

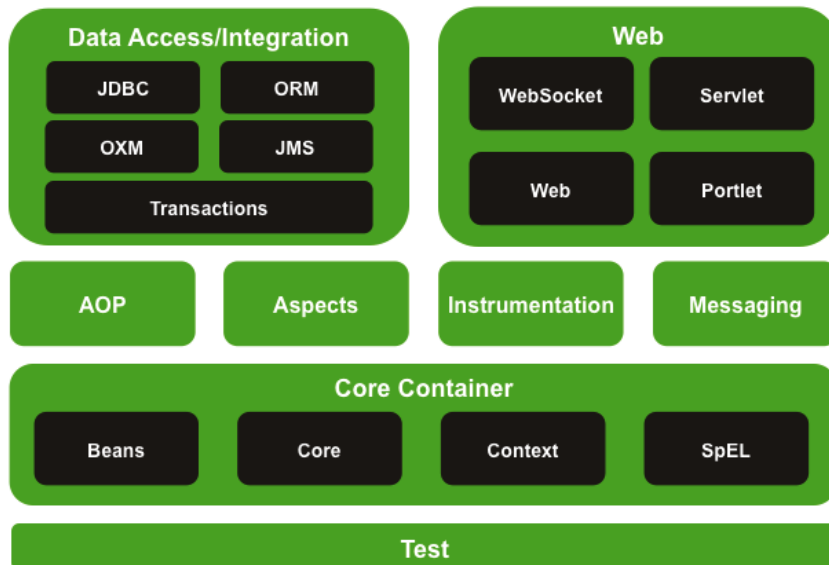


GUSTAVO CORONEL
DESARROLLA SOFTWARE

DESARROLLO WEB CON SPRING BOOT



Spring Framework Runtime



UNIDAD 02 SPRING FRAMEWORK

Eric Gustavo Coronel Castillo
I N S T R U C T O R
youtube.com/DesarrollaSoftware
gcoronelc@gmail.com



CONTENIDO

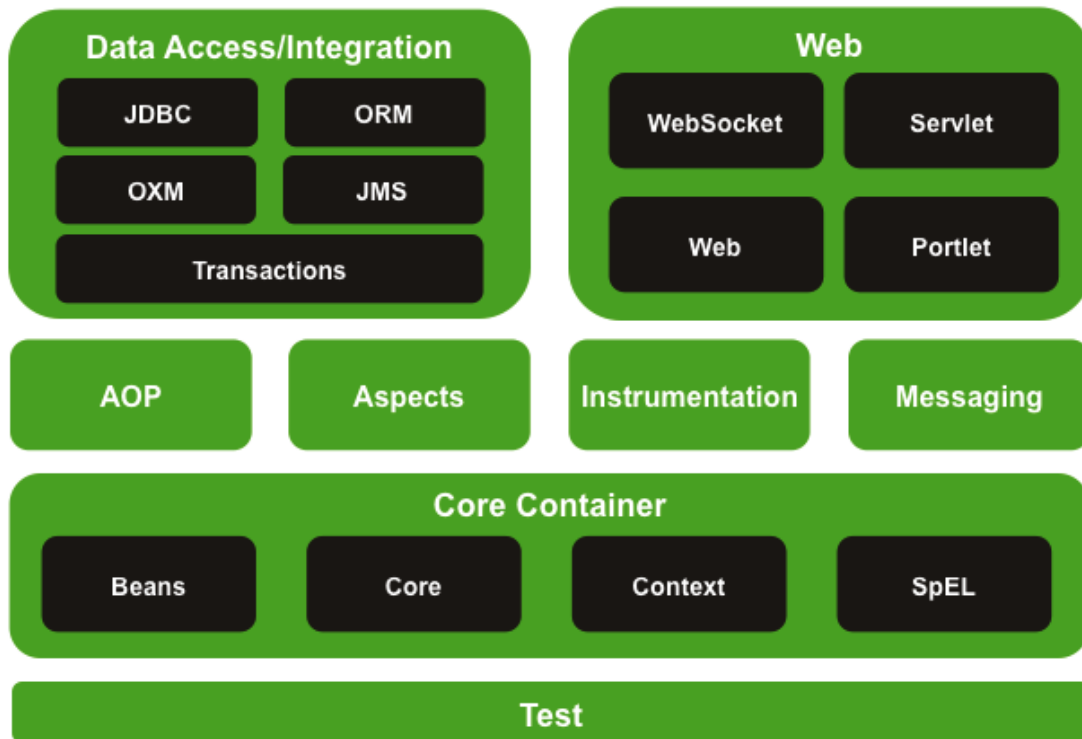
¿QUÉ ES SPRING?	3
UN POCO DE HISTORIA	4
CARACTERÍSTICAS	6
VERSIONES	7
MODULOS	8
CORE CONTAINER	9
APPLICATION CONTEXT MODULE	9
AOP MODULE	9
JDBC ABSTRACTION AND THE DAO MODULE	10
OBJECT RELATIONAL MAPPING INTEGRATION MODULE	11
SPRING'S WEB MODULE	12
SPRING BOOT	13
CARACTERÍSTICAS	13
STARTERS	13
HOLA MUNDO	14
SPRING INITIALIZR	15
CURSOS VIRTUALES	16
ACCESO A LOS CURSOS VIRTUALES	16
FUNDAMENTOS DE PROGRAMACIÓN CON JAVA	16
JAVA ORIENTADO A OBJETOS	17
PROGRAMACIÓN CON JAVA JDBC	18
PROGRAMACIÓN CON ORACLE PL/SQL	19



