



UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

SPRING ROO ADD-ONS PARA PROTOTIPADO RÁPIDO



JAVIER MENÉNDEZ ÁLVAREZ
JULIO 2014



UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

SPRING ROO ADD-ONS PARA PROTOTIPADO RÁPIDO

DOCUMENTO N° I

MEMORIA



JAVIER MENÉNDEZ ÁLVAREZ
JULIO 2014

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

Índice

Resumen.....	4
Summary.....	4
Objetivos.....	5
Alcance.....	6
Trabajo realizado.....	7
Estructura del trabajo realizado.....	8

Resumen

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 (por ejemplo, que la aplicación se encuentre integrada en el portal web de la organización), en caso de que sea necesaria. Se trata siempre del desarrollo de aplicaciones con una interfaz web, que hacen uso de base de datos, y que pueden hacer uso de gestión de usuarios, navegación mediante “migas de pan” (es decir, botones de “Anterior”, “Siguiente”, etc.), así como de otras funcionalidades que puedan proporcionar otros sistemas software ya existentes en la organización, siempre que se ajusten al tipo de plataforma utilizado por Roo++. De este modo, se simplifica mucho el desarrollo de este tipo de proyectos, reduciendo la necesidad de editar manualmente multitud de archivos y de ejecutar comandos complejos.

Concretamente, el software está dirigido al desarrollo de aplicaciones Java haciendo uso de la herramienta de desarrollo rápido Spring Roo y del sistema de automatización de compilación Apache Maven, asumiendo que el software de la empresa con el que interactuará con las aplicaciones web desarrolladas también está desarrollado con Java y Roo.

El software Roo++ se ha implementado mediante un add-on que se integra en Spring Roo mediante el sistema de gestión de módulos OSGi. De este modo, se puede hacer uso de toda la funcionalidad proporcionada por Roo, cuando se trata de obtener la máxima flexibilidad posible, y al mismo tiempo de las características que proporciona Roo++, cuando se trata de maximizar la facilidad y comodidad de uso y la agilidad.

Summary

This software's goal is complex web application development automation, usually in a business (or another organization type), and its integration with other of its computer systems (for example, application integration in organization's web portal), if that's necessary. It's always about web application development, using a database, and which can use user management systems, “breadcrumb” navigation (for example, “back” and “forward” buttons), and possibly other functionality from organization's existing computer systems, if they are conform with platform type used by Roo++. In this way, this type of project development is greatly simplified, without having to manually edit lots of files and running complex commands.

Specifically, this software is oriented towards Java application development using Spring Roo rapid development tool, and Apache Maven compilation automation system, assuming company's software development was also done using Java and Roo, so Roo++-developed web software can interact with it.

Roo++ software has been implemented through a Spring Roo add-on, integrated within it using OSGi module management system. Thereby, both all Roo functionality can be used, if you want maximum possible flexibility, and also Roo++, if the target is comfort, quickness and ease of use maximization.

Objetivos

El principal objetivo del software a desarrollar es la automatización del desarrollo de un tipo concreto de aplicaciones: empresariales, basadas en la web, que se pueden integrar en la plataforma software y la gestión de usuarios de la empresa, compuestas de múltiples módulos, y desarrolladas haciendo uso de la plataforma Java y de la herramienta de desarrollo rápido de aplicaciones Spring Roo.

El desarrollo de este tipo de aplicaciones presenta una serie de dificultades importantes: es necesario crear y editar manualmente multitud de archivos, crear cada uno de los módulos de cada proyecto del mismo tipo mediante comandos Roo, y programar y configurar manualmente aspectos tales como el acceso a la base de datos, la inclusión de dependencias entre módulos o respecto a componentes externos, o la gestión de usuarios, todo lo cual implica una dificultad y una duración en el tiempo bastante elevadas.

