# **Portada**

Número de grupo: E3.08

Url al repositorio: <https://github.com/antsermen/Acme-Toolkits-E3.08.git>

Miembros (+ emails):

- José Ramón Arias Expósito ([jrae1112jrae@gmail.com](mailto:rae1112jrae@gmail.co)),

- Manuel Carnero Vergel ([mancarver1@alum.us.es](mailto:mancarver1@alum.us.es)),

- Juan Carlos Moreno Pérez ([morenoperezjuancarlos@gmail.com](mailto:morenoperezjuancarlos@gmail.com)),

- Pablo Santos Pérez ([pablo.santos.232001@gmail.com](mailto:pablo.santos.232001@gmail.com)),

- David Sabugueiro Troya ([dsabugueiro14@gmail.com](mailto:dsabugueiro14@gmail.com)),

- Antonio Roberto Serrano Mena ([robertoserranomena@gmail.com](mailto:robertoserranomena@gmail.com))

Fecha: 31/05/2022

# **Tabla de contenidos**

Contenido

[**1.** **Portada** 1](#_Toc104914580)

[**2.** **Tabla de contenidos** 2](#_Toc104914581)

[**3.** **Resumen ejecutivo** 3](#_Toc104914582)

[**4.** **Tabla de revisión** 4](#_Toc104914583)

[**5.** **Introducción** 4](#_Toc104914584)

[**6.** **Contenidos** 5](#_Toc104914585)

[**7.** **Conclusiones** 7](#_Toc104914589)

[**8.** **Bibliografía** 7](#_Toc104914591)

# **Resumen ejecutivo**

En este documento podemos encontrarnos con 8 apartados: una portada, que contiene el número de grupo, la url al repositorio en github, los miembros (junto a sus emails) y la fecha de creación del documento; una tabla de contenidos, que sería un índice típicamente conocido para un documento; un resumen ejecutivo, en el que se resumen los contenidos de cada apartado del documento; una tabla de revisión, formada por los datos de las revisiones realizadas sobre el documento (número de revisión, fecha y descripción); una introducción, que indica lo que se encuentra en el cuerpo del documento, es decir, un resumen de lo que es el plugin con el que hicimos el componente .jar del proyecto “Acme Framework 22.7”; los contenidos, que recoge el cuerpo del documento, y concretamente contendrá los pasos seguidos para la elaboración de la tarea que solicitaba el empaquetado de “Acme Framework” dentro de nuestro proyecto finalizado en el pasado entregable como un componente .jar independiente; unas conclusiones, extraídas a partir de la finalización del documento y una bibliografía, que contiene las fuentes que nos sirvieron de apoyo para el documento (en caso de no encontrarse ninguna, aparecerá “Intencionalmente en blanco.”).

# **Tabla de revisión**

|  |  |  |
| --- | --- | --- |
| **N.º** | **Fecha** | **Descripción** |
| 1 | 26/05/2022 | Creación de documento y relleno parcial |
| 2 | 30/05/2022 | Añadiendo las horas respectivas y haciendo los últimos retoques |
| 3 | 31/05/2022 | Preparación para entrega |

