



UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

ROO++ SPRING ROO ADD-ON



JAVIER MENÉNDEZ ÁLVAREZ
JULIO 2014



UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

ROO++ SPRING ROO ADD-ON

DOCUMENTO N° III

ESTUDIO DE VIABILIDAD DEL SISTEMA



JAVIER MENÉNDEZ ÁLVAREZ
JULIO 2014

**ÁREA DE CIENCIAS DE LA
COMPUTACIÓN E INTELIGENCIA
ARTIFICIAL**
TUTOR: M^a JOSÉ SUÁREZ CABAL

Índice

Introducción.....	4
Ámbito y alcance.....	5
Descripción del sistema actual.....	6
Lista de problemas y necesidades.....	6
Catálogo de requisitos.....	7
Requisitos funcionales.....	7
Requisitos no funcionales.....	8

Introducción

El objetivo de este software es la automatización del desarrollo de aplicaciones web complejas, típicamente dentro de una empresa (o una organización de otro tipo), y su integración con el resto de sistemas informáticos de ésta, en caso de que sea necesaria.

Concretamente, el software está dirigido al desarrollo de aplicaciones Java mediante Spring Roo¹ y Apache Maven², asumiendo que el software de la empresa con el que interactuará con las aplicaciones web desarrolladas también está desarrollado con Java y Roo. La implementación del software será un add-on que se integrará en Spring Roo mediante OSGi³.

En este documento, Estudio de Viabilidad del Sistema, veremos cuáles son las necesidades a cubrir y los requisitos con que debe cumplir el software desarrollado.

1 Software de desarrollo rápido de aplicaciones en Java.

2 Sistema de automatización de la compilación de proyectos Java.

3 Sistema de gestión de módulos y servicios para Java.

Ámbito y alcance

El add-on Roo++ debe proporcionar a Spring Roo comandos sencillos, con la menor cantidad posible de parámetros, indicándose cualquier otro dato necesario mediante archivos de configuración. Estos comandos deben permitir ahorrar el mayor tiempo posible al desarrollador, de forma que éste pueda centrarse en las características únicas de la aplicación desarrolladas, en lugar de realizar tareas repetitivas, comunes a todos los proyectos similares desarrollados en la empresa u organización.

El objetivo fundamental es que estos comandos permitan crear la aplicación con sus diferentes módulos, conectarla a una base de datos, configurar su gestión de usuarios, e integrarla con otros sistemas de la empresa, todo ello en un único proyecto con diferentes módulos.

Descripción del sistema actual

En una empresa que desarrolla aplicaciones web con estas características se ha detectado que el desarrollo de las mismas haciendo uso de Roo es largo y complejo. En cada proyecto, similar al anterior, hay que realizar manualmente multitud de ajustes, fundamentalmente relacionados con la integración entre la aplicación desarrollada y los sistemas de la empresa, pero también con otros muchos aspectos, desde la introducción de la licencia hasta la conexión con la base de datos, pasando por, entre otros, la configuración del despliegue en un servidor web. Estas tareas consumían buena parte del tiempo total dedicado a cada proyecto.

El problema estaba agravado por el hecho de que la empresa ni tan siquiera había elaborado una lista de pasos a seguir, ya que sus sistemas evolucionaban constantemente, y los ajustes necesarios no eran exactamente los mismos para cada proyecto web. Por tanto, no sólo aumentaba el tiempo necesario para cada desarrollo, sino también la posibilidad de cometer errores.

Los desarrolladores de esta empresa identificaron el problema, y establecieron que la mejor solución sería el desarrollo de un add-on para Roo que automatizara las tareas repetitivas, pudiendo ajustarse su comportamiento a las necesidades de cada momento y cada proyecto mediante archivos de configuración (preferentemente), o parámetros en los comandos.

A partir de las necesidades de esta empresa, y por tanto también de otras empresas u organizaciones que desarrollen aplicaciones web mediante Java y Roo, surge Roo++.

Lista de problemas y necesidades

Problemas:

- Desarrollo de aplicaciones web mediante Spring Roo excesivamente complejo
- Multitud de tareas repetitivas
- Dificultad de adaptación de los proyectos a cambios en la plataforma software de la empresa, o en otro software de terceros utilizado
- Facilidad de cometer errores (especialmente olvidar alguno de los pasos a seguir)

Necesidades:

- Poder llevar a cabo las diferentes tareas mediante la ejecución de una serie de comandos sencillos
- Permitir la automatización del desarrollo de proyectos Roo con múltiples módulos
- Integración automática de la aplicación en desarrollo con los sistemas ya existentes en la empresa
- Simplificar la realización de otras tareas (gestión de usuarios, cambio de la apariencia de la interfaz de la aplicación web, conexión con la base de datos, documentación del proyecto ...)
- Flexibilidad, mediante el empleo de parámetros en los comandos, cuando sea necesario, y siempre que sea posible mediante archivos de configuración

