15ª Emisión

DIPLOMADO

Desarrollo de Sistemas con Tecnología Java

Módulo 7

Introducción de Aplicaciones Empresariales con Spring Framework.

Mtro. ISC Miguel Ángel Sánchez Hernández

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Dirección General de Cómputo y de Tecnologías de información y Comunicación
Dirección de Docencia en TIC

DGTIC

Educación Continua
1971 - 2021

Convenios

- Tolerancia de inicio de clases 15 minutos
- 20 minutos de receso
  - Viernes 18:30-18:50
  - Sábados 11:30-11:50

Evaluación

• Ejercicios	30%
• Prácticas	40%
• Problema	30%
	100%

Objetivo

Conocer el funcionamiento interno del Framework Spring, así como sus ventajas al ocuparlo para el desarrollo de aplicaciones empresariales.

Lo que veremos

- ¿Qué es el Framework Spring?
- Problema del acoplamiento fuerte
- Como hacer acoplamiento ligero
- Módulos de Spring
- Contenedor núcleo de Spring

¿Qué es Spring?

Spring es una Framework (Marco de Trabajo) de código abierto realizado por Rod Johnson, para resolver la complejidad del desarrollo de aplicaciones empresariales.

Cualquier aplicación en Java obtiene los siguientes beneficios:

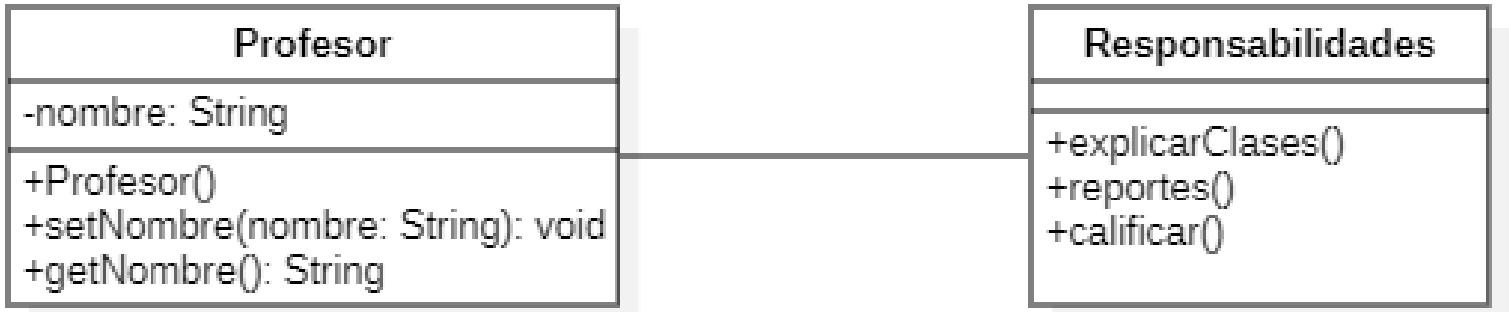
- Simplicidad en el desarrollo
- Compatibilidad
- Acoplamiento ligero

¿Cómo logra eso Spring?

Spring logra esto gracias a sus partes básicas que son:

- **Contenedor**: Podemos decir que Spring es un contenedor, porque contiene, configura, asocia y gestiona el ciclo de vida de los objetos en una aplicación.
- **Marco de Trabajo**: Sus bibliotecas ayudan a una funcionalidad de infraestructura, porque gestiona transacciones, persistencias e inyección de dependencias, dejando a nosotros dejando el desarrollo de la lógica de la aplicación.
- **Ligero**: Es ligero en tamaño y tiempo de procesamiento, el corazón de Spring no pesa más de 4.5 Mb.
- **Inyección de Dependencia**: Fomenta el acoplamiento ligero, utilizando la Inyección de Dependencia (DI).
- **Orientado a Aspectos**: Tiene un amplio soporte a la Programación Orientada a Aspectos (AOP).

