

Configurando Spring tool suite

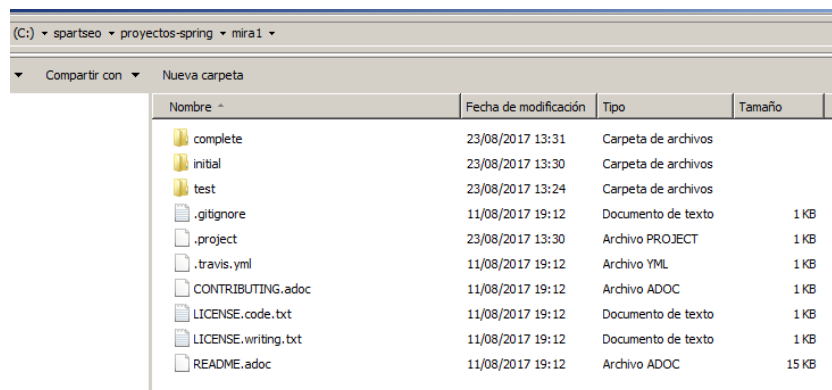
Instalamos SPRING TOOL SUITE que es una versión personalizada de ECLIPSE para poder trabajar mucho mas sencillo sin tener que configurar n-mil PLUGINS de ECLIPSE.

<https://spring.io/tools>

Nos bajamos el ZIP que hay en la demo inicial de SPRING BOOT.

<https://spring.io/guides/gs/spring-boot/>

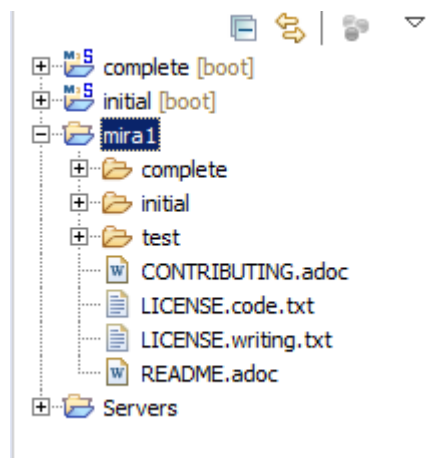
Lo descomprimos y lo dejamos en un directorio donde tendremos todos nuestros proyectos de SPRING.



(C:) > spartseo > proyectos-spring > mira1

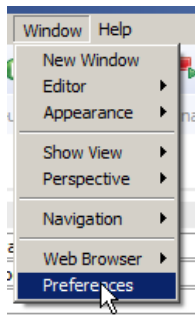
Compartir con Nueva carpeta

Nombre	Fecha de modificación	Tipo	Tamaño
complete	23/08/2017 13:31	Carpeta de archivos	
initial	23/08/2017 13:30	Carpeta de archivos	
test	23/08/2017 13:24	Carpeta de archivos	
.gitignore	11/08/2017 19:12	Documento de texto	1 KB
.project	23/08/2017 13:30	Archivo PROJECT	1 KB
.travis.yml	11/08/2017 19:12	Archivo YML	1 KB
CONTRIBUTING.adoc	11/08/2017 19:12	Archivo ADOC	1 KB
LICENSE.code.txt	11/08/2017 19:12	Documento de texto	1 KB
LICENSE.writing.txt	11/08/2017 19:12	Documento de texto	1 KB
README.adoc	11/08/2017 19:12	Archivo ADOC	15 KB

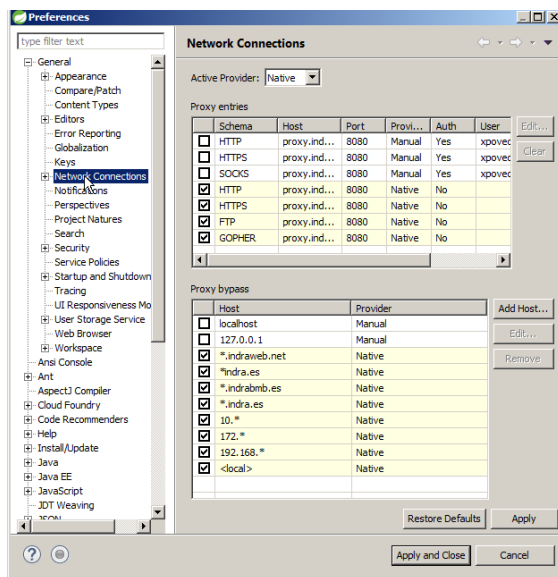


Si estamos en EMPRESA tendremos que preparar el proxy.