¿QUÉ ES SPRING?



Spring Framework Runtime



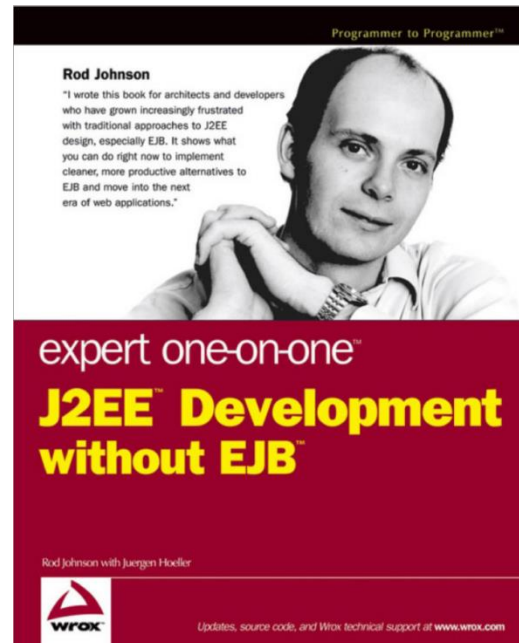
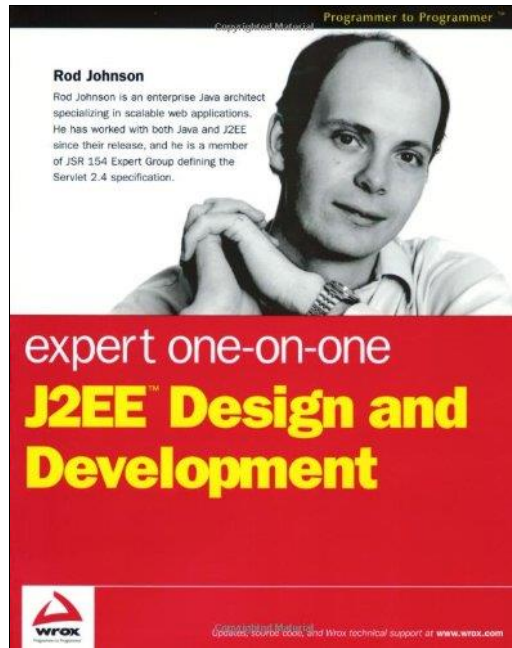
Spring es un framework creado por **Rod Johnson** y descrito en su libro **Expert One-on-One: J2EE Design and Development**. Fue creado para solucionar la complejidad en el desarrollo de aplicaciones empresariales.

Spring hace posible el uso de simples JavaBeans para conseguir cosas que antes era sólo posible con los EJBs.

Está basado en el patrón inyección de dependencia (DI), está compuesto por librerías y puede integrarse con muchas otras, permitiendo de esta manera crear aplicaciones flexibles, altamente cohesivas y débilmente acopladas, permitiendo de esta manera un desarrollo rápido y fácil mantenimiento.



UN POCO DE HISTORIA



Fuente Wikipedia: https://es.wikipedia.org/wiki/Spring_Framework#Historia.