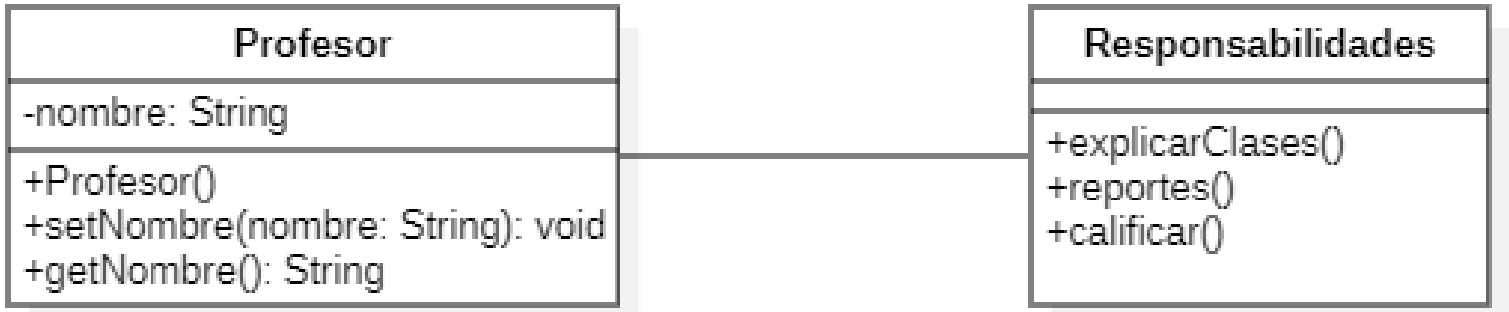
¿Acoplamiento fuerte?



1.- El objeto Responsabilidades se pasa al objeto Profesor como un parámetro.

Ejercicio 1: Método uno de acoplamiento fuerte

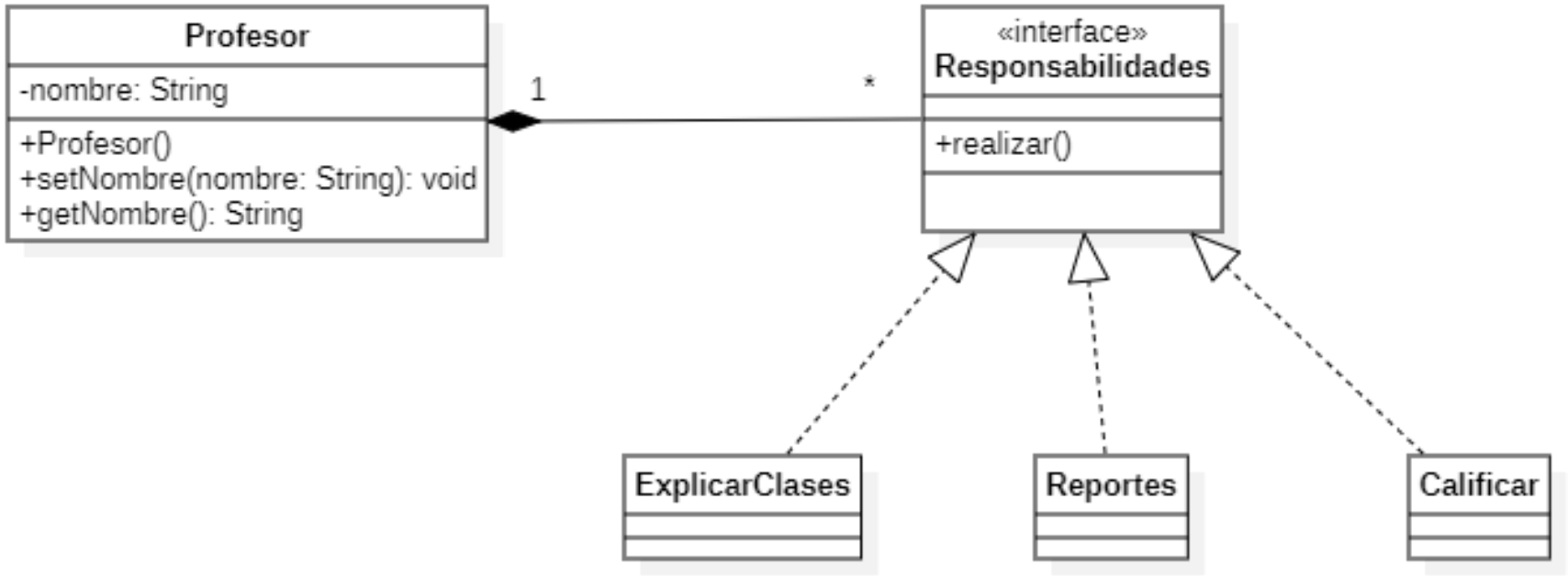
¿Acoplamiento fuerte?



