

Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

# Spring

Juan Carlos Figueroa Duran

Primavera -2019

Departamento de Tecnologías de la Información  
y Ciencias de la Computación  
Universidad del Bío-Bío

# Outline

## Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

- 1 Understanding the context
- 2 Spring history
- 3 See Spring from outside the forest
- 4 See Spring from inside the forest
  - The Life Cycle

# What is a framework?

## Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest  
The Life Cycle

Un framework es la pieza principal de una aplicación web. Se le dice Principal, porque, definir la estructura de una aplicación web es quizás mas importante que la especificación de los requerimientos funcionales. Un framework intenta hacer generalizaciones de tareas comunes y de flujos de trabajo de un dominio específico.

# What is a MVC?

Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

El patron arquitectonico MVC provee una separación de los componentes del software que se aplica muy bien a las aplicaciones web. La separación de los componentes permite manejar la complejidad de sistemas de software de gran envergadura, dividiéndolos en componentes de alto nivel. El diseño del patron MVC identifica tres distintos componentes: modelo, vista y controlador.

# Spring history

## Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest  
The Life Cycle

## Spring 0.9

En el 2003, a partir de interface21 (un primer intento de framework), se forma Spring, ya integrando en su versión 0.9 con soportes a JDBC y sus primeros pasos en seguridad. Liberado bajo licencia Apache.

Durante su existencia Spring ha regido por una sola palabra, evolución.

## Spring 2.5

Para el 2006, Spring (2.0) ya tenía 1 millón de descargas. Fue adquirida por VMware y de aquí en adelante su crecimiento y continuos cambios fueron vertiginosos. En el 2007 Spring 2.5 dio soporte a JPA.

En el 2018 se puede disfrutar de la versión 5, con JDK 8 como mínimo.

# Spring Framework Services

## Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

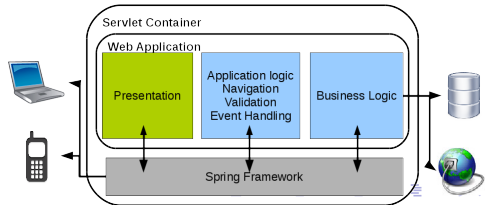
See Spring  
from outside  
the forest

See Spring  
from inside  
the forest  
The Life Cycle

Si bien Spring provee las herramientas para enlazar la interfaz de usuario, lógica de la aplicación y la lógica de negocio de una aplicación web. El ámbito de Spring no esta restringido solo a la capa de presentación.

Alguna de las herramientas que provee son:

- IoC: Inversion of control
- Dependency injection
- Data conversion.
- Validation and error handling
- Internationalization
- Security
- Integration whit



# Component

## Spring

Juan Carlos  
Figueroa  
Duran

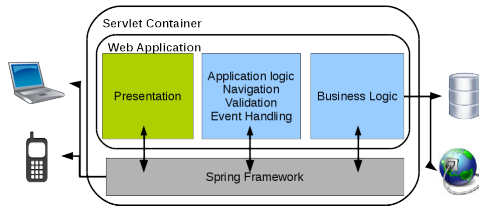
Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest  
The Life Cycle

Para Spring todos los elementos necesario para construir una aplicación serán **Components** o **beans**, y todos los componentes los administrará a través de su **Context** o también conocido como **Spring application context**. Los cuales instanciará, destruirá, actualizará, conectará y proveerá. A medida que la aplicación lo vaya solicitando. Para esto se basará en dos patrones fundamentales IoC y DI.



# IoC: Inversion of control

Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from inside  
the forest

See Spring  
from inside  
the forest  
The Life Cycle

## Patron de diseño

Ahora el control del flujo de ejecución no lo establece el programador, lo define el framework.

Así es, el framework es el que ahora lleva el control indicando que métodos son los que se van a ejecutar y en que secuencia. Dependiendo de las necesidades de información del usuario, esto se logra mediante la inyección de dependencias.



# Dependency injection

Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest  
The Life Cycle

## Patron de diseño

Nunca mas el programador creará un objeto, el framework lo hará por ti.

Tan loco como se escucha, nuestras clases estarán contenidas por otra clase, la cual se encargará de instanciar e inyectar todos los objetos que definamos. Eliminando así problemas de mantención de estos objetos y eliminado instancias de compilación de nuestro sistema. Pero para esto nuestras clase deberán contener los métodos que permitan acceder a los atributos, o constructores que permitan instanciar las mismas.

