Desarrollo avanzado de aplicaciones II

Tema Nº1:Introducción al curso.

Indicador de logro Nº1:Reconoce conceptos básicos de Spring mediante la teoría y casos prácticos.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº1:**

Introducción al curso.

**Subtema 1.1:**

Ediciones de Java para el desarrollo de software

Java es uno de los lenguajes de programación más importantes de la industria de desarrollo de software. Odiado por muchos, pero amado por muchos más, este lenguaje lidera el mercado de desarrollo gracias a sus grandes productos estables, robustos y escalables. Las empresas usan este lenguaje como su principal backend, sin olvidar que la gran mayoría de desarrollos Android están hechos con Java. Y al contrario de lo que muchos piensan, Kotlin no es un lenguaje independiente, ya que este lenguaje requiere de la máquina virtual de Java para ejecutarse.

Para quien se acerca por primera vez al mundo de Java, le puede parecer muy confuso las diferentes ediciones de Java que existen (SE, EE, etc), así que en este artículo te explicaré cuales son las tecnologías (ediciones) que ofrece Java en su catálogo. Debo aclarar que hablaremos de las tecnologías, no de los frameworks o librerías.

**Java SE (Standard Edition)**

Esta edición es por donde se comienza en Java. Con ella se pueden crear aplicaciones de escritorio o de servidores. Sus librerías / bibliotecas permiten crear aplicaciones completas de escritorio. Puedes crear pantallas que se ejecutan en windows, linux y mac. También se puede conectar a las bases de datos, crear sockets o aplicaciones CLI para ejecutar procesos de forma automática.

**Java SE Embedded**

Con esta edición podemos crear código que pueda controlar dispositivos del Internet de las cosas (IoT). Muchos dispositivos vienen preparados para ejecutar código Java, por lo que podemos crear nuestros propios procesos.

**Java ME (Micro Edition)**

Aquí es donde puedes crear aplicaciones completas para dispositivos móviles (no confundir con aplicaciones Android), internet de las cosas, micro controladores, sensores, gateways, TV set-top boxes, impresoras y mucho más. Java ME incluye interfaces para protocolos de red, aplicaciones offline que se pueden descargar automáticamente. Estas aplicaciones pueden utilizar las capacidades de los dispositivos nativamente.

**Java EE (Enterprise Edition)**

Con esta edición podemos crear aplicaciones web con todas las de la ley. Creación de JSP (Java Server Pages), Servlets, JSF (Java Server Faces), Beans, WebServices (SOA y Rest), WebSockets, JSON-Processing y mucho más.

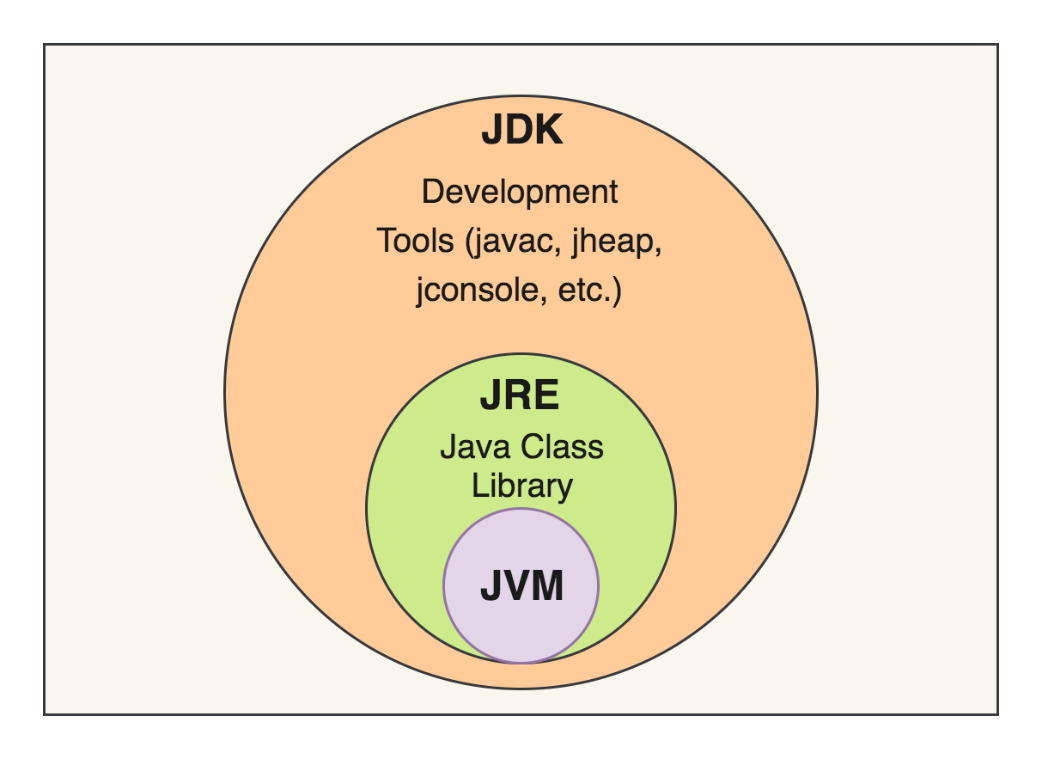
Para una correcta instalación de tu entorno de desarrollo, se debe tener en cuenta lo siguiente:

Recuerda que Java es un lenguaje que se compila a *bytecode* lo que significa que este código debe ser interpretado por la máquina virtual de java (JVM) para poder ejecutarse.

Por lo tanto, debemos saber que:

* Se necesita una máquina virtual (JVM) para poder ejecutar el código Java. No te confundas con las máquinas virtuales que creamos con herramientas como VirtualBox, sino que es la instalación de los paquetes de java necesarios para ejecutar los programas escritos en java. Por eso en los clientes (es decir, en las máquinas donde se va a ejecutar el código) se debe instalar el Java Runtime Environment (JRE).
* Para las máquinas donde se va a desarrollar, además del JRE se debe tener el kit de desarrollo de java, esto es el JDK.
* Lo más importante de todo, es que si estás haciendo una aplicación empresarial (JEE), en el servidor donde vas a alojar el programa, debes instalar el JRE y el JDK, ya que algunos procesos necesitan del JDK para poder ejecutarse.

En la siguiente imagen se muestra la relación entre JVM, JRE y JDK, asimismo en la última parte de la guía se indicará las pautas para la instalación.



**Subtema 1.2:**

Fundamentos Spring Framework.

Escrito inicialmente por Rod Johnson, fue lanzado por primera vez en el mes de Junio del año 2003 bajo la licencia Apache 2.0, siendo una plataforma Java de código abierto. Convirtiéndose desde entonces en el framework más popular para Java empresarial, para crear código de alto rendimiento, liviano y reutilizable. Ya que su finalidad es estandarizar, agilizar, manejar y resolver los problemas que puedan ir surgiendo en el trayecto de la programación.

Spring, ofrece como elemento clave el soporte de infraestructura a nivel de aplicación, brindando un completo modelo tanto para la configuración como para la programación de aplicaciones empresariales desarrolladas bajo Java, sin discriminación en cuanto al despliegue de la plataforma.

Todo esto trae consigo una gran ventaja, ya que permite que los equipos de desarrollo puedan enfocarse directamente en la lógica empresarial que requiere la aplicación, haciendo el proceso más corto, rápido y eficaz, ahorrando líneas de código evitando tareas repetitivas.

Spring se puede considerar como el padre del los frameworks Java, ya que da soporte a varios frameworks como: Hibernate, Struts, Tapestry, EJB, JSF, entre otros.

Entre las características de Spring, tenemos las siguientes que ofrecen una cantidad considerable de servicios:

* Tecnologías: como la inyección de dependencias, eventos, recursos, i18n, validación, enlace de datos, conversión de tipo, SpEL.
* Acceso a datos: soporte DAO, JDBC, ORM, Marshalling XML.
* Gestión de transacciones.
* Integración: comunicación remota, JMS, JCA, JMX, correo electrónico, tareas, programación, caché.
* Pruebas (Testing): simulacro de objetos, el framework TestContext, Spring MVC prueba, WebTestClient.
* Programación orientada a aspectos (AOP): permite la implementación de rutinas transversales.
* MVC (Modelo Vista Controlador).
* Seguridad.
* Frameworks web: Spring WebFlux y Spring MVC.
* Procesamiento de datos por lotes.
* Administración Remota: a través de este módulo se puede configurar la visibilidad y gestión de los objetos Java para la configuración local o remota vía JMX.
* Es un framework liviano debido a su implementación POJO (Plain Old Java Object), Spring Framework no obliga al programador a heredar ninguna clase ni a implementar ninguna interfaz.