2.- El objeto Profesor crea el objeto Responsabilidades

Ejercicio 1: Método dos de acoplamiento fuerte

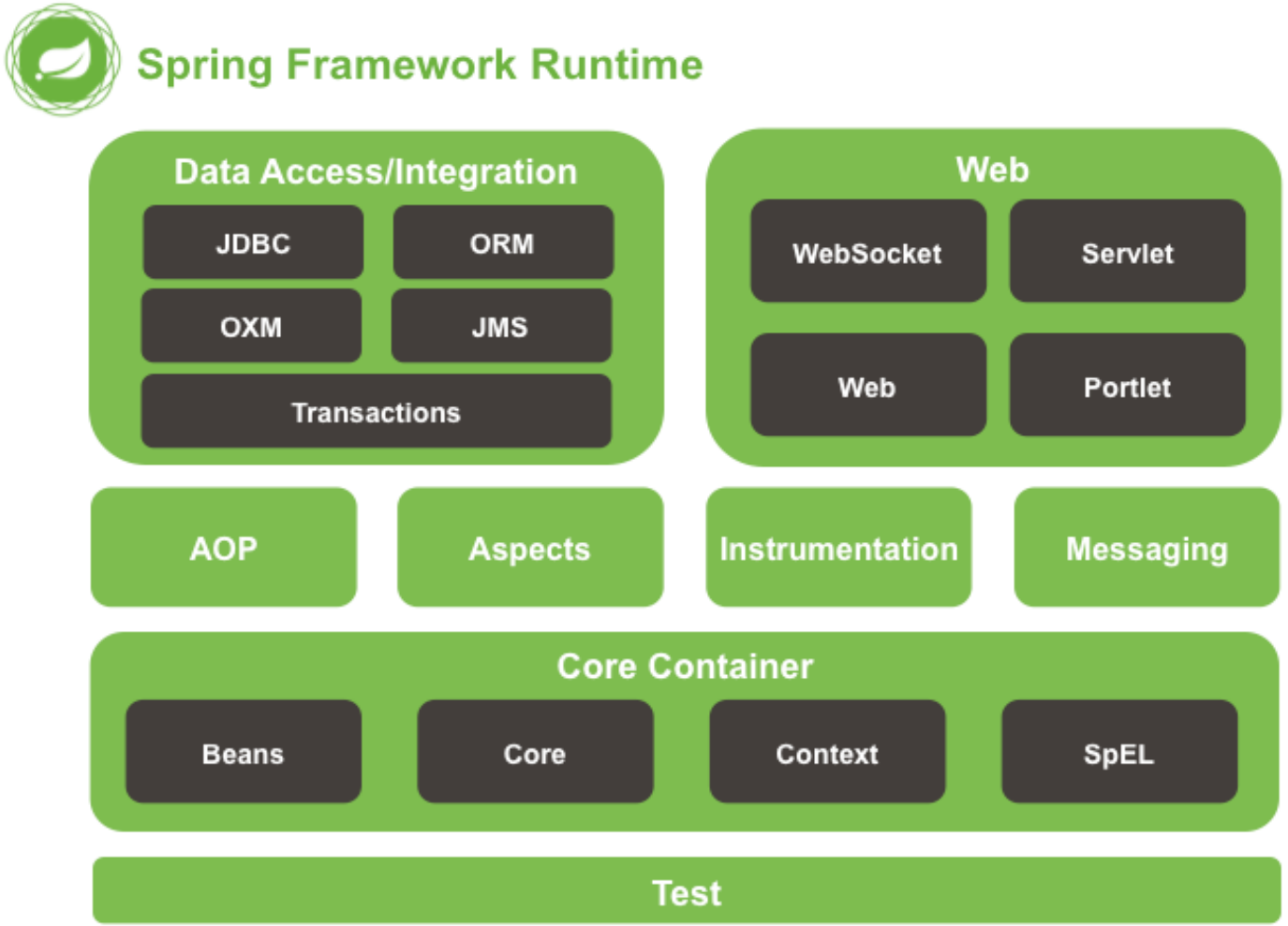
¿Acoplamiento ligero?