Los primeros componentes de lo que se ha convertido en Spring Framework fueron escritos por **Rod Johnson** en el año 2000, mientras trabajaba como consultor independiente para sus clientes en la industria financiera en Londres. Mientras escribía el libro Expert One-on-one J2EE Design And Development (Programmer to programmer), Rod amplió su código para sintetizar su visión acerca de cómo las aplicaciones que trabajan con varias partes de la plataforma J2EE podían llegar a ser más simples y más consistentes que aquellas que los desarrolladores y compañías estaban usando por aquel entonces.

En el año 2001 los modelos dominantes de programación para aplicaciones basadas en web eran ofrecidas por el API Java Servlet y los Enterprise JavaBeans, ambas especificaciones creadas por Sun Microsystems en colaboración con otros distribuidores y partes interesadas que disfrutaban de gran popularidad en la comunidad Java. Las aplicaciones que no eran basadas en web, como las aplicaciones basadas en cliente o aplicaciones en batch, podían ser escritas con base en herramientas y proyectos de código abierto o comercial que proveyeran las características requeridas para aquellos desarrollos.

Se formó un pequeño equipo de desarrolladores que esperaba trabajar en extender el framework y un proyecto fue creado en Sourceforge en febrero de 2003. Después de trabajar en su desarrollo durante más de un año lanzaron una primera versión (1.0) en marzo de 2004. Después de este lanzamiento Spring ganó mucha popularidad en la comunidad Java, debido



en parte al uso de Javadoc y de una documentación de referencia por encima del promedio de un proyecto de código abierto.

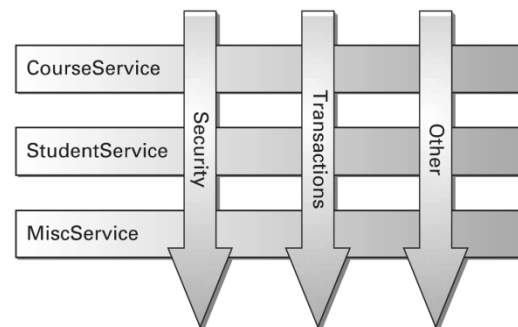
Sin embargo, Spring Framework también fue duramente criticado en 2004 y sigue siendo el tema de acalorados debates. Al tiempo en que se daba su primer gran lanzamiento muchos desarrolladores y líderes de opinión vieron a Spring como un gran paso con respecto al modelo de programación tradicional; esto era especialmente cierto con respecto a Enterprise JavaBeans. Una de las metas de diseño de Spring Framework es su facilidad de integración con los estándares J2EE y herramientas comerciales existentes. Esto quita en parte la necesidad de definir sus características en un documento de especificación elaborado por un comité oficial y que podría ser criticado.

Spring Framework hizo que aquellas técnicas que resultaban desconocidas para la mayoría de programadores se volvieran populares en un periodo muy corto de tiempo. El ejemplo más notable es la inversión de control. En el año 2004, Spring disfrutó de unas altísimas tasas de adopción y al ofrecer su propio framework de programación orientada a aspectos (aspect-oriented programming, AOP) consiguió hacer más popular su paradigma de programación en la comunidad Java.

En 2005 Spring superó las tasas de adopción del año anterior como resultado de nuevos lanzamientos y más características fueron añadidas. El foro de la comunidad formada alrededor de Spring Framework (The Spring Forum) que arrancó a finales de 2004 también ayudó a incrementar la popularidad del framework y desde entonces ha crecido hasta llegar a ser la más importante fuente de información y ayuda para sus usuarios.



CARACTERISTICAS



Spring te permite desarrollar aplicaciones flexibles, altamente cohesivas y con un bajo acoplamiento.

Spring te permite simplificar el desarrollo JEE al utilizar clases Java Simples (POJO – Plain Old Java Object) para la configuración de servicios.