A continuación listamos algunos de los beneficios de implementar Spring Framework:

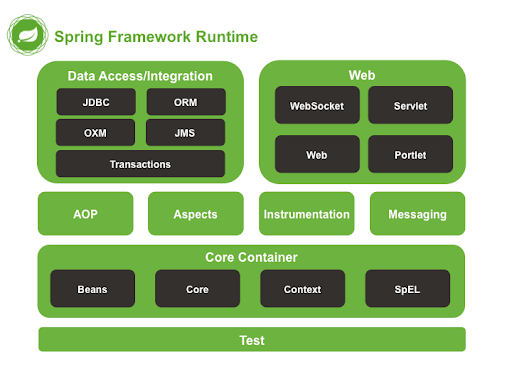
* Spring está organizado de forma modular. A pesar de la cantidad de paquetes y clases que tiene, solo debemos ocuparnos de aquellos que necesitemos para nuestro desarrollo e ignorar el resto.
* Utiliza algunas de las tecnologías existentes, como varios frameworks ORM, JEE, temporizadores Quartz y JDK, frameworks de registro y otras tecnologías de visualización.
* Probar una aplicación escrita con Spring es un proceso simple, porque el código dependiente del entorno se traslada a este framework. Además, mediante el uso de JavaBeanstyle, se vuelve más fácil utilizar la inyección de dependencia para hacer pruebas, para ello podemos hacer uso de datos dummies o mocks, para ver las respuestas.
* El framework web de Spring es un framework MVC web bien diseñado, que proporciona una excelente alternativa a los frameworks web como Struts u otros frameworks web sobre diseñados o menos populares.
* Spring proporciona una API para traducir excepciones específicas de la tecnología (como por ejemplo las generadas por JDBC, Hibernate o JDO) en excepciones consistentes y no verificadas.
* Los contenedores de IoC (Inversion of Control) tienden a ser livianos, especialmente cuando se comparan con los Enterprise JavaBeans (EJB). Esto es ideal para desarrollar y desplegar aplicaciones en maquinas con memoria y recursos limitados.
* Spring proporciona una interfaz de gestión de transacciones coherente que puede reducirse a una transacción local (utilizando una única base de datos) y ampliarse a transacciones globales.
* Permite separar el registro, la auditoría, las transacciones declarativas, la seguridad, el almacenamiento en caché, de la lógica comercial a través de la AOP (Programación Orientada a Aspectos).
* Cuenta con plantillas para diversas tecnologías entre la cuales podemos destacar las siguientes: JDBC, Hibernate y JPA, de forma tal que no hay necesidad de escribir un código extenso, ya que con estas plantillas simplifica el trabajo en cuanto a los pasos básicos a implementar de estas tecnologías.

**Subtema 1.3:**

Principales módulos de Spring Framework.

Spring, tiene alrededor de 20 módulos y, además, se pueden elegir libremente cuáles de los componentes utilizar para sus aplicaciones y cuáles no. En la arquitectura estándar, los diferentes módulos se dividen en seis categorías principales:

* **Core Container**: el contenedor núcleo proporciona los módulos básicos: spring-core, spring-beans, spring-context, spring-context-support y spring-expression. Con esto, Core y Beans forman la columna vertebral del framework, conteniendo la función de inyección de dependencia y soportando los objetos POJO, por ejemplo. El módulo Context hereda sus características del módulo Beans y puede complementarlas con funciones para la internalización o para la carga de recursos. Adicionalmente, es compatible con funciones de Java Enterprise como EJB y JMX (Java Management Extensions). Context-support facilita la integración de bibliotecas de otros proveedores en Spring. Spring-expression contiene el Spring Expression Language (SPeL), una extensión del Unified Expression Language de la especificación JSP-2.1 (Java Server Pages).
* **AOP e instrumentación**: para habilitar la programación orientada a aspectos, el Spring Framework contiene el módulo spring-aop, así como el módulo spring-aspects para la integración del anteriormente mencionado lenguaje AspectJ. El componente spring-instrument permite instrumentalizar clases Java, es decir, cambiar el tiempo de ejecución del código de bytes y añadir implementaciones de ClassLoader para diferentes servidores de aplicaciones.
* **Mensajería**: para servir como base a las aplicaciones basadas en mensajes, Spring Framework cuenta con algunas de las funciones de encriptación del proyecto Spring Integration como, por ejemplo, “Message”, “MessageChannel” o “MessageHandler”. Dicho módulo tiene el nombre de spring-messaging.
* **Data Access/Integration**: los módulos de esta categoría están diseñados para ofrecer a las aplicaciones Java los instrumentos necesarios para que interactúen con otras aplicaciones, así como para controlar el acceso a los datos. Por ejemplo, el módulo spring-jdbc pone a disposición una capa de abstracción que define la manera en la que un cliente accede a la base de datos y ahorra la complicada codificación típica de JDBC. Por su parte, spring-orm ofrece capas de integración para acceder a bases de datos relacionales en las populares interfaces ORM. Otros componentes son spring-tx (soporta la gestión programática de transacciones para todas las clases y POJO), spring-oxm (capa de abstracción para el mapeo de objetos o de XML) y spring-jms, un módulo con características para la producción y el procesamiento de mensajes.
* **Web**: en esta categoría se encuentran los módulos específicos para las aplicaciones web: spring-web, spring-webmvc y spring-websocket. Adicionalmente, esta aplicación de Java ha añadido las funciones típicas de integración orientada a la web como una función de carga para datos múltiples (multipart) o un cliente HTTP. El módulo spring-webmvc también se conoce como web servlet y es quien contiene las implementaciones de Spring para la ejecución del patrón de arquitectura modelo-vista-controlador (MVC) y de los servicios web REST.
* **Test**: el módulo spring-test permite comprobar la funcionalidad de los componentes de tu aplicación Java. Así, con la ayuda de frameworks adicionales como JUnit o TestNG, puedes realizar pruebas unitarias o unit tests (centrándote en un solo componente) así como pruebas de integración.



