**Actualización de versión de Grails. De 2.1.1 a 2.3.6**

1. Descargar Nueva versión.

La podemos descargar desde el siguiente enlace.

<http://dist.springframework.org.s3.amazonaws.com/release/GRAILS/grails-2.3.6.zip>

1. Descomprimir el zip descargado en un *path* sin espacios.

Es ¡MUY IMPORTANTE! para el correcto funcionamiento de esta versión, que la ruta no contenga ningún espacio en blanco. Ruta de Ejemplo válida:

“C:\springsource\grails-2.3.6”

Ruta incorrecta:

“C:\Archivos de programa\springsource\grails-2.3.6”

Si esta contuviera algún espacio en blanco, en el momento de la compilación y/o ejecución de nuestros programas podríamos tener problemas para iniciar la JVM.

1. Cambio de variables de entorno.

Debemos modificar la variable de entorno GRAILS\_HOME para que apunte a nuestra nueva versión. El valor de esta variable ahora ha de ser.

“C:\springsource\grails-2.3.6”

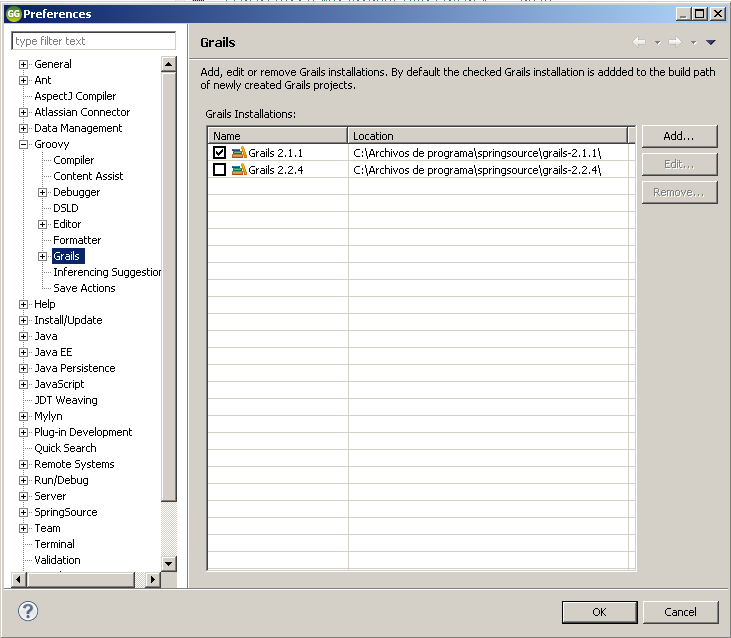
En este caso no debería ser necesario modificar la variable PATH siempre que tenga la referencia a:

“%GRAILS\_HOME%\bin”

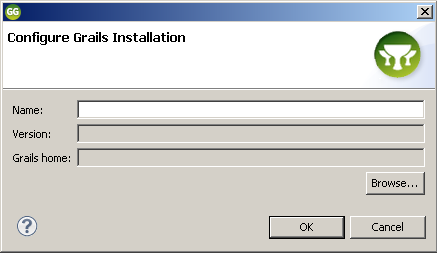
Llegados a este punto y puesto que se ha modificado una variable de sistema, para un óptimo funcionamiento se debería reiniciar el PC.

1. Agregar la versión a 2.3.6 a la configuración de nuestro *Workspace*.

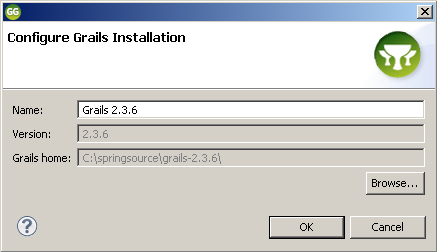
Para agregar a nuestro entorno esta nueva versión, con el GGTS abierto pulsaremos en el menú superior de la pantalla sobre la opción *Window*, y a continuación *Preferences* y nos aparecerá una pantalla similar a que vemos a continuación. En esta pantalla navegaremos por el menú de desplegables de la izquierda donde expandiremos *Groovy* i seleccionaremos la opción *Grails,* quedando como vemos en la imagen.



