



# **UNIVERSIDAD INTERAMERICANA PARA EL DESARROLLO**

**CAMPUS CANCÚN**

**“Frameworks”**

**“Spring Framework”**

**“Model View Controller”**

**“Formato de versiones”**

**DESARROLLO ORIENTADO A OBJETOS**

**ELABORADO POR:**

**MARIO DAVID MORALES CASTRO**

**LIC. INGENIERÍA EN TECNOLOGÍAS DE  
INFORMACIÓN Y COMUNICACIÓN**

**ENERO 2019**

## Frameworks

---

Framework es un término usado muy frecuentemente dentro la ingeniería de software, especialmente cuando se habla del diseño orientado a objetos. En general, un framework puede definirse como un generador de aplicaciones, o para ser más específicos, este representa el esqueleto de una aplicación, este incluye el código completo que realiza las funciones básicas del sistema y se encarga de cumplir con necesidades específicas de la aplicación.

Una definición ampliamente aceptada para un framework orientado a objetos es la dada por Johnson & Foote:

"Un conjunto de clases que incorporan un diseño abstracto para soluciones a una familia de problemas relacionados, y soporta re-uso a un nivel de granularidad mayor que el de las clases"<sup>1</sup>

Una de las principales razones que dan viabilidad al desarrollo de aplicaciones mediante Frameworks es el re-uso de código. Los Frameworks están desarrollados para hacer más fácil el re-uso de código.

El desarrollo de software convencional difiere del desarrollo mediante Frameworks. Una de las mayores diferencias se encuentra en la fase de análisis de un desarrollo de software convencional, en la que solo un dominio de aplicación (aquellos que maneja la aplicación<sup>2</sup>) es analizado. Por otra parte, El análisis de dominio mediante un framework involucra el dominio completo. El resultado del desarrollo convencional es una aplicación en concreto, mientras que el desarrollo mediante Frameworks puede dar como resultado 'n' número de aplicaciones.

El uso de un framework requiere tener un conocimiento amplio acerca de su arquitectura o haber leído su documentación detalladamente. Existen dos tipos de Frameworks:

### **Caja blanca**

En este caso el programador debe crear clases concretas para un conjunto de objetos de los que solo se define su interface. Suelen tener una visión más general, por lo que suelen aparecer porciones similares de una aplicación a otra. Suelen emplearse en desarrollo de aplicaciones son las MFC's del "Microsoft Visual C++"<sup>3</sup>

### **Caja Negra**

Podría considerarse como un framework a medida, en la que el programador elige la combinación adecuada para su desarrollo, dando lugar a muchas aplicaciones distintas. El usuario en general configura el framework.

## Spring Framework

---

Spring framework es una aplicación framework open-source destinada a la plataforma de Java. Spring Framework puede ser usado en el núcleo de cualquier aplicación Java, este no impone un modelo específico de programación lo que lo ha vuelto popular dentro de la comunidad de desarrolladores en Java.

Spring framework permite construir aplicaciones de manera simple, rápida y flexible gracias a que se compone de módulos los cuales permiten la implementación de características a una aplicación a elección del desarrollador o las necesidades del proyecto.

La tecnología que más identifica a Spring es su Dependency Injection (DI). Cuando se crea una aplicación compleja en Java, las clases de la aplicación deben de ser lo más independientes posibles de otras clases para incrementar la posibilidad de reutilizarlas y probarlas independiente mente en otras clases. Dependency Injection ayuda a unir estas clases y al mismo tiempo mantenerlas independientes entre sí. Se podría decir que el Dependency Injection es el corazón de Spring Framework.

El modulo principal de Spring se conoce como Spring Core Container. Este consiste en 4 módulos: Core, Beans, Context y SpEL.

**Core:** Provee las parte fundamentales del framework incluida la característica DI

**Bean:** Es una sofisticada implementación de del patrón fabrica.

**Context:** Construye una base sólida provista por los módulos Core y Beans y constituye un acceso medio a la definición y configuración de objetos.

**SpEL:** Provee un poderoso lenguaje de expresión para consultar y manipular objetos.

Spring Framework ofrece además cerca de 20 módulos los cuales pueden ser utilizados en base a los requerimientos de la aplicación:

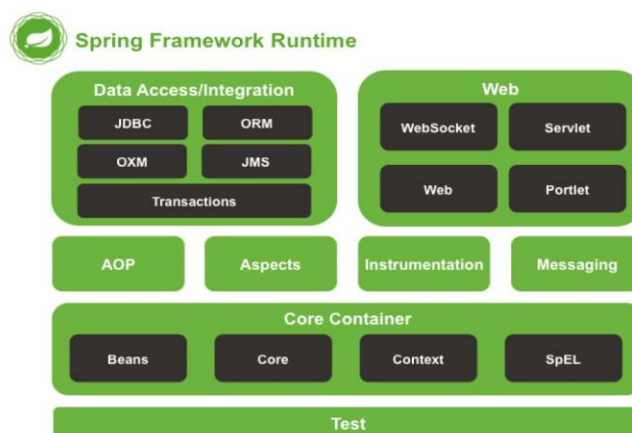


Imagen de: <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/images/spring-overview.png>

## Model View Controller (MVC)

---

El modelo vista controlador es un patrón arquitectónico para el desarrollo de aplicaciones, Su mayor ventaja se encuentra en la reutilización de código. El las divide en tres componentes fundamentales.

**El modelo:** Contiene la finalidad central y los datos de la aplicación. También es el componente encargado de recuperar datos de una base de datos u alguna otra fuente de datos. Puede interpretarse que el modelo es la parte que interactúa con los datos, la lógica y las reglas de negocio.