**Subtema 1.4:**

Spring Boot.

Spring Boot, publicado en 2012, es una solución para el framework Spring de Java que sigue el principio de “convención sobre configuración” y reduce la complejidad del desarrollo de nuevos proyectos basados en Spring. Para ello, Spring Boot proporciona la estructura básica configurada del proyecto, que incluye las pautas para usar el marco y todas las bibliotecas de terceros relevantes para la aplicación, lo que nos allana el camino para comenzar a desarrollarla lo más rápidamente posible. De esta manera se simplifica mucho la creación de aplicaciones independientes y reproducibles, por lo que la mayoría de las nuevas aplicaciones basadas en Springse desarrollan con Spring Boot.

Las características de Spring Boot pueden resumirse de la siguiente manera:

* Incorporación directa de aplicaciones de servidores web/contenedores como Apache Tomcat o Jetty, eliminando la necesidad de incluir archivos WAR (Web Application Archive)
* Simplificación de la configuración de Maven gracias a los POM (Project Object Models) “starter”
* Configuración automática de Spring en la medida de lo posible
* Características no funcionales, como métricas o configuraciones externalizadas

La empresa desarrolladora Pivotal Software dio al framework de Spring, publicado ya en 2005, un enfoque moderno y con visión de futuro cuando lanzó Spring Boot. Para desarrollarlo, se recurrió a la sólida tecnología de base del framework de Spring, madurada durante años, que contribuyó mucho a su funcionalidad.

**Subtema 1.5:**

Herramientas por utilizar en el curso.

Es necesario contar con las siguientes herramientas:

* JDK. <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

* Spring Tool Suite. <https://spring.io/tools>

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* IntelliJ Idea. <https://www.jetbrains.com/idea/download/?gclid=Cj0KCQjw9O6HBhCrARIsADx5qCTs6mSCuoJ0z4TYK9jVifccl0IiQuTHnW51jTBIIYnKaBN2rICP3AgaAuYLEALw_wcB#section=windows>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* MySQL Workbench. <https://dev.mysql.com/downloads/workbench/>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Subtema 1.6:**

Creando un proyecto Web con Spring Boot.

**Ejemplos:**

Ejecutamos Spring Tool Suite.

Seleccionamos la opción Project.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Buscamos Spring Boot y seleccionamos Spring Starter Project, dar clic en Next.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Agregamos un nombre al proyecto.

Quitamos el check para cambiar la ubicación, y dar clic en Browser.

Colocar un nombre a Group y Package y dar clic en Next.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Buscar las siguientes dependencias de Spring Boot Dev Tools y Spring Web y dar clic en Finish.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Obtenemos nuestro primer proyecto con Spring Boot.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Actividad:**

a) CUESTIONARIO TÉCNICO

* + Describa con sus propias palabras las ediciones de Java.
  + Explique la relación entre JVM, JRE y JDK.
  + Indique dos características de Spring Framework que para usted son importantes.
  + Indique dos beneficios de Spring Framework que para usted son importantes.
  + Defina con sus propias palabras Spring Boot.

b) CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_