Spring te permite resolver de una manera sencilla muchas de las tareas repetitivas que tienes en tus proyectos, tales como Localización de Servicios, Seguridad, Manejo de Transacciones, Manejo de Excepciones, Parametrización de la aplicación, entre muchos más. Para lograrlo se basa en dos conceptos fundamental:

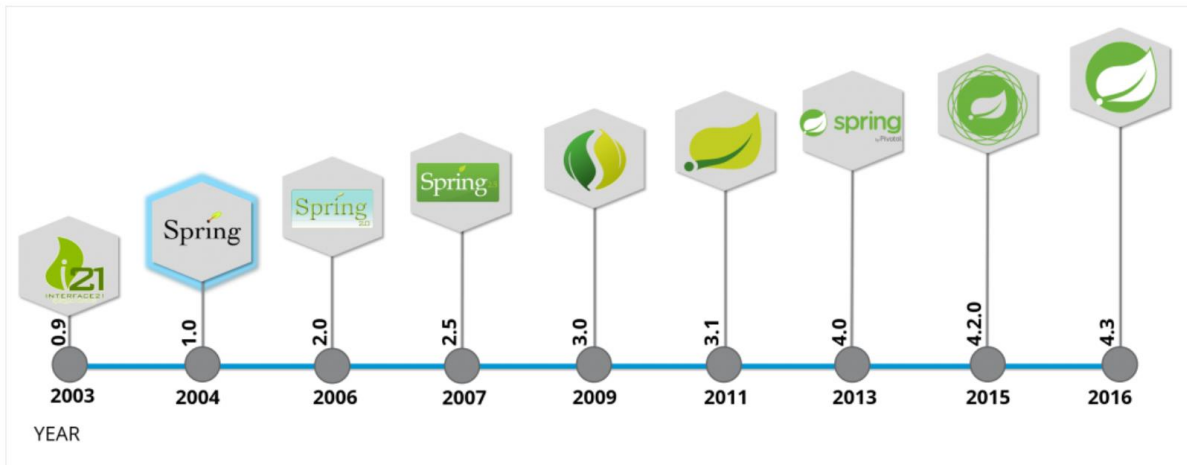
- **DI (Dependency Injection):** Este patrón de diseño permite suministrar objetos a una clase (POJO) que tiene dependencias, en lugar de ser ella misma sea quien los proporcione.
- **AOP (Aspect Oriented Programming):** AOP es un paradigma de programación que permite modularizar las aplicaciones y mejorar la separación de responsabilidades entre módulos y/o clases.

Las características anteriores son la base para la creación de contenedores ligeros (lightweight containers). Spring es uno de los contenedores ligeros más completos y utilizados en proyectos Java.

Puedes consultar el siguiente enlace: <https://spring.io/projects/spring-framework>



VERSIONES



Fuente: https://en.wikipedia.org/wiki/Spring_Framework

- La primera versión fue escrita por Rod Johnson
- Expert One-on-One J2EE Design and Development: Octubre 2002
- Primer reléase en Junio del 2003
- Spring 1.0: Marzo 2004
- Spring 2.0: Octubre 2006
 - ✓ Java 1.3+
 - ✓ Soporte a AspectJ, JPA
- Spring 2.5: Noviembre 2007
 - ✓ Java 1.4+
 - ✓ XML namespaces, anotaciones
- Spring 3.0: Diciembre 2009
 - ✓ Java 1.5+
 - ✓ Soporte REST, SpEL, mas anotaciones, JavaConfig.
- Spring 3.1: Diciembre 2011
- Spring 3.2.5: Noviembre 2013
- Spring 4.0: Diciembre 2013
 - ✓ Incluye soporte para Java 8
- Spring 4.2: Julio 2015

