar el siguiente enlace, de donde se ha extraído la información presentada.



<https://cutt.ly/w0ygVkv>