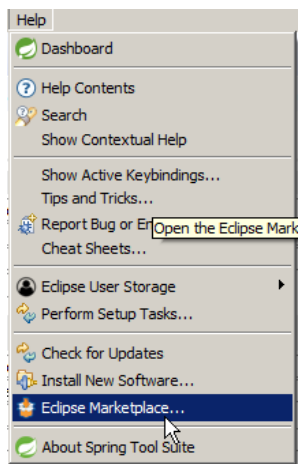
Primero tendremos que dejar como “Native” el proxy de STS.



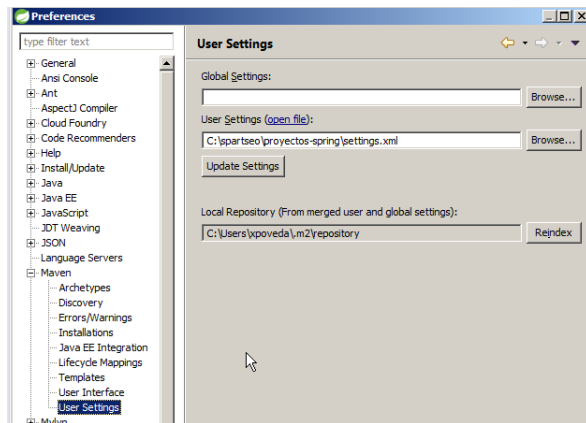
Nos cogerá la configuración directa que hay en la configuración de red del sistema operativo.



Comprobaremos que la salida es correcta yendo al MARKETPLACE, comprobándose así que la conectividad de eclipse es la correcta.