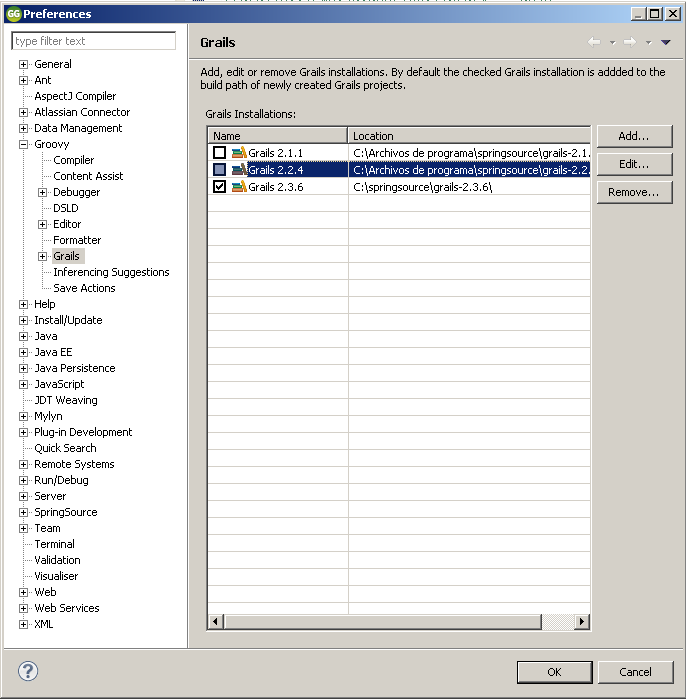
Pulsamos sobre el botón *Add* para incorporar la nueva versión. Al pulsar sobre este nos aparece una ventana que nos solicita el nombre, versión y la ruta de nuestro *Grails*.



Ahora pulsamos sobre el botón llamado *Browse*… y navegamos por la pantalla que aparece hasta la carpeta donde hemos descomprimido *Grails* 2.3.6. Seleccionamos la carpeta raíz de la versión, NO el *bin*. Después de aceptar quedará como lo vemos en la siguiente imagen.



Después de pulsar *OK*, vemos que nos ha aparecido una nueva línea de instalaciones de *Grails*, la *Grails 2.3.6*, esta será la que seleccionaremos tal y como podemos ver en la siguiente imagen. Y a continuación pulsamos *OK* para confirmar el cambio.



1. Adaptación de ficheros de configuración a versión 2.3.6

Para adaptar la configuración de nuestro proyecto a la versión 2.3.6 sólo hemos de modificar el fichero *BuildConfig.groovy*, *Config.groovy*, y *application.properties*.

*BuildConfig.groovy*

Para esta versión el cambio más significativo en nuestro proyecto es referente a la instalación y uso de los plugins, estos a partir de ahora deben ir indicados en el fichero *BuildConfig* y no se pueden instalar por el menú contextual.

Para ello añadiremos las siguientes líneas dentro de la llave de plugins de *grails.project.dependency.resolution:*

build ":tomcat:7.0.50.1"

compile ":scaffolding:2.0.2"

compile ':cache:1.1.1'

compile ":mail:1.0"

compile ":spring-security-core:1.2.7.3"

compile ":spring-security-cas:1.0.5"

runtime ":hibernate:3.6.10.8"

runtime ":database-migration:1.3.8"

runtime ":jquery:1.11.0"

runtime ":resources:1.2.1"

Como se puede ver en el anterior listado, podemos seguir usando los mismos plugins de *spring-segurity* y *mail*, minimizando así el impacto sobre la aplicación, aunque como veremos hay que modificar el repositorio donde encontraremos estos plugins, estos siguen siendo los mismos.

Existen otros plugins para los que si hemos de actualizar su versión, es son el de *tomcat*, *database-migration* e *hibérnate*, aunque permiten seguir trabajando como hasta ahora y no se tendrá que adaptar ni una sola línea de código.

Para que estas actualizaciones de plugins puedan ser correctamente descargadas tenemos que adaptar el fichero *BuildConfig* a los nuevos repositorios, para ello en la llave de *repositories* dentro de *grails.project.dependency.resolution*, descomentaremos o añadiremos las siguientes líneas.

mavenLocal()

mavenCentral()

mavenRepo 'http://repo.spring.io/milestone'

Para todas las versiones 2.3.X se necesita añadir un tipo especial de dependencia, lo haremos añadiendo la línea:

grails.project.dependency.resolver = "maven" // or ivy

Otra nueva característica que nos aporta esta versión de *Grails* y que nos obliga a modificar el fichero *BuildConfig* es referente a la gestión de la maquina virtual de Java permitiendo la gestión de la memoria, las conexiones máximas permitidas, etc., de este modo para cada uno de los entornos, hemos de especificar la horquilla que permitiremos en cada caso. Los valores que se copian a continuación y que servirán como ejemplo son los proporcionados en la web oficial.

grails.project.fork = [

// configure settings for compilation JVM, note that if you alter the Groovy version forked compilation is required

// compile: [maxMemory: 256, minMemory: 64, debug: false, maxPerm: 256, daemon:true],

