**Apuntes sobre migración a Java 13**

1. Instalar openJDK 13
2. Compilar JRE

Ejecutar el siguiente comando en la carpeta *bin* de la instalación del JDK:

* **MacOS**

./jlink --output jre-13.0.2 --compress=2 --no-header-files --no-man-pages --module-path ../jmods --add-modules java.base,java.compiler,java.datatransfer,java.desktop,java.instrument,java.logging,java.management,java.management.rmi,java.naming,java.net.http,java.prefs,java.rmi,java.scripting,java.se,java.security.jgss,java.security.sasl,java.smartcardio,java.sql,java.sql.rowset,java.transaction.xa,java.xml,java.xml.crypto,jdk.unsupported

* **Windows**

.\jlink --output jre-13.0.2 --compress=2 --no-header-files --no-man-pages --module-path ../jmods --add-modules java.base,java.compiler,java.datatransfer,java.desktop,java.instrument,java.logging,java.management,java.management.rmi,java.naming,java.net.http,java.prefs,java.rmi,java.scripting,java.se,java.security.jgss,java.security.sasl,java.smartcardio,java.sql,java.sql.rowset,java.transaction.xa,java.xml,java.xml.crypto,jdk.unsupported

* **Linux**

./jlink --output jre-13.0.2 --compress=2 --no-header-files --no-man-pages --module-path ../jmods --add-modules java.base,java.compiler,java.datatransfer,java.desktop,java.instrument,java.logging,java.management,java.management.rmi,java.naming,java.net.http,java.prefs,java.rmi,java.scripting,java.se,java.security.jgss,java.security.sasl,java.smartcardio,java.sql,java.sql.rowset,java.transaction.xa,java.xml,java.xml.crypto,jdk.unsupported

Se creará una carpeta con el nombre jre-13.0.2

1. Configurar servidores
   1. **Wildfly 11**

- Seguir las instrucciones descritas en el documento “Configuración Wildfly 11”.

- Reemplazar fichero “standalone.xml” en “standalone/configuration”

- Editar archivo bin/standalone.conf y definir la ruta a la carpeta jre-13.0.2:

JAVA\_HOME=//Aquí se pone la ruta a la carpeta jre-13.0.2

* 1. **Tomcat 8**

Las carpetas y/o archivos para agregar al nuevo servidor son:

Desde el empaquetado apache-tomcat-8.5.23

* Carpeta *conector*
* Carpeta *derby*
* De la carpeta *lib*:
  + derby.jar
  + derbyLocale\_cs.jar
  + derbyLocale\_de\_DE.jar
  + derbyLocale\_es.jar
  + derbyLocale\_fr.jar
  + derbyLocale\_hu.jar
  + derbyLocale\_it.jar
  + derbyLocale\_ja\_JP.jar
  + derbyLocale\_ko\_KR.jar
  + derbyLocale\_pl.jar
  + derbyLocale\_pt\_BR.jar
  + derbyLocale\_ru.jar
  + derbyLocale\_zh\_CN.jar
  + derbyLocale\_zh\_TW.jar
  + Derbynet.jar
* De la carpeta *conf:* archivos agesic\_v2.0.keystore, connector-web.properties, connector.properties
* De la carpeta bin

Crear un archivo (si existe modificarlo) con el nombre *setenv.sh* y copiar lo siguiente:

export CATALINA\_OPTS="$CATALINA\_OPTS -Dfile.encoding=UTF-8 -Dderby.system.home=$CATALINA\_HOME/derby -DuploadFolder=$CATALINA\_HOME/connector -Dconnector.integration.configLocation=$CATALINA\_HOME/conf -Dconnector.web.configLocation=$CATALINA\_HOME/conf -Dconnector.web.log4jLocation=$CATALINA\_HOME/logs -Dconnector.integration.log4jLocation=$CATALINA\_HOME/logs"

export JRE\_HOME=//Aquí se pone la ruta a la carpeta jre-13.0.2

* Dar permisos de ejecución a los archivos catalina.sh y setenv.sh

chmod +x catalina.sh

chmod +x setenv.sh

* En la carpeta *webapps* van los ficheros .war, connector-runtime.war y connector-web.war

1. Compilar con gradle 6.0 o posterior (A partir de la versión 6 es compatible con Java 13)

Abrir una consola en la raíz del proyecto y ejecutar:

gradle build -x pmdTest -x pmdMain -x test -x pmdIntegrationTest