


Bootcamp Java Developer

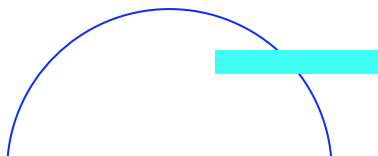
Fase 3 - Java Architect
Módulo 21



- **Aspectos:** un componente de la programación orientada a aspectos que encapsula el comportamiento que se puede aplicar a múltiples objetos en una aplicación. En Spring Framework, los aspectos se definen y se configuran a través de la configuración de aspectos.
 - **Beans:** un objeto gestionado por el contenedor de aplicaciones en Spring Framework. Los *beans* son objetos que se crean, configuran y gestionan a través del `ApplicationContext`.
 - **Concern:** en la Programación Orientada a Aspectos (POA), un *concern* refiere a una preocupación o aspecto transversal que afecta a múltiples partes de un programa, a menudo no relacionadas entre sí. Un *concern* puede ser visto como un tema o una función específica que atraviesa varias partes de un programa, y puede ser abordado mediante la definición de un aspecto específico.
 - **Configuración basada en anotaciones:** técnica de configuración de Spring que utiliza anotaciones para configurar objetos, en lugar de archivos XML.
- 

- **Contenedor de aplicaciones:** es el objeto que gestiona y mantiene el ciclo de vida de los objetos de una aplicación en Spring *framework*. Proporciona servicios como la inversión de control, la inyección de dependencias, y la configuración de la aplicación.
- **Controlador:** clase de Java que procesa las solicitudes HTTP y devuelve la respuesta adecuada.
- **Crosscutting:** término utilizado en la Programación Orientada a Aspectos (POA) para describir las preocupaciones que no son específicas de un solo módulo o función de una aplicación, sino que se cortan a través de varios módulos o funciones.
- **Hibernate:** *framework* de persistencia de objetos relacionales (ORM) que se integra con JPA para proporcionar una implementación más fácil y flexible del acceso a datos.

- **Inversión de control (IoC):** técnica de diseño en la que el control de la aplicación se invierte, lo que significa que el marco de trabajo es responsable de invocar los métodos de la aplicación en lugar de que la aplicación tenga que invocar los métodos del marco.
- **Inyección de dependencias:** patrón de diseño en el que los objetos dependen de otros objetos para su funcionamiento, y en lugar de crearlos directamente, se inyectan en el objeto a través de una configuración de contenedor.
- **JDBC (*Java Database Connectivity*):** es una API de Java que sirve para acceder a bases de datos relacionales. Spring *framework* proporciona soporte para JDBC a través del módulo Spring JDBC.
- **JPA (*Java Persistence API*):** Una API de Java que proporciona una interfaz para trabajar con bases de datos relacionales.
- **MVC (*Modelo Vista Controlador*):** es un patrón de diseño arquitectónico que separa la lógica de la aplicación en tres componentes: modelo, vista y controlador. Spring *framework* proporciona soporte para el patrón MVC a través del módulo Spring MVC.



- **POA (Programación Orientada a Aspectos):** una técnica de programación que permite la separación de preocupaciones (*concerns*) en el código. Spring *framework* proporciona soporte para la programación orientada a aspectos a través de la configuración de aspectos.
- **Spring Boot:** un subproyecto de Spring que facilita la creación de aplicaciones Java autónomas y listas para la producción.
- **Spring Cloud:** subproyecto de Spring que proporciona un marco de trabajo para la construcción de aplicaciones en la nube.
- **Spring Framework:** *framework* de desarrollo de aplicaciones de Java que proporciona una infraestructura integral para construir aplicaciones empresariales escalables y de alta calidad. Proporciona una variedad de características para la programación de aplicaciones Java, como un contenedor de inversión de control (IoC), un marco de inyección de dependencia (DI), un marco de programación de aspectos (AOP), un marco de transacciones, y un marco de acceso a datos.

- **Spring MVC:** módulo de Spring *framework* que proporciona soporte para la arquitectura MVC en el desarrollo de aplicaciones web. Proporciona una capa de controlador para manejar las solicitudes HTTP entrantes y generar respuestas, y utiliza una vista para renderizar la respuesta en el navegador del usuario.
- **Spring Security:** subproyecto de Spring que proporciona una capa de seguridad para aplicaciones web.



**Ahora sí,
¡Comencemos!**

