

Investigación de Spring

En todo proyecto de tecnología, sin importar la infraestructura, el lenguaje, entorno de desarrollo, existen numerosos recursos para crear y lograr un sistema más complejo o una aplicación funcional. Actualmente, existen un sin fin de herramientas que nos apoyan para mejorar la productividad, tales como las librerías y los frameworks.

¿Qué es una librería en Java?

En Java, una librería es conocida por ser una colección de clases, interfaces y métodos que los mismos desarrolladores pueden utilizar para realizar tareas específicas. Asimismo, pueden darnos mayor eficiencia, funcionalidad para poder usarlas en nuestros códigos o proyectos que deseamos trabajar.

Ejemplos de Librerías en Java.

Tenemos varias librerías en Java, cada una tiene un propósito que va de acuerdo a nuestros objetivos que queremos lograr a lo largo de nuestros códigos o proyectos que deseamos trabajar, esto aumentando nuestra productividad y eficiencia al momento de usarlas, tales como:

1. Apache Commons: Ofrece componentes Java reutilizables, esta dedicada a clases, colecciones, bases de datos y más.
2. Guava (Google): Desarrollado por Google, ofrece más posibilidades como nuevas colecciones, modelación de datos en grafos, mayor posibilidad de manejo concurrencias y más.
3. Log4j2: Permite entender el funcionamiento y ayuda a localizar la causa de errores cuando se producen en los loggins asíncronos, mejores filtrados y más.

Objetivo de la librería

Su objetivo o propósito en general, es el ejecutar una o más acciones concretas para reutilizar código que creamos, se trata como una interfaz o un contrato que define una serie de métodos, que facilita a nuestras aplicaciones o sistemas que trabajamos hoy en día.

¿Qué es un framework en Java?

Un framework, por su parte, es un marco de trabajo que proporciona una base sólida para el desarrollo de nuestras aplicaciones o proyectos en general. A diferencia de las librerías, estas establecen los flujos de trabajo o de control para nuestros programas y coexisten gracias a la estructura predefinida en la que los desarrolladores podemos empezar a distribuir nuestro código.

Ejemplos de frameworks

Al igual que las librerías, tenemos distintos frameworks en Java, cada una con un propósito en la cual nosotros decidiremos emplear a lo largo de nuestros códigos o proyectos que deseamos trabajar, esto aumentando nuestra productividad y eficiencia al momento de usarlas, tales como:

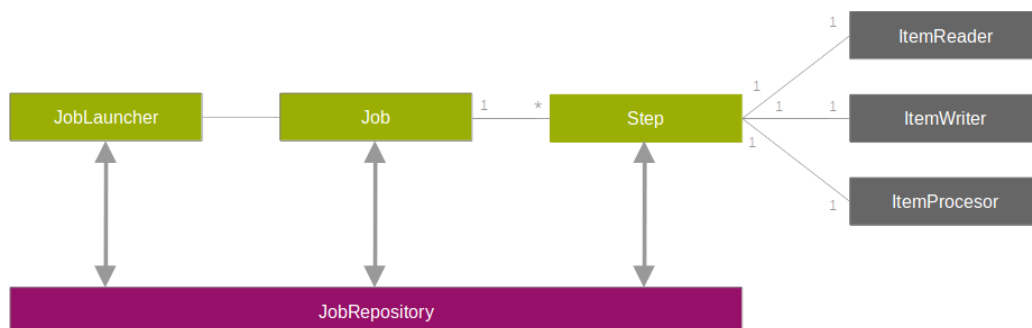
1. Spring: Destinado al desarrollo de aplicaciones empresariales, incluye una gestión de dependencias, soporta transacciones, acceso a datos, seguridad y mucho más.
2. Spring Boot: Derivado de Spring, esta mejora considerablemente el proceso de desarrollo de aplicaciones, adicionalmente da configuraciones automáticas, dependencias más fáciles de manejar, destinada a los microservicios y aplicaciones empresariales de manera más optimizadas y más.
3. Hibernate: Este dedicado al mapeo de ORM para el acceso a bases de datos.

¿Qué es Spring Batch?

A lo largo de lo que hemos visto de librerías y frameworks, debemos conocer este framework. Diseñado para el desarrollo de aplicaciones en la creación de procesos batch. Estos batchs, suelen ser procesos relativamente pesados, ya que son volúmenes de datos que se procesan, permitiendo la gestión y ejecución de trabajos de forma sencilla y robusta. Este framework surgió en colaboración de SpringSource (ahora Pivotal) y Accenture. No es un planificador de tareas, aunque se puede incluir un planificador en los procesos batch. Lo podemos ver importaciones/exportaciones de datos, generación de informes, migración de datos y mucho más.

Componentes y/o elementos de Batch

Gracias a Spring Batch nos propone un diseño como el que se mostrara a continuación para poder construir nuestros procesos y los conceptos importantes de cada uno.



1. JobRepository: Es nuestro componente encargado que guarda el comportamiento de más elementos. Es como un repositorio en donde se usa la escritura, la consulta, preprocesar o estados de ejecución.
2. JobLauncher: Es encargado de lanzar los procesos suministrando los parámetros de entrada deseados y así iniciar otros componentes
3. Job: Es el componente que actúa como la representación de un proceso, así mismo este se compone de un contenedor de pasos (steps), este también tiene unas cosas internas
 - a. JobInstance: Representación lógica de un job especificado con ciertos parámetros de ejecución
 - b. JobParameters: Conjunto de valores para comenzar la ejecución de un Job, que se pasan a una instancia del trabajo, sirve para identificar una ejecución del mismo.
 - c. JobExecution: Aquí la información referente de a cada ejecución del Job, sirve para ver el estado de ejecución y también nos sirve para ver la duración del Job.
4. Step: Es el elemento que hace relación con Job, representa una de las fases de las que está compuesto dicho proceso. Cada uno representa una tarea independiente que puede ser configurada y ejecutada de manera individual, también está conformado por tres partes.
 - a. ItemReader: Se encarga de la lectura del procesamiento de lotes, es responsable de leer los datos de una fuente y proporcionar estos datos al procesador
 - b. ItemProcessor: Representa la lógica de negocio implementada para realizar el proceso de la información, se encarga de transformar los datos previamente leídos, además de procesar, incluye cambios en el formato puede incluir filtrado de datos o la lógica en sí.
 - c. ItemWriter: Este elemento es lo opuesto a ItemReader. Se encarga de la escritura de los ítems, generalmente no sabe qué información recibirá, sino únicamente del elemento que se encuentra procesando en cada instante

Diferencias entre Librería y Framework

Para ser concretos, cada uno responde a las necesidades que necesitamos resolver para poder concretar nuestros códigos o proyectos. Poniendo en términos más sencillos, una librería nos permite solucionar un problema en concreto, mientras que un framework nos brinda un set de herramientas para desarrollar sistemas o aplicaciones más avanzadas.