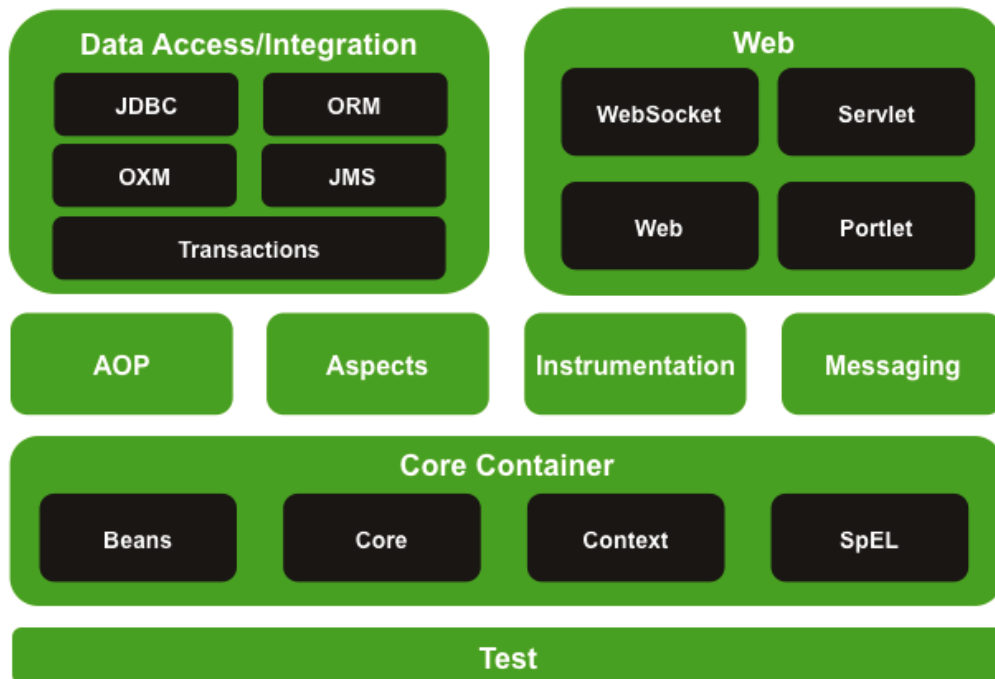
Actualmente Spring Framework se encuentra en la versión 5.3.



MODULOS



Spring Framework Runtime



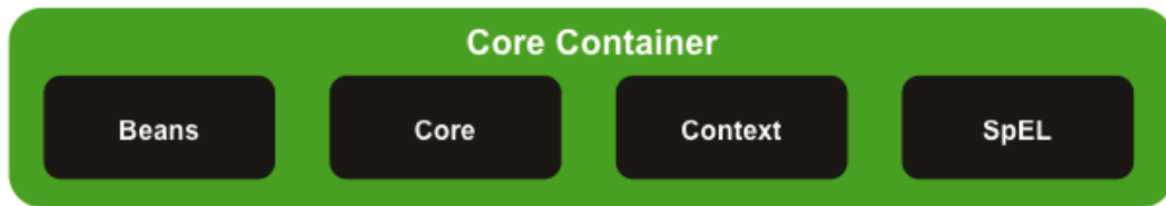
Spring está compuesto por diferentes y bien definidos módulos, contruidos sobre la base del Core Container.

Ser modular, es precisamente la característica que hace posible el uso de todo o una parte de Spring Framework, según las necesidades de una aplicación en particular.

Puedes revisar: <https://docs.spring.io/spring-framework/docs/current/reference/html/>



Core Container



El contenedor de Spring provee la funcionalidad más importante del Spring Framework.

En este módulo encontraras el BeanFactory de Spring, que es la base cualquier aplicación que desarrolles basada Spring.

El BeanFactory es una implementación del patrón Factory, aplica DI para separar la configuración de tu aplicación y sus dependencias.

Application Context Module

Este módulo extiende del BeanFactory, añadiendo soporte para mensajes (I18N), eventos de aplicación, validaciones.

Provee muchos de los servicios empresariales como email, acceso vía JNDI, integración EJB, remoting y scheduling.

Incluye del soporte para integrarse con otros frameworks de vista como Velocity y FreeMarker.

AOP Module



Spring provee un soporte muy enriquecido para la programación orientada a aspectos con su módulo AOP.

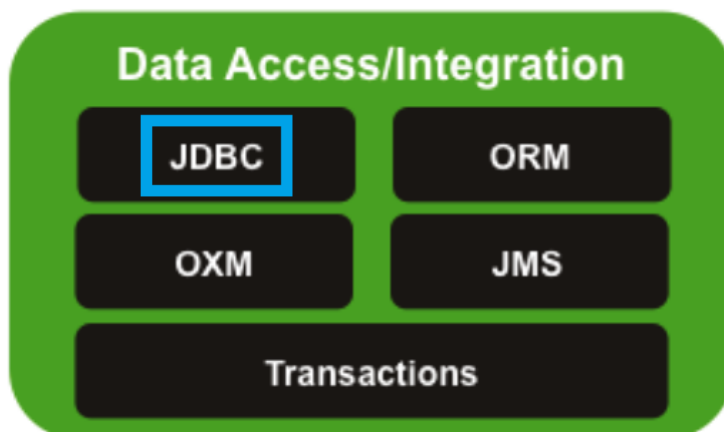
Te permite desarrollar tus propios aspectos, y, así como DI, AOP te permite desacoplar los componentes de tu aplicación.

