Manual para la generación del instalador Windows de AutoFirma

Índice de contenidos

[1 Introducción 3](#_Toc472669382)

[2 Requisitos mínimos 4](#_Toc472669383)

[3 Proceso de generación 5](#_Toc472669384)

[3.1 Preparación del entorno 5](#_Toc472669385)

[3.2 Generación del instalador .exe 5](#_Toc472669386)

[3.3 Generación del instalador .msi 10](#_Toc472669387)

[3.4 Firma de los instaladores 12](#_Toc472669388)

[4 Cambio de versión 13](#_Toc472669389)

# Introducción

En el proceso de instalación de una aplicación Windows es necesaria la existencia de un ejecutable que contenga una serie de instrucciones para ejecutar el programa y un conjunto de recursos como librerías, imágenes u otros ejecutables que lo hagan funcionar correctamente.

En este caso la aplicación AutoFirma cuenta con dos tipos de instaladores:

* **Fichero .exe**: Su función es la de ejecutar una serie de instrucciones en un sistema Windows. Este tipo de archivos se utilizan para instalar, arrancar programas y ejecutar aplicaciones de forma manual.
  + El instalador .exe de AutoFirma está pensado para ser utilizado por los usuarios de la aplicación.
* **Fichero .msi:** Permite automatizar una instalación en Windows, reduciendo al mínimo la intervención manual, ya que toda la información necesaria se encuentra en el propio fichero.
  + El instalador .msi de AutoFirma está pensado para ser utilizado por administradores de sistemas que deseen realizar despliegues masivos de la aplicación.

Este documento tiene como objetivo la explicación en detalle de los pasos necesarios para la generación de ambos ficheros.

Los instaladores Windows de AutoFirma incluyen la JRE con la que se va a ejecutar la aplicación en el equipo del usuario. Para abarcar los equipos de 32 y 64 bits, se creará una versión para cada una de las arquitecturas en donde la aplicación será la misma y cambiará únicamente algunas propiedades del empaquetado y la versión de Java con la cual se empaqueta.

Así pues, obtendremos 4 versiones del instalador de AutoFirma para Windows:

* AutoFirma\_32\_vX\_Y\_Z\_installer.exe
* AutoFirma\_64\_vX\_Y\_Z\_installer.exe
* AutoFirma\_32\_vX\_Y\_Z\_installer.msi
* AutoFirma\_64\_vX\_Y\_Z\_installer.msi

Los proyectos java utilizados (AutoFirma y AutoFirmaConfigurador) son los mismos para todos los instaladores.

# Requisitos mínimos

Sistema operativo:

* Windows XP SP3 con arquitectura 32 (x86) o 64 (x64) bits.

Adicionalmente para la construcción de los instaladores se hace uso de las siguientes herramientas:

* Java SE Runtime Environment 8 o 9 (32 y 64 bits) (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
  + JREs con la que se empaquetarán los instaladores.
* Launch4j ([https://sourceforge.net/projects/launch4j/files/launch4j-3/](https://sourceforge.net/projects/launch4j/files/launch4j-3/3.8/))
  + Herramienta para la envoltura de los JAR de AutoFirma a modo de EXE.
* WiX (<http://wixtoolset.org/releases/>)
  + Herramienta para la generación del instalador MSI.
* NSIS (<http://nsis.sourceforge.net/Download>)
  + Herramienta para la generación de los EXE de instalación.
* NSIS NsProcess Plugin (<http://nsis.sourceforge.net/NsProcess_plugin>)
  + Plugin de NSIS para permitir la detección y manipulación de procesos durante la instalación.
  + Deberán localizarse los ficheros .nsh y .dll según la siguiente jerarquía:

**NSIS 2.x:**

NSIS/  
├── Includes/  
│ └── nsProcess.nsh  
└── Plugins/  
    └── nsProcess.dll

**NSIS 3.x:**

NSIS/  
├── Includes/  
│ └── nsProcess.nsh  
└── Plugins/  
    ├── x86-ansi/  
    │ └── nsProcess.dll  
    └── x86-unicode/  
        └── nsProcess.dll

# Preparación del entorno

Para la generación del instalador de AutoFirma será necesario disponer del software listado en el apartado de requisitos mínimos.

Los distintos instaladores de AutoFirma se construirán en base a la estructura de directorios y los recursos alojados en el módulo **afirma-simple-installler** del proyecto **clienteafirma** en GitHub (<https://github.com/ctt-gob-es/clienteafirma/tree/master/afirma-simple-installer>).

## Estructura de ficheros

Los directorios y ficheros de utilidad que encontraremos dentro de este módulo son:

* AutoFirma32
  + Directorio en el que se generarán los ejecutables de 32bits.
* AutoFirma64
  + Directorio en el que se generarán los ejecutables de 64bits.
* Certificados
  + Directorio con los certificados SSL que deben considerarse de confianza.
* jar
  + Directorio en el que guardar los JAR de AutoFirma.
* java32
  + Directorio en el que guardar la JRE que se dese empaquetar con AutoFirma 32bits.
* java64
  + Directorio en el que guardar la JRE que se dese empaquetar con AutoFirma 64bits.
* AutoFirma\_32\_installer.wxs
  + Fichero para la construcción del instalador MSI de 32bits.
* AutoFirma\_64\_installer.wxs
  + Fichero para la construcción del instalador MSI de 64bits.
* AutoFirma\_launch4J\_Configurador\_project\_32.xml
  + Fichero para el empaquetado EXE 32bits de AutoFirmaConfigurador.
* AutoFirma\_launch4J\_Configurador\_project\_64.xml
  + Fichero para el empaquetado EXE 64bits de AutoFirmaConfigurador.
* AutoFirma\_launch4J\_project\_32.xml
  + Fichero para el empaquetado EXE 32bits de AutoFirma.
* AutoFirma\_launch4J\_project\_64.xml
  + Fichero para el empaquetado EXE 64bits de AutoFirma.
* AutoFirma\_NSIS\_project\_EXE\_32.nsi
  + Fichero para la generación del instalador EXE de 32bits.
* AutoFirma\_NSIS\_project\_EXE\_64.nsi
  + Fichero para la generación del instalador EXE de 64bits.
* AutoFirma\_NSIS\_project\_MSI\_32.nsi
  + Fichero para generar el EXE de instalación a partir del cual se construye el instalador MSI de 32bits.
* AutoFirma\_NSIS\_project\_MSI\_64.nsi
  + Fichero para generar el EXE de instalación a partir del cual se construye el instalador MSI de 64bits.
* AutoFirmaCommandLine\_launch4J\_project\_32.xml
  + Fichero para el empaquetado EXE 32bits de AutoFirma para línea de comandos.
* AutoFirmaCommandLine\_launch4J\_project\_64.xml
  + Fichero para el empaquetado EXE 64bits de AutoFirma para línea de comandos.
* AutoFirmaConfigurator.manifest
  + Fichero de permisos para que ejecución de AutoFirmaConfigurador con permisos de administrador.
* ic\_firmar.ico
  + Icono de la aplicación.
* ic\_head.bmp
  + Imagen de cabecera del instalador EXE.
* ic\_install.bmp
  + Imagen lateral del instalador EXE.
* ic\_launcher.ico
  + Icono del lanzador de la aplicación.
* licencia.txt
  + Información de la licencia del producto.
* no\_ejecutar\_x64.exe
  + Desinstalador de la aplicación para el MSI de 64bits.
* no\_ejecutar\_x86.exe
  + Desinstalador de la aplicación para el MSI de 32bits.
* splash.bmp
  + Imagen *splash* que se muestra al abrir la aplicación antes de que aparezca la primera pantalla.

## Configuración de las JRE empaquetadas

Las JRE con las que se empaquetarán los instaladores de AutoFirma en Windows no están incluidos en el directorio cargado en el repositorio de código. El desarrollador encargado de empaquetarlo deberá introducirlos en una copia local de esta estructura, si subir nunca su contenido al repositorio debido a su tamaño. Las instrucciones para la configuración de las JRE son las siguientes:

1. Copiar las JRE

* El usuario deberá copiar a los directorios “java32” y “java64” una JRE de 32 y 64 bits respectivamente. Los datos a copiar son el directorio “jre” con los ficheros de la JRE correspondiente.
* La estructura resultante sería:
  + java32
    - jre
      * *<ficheros\_jre>*
  + java64
    - jre
      * *<ficheros\_jre>*
* Podemos, por ejemplo, instalarnos una JRE en nuestro equipo y luego copiar el directorio “jre” del directorio de instalación en el directorio “java32” o “java64” según corresponda.
* Se puede utilizar una JRE 8 o superior, aunque se recomienda utilizar siempre la última disponible en el momento de empaquetar.
* Las JRE de 32 y 64 bits utilizadas pueden tener distinta versión.

1. Eliminar lo innecesario

* Opcionalmente, se pueden eliminar los ficheros de las JRE a utilizar que se sepan que son innecesarios para el correcto funcionamiento de la JVM durante la ejecución de AutoFirma.
* Esto sirve para reducir el tamaño del instalador final y debería hacerse única
* Por ejemplo, se pueden eliminar ficheros relativos al uso de JavaFX, ya que este API no se utiliza en AutoFirma y no son parte imprescindible de la JRE. Así pues, se eliminarían, entre otros, la extensión “jfxrt.jar” (17 Mb) y la biblioteca “jfxwebkit.dll” (42 Mb).

1. Instalar certificados SSL

* Para el correcto funcionamiento de AutoFirma frente a determinados servicios sobre SSL es necesario que se inserten en el almacén de confianza de las JRE los certificados de las autoridades susceptibles a utilizar por los servicios y que no estén reconocidas por defecto en Java.
* En el directorio “certificados” se han guardado los certificados de raíces e intermedios de varias autoridades españolas. Si se desean agregar otros certificados como de confianza para AutoFirma, se deberían copiar los correspondientes certificados a este directorio.
* Para insertar los certificados del directorio “certificados” en los almacenes de confianza de las dos JRE configuradas, se deberá ejecutar el script “insert\_cacerts.bat” del directorio certificados. Este script utilizará la herramienta “keytool” de las JRE para insertar todos los certificados del directorio “certificados” en el almacén de confianza (cacerts) de las mismas JRE.