**La vista:** Es el componente responsable de presentar la aplicación al usuario. La vista controla como los datos son mostrados y como lo usuarios interactúan con ellos. Por regla general, la vista no debe contener elementos de la lógica de la aplicación, por lo que estos se deben mantener al mínimo. Adicionalmente cuando el modelo cambia, la vista debe redibujar la zona afectada para indicar el cambio al usuario.

**El controlador:** Es el componente sirve para llamar la función del modelo y enviar los resultados a través de la vista. El controlador también es responsable de marcar como la aplicación debe responder en diferentes situaciones, enviando comandos al modelo o mapeando secciones de la vista para el usuario final.

Como se puede leer, MVC sigue un enfoque más común para realizar el acomodo de una aplicación. Aplicando una lógica que divide el código en capas independientes las cuales cumplen funciones diferentes que permiten hacer uso de la aplicación.

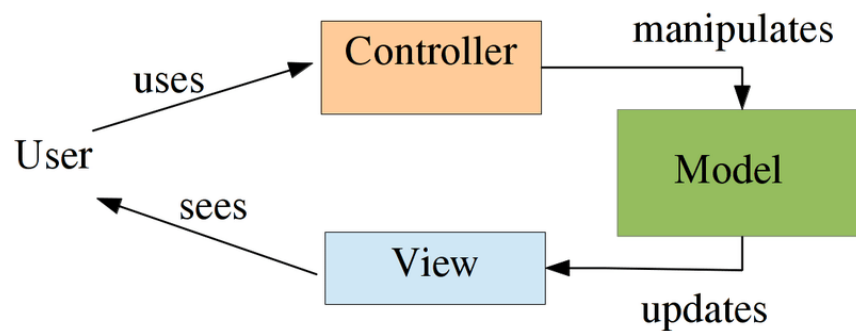


Imagen de: [https://www.researchgate.net/figure/Model-View-Controller-Design-Pattern\\_fig3\\_259922547](https://www.researchgate.net/figure/Model-View-Controller-Design-Pattern_fig3_259922547)

## Formato de versiones

---

El control de versiones, es el manejo de métodos y herramientas para la enmarcación de los cambios en el tiempo de desarrollo de un software. Esto facilita su desarrollo futuro y permite mantener un registro de las modificaciones realizadas.

### Versiones X.Y.Z

Es el método más común para numerar las versiones, se emplean dos o tres cifras decimales para indicar la importancia de los cambios realizados.

**X:** Es la versión mayor del código, denota cambios en mayores realizados en la aplicación, tales como reescritura de código.

**Y:** Indica la versión mayor del documento. Y representas cambios en la funcionalidad de la aplicación, pero no a tal grado que generen incompatibilidad entre versiones.

**Z:** Indica la segunda versión menor, Puede referirse a simples correcciones en las que no se quita o cambia nada relevante para el código.

### Versiones numeradas

Este método implica enumerar la versión del código con solamente una cifra decimal, Ej. 1 o 2 Y sucesivamente. Suele utilizarse cuando las versiones se llevan de manera automática mediante un control de versiones.

## Conclusiones

---

Para concluir, podemos notar como el desarrollo de software a logrado adaptarse a los requerimientos de una industria más demandante, permitiendo la implementación de nuevas tecnologías y modelos de diseño más avanzados y versátiles de los que existían anteriormente.

La implementación de frameworks en el desarrollo de software permite las múltiples ventajas tales como la reutilización de código y una mayor versatilidad y modularidad a los proyectos de desarrollo.

Spring framework es un gran ejemplo de ello, proporcionado una plataforma para el desarrollo de aplicaciones con capacidades de reutilización y modularidad, permitiendo la creación de múltiples aplicaciones siguiendo este mismo modelo de desarrollo.

## Bibliografía

- <sup>1</sup>R.E Johnson and B. Foote. (Junio/Julio 1988). "Designing Reusable Classes", Journal of Object. Oriented Programming.
  - <sup>2</sup>Ingeniería de software II. (17/01/2019), Recuperado de sitio web: [https://ldc.usb.ve/~suarez/pdf/Ingenieria\\_de\\_Software\\_II.pdf](https://ldc.usb.ve/~suarez/pdf/Ingenieria_de_Software_II.pdf).
  - Ojislav Stanojević, Siniša Vlajić, Milić Miloš, Marina Ognjanović. (Octubre 2011). "Guidelines for framework development process". Software engineering laboratory faculty of organizational sciences -, university of Belgrade.
  - <sup>3</sup>Conceptos Básicos en Ingeniería de Software III. (17/01/2019), Recuperado de sitio web: <https://iie.fing.edu.uy/investigacion/grupos/bicoti/bicoti1/SoftEngineering/softeng03.htm>.
  - Federico Julián Gutiérrez Faraoni. (2015). Desarrollo de una aplicación web con spring framework para un gestor de un recetario. Madrid: Universidad Politécnica de Madrid, Escuela técnica superior de ingeniería de sistemas informáticos.
  - Spring Framework. (17/01/2019), Recuperado de sitio web: [https://www.tutorialspoint.com/spring/spring\\_tutorial.pdf](https://www.tutorialspoint.com/spring/spring_tutorial.pdf).
  - Kembang Hapsari, Rinci, Azinar, Azmuri, Sugiyanto, Sugiyanto. (2017). Architecture Application Model View Controller (MVC) in Designing Information System of MSME Financial Report. Surabaya, Indonesia: Quest Journals Journal of Software Engineering and Simulation.
- 
-