

## GRAILS EN SISTEMAS DE CONTROL DE VERSIONES

**Ing. Yannier Estévez Dieppa**

Investigador del Centro Nacional de Sanidad Agropecuaria. San José de las Lajas, Mayabeque, Cuba.

[yannier@censa.edu.cu](mailto:yannier@censa.edu.cu)

### Resumen

El objetivo fundamental de este artículo es detallar la configuración necesaria en la estructura de carpetas que maneja el framework de desarrollo web Grails, cuando se trabaja en un ambiente colaborativo de desarrollo y de control de versiones. Se brinda una posible solución para eliminar los conflictos que surgen, cuando los ficheros del sistema sufren cambios debido a las modificaciones sobre cada copia local. A pesar de utilizar Subversion como sistema de control de versiones y TortoiseSVN como cliente de este para ilustrar las configuraciones, la solución que brinda este trabajo es genérica y puede ser tomada en cuenta sin importar las herramientas definidas por cada proyecto que se pretenda llevar a cabo.

**Palabras clave:** grails, ambiente colaborativo, control de versiones, subversion, tortoissvn.

### Introducción

El vertiginoso mejoramiento tecnológico que sufre el desarrollo de software ha llevado a las personas involucradas en él a brindar soluciones cada vez más eficientes y en un menor tiempo posible, para ello se hace necesaria la creación de nuevos y más flexibles ambientes de desarrollo integrado, así como más potentes herramientas. Grails es un framework de aplicaciones web libre que brinda un marco de trabajo altamente productivo. A su vez los sistemas informáticos cada día son más complejos por lo que se necesitan muchos desarrolladores programando simultáneamente para que el resultado esté disponible en el menor tiempo posible, esta es una de las razones por las cuales empieza a hablarse de desarrollo colaborativo en la industria del software. Por lo que se necesita manejar y controlar el código fuente y un gran conjunto de da-

tos a través del tiempo, permitiendo que muchas personas trabajen sobre el mismo sistema y que en cierto punto todo sea capaz de integrarse con facilidad.

Subversion es un sistema de control de versiones que permite lograr esto y puede acoplarse con muchos clientes como TortoiseSVN, pero se necesita tener en cuenta una serie de aspectos en la configuración cuando se desarrolla sobre Grails, ya que el framework genera y/o modifica automáticamente ficheros que deben estar incluidos en cada copia local y que los desarrolladores no tienen permisos para modificarlos en el repositorio cuando decidan subir a este parte de su trabajo.

Esta investigación especifica el conjunto de reglas necesarias para configurar una aplicación basada en Grails en un sistema de control de versiones de forma genérica (aunque se tome Subversion y TortoiseSVN como referencia) y está dirigida fundamentalmente a personas con experiencia en el trabajo con Grails y herramientas de control de versiones.

### **Grails 1.3.7**

Grails es un framework para aplicaciones web libre desarrollado sobre el lenguaje de programación Groovy (el cual a su vez se basa en la plataforma Java). Grails pretende ser un marco de trabajo altamente productivo siguiendo paradigmas tales como convención sobre configuración o no te repitas (DRY), proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador [1] [2].

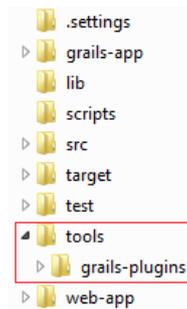
### **Configurando estructura básica de carpetas**

Grails provee una estructura básica de carpetas a la cual es necesario hacerle algunos cambios. Primero debemos indicarle al framework que cada uno de los plugins instalados debe residir dentro de la aplicación, así cada desarrollador al descargar su copia local ya tendrá en ella todos los plugins escogidos por el equipo de arquitectura y no tendrá que instalar ni configurar los mismos, ya que no es recomendable pues podría dañar la integridad del sistema.