## Configuración de la versión de AutoFirma

Cada nueva versión de AutoFirma puede requerir que se actualicen diversos recursos de los ficheros de instalación. Ejemplos claros son el número de versión y la fecha del *copyright*. Los textos susceptibles de actualizarse en cada nueva versión de AutoFirma se encuentran en los siguientes ficheros de la estructura de directorios:

* AutoFirma\_32\_installer.wxs
  + Número de versión de AutoFirma 32bits.
* AutoFirma\_64\_installer.wxs
  + Número de versión de AutoFirma 64bits.
* AutoFirma\_launch4J\_project\_32.xml
  + Número de versión del configurador de 32bits.
* AutoFirma\_launch4J\_project\_64.xml
  + Número de versión del configurador de 64bits.
* AutoFirmaCommandLine\_launch4J\_project\_32.xml
  + Número de versión de AutoFirma 32bits.
* AutoFirmaCommandLine\_launch4J\_project\_64.xml
  + Número de versión de AutoFirma 32bits.
* AutoFirma\_NSIS\_project\_EXE\_32.nsi
  + Número de versión y nombre del instalador EXE de 32bits.
* AutoFirma\_NSIS\_project\_EXE\_64.nsi
  + Número de versión y nombre del instalador EXE de 64bits.
* AutoFirma\_NSIS\_project\_MSI\_32.nsi
  + Número de versión de AutoFirma 32bits.
* AutoFirma\_NSIS\_project\_MSI\_64.nsi
  + Número de versión de AutoFirma 64bits.
* AutoFirma\_to\_msi\_32.bat
  + Número de versión del nombre del instalador MSI de 32bits.
* AutoFirma\_to\_msi\_64.bat
  + Número de versión del nombre del instalador MSI de 64bits.
* licencia.txt
  + Texto de licencia, versión y año de *copyright*.
* splash.bmp
  + Año del *copyright*.

## Generación de los instaladores EXE

En primer lugar se compilan cada uno de los proyectos ejecutando en el directorio principal de cada uno de ellos el siguiente comando:

mvn clean install –Dmvn.skip.tests

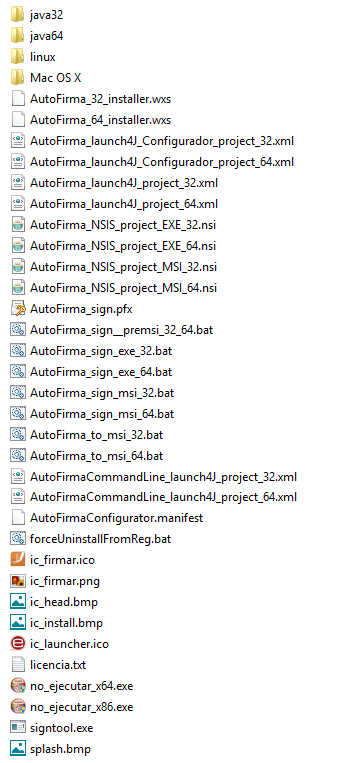
El orden de ejecutción debe ser:

* Proyecto clienteafirma-external
* Proyecto jmulticard
* Proyecto clienteafirma

Una vez hecho esto se podrá generar el JAR de los binarios que contendrá el componente instalador. Para ello, se deberá ejecutar el mismo comando desde los directorios de los módulos:

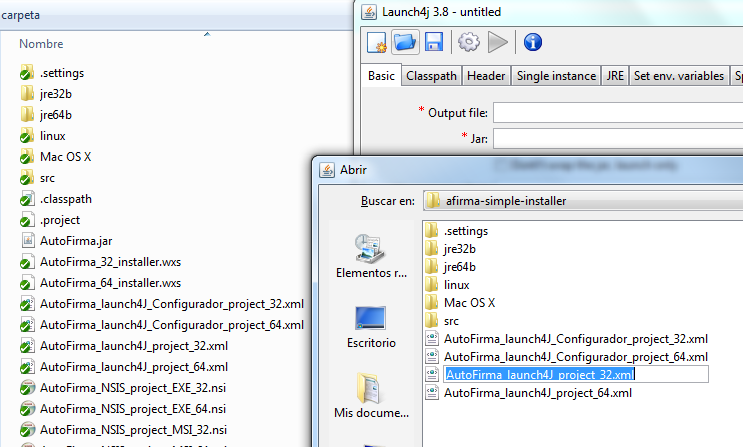
* **afirma-simple:** Genera afirma-ui-simpleafirma-X.X.X-SNAPSHOT.jar
* **afirma-ui-simple-configurator:** Genera afirma-ui-simpleafirma-configurator-X.X.X-SNAPSHOT.jar

Una vez generados estos JAR se renombrarán como “AutoFirma.jar” y “AutoFirmaConfigurador.jar”, respectivamente, y se copiarán al directorio del módulo **afirma-simple-installer**

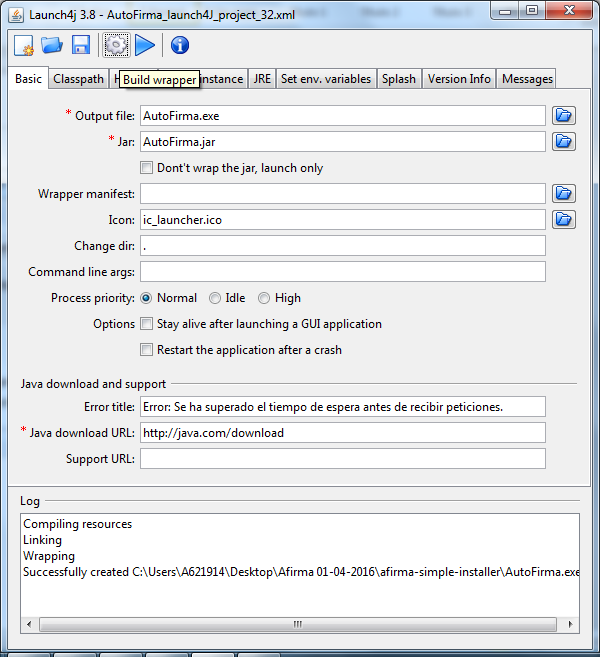


A continuación se generarán los distintos EXE de la aplicación a partir de estos archivos JAR. Para ello se hará uso de la aplicación Launch4j. Abriremos la aplicación, cargaremos uno de los ficheros de configuración y daremos la orden de construir el archivo EXE asociado.