Por otro lado hemos de configurar el settings.xml de MAVEN para que también pueda salir por el proxy ya que no esta directamente enlazado con la configuración de ECLIPSE.



```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
    https://maven.apache.org/xsd/settings-1.0.0.xsd">

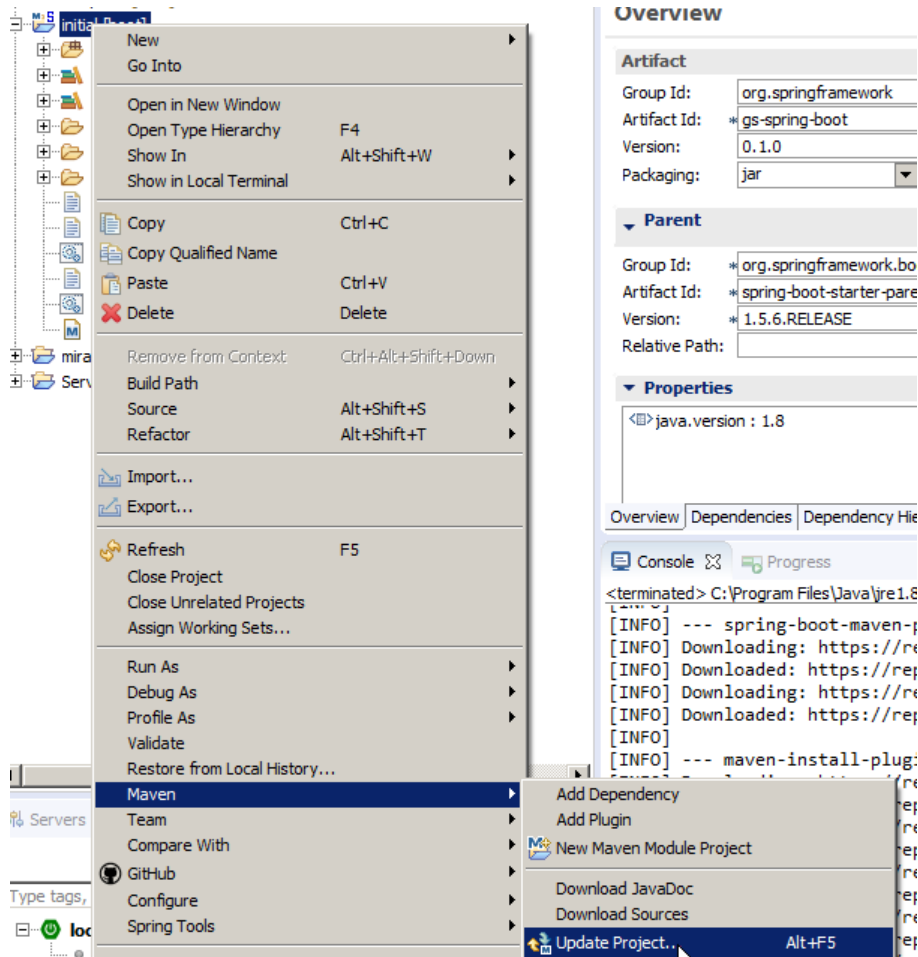
<!-- proxies
  | This is a list of proxies which can be used on this machine to connect to the
  | network.
  | Unless otherwise specified (by system property or command-line switch), the first
  | proxy
  | specification in this list marked as active will be used.
  |-->
<proxies>
  <proxy>
    <id>proxy1</id>
    <active>true</active>
    <protocol>http</protocol>
    <username>xpoveda@empresa.es</username>
    <password>MIPASSWORD</password>
    <host>proxy.empresa.es</host>
    <port>8080</port>
    <nonProxyHosts>local.net|some.host.com</nonProxyHosts>
  </proxy>

  <proxy>
    <id>proxy2</id>
    <active>true</active>
    <protocol>https</protocol>
    <username>xpoveda@empresa.es</username>
    <password>MIPASSWORD</password>
    <host>proxy.empresa.es</host>
    <port>8080</port>
    <nonProxyHosts>local.net|some.host.com</nonProxyHosts>
  </proxy>

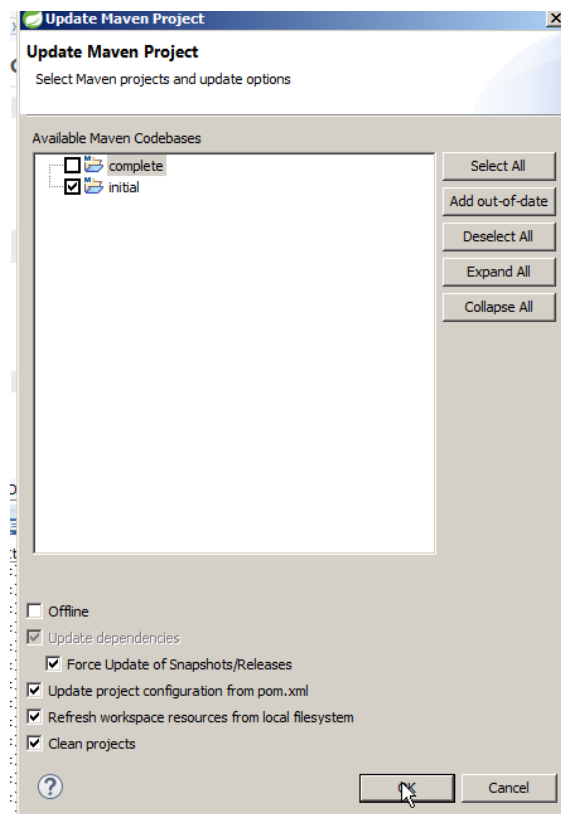
</proxies>
</settings>
```

Con esto ya se podrá empezar a bajar los ARTIFACTS que son las dependencias que manejará MAVEN en formato JAR.