El software Roo++ permite, a través de un add-on que proporciona una serie de comandos adicionales a Spring Roo, automatizar gran parte del desarrollo de este tipo de proyectos. Con el fin de simplificar lo máximo posible el desarrollo de aplicaciones con Roo++, se ha decidido reducir en la medida de lo posible los parámetros en los comandos, llevándose a cabo el resto de ajustes a través de archivos de configuración, especialmente aquellos aspectos que no cambiarán habitualmente dentro de una organización que haga uso de Roo++.

Roo++ está orientado exclusivamente al desarrollo sencillo y rápido de un tipo muy concreto de aplicaciones; para cualquier otro tipo de característica requerida, tal como la creación de módulos adicionales para la aplicación, se continúa disponiendo de toda la flexibilidad proporcionada por Spring Roo.

Alcance

Este software permite automatizar las siguientes tareas en el desarrollo de una aplicación web empresarial:

- Crear un proyecto software con un conjunto de módulos predefinido (*core, docs, ear, package, scripts, web*).
- Incluir en el proyecto la/s licencia/s del software que se está desarrollando.
- Preparar la aplicación en desarrollo para su despliegue en los servidores web Tomcat y Jetty.
- Configurar la conexión de la aplicación en desarrollo con una base de datos.
- Configurar la gestión de usuarios de la aplicación en desarrollo.
- Integrar el software en desarrollo en el portal web y en la plataforma software de una organización.
- Establecer la apariencia y hojas de estilos utilizadas en la interfaz web del software que se está desarrollando.
- Definir las vistas que tendrá en la aplicación web cada usuario según su perfil y sus permisos.
- Configurar la aplicación web en desarrollo para la navegación mediante “migas de pan” (es decir, botones de “Anterior”, “Siguiente”, etc.).

Trabajo realizado

El desarrollo de este proyecto ha consistido en una serie de fases, tal y como es habitual en cualquier desarrollo de software: análisis, diseño, desarrollo, documentación y pruebas. Además de estas fases, también se han elaborado documentos de Planificación y presupuesto y Memoria.

La principal dificultad en el desarrollo de este proyecto ha sido la poca documentación existente sobre el desarrollo de add-ons para Spring Roo, siendo necesaria por ello mucha experimentación con el fin de adquirir experiencia. Otras dificultades surgidas durante el desarrollo han sido un *bug* en la versión por defecto de la librería Hibernate, siendo necesario forzar el uso de una versión anterior de dicha librería, el análisis detallado de las acciones que el mismo Spring Roo realiza cuando se ejecutan sus comandos para tareas tales como la creación de un módulo del proyecto (ya que Roo++ en algunos de sus comandos ha de realizar las mismas tareas que Roo realiza, aunque de otro modo), y la propia identificación de la lista de acciones que el software ha de realizar durante la ejecución de cada uno de los comandos, ya que ésta generalmente es extensa y no siempre trivial.

Debido a estas dificultades, se han producido las siguientes discrepancias entre la planificación temporal realizada y la duración real de las fases del proyecto:

Fase	Duración prevista	Duración real
Estudio de Viabilidad del Sistema	1 a 7 de marzo	1 a 7 de marzo
Análisis del Sistema de Información	8 a 25 de marzo	8 a 20 de marzo
Diseño del Sistema de Información	26 de marzo a 26 de abril	21 de marzo a 2 de mayo
Programación	29 de abril a 29 de mayo	3 a 31 de mayo
Documentación	30 de mayo a 21 de junio	3 a 25 de junio
Pruebas	24 de junio a 23 de julio	26 de junio a 19 de julio

Como se puede apreciar en esta tabla, los retrasos en algunas de las fases del proyecto no han supuesto un retraso en la fecha de finalización de éste, ya que dichos retrasos fueron compensados por la finalización antes de lo previsto de otras de las fases, consiguiéndose finalmente completar el proyecto a tiempo para la fecha final prevista.

Estructura del trabajo realizado

Los resultados del trabajo realizado se estructuran en los siguientes productos:

- Memoria
- Planificación y presupuesto
- Estudio de Viabilidad del Sistema (EVS)
- Análisis del Sistema de Información (ASI)
- Diseño del Sistema de Información (DSI)
- Documento de pruebas
- Código fuente
- Manual del usuario