Para ello, abriremos la aplicación desde el menú Inicio de Windows, se abre el fichero de configuración en cuestión y se hace pincha en “Build wrapper” (Figuras 1 y 2).



**Figura 1**



**Figura 2**

El proceso de construcción de los archivos EXE se deberá realizar con cada uno de los siguientes ficheros de configuración:

* AutoFirma\_launch4J\_project\_32.xml
* AutoFirmaCommandLine\_launch4J\_project\_32.xml
* AutoFirmaConfigurator\_launch4J\_project\_32.xml

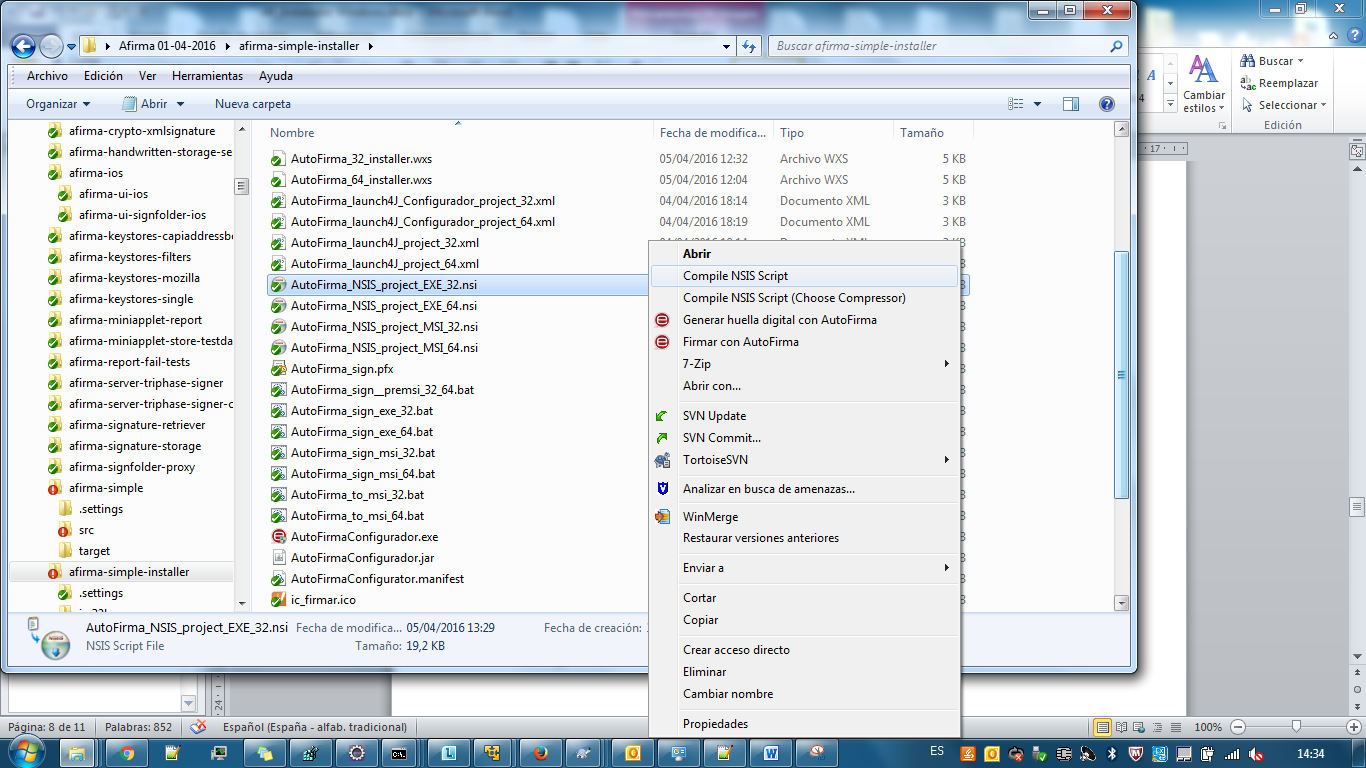
Los ficheros anteriores permiten generar el instalador de 32 bits de AutoFirma. En caso de desear empaquetar la versión de 64 bits, se utilizarán los ficheros “AutoFirma\_launch4J\_project\_64.xml” y “AutoFirmaCommandLine\_launch4J\_project\_64.xml” en lugar de los 2 primeros listados.

