Jargon:

* Step: un “tasklet” es un tipo especial de “step” que se usa para ejecutar alguna función sin Reader ni Writer
* Job
* Item processor

“The business world runs on batch”, dicen.

Core:

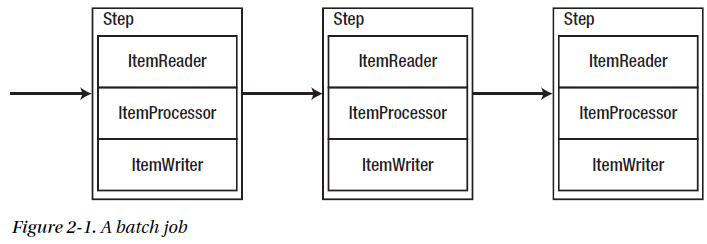
Interfaces:

* **org.springframework.batch.core.Job**: es un proceso que consiste en una serie de pasos (steps).
* **org.springframework.batch.core.Step**
* **org.springframework.batch.item.ItemReader<T>** (ver patrón Stragegy)
* **org.springframework.batch.item.ItemProcessor<T>**
* **org.springframework.batch.item.ItemWriter<T>** (ver patrón Strategy)
* **JobLauncher**
* **JobParameter**

Infrastructure:

**JOBS:**

Un “job” es una serie de pasos (“steps”) con un orden determinado que deben ser ejecutados como parte de un proceso predefinido:

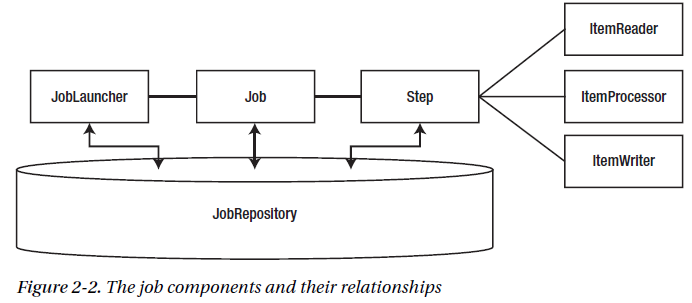


Como se puede ver, la unidad de trabajo autocontenida es el “step”. Son los ladrillos con los que se construye el job.

Los ItemXXX de cada paso se ejecutan sobre cada “record” (supongo que es la fuente de datos que hay que procesar, que se divide en registros). No es necesario que un step tenga esos tres items.

Al desacoplar los “steps” (cada uno tiene su propio procesador autocontenido), los “jobs” tienen una serie de características ventajosas: *flexibilidad*, *mantenibilidad*, *escalabilidad* y *fiabilidad*.

**Ejecución de Jobs:**



La ejecución de un “job” comienza en el JobLauncher