Accediendo al fichero **Config.groovy** que se encuentra en **grails-app\conf** se incluye la siguiente línea de código:

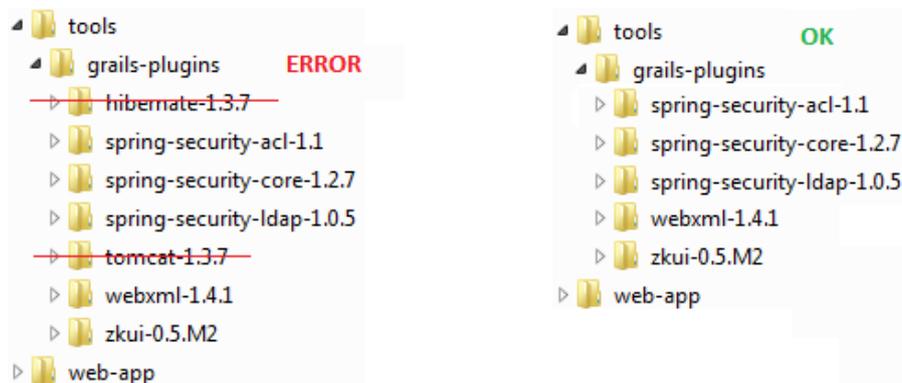
```
4 grails.project.plugins.dir = "tools/grails-plugins"
```

Esto garantiza que todos los plugins van a residir en una carpeta creada automáticamente llamada *tools*.



Se ejecuta la aplicación y se podrá ver que la carpeta **tools** ha sido creada. Se instala el resto de los plugins que se van a utilizar y los mismos automáticamente pasarán a residir dentro de la aplicación en la carpeta **tools/grails-plugins**.

Es muy importante que se eliminen de la carpeta **tools/grails-plugins** los plugins **hibernate** y **tomcat** ya que estos se generan y/o modifican automáticamente en cada copia local una vez que el desarrollador ejecute el comando **grails upgrade** y no es necesario que residan en el servidor para evitar conflictos en el momento que el programador decida subir su código.



Se chequea que las carpetas **target**, **web-app/META-INF**, **web-app/WEB-INF** estén completamente vacías, en caso de que contengan algún otro fichero y/o carpeta se procede a eliminarlos ya que el framework genera archivos propios para cada copia local cada vez que compila la aplicación evitando que entren en conflictos al momento de subir código y que residan innecesariamente en el sistema de control de versiones.

A partir de este momento es posible subir la aplicación al repositorio por primera vez utilizando un cliente (TortoiseSVN). Clic derecho sobre la carpeta padre de la aplicación, **SVN/Commit**.

Lo siguiente es indicarle al Subversion a través del cliente cuales de los archivos serán ignorados siempre que un desarrollador o el framework realicen cambios sobre los mismos.

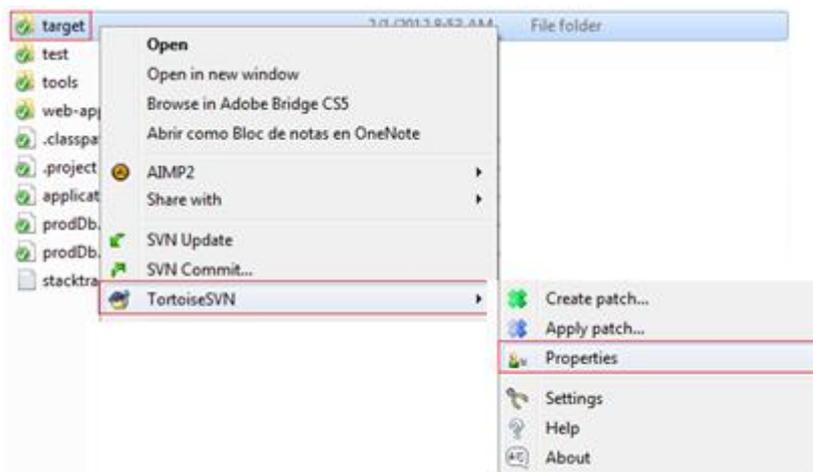
### Ignorando archivos

Es necesario que la aplicación ya se encuentre residiendo en un sistema de control de versiones (Subversion) y tener instalado un cliente (TortoiseSVN) en la estación de trabajo. Se descarga el repositorio (todo esto debe realizarse con una cuenta de administración) y se procede a ignorar los siguientes directorios.