1.- Ocupar Interfaces

Ejercicio 2: Método acoplamiento ligero con interfaces

Módulos de Spring



Modulo Core Container

Es el contenedor núcleo de Spring, proporciona la funcionalidad fundamental al Framework de Spring.

- **spring-core,spring-beans:** Incluye la inyección de dependencia, Inversion of Control (IoC), implementación del Patrón de Fabrica, logrando desacoplar la configuración y dependencias de la lógica de negocio de la aplicación.
- **spring-context,spring-context-support:** Garantiza acceder a los objetos en forma similar a un JNDI, soporta internacionalización, carga de recursos, programación de eventos, comunicación remota básica e integración de bibliotecas de terceros.
- **spring-expression:** Da la posibilidad de ocupar un lenguaje de expresiones para manipular y consultar un gráfico de objetos en tiempo de ejecución, se puede decir que es una extensión del lenguaje de extensión unificado (JSP 2.1).

Modulos AOP,Aspects,Instrumentation y Messaging

- **spring-oap:** Implementa la programación orientada a aspectos, definiendo interceptores de métodos con puntos de corte.
- **spring-aspects:** Facilita una integración con AspectJ.
- **spring-instrument:** Da soporte para la instrumentación de clases e implementación de cargadores de clases para usar servidores de aplicación como Tomcat (**spring-instrument-tomcat**).
- **spring-messaging**: Para servir como aplicaciones fundamentadas en mensajería.

Modulos Data Access/Integration

- **spring-jdbc:** Elimina la necesidad de hacer cansadas codificaciones JDBC y un análisis de errores según el proveedor de base de datos.
- **spring-tx:** Admite la gestión de transacciones.
- **spring-orm:** Posibilita la integración de las API que hagan el mapeo relación de objetos como JPA y Hibernate.
- **spring-oxm:** Implementación del mapeo Objecto a XML como JAXB y XStream.
- **spring-jms:** Permite producir y consumir mensajes.