// configure settings for the test-app JVM, uses the daemon by default

test: [maxMemory: 768, minMemory: 64, debug: false, maxPerm: 256, daemon:true],

// configure settings for the run-app JVM

run: [maxMemory: 768, minMemory: 64, debug: false, maxPerm: 256, forkReserve:false],

// configure settings for the run-war JVM

war: [maxMemory: 768, minMemory: 64, debug: false, maxPerm: 256, forkReserve:false],

// configure settings for the Console UI JVM

console: [maxMemory: 768, minMemory: 64, debug: false, maxPerm: 256]

]

*Config.groovy*

Las modificaciones que hay que realizar para este fichero son muy pocas. Sólo se requiere modificar (añadiendo propiedades) la configuración del Cas para que la validación se pueda realizar con éxito. Para cada uno de los entornos en los que nos conectaremos hemos de anexar las siguientes líneas a las que ya tenemos de *springsecurity*.

grails.plugins.springsecurity.cas.active = **true**

grails.plugins.springsecurity.cas.sendRenew = **false**

grails.plugins.springsecurity.cas.serverUrlEncoding = 'UTF-8'

grails.plugins.springsecurity.cas.key = 'grails-spring-security-cas'

grails.plugins.springsecurity.cas.artifactParameter = 'ticket'

grails.plugins.springsecurity.cas.serviceParameter = 'service'

grails.plugins.springsecurity.cas.filterProcessesUrl = '/j\_spring\_cas\_security\_check'

*application.properties*

Hemos de modificar nuestro fichero *application.properties* para indicar que estamos usando una nueva versión de *Grails* y los plugins que debe hacer servir. Este fichero debería quedar después de la modificación de la siguiente forma.

app.grails.version=2.3.6

app.name=SIG

app.version=0.1

plugins.export=1.5

plugins.hibernate=3.6.10.8

plugins.mail=1.0

plugins.spring-security-cas=1.0.5

plugins.spring-security-core=1.2.7.3

plugins.webxml=1.4.1

Siguiendo todos los pasos mencionados nuestro proyecto ya está preparado para poder ser ejecutado con esta versión de *Grails*.

**Anexo 1 BuildConfig.groovy**

grails.servlet.version = "2.5"

grails.project.**class**.dir = "target/classes"

grails.project.test.**class**.dir = "target/test-classes"

grails.project.test.reports.dir = "target/test-reports"

grails.project.target.level = 1.6

grails.project.source.level = 1.6

grails.project.war.file = "target/${appName}.war"

grails.project.fork = [

// configure settings for compilation JVM, note that if you alter the Groovy version forked compilation is required

// compile: [maxMemory: 256, minMemory: 64, debug: false, maxPerm: 256, daemon:true],

// configure settings for the test-app JVM, uses the daemon by default

test: [maxMemory: 768, minMemory: 64, debug: **false**, maxPerm: 256, daemon:**true**],

// configure settings for the run-app JVM

run: [maxMemory: 768, minMemory: 64, debug: **false**, maxPerm: 256, forkReserve:**false**],

// configure settings for the run-war JVM

war: [maxMemory: 768, minMemory: 64, debug: **false**, maxPerm: 256, forkReserve:**false**],

// configure settings for the Console UI JVM

console: [maxMemory: 768, minMemory: 64, debug: **false**, maxPerm: 256]

]

grails.project.dependency.resolver = "maven" // or ivy

grails.project.dependency.resolution = {

// inherit Grails' default dependencies

inherits("global") {

}

log "warn"

repositories {

inherits **true**

grailsPlugins()

grailsHome()

grailsCentral()

// from public Maven repositories

mavenLocal()

mavenCentral()

mavenRepo 'http://repo.spring.io/milestone'

}

dependencies {

// specify dependencies here under either 'build', 'compile', 'runtime', 'test' or 'provided' scopes eg. runtime 'mysql:mysql-connector-java:5.1.13'

}

plugins {

build ":tomcat:7.0.50.1"

// plugins for the compile step

compile ":scaffolding:2.0.2"

compile ':cache:1.1.1'

compile ":mail:1.0"

compile ":spring-security-core:1.2.7.3"

compile ":spring-security-cas:1.0.5"

// plugins needed at runtime but not for compilation

runtime ":hibernate:3.6.10.8" // or ":hibernate4:4.3.1.1"

runtime ":database-migration:1.3.8"

runtime ":jquery:1.11.0"

runtime ":resources:1.2.1"

}

}

CREAR PLUGIN E IMPORTARLO

- creas un proyecto nuevo de tipo plugin

- creas o copias las clases de dominio

- ejecutas:

grails package-plugin

- en el proyecto que quieres el plug-in ejecutas:

install-plugin D:\grails-sigris-domain-0.1.zip (path y nombre del plugin empaquetados)