Como resultado se generarán los ficheros:

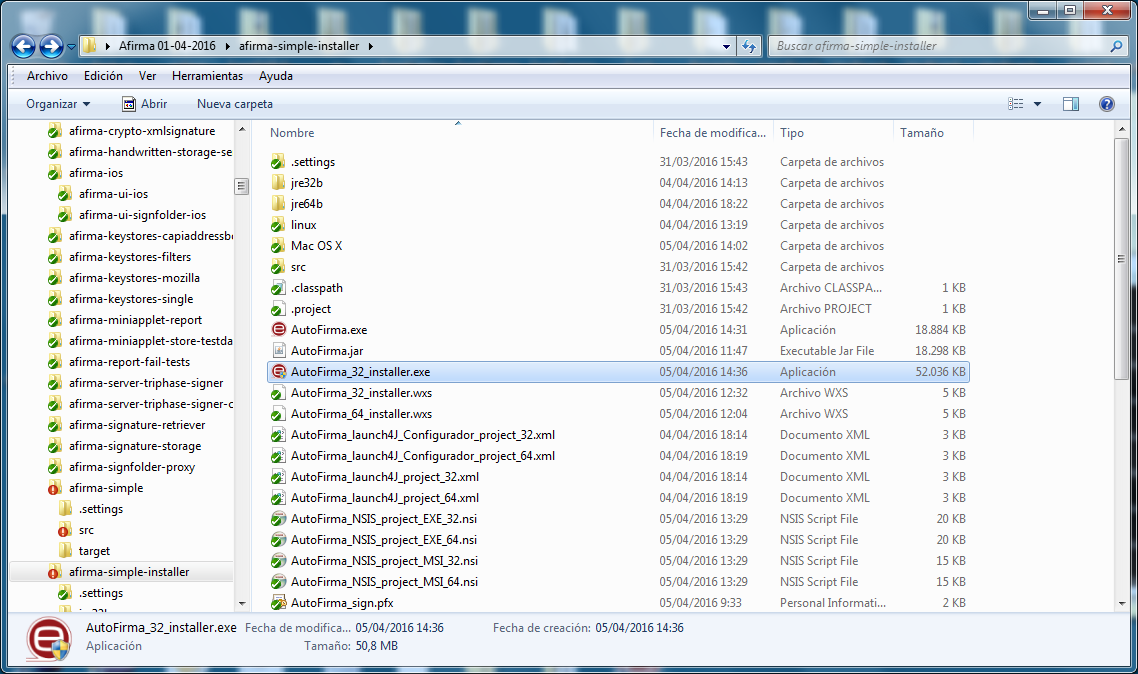
* AutoFirma.exe
* AutoFirmaCommandLine.exe
* AutoFirmaConfigurador.exe

Una vez disponemos de los ficheros ejecutables EXE se generará el instalador mediante la herramienta NSIS.

Para esto, se pincha con botón derecho en el fichero AutoFirma\_NSIS\_project\_EXE\_32 .nsi y pulsa en la opción “Compile NSIS Script” (Figura 3). Este proceso generará en el mismo directorio el instalador AutoAfirma\_32\_installer.exe (Figura 4).



