

- **Principio de Inversión de Control (IoC):** IoC permite que el control sobre la creación y manejo de objetos sea transferido al contenedor de Spring. Esto significa que en lugar de que el código personalizado instancie directamente las dependencias, el contenedor de Spring las inyecta automáticamente. Esto promueve la modularidad y facilita la sustitución de implementaciones, lo cual es fundamental para el desarrollo ágil y mantenible de aplicaciones.
- **Inyección de Dependencias (DI):** DI es un patrón específico para implementar IoC en Spring. Permite conectar objetos entre sí sin que los objetos mismos gestionen sus dependencias directamente. En Spring, las dependencias pueden inyectarse a través de constructores, métodos setters o directamente en campos anotados con `@Autowired`. Esta flexibilidad ayuda a mejorar la legibilidad del código y facilita las pruebas unitarias al permitir la fácil sustitución de dependencias por mocks.
- **Contenedor IoC de Spring:** El contenedor IoC de Spring, representado por la interfaz `ApplicationContext`, es responsable de la creación y administración de los objetos conocidos como beans. Puede configurarse mediante metadatos en forma de anotaciones o archivos XML. Además, proporciona varios modos de inyección automática de dependencias (`@Autowired`, por tipo, por nombre, etc.) que simplifican el desarrollo al reducir la cantidad de código boilerplate necesario para configurar y conectar componentes de la aplicación.