Aspectos como manejo de transacciones, seguridad, etc, puedes desacoplarlos de los componentes donde se debería aplicar.

También te permite trabajar con otros framework con AspectJ, que es un framework bastante aceptado en la comunidad de Java.



JDBC Abstraction and the DAO Module



Cuando has programado con JDBC, crear un Statement o PreparedStatement, procesar un ResultSet, administrar transacciones, cerrar conexiones a base de datos, siempre te ha sido un tema crítico y repetitivo al momento de programar los componentes de persistencia.

Spring JDBC y Data Access Objects (DAO) abstrae la complejidad que resulta de programar con JDBC, de tal manera que tú código sea limpio, simple y fácil para darle mantenimiento.

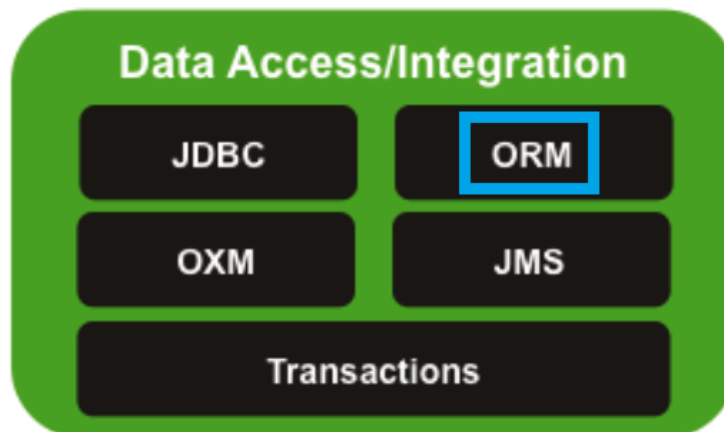
Ya no tienes que preocuparte de los errores comunes de liberación de los recursos de base de datos.

También tienes un soporte significativo para el manejo de excepciones y mensajes de error dados por diferentes servidores de base de datos. No tendrás que lidiar con mensajes SQL propietarios.

Adicionalmente, en este módulo encuentras el servicio de manejo de transacciones que puedes aplicar de una manera muy sencilla en todas tus aplicaciones.



Object Relational Mapping Integration Module



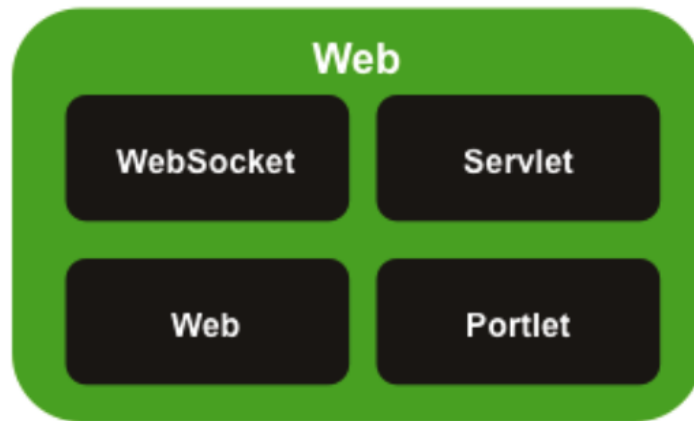
Este módulo te permite integrar tu aplicación con framework ORM para programar tus componentes DAO.

Entre los diferentes frameworks ORM que puedes utilizar tienes a Hibernate, Java Persistence API y MyBATIS SQL Maps.

Para el manejo de tus transacciones, Spring Transaction Management soporta cada uno de estos Frameworks ORM, así como JDBC.