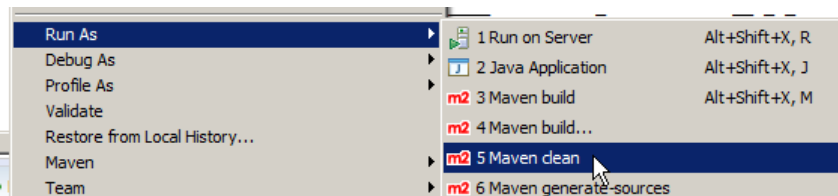
Si tenemos algún problema de compilado nos lo dirá el sistema, podemos hacer una actualización del proyecto para que coja los últimos elementos de MAVEN y borre memoria interna.



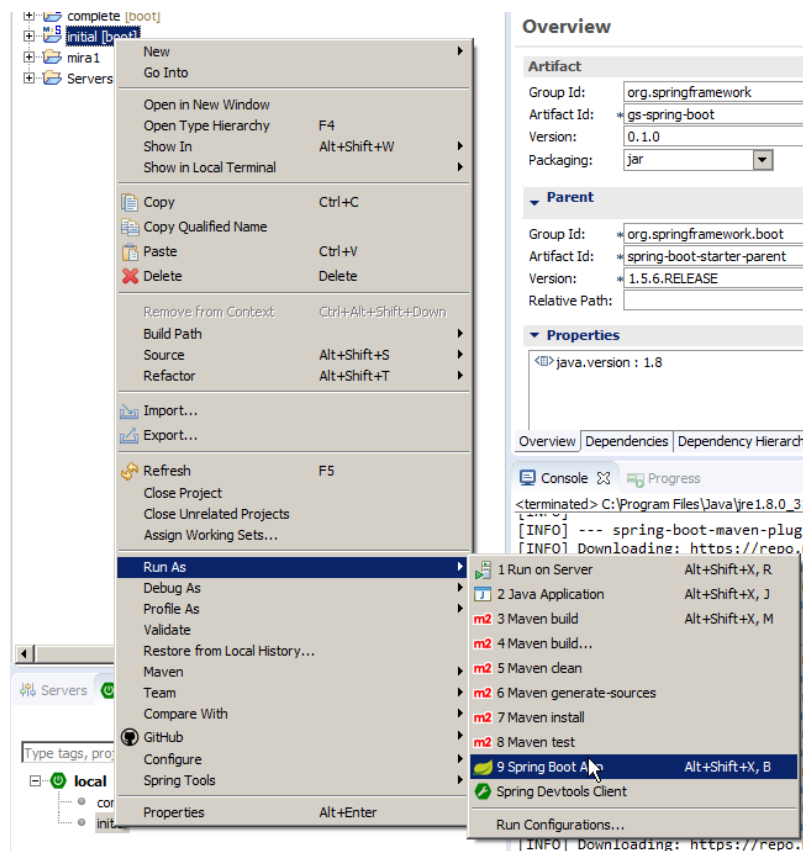
Marcando sobretodo el “Force update of Snapshots releases”.



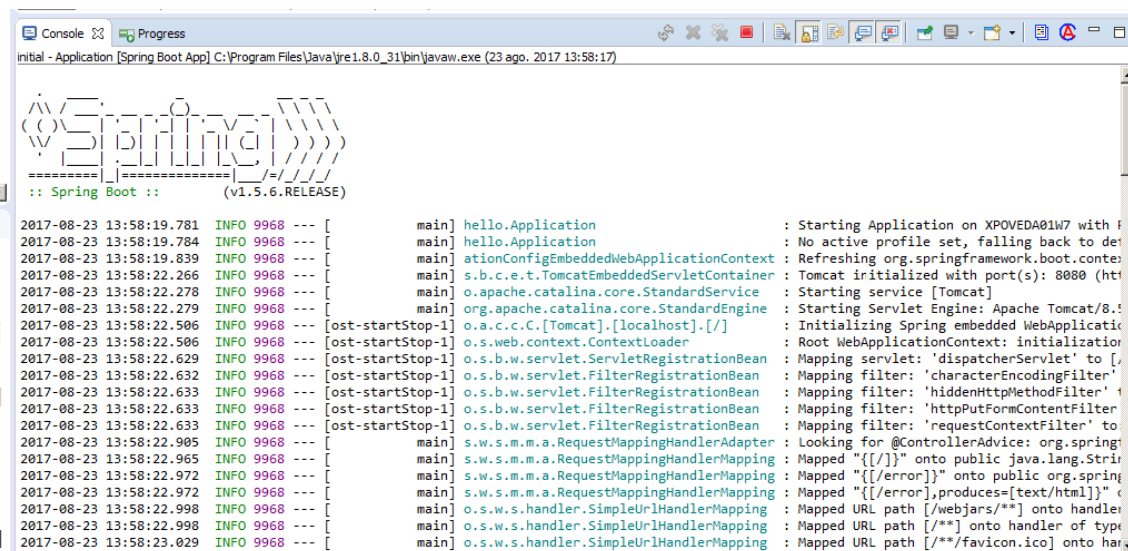
También podemos hacer un “maven clean” del proyecto con MAVEN directamente por si tuviéramos cualquier problema con la compilación que realiza eclipse.