**Figura 3**



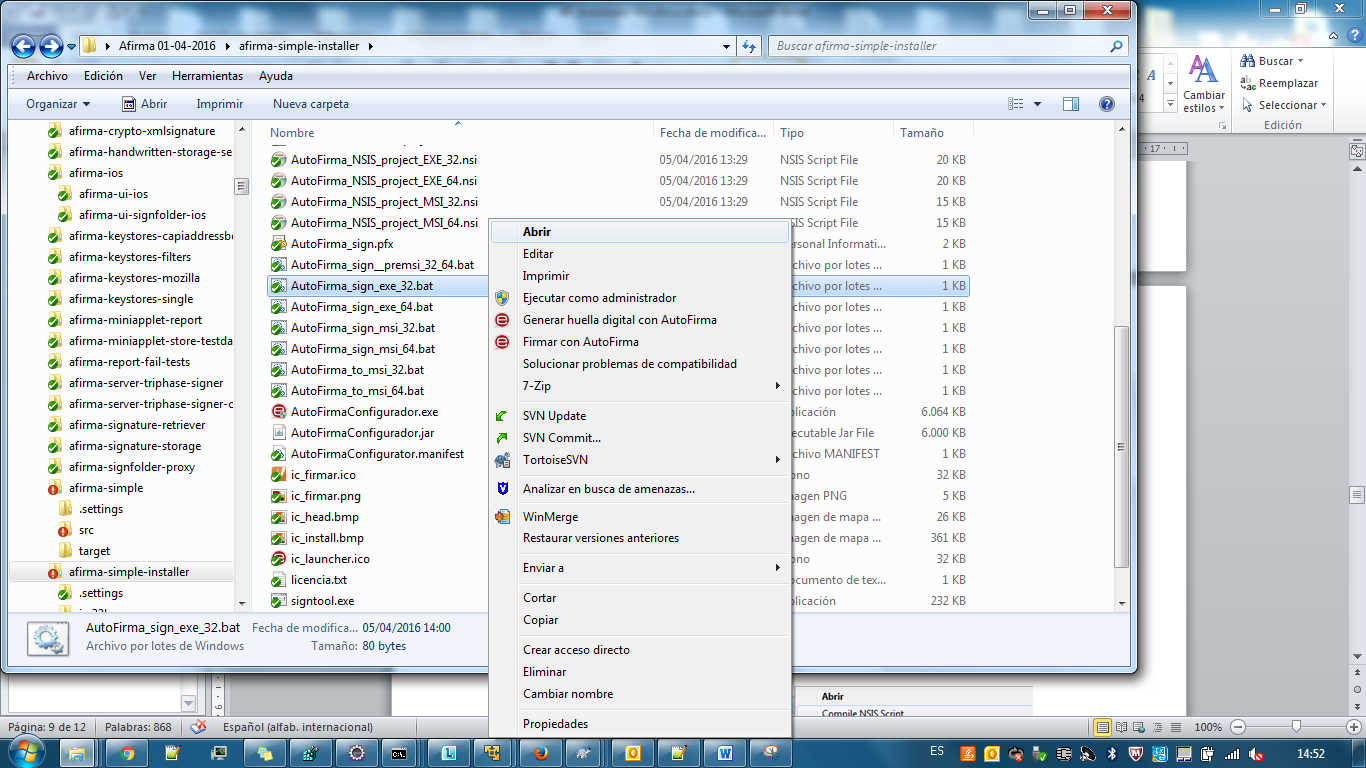
**Figura 4**

Para generar el instalador EXE de 64 bits, se operará de igual manera pero generando los ejecutables de 64 bits mediante los ficheros apropiados de Launch4J y haciendo uso del script AutoFirma\_NSIS\_project\_EXE\_64.nsi.

Si se desea generar un instalador con un número distinto de versión de AutoFirma, se puede modificar la siguiente línea del fichero NSIS.

**!define VERSION "X.X.X"**

En caso de querer firmar el EXE del instalador se ejecutará el script Afirma\_sign\_exe\_32.bat (o Afirma\_sign\_exe\_64.bat para el instalador de 64 bits). Este script hace uso de las claves de firma del almacén AutoFirma\_sign.pfx.

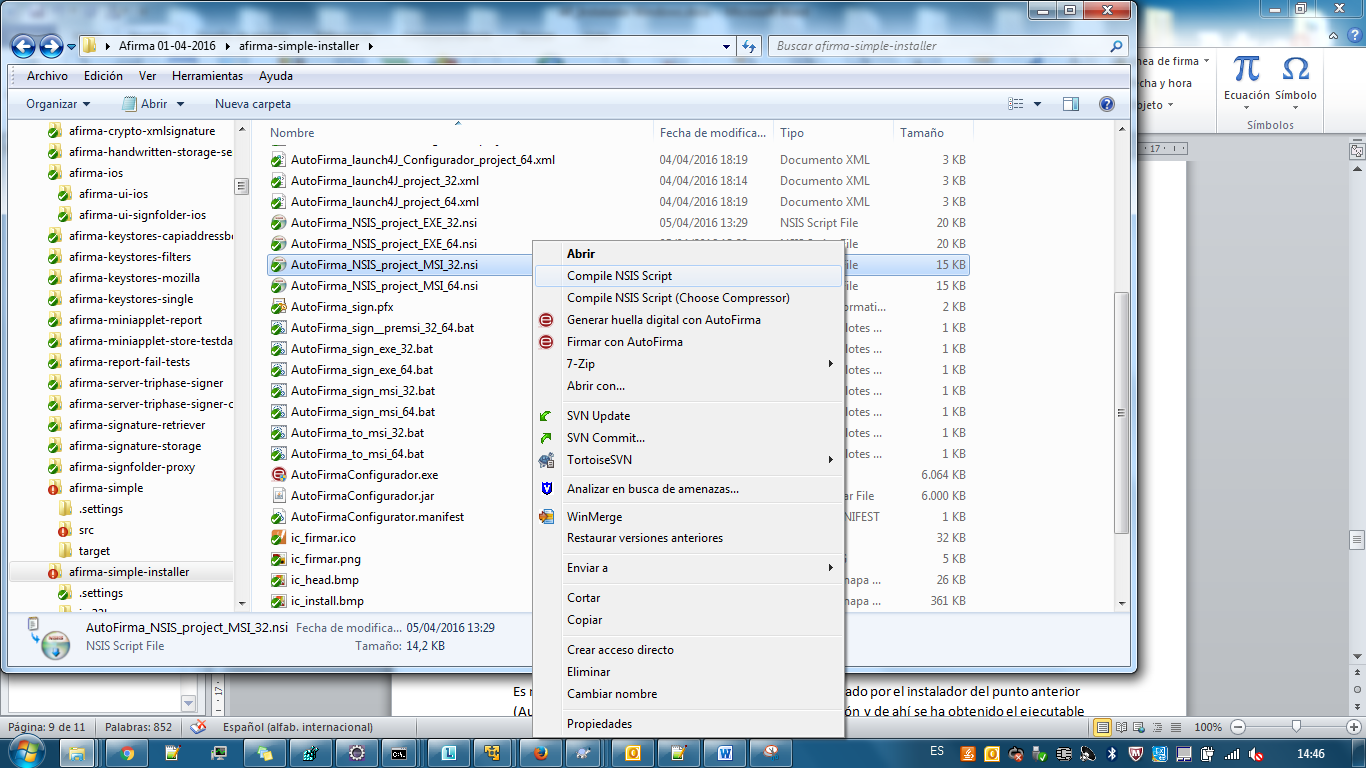


## Generación de los instaladores MSI

Para la generación del instalador MSI se hace uso de las herramientas de WiX que deben descargarse, instalarse y agregarse su ruta de binarios a la variable de entorno PATH. La ruta en cuestión para la instalación por defecto es C:\Program Files (x86)\WiX Toolset v3.11\bin.

El proceso de generación de los ejecutables EXE a partir de los archivos JAR se realizará igual que en el caso de los instaladores EXE, utilizando los archivos de configuración de Launch4J para la versión de 32 o 64 bits según se desee.