Spring's Web Module

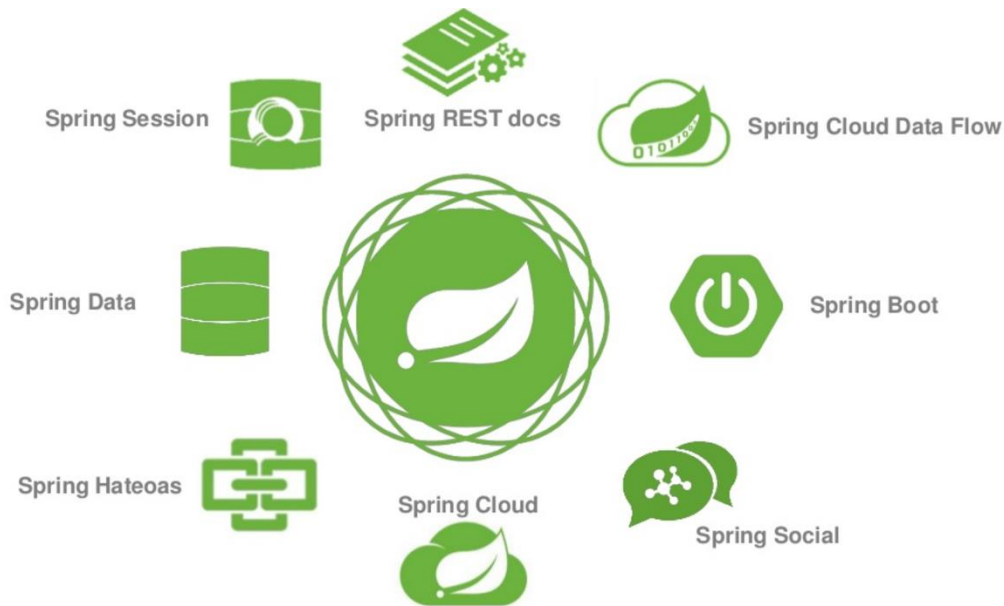


Con Spring, tienes un módulo para aplicar MVC, este módulo te permite utilizar técnicas de desacoplamiento entre el controlador y las vistas.

Sin embargo, si prefieres usar por ejemplo Java Server Faces u otro framework, Spring te permite hacer la integración de una manera muy fácil.



SPRING BOOT



Spring Boot facilita la creación de aplicaciones basadas en Spring para que te concentres solo en tu código puedas luego "simplemente ejecutar".

Toma la plataforma Spring y las librerías de terceros para que pueda comenzar con el mínimo esfuerzo el desarrollo de una aplicación. La mayoría de las aplicaciones Spring Boot necesitan una configuración mínima de Spring.

Características

- Cree aplicaciones Spring independientes
- Cuenta con Tomcat embebido (no es necesario implementar archivos WAR)
- Configure automáticamente las librerías de Spring y de terceros siempre que sea posible
- Absolutamente sin generación de código y sin requisitos de configuración XML

Starters

Spring Boot nos proporciona una serie de dependencias, llamadas starters, que podemos añadir a nuestro proyecto dependiendo de lo que necesitemos: crear un controlador REST, acceder a una base de datos usando JDBC, conectar con una cola de mensajes Apache ActiveMQ, etc.



Una vez añadimos un starter, éste nos proporciona todas las dependencias que necesitamos, tanto de Spring como de terceros. Además, los starters vienen configurados con valores por defecto, que pretenden minimizar la necesidad de configuración a la hora de desarrollar.

Al igual que con Spring Framework, cualquier configuración puede ser modificada de ser necesario: desde el puerto en el que la aplicación escucha peticiones, hasta el banner que sale por consola al arrancar la aplicación.

Hola Mundo

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@RestController
public class HolaMundoApplication {

    public static void main(String[] args) {
        SpringApplication.run(HolaMundoApplication.class, args);
    }

    @GetMapping("/hello")
    public String hello() {
        return "Hola mundo!";
    }