Modulo WEB

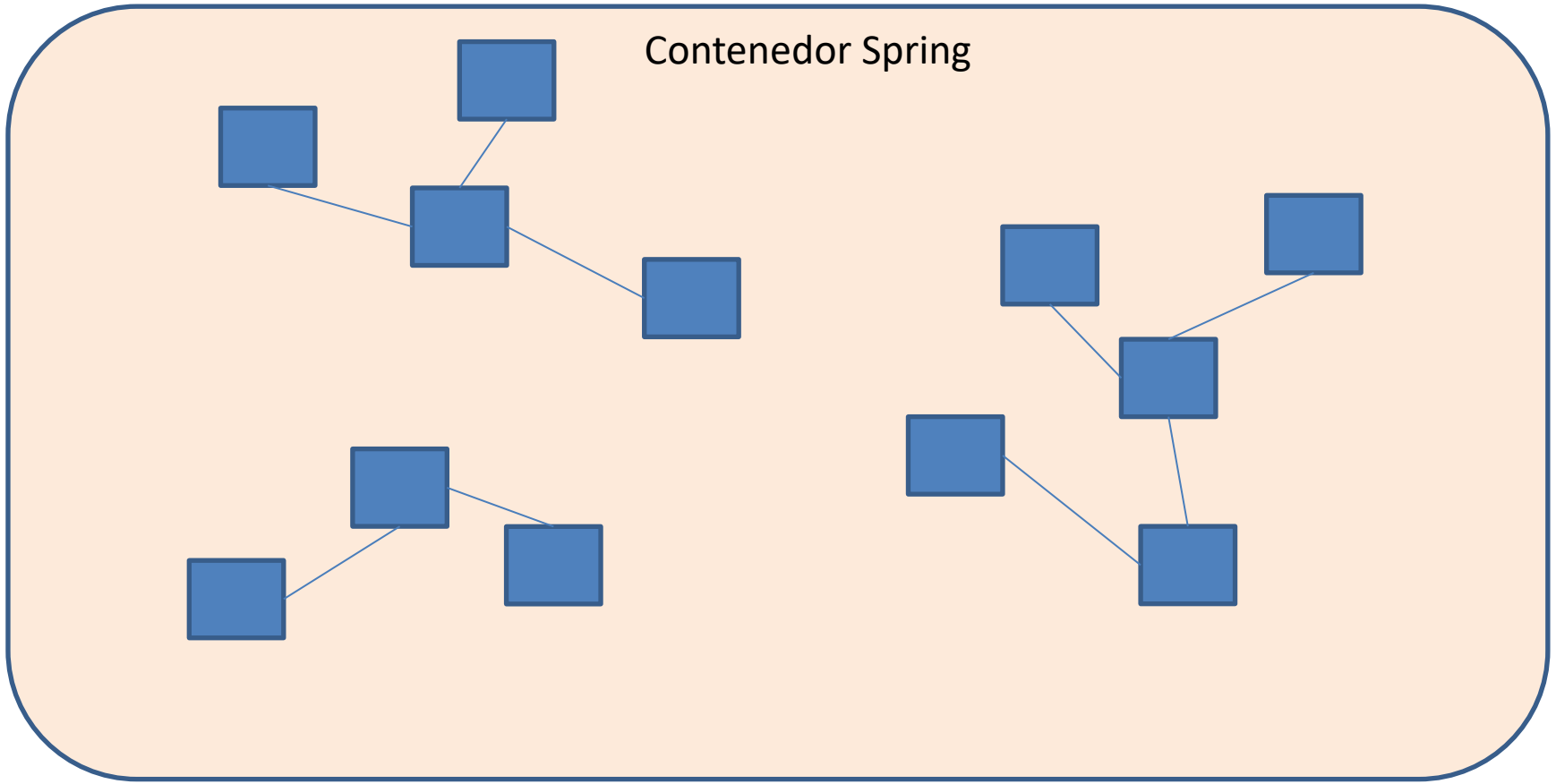
- **spring-web:** Carga de archivos, iniciar el contenedor IoC para que escuche por Servlet y tener el contexto de aplicación orientado a la WEB.
- **spring-webmvc:** Contiene el Modelo-Vista-Controlador(MVC), implementación de los servicios web REST para aplicaciones web, este como también se conoce como Web-Servlet.
- **spring-websocket:** Sirve para ocupar web-socket en una aplicación web.

Modulo Test

- **spring-text:** Garantiza poder hacer pruebas unitarias y pruebas de integración de componentes Spring con JUnit y Mockito.

Contenedor núcleo de Spring

En una aplicación Spring, los objetos de la aplicación vivirán dentro del contenedor de Spring.



Inversión of Control (IoC)

Se conoce también como Dependency Injection (DI). Este es el proceso mediante los objetos se definen sus dependencias.

Al decir dependencias, es la relación que necesita un objeto con los demás objetos con los que trabajara. La relación se realizará por medio de los argumentos del constructor, argumentos de un método de fabrica o propiedades que contenga la instancia del objeto.

Los objetos que estarán en el contenedor Spring IoC y formarán la columna vertebral de la aplicación los llamaremos **beans**.

Cuando creamos un bean el IoC inyecta sus dependencias, logrando así que el contendedor controle la asociación entre los beans colaboradores.