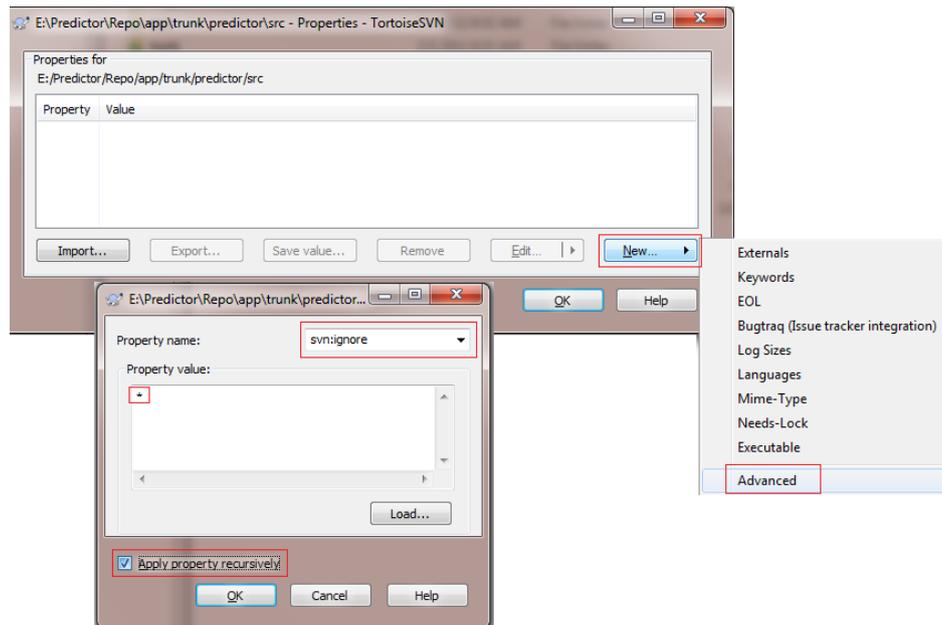
***tools, target, web-app/META-INF, web-app/WEB-INF***

El procedimiento descrito a continuación es el mismo para cada una de las carpetas que deben ser ignoradas.

Clic derecho sobre la carpeta, **TortoiseSVN/propiedades**



Clic en **Nuevo/Avanzada**, en la ventana emergente se escoge **svn:ignore** como nombre de la propiedad y por valor se introduce \* (asterisco), seleccionando **aplicar propiedad recursivamente**. Por último se aceptan todos los cambios.



De igual modo se crea una regla sobre la carpeta **web-app/images**, aplicando la siguiente propiedad recursivamente **thumbs.db Thumbs.db**, garantizando así que estos ficheros sean tomados en consideración en el momento de subir cambios.

Una vez terminado el procedimiento para todos los directorios necesarios se envían al repositorio todos los cambios. Clic derecho sobre la carpeta padre de la aplicación, **SVN/Commit** y de esta manera cada vez que el programador descargue el repositorio podrá programar sin problemas y en el momento de subir sus cambios no encontrará conflictos utilizando este poderoso framework que es Grails.

## Conclusiones

El artículo muestra las configuraciones necesarias para poner en marcha el desarrollo en ambiente colaborativo de aplicaciones basadas en el framework libre de aplicaciones web Grails. Brinda una solución genérica que debe ser tomada en cuenta independientemente del sistema de control de versiones y el cliente que se utilice en el equipo de desarrollo.

## Bibliografía

1. Brown, Jeff; Rocher, Graeme. 2009. "The Definitive Guide to Grails" (2ª edición), pp. 648, ISBN1590599950

2. Smith, Glen; Ledbrook, Peter. 2009. "Grails in Action" (1ª edición), pp. 520, ISBN1933988932
3. Dickinson, Jon. 2009. "Grails 1.1 Web Application Development" (1ª edición), pp. 328, ISBN1847196683
4. Abdul-Jawad, Bashar . 2008. "Groovy and Grails Recipes "(1ª edición), pp. 424, ISBN 143021600X
5. Fischer, Robert .2009. Grails Persistence with GORM and GSQL (1ª edición), pp. 125, ISBN 1430219262
6. M. Judd, Christopher; Nusairat, Joseph Faisal; Shingler, Jim. 2008. "Beginning Groovy and Grails: From Novice to Professional" (1ª edición) pp. 440, ISBN 1430210451
7. Rudolph, Jason. 2007. "Getting Started with Grails" (2ª edición), pp. 132, ISBN 143030782X
8. Grails. Search is over. 2012. <http://grails.org/>
9. Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato. "Control de versiones con Subversion".
10. TortoiseSVN.2012. <http://tortoisesvn.net/>

# SOCIEDAD DE LA INFORMACION

[www.sociedadelainformacion.com](http://www.sociedadelainformacion.com)

Edita:



Director: José Ángel Ruiz Felipe  
Jefe de publicaciones: Antero Soria Luján  
D.L.: AB 293-2001  
ISSN: 1578-326x