}
```



SPRING INITIALIZR

<https://start.spring.io/>

The screenshot shows the Spring Initializr web application interface. It has a header with the 'spring initializr' logo. The main content area is divided into two columns. The left column contains sections for 'Project' (with radio buttons for Maven Project, Gradle Project, and Language: Java, Kotlin, Groovy), 'Spring Boot' (with radio buttons for versions 2.4.1, 2.4.0, 2.3.7, 2.3.6, 2.2.12, and 2.2.11), 'Project Metadata' (with input fields for Group, Artifact, Name, Description, and Package name), and 'Packaging' (with radio buttons for Jar and War, and Java versions 15, 11, and 8). The right column contains a 'Dependencies' section with a button 'ADD DEPENDENCIES... CTRL + B' and a 'Spring Web' section with a 'WEB' button and a description: 'Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.'

El equipo de Spring también ha creado Spring Initializr, una herramienta web que puedes utilizar para autogenerar el esqueleto de tu aplicación Spring Boot.

Se trata de un inicializador muy útil, ya que genera el pom.xml (o build.gradle), el main de la aplicación, y un test que comprueba que la aplicación arranca correctamente.



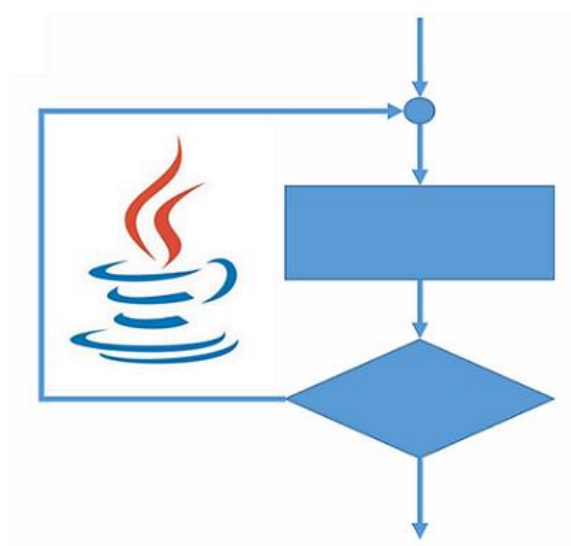
CURSOS VIRTUALES

Acceso a los Cursos Virtuales

En esta URL tienes los accesos a los cursos virtuales:

<http://gcoronelc.github.io>

Fundamentos de Programación con Java



Tener bases sólidas de programación muchas veces no es fácil, creo que es principalmente por que en algún momento de tu aprendizaje mezclas la entrada de datos con el proceso de los mismos, o mezclas el proceso con la salida o reporte, esto te lleva a utilizar malas prácticas de programación que luego te serán muy difíciles de superar.

En este curso aprenderás las mejores prácticas de programación para que te inicies con éxito en este competitivo mundo del desarrollo de software.

URL del Curso: <https://n9.cl/gcoronelc-java-fund>

Avance del curso: <https://n9.cl/gcoronelc-fp-avance>

Cupones de descuento: <http://gcoronelc.github.io>



Java Orientado a Objetos



CURSO PROFESIONAL DE JAVA ORIENTADO A OBJETOS

Eric Gustavo Coronel Castillo

www.desarrollasoftware.com

I N S T R U C T O R

En este curso aprenderás a crear software aplicando la Orientación a Objetos, la programación en capas, el uso de patrones de software y Swing.

Cada tema está desarrollado con ejemplos que demuestran los conceptos teóricos y finalizan con un proyecto aplicativo.

URL del Curso: <https://bit.ly/2B3ixUW>

Avance del curso: <https://bit.ly/2RYGXIt>

Cupones de descuento: <http://gcoronelc.github.io>



Programación con Java JDBC



PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JAVA JDBC

Eric Gustavo Coronel Castillo

www.desarrollasoftware.com

I N S T R U C T O R

En este curso aprenderás a programar bases de datos Oracle con JDBC utilizando los objetos Statement, PreparedStatement, CallableStatement y a programar transacciones correctamente teniendo en cuenta su rendimiento y concurrencia.

Al final del curso se integra todo lo desarrollado en una aplicación de escritorio.

URL del Curso: <https://bit.ly/31apy0O>

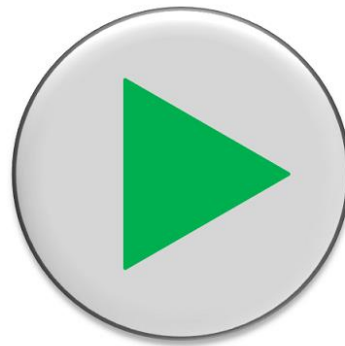
Avance del curso: <https://bit.ly/2vatZOT>

Cupones de descuento: <http://gcoronelc.github.io>



Programación con Oracle PL/SQL

ORACLE PL/SQL



En este curso aprenderás a programar las bases de datos ORACLE con PL/SQL, de esta manera estarás aprovechando las ventajas que brinda este motor de base de datos y mejorarás el rendimiento de tus consultas, transacciones y la concurrencia.

Los procedimientos almacenados que desarrolles con PL/SQL se pueden ejecutarlos de Java, C#, PHP y otros lenguajes de programación.

URL del Curso: <https://bit.ly/2YZjfxT>

Avance del curso: <https://bit.ly/3bcigYb>

Cupones de descuento: <http://gcoronelc.github.io>