Cuando ya el proyecto esté en disposición de lanzarlo (no haya marcas rojas en los fuentes ni en ningún otro elemento del proyecto) lo podemos hacer con un “run as spring boot app”.



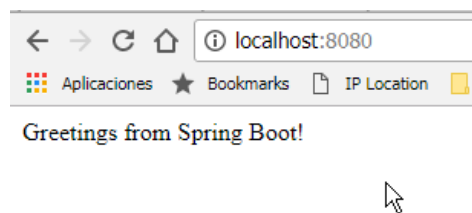
Que veremos en la consola.



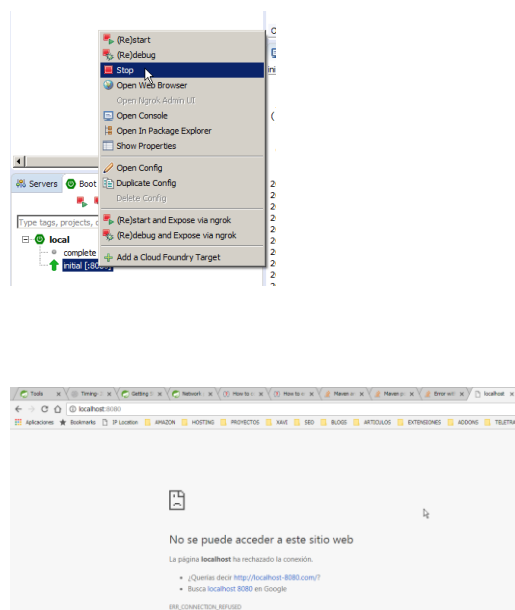
En el BOOT DASHBOARD vemos como hay un servidor a la escucha en el puerto 8080.



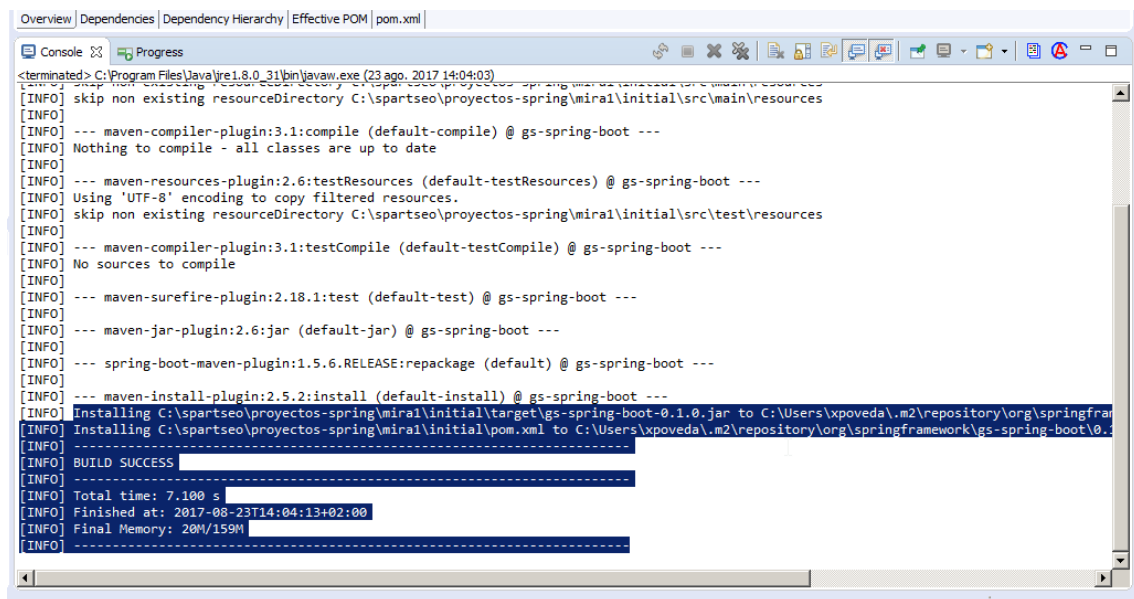
Y podemos ver que funciona.