# Understanding MVC in Spring

## Spring

Juan Carlos  
Figueroa  
Duran

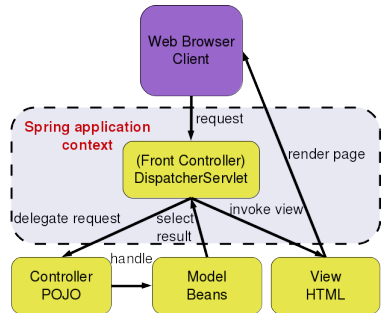
Understanding  
the context

Spring  
history

See Spring  
from inside  
the forest

See Spring  
from inside  
the forest  
The Life Cycle

Desde un diseño de alto nivel Spring MVC sigue un patron MVC bien definido. Podemos ver que se identifican claramente los tres componentes del patrón arquitectónico MVC: modelo, vista y controlador. En Spring, estos estan implementados por Bean (Components), Pages (Thymeleaf), Controllers (Specific type of components) respectivamente.



# Front controller-DispatcherServlet

## Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

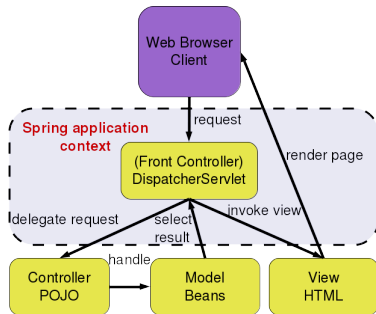
Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

El trabajo del controlador frontal es mapear las peticiones al controlador solicitado.



# Controller-Controller

## Spring

Juan Carlos  
Figueroa  
Duran

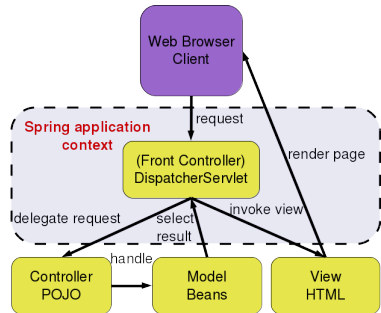
Understanding  
the context

Spring  
history

See Spring  
from inside  
the forest

See Spring  
from inside  
the forest  
The Life Cycle

El trabajo del controlador es mapear las peticiones a beans (Services - Model). En una aplicación web, la petición entrante puede ser considerados comandos que los usuarios ejecutan en la aplicación. Una de las tareas fundamentales de las aplicaciones web es redirigir estas peticiones al conjunto apropiado de beans que tomarán el control de estas dentro de la aplicación.



# Model-Beans

## Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

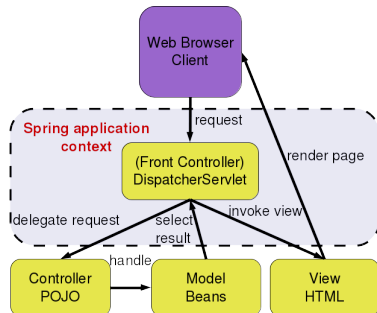
Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

En términos más técnicos, el modelo representa el estado interno de la aplicación. Este estado es compuesto de el **modelo de datos** y la **lógica de negocio**. Desde un nivel de abstracción de caja negra, los datos y la lógica de negocio se combinan en un estado monolítico de la aplicación. Los beans encapsulan las llamadas a la lógica de negocio en un sola unidad de trabajo.



# View-Result

## Spring

Juan Carlos  
Figueroa  
Duran

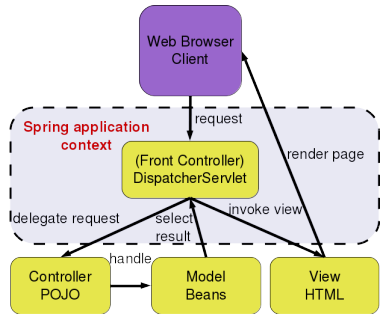
Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest  
The Life Cycle

La vista es la presentación de la pagina retornada al web browser. Esta página es la interfaz de usuario que presenta al mismo una representación del estado de la aplicación.



# Behind the scenes

## Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

Ahora que ya hemos visto el **Que** y el **Porque** de Spring, escarbemos en el **Como**.

Todo comienza con la primera conexión como en el ejemplo `http://localhost:8080/login/index.jsp`

Es aquí cuando el framework lee la pagina y encuentra tags como `th:scr` y `th:text`. Cada unos de estos tags tienen asociados una clase tag handler, que cooperan entre si para construir la pagina.

# Rendering pages

## Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

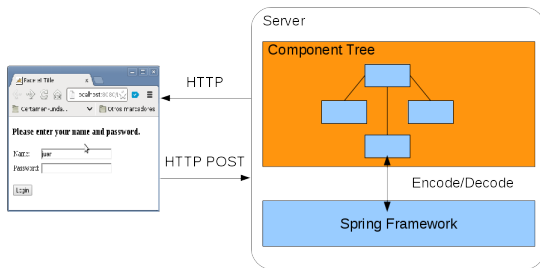
Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

A continuación es cuando la página es interpretada. Todo texto que no corresponda a un Thymeleaf tag es traspasado directamente a una página HTML. En cambio los tags Thymeleaf antes de incluirlos en la página HTML deben ser convertidos en HTML.