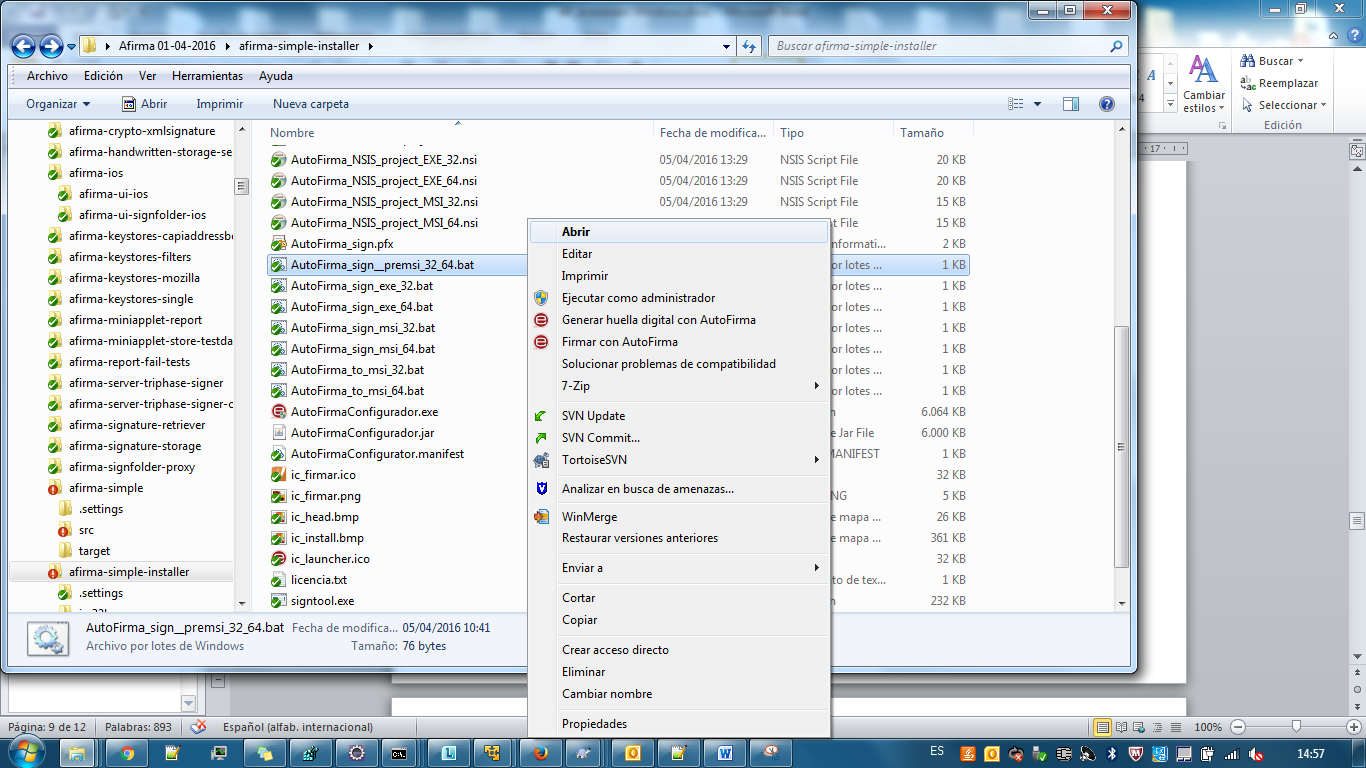
A continuación, se ejecutará el script NSIS AutoFirma\_NSIS\_project\_MSI\_32.nsi (o AutoFirma\_NSIS\_project\_MSI\_64.nsi para la versión de 64 bits).



Esto dará lugar al ejecutable “AutoFirmaGenerator.exe”.

A diferencia de los instaladores EXE generados anteriormente, este instalador no crea ni elimina información del registro, ni tampoco genera un ejecutable de desinstalación. Únicamente crea la estructura de directorios de la aplicación y genera e importa los certificados necesarios.

Este ejecutable puede ser firmado ejecutando el fichero AutoFirma\_sign\_\_premsi\_32\_64.bat (se ejecuta el mismo .bat para las versiones de 32 y 64 bits).



El instalador MSI se genera al ejecutar el script AutoFirma\_to\_msi\_32.bat (o AutoFirma\_to\_msi\_64.bat para el instalador de 64 bits). Este script utiliza los siguientes recursos para la generación del instalador:

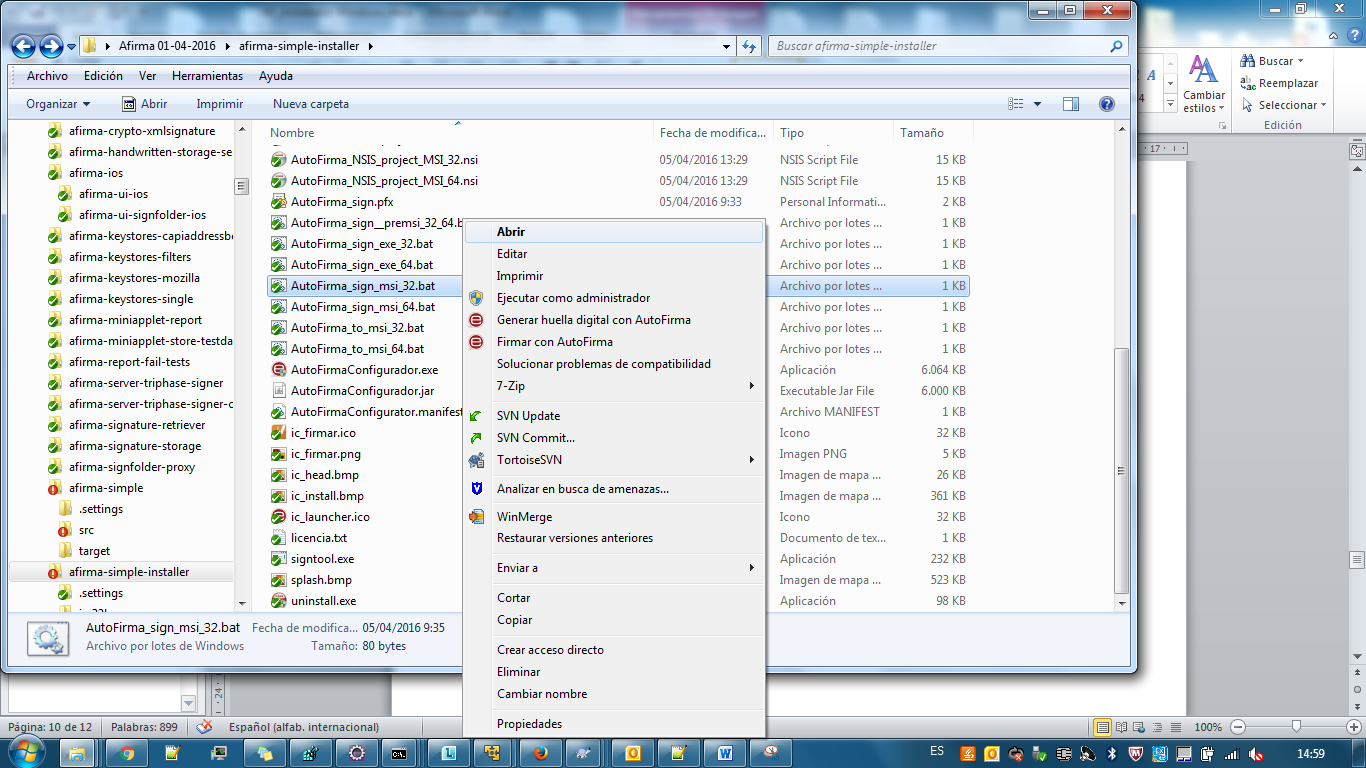
* AutoFirmaGenerator.exe: El pre-instalador generado anteriormente.
* no\_ejecutar\_x86.exe / no\_ejecutar\_x64.exe: Ejecutables para la desinstalación parcial de la aplicación. Se les ha asignado ese nombre para evitar que un usuario lo ejecute, dado que desinstalaría la aplicación pero no eliminaría su registro del panel de aplicaciones. Se encuentran ya en el directorio.
* AutoFirma\_32\_installer.wxs / AutoFirma\_64\_installer.wxs: Fichero de configuración que definen el proceso de generación del instalador MSI.
* Iconos de la aplicación

Es necesario destacar que los ejecutables de desinstalación han sido generados por el instalador EXE (AutoFirma\_32\_installer.exe / AutoFirma\_64\_installer.exe). Se modificó el script de desinstalación para que no mostrasen diálogos gráficos, se generó el instalador, se instaló en el equipo y se tomó el desinstalador. Esto se realizó para las versiones de 32 y 64 bits. Después se les cambió el nombre a no\_ejecutar\_x86.exe y no\_ejecutar\_x64.exe.

En caso de que en un futuro se realizasen cambios en el proceso de instalación/desinstalación, (AutoFirma\_NSIS\_project\_EXE\_32.nsi y AutoFirma\_NSIS\_project\_EXE\_64.nsi) será necesario generar de nuevo los desinstaladores del MSI y subirlos al repositorio.

El instalador MSI creado habrá aparecido en el directorio con nombre **AutoFirma\_32\_installer.msi** (o **AutoFirma\_64\_installer.msi** en el caso del instalador de 64 bits).

Este instalador puede ser firmado con AutoFirma\_sign\_msi\_32.bat o AutoFirma\_sign\_msi\_64.bat, según corresponda.



Por último la instalación del fichero .msi en una máquina puede llevarse a cabo abriendo directamente el fichero o mediante consola de manera silenciosa con la ejecución del siguiente comando:

msiexec /i "\_RUTA\_\AutoFirma\_installer.msi" /quiet

De la misma forma su desinstalación se llevaría a cabo de la siguiente manera:

msiexec /x "\_RUTA\_\AutoFirma\_installer.msi" /quiet

**NOTA:** La ejecución de los instaladores y desinstaladores MSI cerrará los navegadores abiertos del equipo para realizar la instalación y restaurará las pestañas al finalizarla, por lo que es conveniente avisar al usuario en caso de ser un administrador el que instala las aplicaciones remotamente en otros sistemas.

### Crear nuevos desinstaladores

## Firma de los instaladores

Los instaladores EXE y MSI pueden firmarse para garantizar la confianza de los usuarios en estos ejecutables. La firma puede realizarse mediante la herramiento SignTool de Microsoft y el siguiente comando:

signtool sign /f ALMACEN.p12 /p CONTASEÑA /a /fd SHA256 /tr http://sha256timestamp.ws.symantec.com/sha256/timestamp RUTA\_INSTALADOR

[Creative Commons License](#Licencia_Creative_Commons)

Esta obra está bajo una licencia [Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 Unported](#Licencia_Creative_Commons).