Si lo paramos veremos que no va.



Ese proceso java lo encapsulamos en un JAR para poderlo publicar posteriormente de una forma más sencilla (con contenedores DOCKER por ejemplo) mediante el modulo JAR que nos generará el “maven install”.



```
<terminated> C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (23 ago. 2017 14:04:03)
[INFO] skip non existing resourceDirectory C:\spartseo\proyectos-spring\miral\initial\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ gs-spring-boot ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ gs-spring-boot ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\spartseo\proyectos-spring\miral\initial\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ gs-spring-boot ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.18.1:test (default-test) @ gs-spring-boot ---
[INFO]
[INFO] --- maven-jar-plugin:2.6:jar (default-jar) @ gs-spring-boot ---
[INFO]
[INFO] --- spring-boot-maven-plugin:1.5.6.RELEASE:repackage (default) @ gs-spring-boot ---
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ gs-spring-boot ---
[INFO] Installing C:\spartseo\proyectos-spring\miral\initial\target\gs-spring-boot-0.1.0.jar to C:\Users\xpoveda\.m2\repository\org\springframework\gs-spring-boot\0.1.0\gs-spring-boot-0.1.0.jar
[INFO] Installing C:\spartseo\proyectos-spring\miral\initial\pom.xml to C:\Users\xpoveda\.m2\repository\org\springframework\gs-spring-boot\0.1.0\gs-spring-boot-0.1.0.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 7.100 s
[INFO] Finished at: 2017-08-23T14:04:13+02:00
[INFO] Final Memory: 20M/159M
```

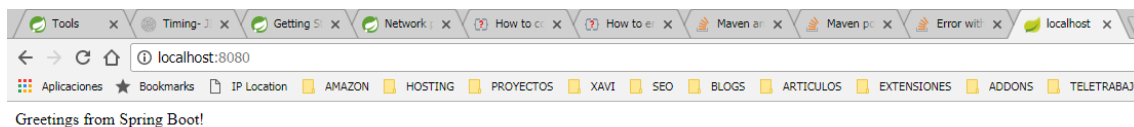
Que podremos ejecutar directamente desde sistema con “java -jar”



```
C:\Users\xpoveda\.m2\repository\org\springframework\gs-spring-boot\0.1.0>java -jar gs-spring-boot-0.1.0.jar
Spring
=====
:: Spring Boot ::
(v1.5.6.RELEASE)

2017-08-23 14:17:41.552 INFO 11488 --- [main] hello.Application : Starting Application v0.1.0 on XP0VEDA001W7 with PID 11488 (C:\Users\xpoveda\.m2\repository\org\springframework\gs-spring-boot\0.1.0\gs-spring-boot-0.1.0.jar started by xpoveda in C:\Users\xpoveda\.m2\repository\org\springframework\gs-spring-boot\0.1.0)
2017-08-23 14:17:41.572 INFO 11488 --- [main] hello.Application : No active profile set, falling back to default profiles: default
2017-08-23 14:17:41.883 INFO 11488 --- [main] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext
307951: startup date [Wed Aug 23 14:17:41 CEST 2017]; root of context hierarchy
```

Vemos que vuelve a ir.



Ese JAR es el que DOCKERIZAREMOS.