

Spring

Spring es un framework para programar en Java. Un framework, como su nombre lo indica, es un marco de referencia, que trae consigo varias herramientas que facilitan la tarea del desarrollador de software. Además, proporciona una arquitectura base, para centrar los esfuerzos en el dominio de la aplicación. Existen muchos frameworks para diferentes tecnologías, en el presente documento hablaremos del framework Spring.

Spring facilita la creación de aplicaciones Java. Además, cuenta con soporte para Groovy y Kotlin, como lenguajes alternativos en la JVM, y con la flexibilidad para crear una variedad de tipos de arquitecturas según las necesidades de una aplicación. Spring es de código abierto, tiene una comunidad grande y activa que proporciona retroalimentación continua basada en una amplia gama de casos de uso del mundo real. Esto ha ayudado a Spring a evolucionar con éxito durante mucho tiempo.

Spring surge en 2003 como respuesta a la complejidad de las primeras especificaciones J2EE. Mientras que algunos autores consideran que Java EE y Spring compiten, Spring es, de hecho, complementario a Java EE. El modelo de programación de Spring no incluye la especificación de la plataforma Java EE; antes bien se integra con especificaciones individuales cuidadosamente seleccionadas del paraguas EE.

Arquitectura de una aplicación WEB



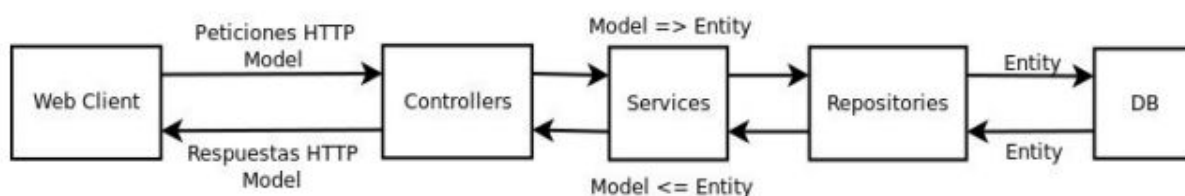
Hablemos un poco de la arquitectura de una aplicación web. Por un lado, en la comunicación entre uno o varios navegadores web y uno o varios servidores web se encuentran peticiones y respuestas HTTP. Ambas son parte un protocolo de comunicación llamado HTTP. Este protocolo especifica un conjunto de reglas que permiten que una comunicación sea efectiva entre el cliente y el servidor. Existen diferentes tipos de peticiones HTTP, todas van a estar conformadas por una dirección URL, por parámetros, por headers para ampliar información de la petición y por su tipo correspondiente. Las dos más comunes son:

- GET: Se utiliza para solicitar recursos.
- POST: Se utiliza para realizar alta, baja o modificación de recursos.

Además de su diferencia en términos de utilización es importante mencionar que también se diferencian por mostrar (GET), o no mostrar (POST), datos de consulta en la dirección URL. Las respuestas HTTP son el resultado que el servidor envía al navegador web. Esta respuesta contiene la información de la solicitud en un formato determinado (HTML, JSON, etc), los headers para ampliar información y el estado de la respuesta con un código de tres dígitos que sirve para poder identificar el tipo de la respuesta.

Del otro lado, se encuentra la comunicación entre el servidor web y la base de datos. Para poder establecer esta comunicación, en el diagrama se usa el concepto de ORM. El mapeo objeto relacional (ORM) permite convertir datos desde la lógica de negocio para luego impactarse en una base de datos, o bien, tomar información de una base de datos y transformarlos en datos de la lógica de negocio. Esto posibilita el uso del paradigma orientado a objetos.

Arquitectura de una aplicación web con Spring



En la arquitectura de una aplicación web con Spring se agregan dos capas más al modelo básico de una aplicación web. Para explicar la misma utilizaremos un ejemplo:

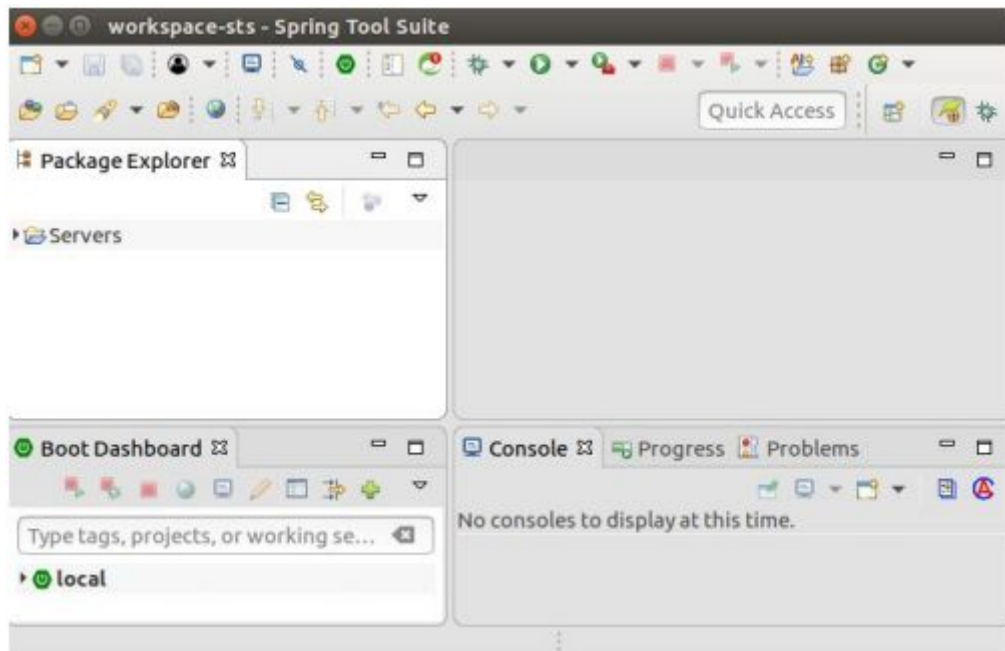
Un usuario final realiza una solicitud al sistema por intermedio de un navegador (Web Client). El flujo se inicia con una petición HTTP por parte del cliente web, dicha petición lleva consigo un modelo (Model) que es la información de la petición. La petición llega a un controlador (Controller) que, a través de un servicio (Service) transforma el modelo en una entidad (Entity) para pasarlo a un repositorio (Repository). Este último impacta esa entidad en la base de datos. Finalmente, la base de datos se comunica con el repositorio, que transforma la entidad recibida en un modelo, a través de un servicio, para luego pasarlo al controlador. Una vez que el controlador haya obtenido su respuesta, podrá realizar otros pedidos a uno o varios servicios, o ejecutar otra lógica de negocio. Cuando el controlador termina la petición, envía una respuesta HTTP al cliente web con el contenido correspondiente. El cliente web sabe como resolver el formato de esa respuesta para mostrarle al usuario final algo significativo con respecto a su solicitud original.

Spring Boot

Spring Boot es una infraestructura ligera que elimina la mayor parte del trabajo de configurar las aplicaciones basadas en Spring. Para ello, podemos utilizar **Spring Tool Suite**, un IDE de desarrollo basado en Eclipse que ya se encuentra customizado para facilitar el desarrollo de aplicaciones con Spring. En caso de no tener instalado Spring Tool Suite o querer cambiar a una versión más reciente se puede seguir la guía de instalación en la página web oficial de Spring:

Pivotal Software. Spring. (2020). Spring Tool SuiteTM. <https://spring.io/tools>.

Una vez instalado y abierto STS debe verse como en la imagen a continuación, la parte importante por ahora es la vista de "Boot Dashboard".



Cuando se genera un proyecto con Spring Boot primero se ingresa a web de Start Spring (2020, <http://start.spring.io>). En ella se puede armar la estructura base de un proyecto. Sólo se agrega la información del proyecto (lenguaje, versión de Spring y nombre del proyecto) y las dependencias que se quieran usar. Haciendo referencia al TFI de Gustavo Siciliano [[link](#)] en dicho ejemplo sólo se agregan dos dependencias, Web “Full-stack web development with Tomcat and Spring MVC” y Template Engines “Thymeleaf”.

Project

☒ Maven Project ☐ Gradle Project

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 2.3.0 M4 ☐ 2.3.0 (SNAPSHOT) ☐ 2.2.7 (SNAPSHOT) ☒ 2.2.6

☐ 2.1.14 (SNAPSHOT) ☐ 2.1.13

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 14 ☐ 11 ☒ 8

Como se puede ver, es un proyecto Maven, con lenguaje Java y versión de Spring Boot 2.2.6. Lo importante de este ejemplo es que el “Artifact” y el “Name” son iguales y además coinciden con el nombre del

repositorio de Github. Hecho esto se puede crear el proyecto con el botón “Generate”, el cual descarga un zip con el proyecto. Se tiene que copiar el contenido de la carpeta del zip, incluyendo lo archivos ocultos, en la carpeta donde se aloja el repositorio.

Luego, se debe abrir el STS, se selecciona la opción de “Import” seguida de la opción “Existing Maven Projects” desde Maven. A continuación se busca y selecciona la ruta del directorio del proyecto. A continuación vemos como queda el ejemplo en el IDE.