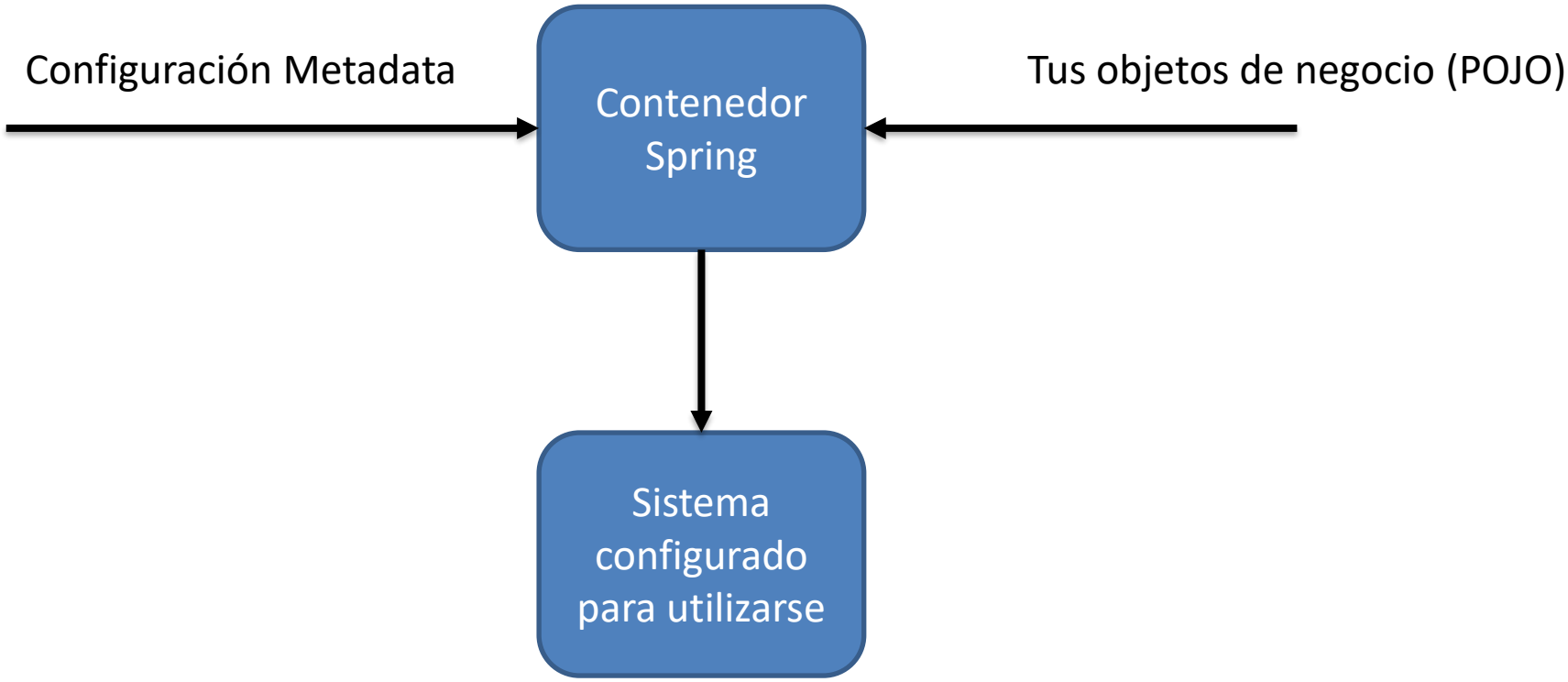
BeanFactory

La interfaz BeanFactory es uno de los contenedores de Spring, proporciona la configuración y la funcionalidad básica. Implementa la Patrón Factory, este consiste en la instanciación y entrega de los beans de la aplicación.

Ya que se conocen muchos beans en la fábrica, crea asociaciones entre ellos, eliminando la configuración desde el bean. Este bean participa en el ciclo de vida (se ve más adelante) de un bean como el método de inicialización y destrucción del mismo.

Configurar Beans para el contenedor Spring IoC

Las clases de la aplicación se combinan con metadatos de configuración para que después los podamos utilizar



Metadatos de configuración

Los metadatos de configuración nos sirven para indicarle a Spring como configurar y ensamblar los beans para crear las instancias en la aplicación. Existen tres maneras de crear estos metadatos.

- Configuración con XML
- Configuración basada en Java (Anotaciones)
- Descubrimiento implícito de beans y Autoconexión.

Ejercicio 3: BeanFactory

ApplicationContext

BeanFactory está bien para aplicaciones sencillas, pero para aplicaciones más complicadas tenemos ApplicationContext, hace la misma función que BeanFactory, pero agrega lo siguiente:

- El contexto de aplicación proporciona resolver mensajes de texto con internacionalización (I18N).
- Forma genérica para abrir recursos como archivos o imágenes.
- Aplicación de eventos a beans que se registren como interceptores.

Por este motivo ocuparemos siempre ApplicationContext.

Ejercicio 4: ApplicationContext