Catálogo de requisitos

NOTA: los requisitos se muestran acompañados de su correspondiente prioridad, según sea la importancia de su cumplimiento. Los posibles valores de este campo son (de mayor a menor):

1. Muy Alta
2. Alta
3. Media
4. Baja

Requisitos funcionales

Id	Descripción	Prioridad
RF1 Gestión de proyectos		
RF 1.1	Se debe permitir la creación de proyectos Maven con múltiples módulos	Muy Alta
RF 1.2	Se debe permitir especificar el número de versión del proyecto a desarrollar	Media
RF 1.2.1	Si no se especifica el número de versión del proyecto a desarrollar, éste debe ser 1.0.BUILD-SNAPSHOT	Baja
RF 1.3	Tras la compilación del proyecto, la aplicación web ha de encontrarse en un archivo EAR listo para su despliegue en un servidor web	Muy Alta
RF 1.4	Se debe permitir la introducción de licencias en el proyecto en desarrollo	Alta
RF 1.5	Se debe permitir preparar un proyecto para su despliegue en los servidores web Tomcat y Jetty durante su desarrollo, con el fin de realizar pruebas	Muy Alta
RF 1.6	Se debe permitir la generación automática de los manuales de instalación y de usuario de la aplicación, mediante DocBook, a partir de una serie de archivos fuente	Alta
RF 2: Base de Datos		
RF 2.1	Se debe permitir configurar la conexión de la aplicación en desarrollo con bases de datos Oracle, MySQL, PostgreSQL y Microsoft SQL Server	Muy Alta
RF 2.1.1	Se debe permitir configurar el número de versión del sistema de gestión de bases de datos a utilizar, así como el nombre de su dialecto SQL	Media
RF 2.1.2	Si el usuario no indica el sistema de gestión de bases de datos a utilizar por la aplicación en desarrollo, éste ha de ser Oracle	Baja
RF 2.2	El usuario ha de indicar obligatoriamente el nombre de usuario y la contraseña necesarios para el acceso a la base de datos, no permitiéndose su almacenamiento en ningún archivo de configuración	Alta
RF 2.3	Se ha de permitir indicar en qué máquina y puerto se encuentra la base de datos que la aplicación en desarrollo ha de utilizar, además de su nombre	Alta
RF 3: Plataforma software de la organización		
RF 3.1	Se debe permitir configurar el proyecto en desarrollo para el uso de	Muy Alta

	utilidades proporcionadas por una plataforma software usada por la empresa	
RF 3.2	Se debe permitir la integración de la aplicación web en desarrollo dentro del portal web de la organización	Muy Alta
RF 3.3	Todo lo relativo a las particularidades de la plataforma software de la organización y a cualquier uso que se haga de ella ha de encontrarse establecido en archivos de configuración	Muy Alta
RF 4: Gestión de usuarios		
RF 4.1	Se debe permitir establecer un sistema de gestión de usuarios para la aplicación en desarrollo	Muy Alta
RF 4.1.1	Se podrá elegir entre el sistema de gestión de usuarios proporcionado por la plataforma software usada por la organización, y el proporcionado por las librerías de Spring Security, en combinación con un sistema CAS (Central Authentication Service)	Alta
RF 5: Interfaz de la aplicación web en desarrollo		
RF 5.1	Se debe permitir modificar la apariencia de la aplicación web (imágenes, hojas de estilos, ubicación de los elementos en la interfaz)	Alta
RF 5.2	Se debe permitir introducir la navegación mediante “migas de pan” ⁴ en la aplicación web	Media
RF 5.3	Se debe permitir establecer diferentes conjuntos de elementos en la aplicación web, que se mostrarán a cada usuario según su perfil y sus permisos	Media

Requisitos no funcionales

Id	Descripción	Prioridad
RNF 1	El sistema debe implementarse como un add-on de Spring Roo, integrándose en éste mediante el framework OSGi	Muy Alta
RNF 2	Todas las acciones deben proporcionarse mediante comandos Roo adicionales	Muy Alta
RNF 2.1	Ningún comando debe poder ser ejecutado antes de que lo hayan sido todos los comandos necesarios para su ejecución	Muy Alta
RNF 2.2	Si la ejecución de un comando en repetidas ocasiones puede acarrear problemas, ésta no ha de ser permitida	Alta
RNF 2.2.1	Si se considera que se ha de permitir repetidas ejecuciones de un comando a pesar de ello, este hecho debe estar debidamente documentado en el Manual de Usuario	Alta

⁴ Enlaces de *Inicio*, *Anterior*, *Siguiente*, etc.