# Decoding Request

Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

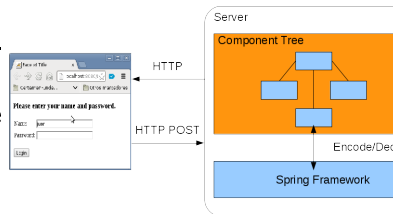
Spring  
history

See Spring  
from inside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

Después que la pagina es desplegada en el navegador, el usuario llena los campos y realiza un click en el botón. Luego, los datos son enviados al web server. En el web server los datos que venían en formato POST son *decodificados*.



# The Life Cycle

## Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

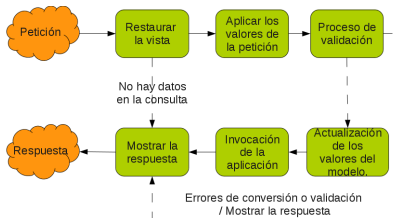
See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

Spring define seis distintas faces en su ciclo de vida:

- Restaurar la vista.
- Aplicar los valores de la petición.
- Proceso de validación.
- Actualización de los valores del modelo.
- Invocación de la aplicación.
- Mostrar la respuesta.





## Spring

Juan Carlos  
Figueroa  
Duran

## Understanding the context

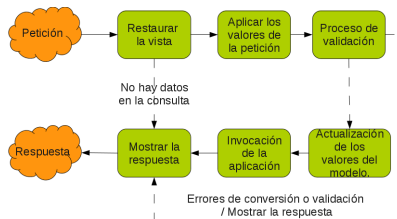
## Spring history

See Spring  
from inside  
the forest

**The Life Cycle**

En esta fase Spring itera sobre los objetos de los componentes del árbol de componentes. Cada objeto componentes verifica que valor de la petición le pertenece y actualiza sus propiedades..

Los valores guardados en los componentes son llamados “valores locales”. Cuando se diseña una pagina Spring, podemos definir que se lleve a cabo una validación de los valores locales.





# Update model values

Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

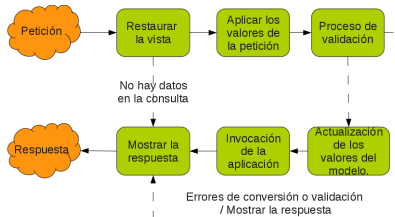
See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

## Validados y convertidos

Una vez que los datos locales han sido validados y convertidos se asume que es seguro actualizar el modelo de datos (Beans), que esta asociado al árbol de componentes.



# Invocación de la aplicación

Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

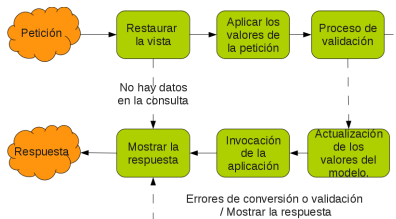
The Life Cycle

## Invocación de la aplicación

En esta fase el método **action** de un componente botón o enlace que haya causado el envío del **form** puede ejecutar un procesamiento arbitrario, retornando un **string** que es entregado al controlador de navegación.

## Controlador de navegación

Con el **string** entregado al controlador de navegación (**navigation handler**) busca un “match” con el nombre de alguna pagina de la aplicación o con un `<navigation-case>` perteneciente al `faces-config.xml`.



# Mostrar la respuesta

Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

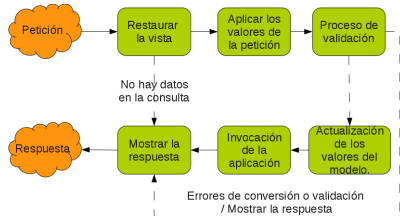
The Life Cycle

## Mostrar la respuesta

Finalmente esta fase codifica la respuesta y la envía al navegador.

### Todo de nuevo

Si el usuario envía un form, hace click en un link o genera una nueva petición, el ciclo comienza nuevamente.





# Special case: Ajax

Spring

Juan Carlos  
Figueroa  
Duran

Understanding  
the context

Spring  
history

See Spring  
from outside  
the forest

See Spring  
from inside  
the forest

The Life Cycle

## Ajax

El ciclo de vida para una petición Ajax es muy similar. En particular los componentes del árbol de componentes se separan y los elementos del tipo **input** quedan en la lista **execute list** y los **output** en **render list**.

Los elementos de la lista **execute list** no ejecutan la ultima fase de mostrar la respuesta, pero los elementos de output si lo hacen.