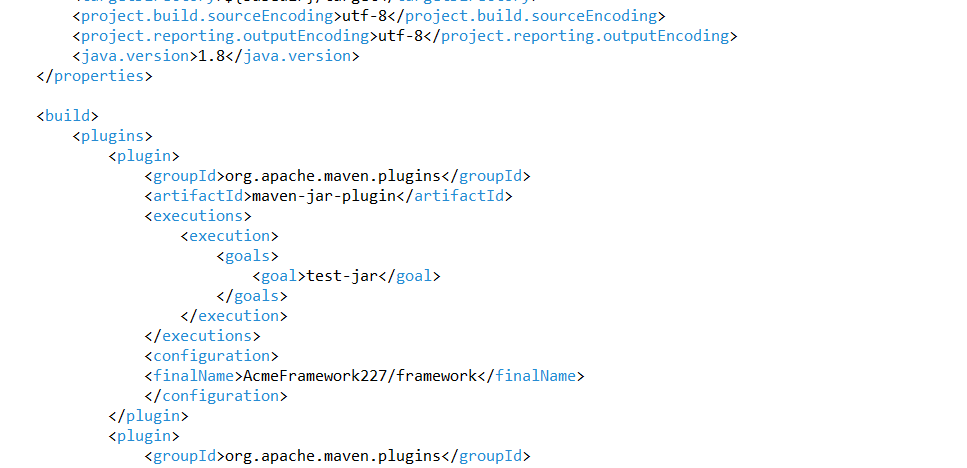
# **Introducción**

El plugin empleado para la creación del jar es el “Apache Maven JAR Plugin” que permite tanto crear el componente del proyecto o de las clases de prueba del proyecto, más conocidas como “tests”. En nuestro caso, empaquetamos “Acme Framework 22.7” completo como un componente .jar, tanto el proyecto como sus tests. Y aunque en el enlace de la bibliografía se puede encontrar toda clase de información acerca de este plugin, nosotros detallaremos a continuación los pasos seguidos para el empaquetado del proyecto completo (nótese que habría ciertas diferencias a la hora de empaquetar el proyecto completo o únicamente las clases de prueba o tests). Actualmente el plugin se encuentra en la versión 3.2.2 y requiere de la versión de Java 1.7, además también se pueden realizar los pasos que detallaremos en el siguiente apartado mediante la consola de comandos, con comandos como “mvn package”.

# **Contenidos**

# El primer paso sería localizar el fichero “pom” del proyecto, que lo podemos encontrar al final del proyecto:

# Para posteriormente añadir el plugin dentro de la etiqueta plugins (“<plugins>”) de dicho fichero “pom.xml”:



# 

# Se le puede añadir la etiqueta version (“<version>”) justo debajo de la etiqueta artifactId, sin embargo, a nosotros nos daba un aviso (o warning) sobre el uso de esta ya que estaría machacando la versión que ya utiliza el framework. Una vez añadido el plugin, tendríamos que actualizar la información del proyecto, para ello, haríamos click derecho sobre “Acme-Framework-22.7” y pulsaríamos sobre “Maven” y sobre la opción “Update Project” (también se podría ir a la pestaña “Problems” de eclipse y solucionar el error que nos aparece haciendo click derecho sobre este y a continuación sobre “Quick Fix”, lo cual nos mandaría a una ventana que solucionaría el error haciendo un update al proyecto). Una vez tenemos la información actualizada, haríamos click derecho sobre el proyecto nuevamente y esta vez iríamos a la opción “Run As” para seleccionar “5 Maven build…”. En la pestaña que nos aparecería tras realizar la acción anterior, tendríamos que escribir “package” en el campo “Goals” (y dejar las demás opciones por defecto) y pulsar sobre “Apply” y “Run” desde esa misma pestaña. Esto haría un build del proyecto y añadiría los archivos .jar a la carpeta “AcmeFramework227” que se crearía justo después de hacer la build del proyecto dentro de la carpeta “target” del proyecto. Los archivos .jar creados serían “project.jar” y “project-test.jar”.

Al importar el framework como un .jar, decidimos realizarlo de igual manera que implementamos el detector de Spam del entregable anterior. Primero configuramos el Build Path y luego añadimos como librería el JAR. Sin embargo nos hemos topado con una cantidad enorme de errores que impedían cualquier funcionalidad del proyecto. Por eso hemos decidido simplemente añadir el JAR dentro del proyecto pero no lo hemos insertado como una librería JAR.

# 

# **Conclusiones**

# A pesar de la información proporcionada por internet y la adquirida en esta asignatura, no fuimos capaces de implementar completamente el framework dentro de nuestro proyecto, sin embargo sí somos capaces de utilizar el plugin de maven para extraer un proyecto como un componente .jar, lo que nos otorga ciertos conocimientos en cuanto al software que nos podrán ser útiles de cara a un futuro. Y aunque no hayamos podido realizar la implementación completa dentro de nuestro proyecto, estamos contentos con los avances conseguidos, ya que seguramente aprendamos a realizar estos pasos con mayor detalle a medida que vayamos adentrándonos en el mundo del código profesional o en su defecto, a resolver los errores que nos encontremos.

# **Bibliografía**

“Apache Maven JAR Plugin”: <https://maven.apache.org/plugins/maven-jar-plugin/index.html>