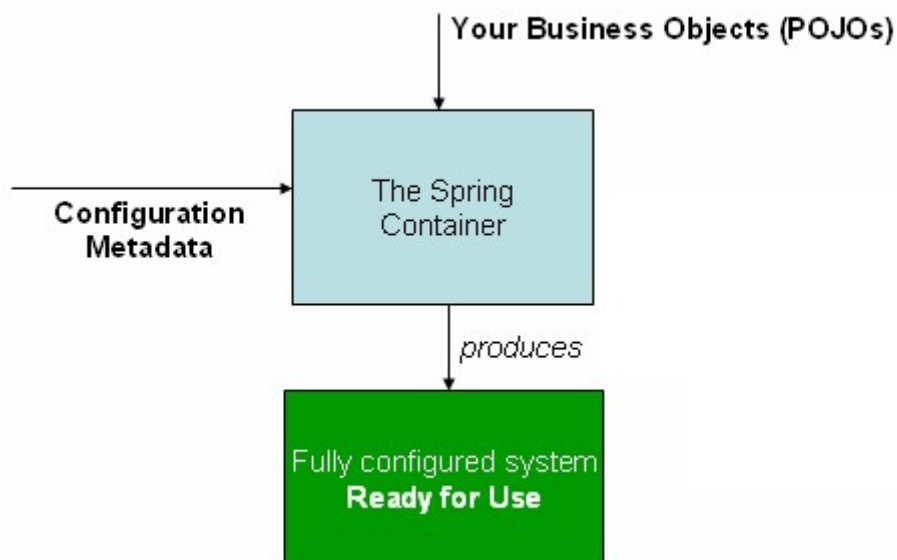


## Spring conceptos:

**Inyección de dependencias:** Es un patrón de desarrollo de software donde los objetos no son responsables de inicializar sus dependencias, sino que estas son provistas a través de otro objeto. En el caso de Spring ese objeto es el contenedor IoC el cual es provisto por módulos Spring-core y Spring-beans.

**Contenedor de Spring:** El contenedor (Inversion of Control principle -> IoC Container) también conocido como inyección de dependencia.



Spring provee tres tipos de inyección de dependencias: inyección por constructor, por setter y por método. Normalmente sólo utilizamos las dos primeras.

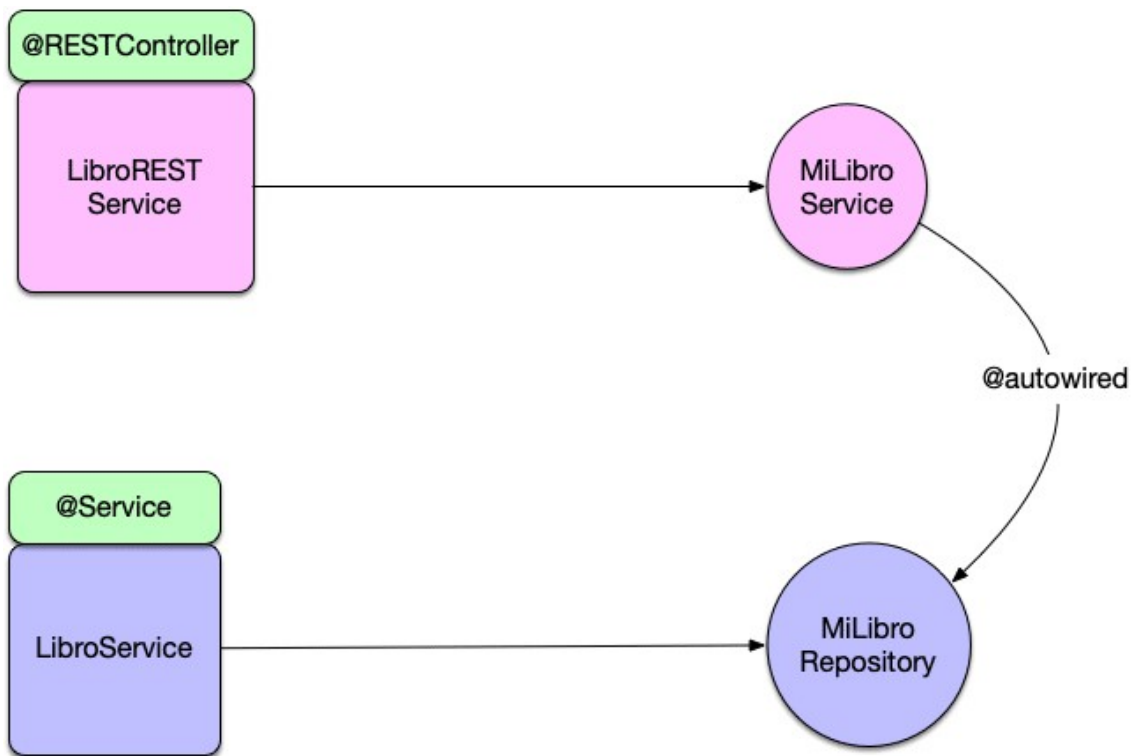
**Configuración de Spring:** Existen diferentes formas de configurar una aplicación de Spring, ya sea a través de XML, anotaciones o código Java. Estas 3 formas no son excluyentes.

**Spring Boot:** Es un proyecto que simplifica la configuración de aplicaciones Spring y facilita la creación de aplicaciones Java autónomas.

**El módulo Spring MVC:** El módulo Spring MVC implementa un patrón de diseño denominado Modelo-Vista-Controlador. Donde el modelo componente central del patrón. Es la estructura de datos dinámica de la aplicación, independiente de la interfaz de usuario. La vista es cualquier representación de información. Son posibles múltiples vistas de la misma información. El controlador acepta entradas y las convierte en comandos para el modelo o vista.

**String Data:** Esto facilita el acceso a datos con base de datos relacionales o no relacionales, este es un proyecto paraguas que tiene varios subproyectos específicos que son especiales para el manejo de datos.

**Anotaciones de Spring:** Como @Autowired, @Component, @Controller y @Service que simplifican la configuración de componentes y la inyección de dependencias.



**Spring Data JPA:** Simplifica el acceso a bases de datos relacionales mediante el uso de interfaces y métodos derivados.

**Testing en Spring:** Las pruebas unitarias e integración en aplicaciones Spring utilizando herramientas como Junit y Mockito.

Métodos ejemplo CRUD:

**jdbcTemplate.update** es un método proporcionado por la clase `JdbcTemplate` de Spring Framework, diseñado específicamente para ejecutar declaraciones de actualización de SQL como INSERT, UPDATE o DELETE, simplificando las operaciones de bases de datos en aplicaciones Spring.

**jdbcTemplate.query** es un método en la clase `JdbcTemplate` de Spring Framework, utilizado para ejecutar consultas SQL SELECT y recuperar datos de una base de datos de una manera estructurada y eficiente.

**jdbcTemplate.queryForObject** es un método dentro de la clase `JdbcTemplate` de Spring Framework, utilizado principalmente para ejecutar consultas SQL SELECT que se espera que devuelvan un único resultado, y simplifica el proceso de buscar y asignar ese resultado a un objeto Java.

**jdbcTemplate.batchUpdate** es un método proporcionado por la clase `JdbcTemplate` de Spring Framework, que permite la ejecución eficiente de múltiples declaraciones de actualización de SQL como una operación por lotes, mejorando el rendimiento al tratar con múltiples actualizaciones de bases de datos simultáneamente.

**SqlParameterSourceUtils.createBatch** es un método de utilidad en Spring Framework que se usa para crear un lote de objetos SqlParameterSource, que se usan comúnmente para actualizaciones e inserciones por lotes parametrizadas en operaciones de bases de datos.