**Índice**

Información general del proyecto …………………………………………………………………………………………… 2

Exportar proyecto a un archivo Jar ………………………………………………………………………………………….. 4

Paso a producción del archivo Jar ………………………………………………………………………………………….. 7

**Información general del proyecto**

El proyecto **ProcesosMCH** está construido pensando en centralizar todas las tareas que se empiecen a hacer de ahora en adelante, en la clase Main se mapea la clase principal de la tarea que se quiere ejecutar, para esto se usa la librería Quartz, esta librería permite ejecutar las clases mapeadas a un determinado tiempo, el proyecto está dividido en tres secciones grandes a nivel de arquitectura de clases:  
1. **Tareas**

2. **Actividades**:

3. **Procesos**

La idea de dividir las clases en esta estructura, es asignar responsabilidades específicas, para no revolver todo con todo y así evitar el código **Espagueti**. A continuación se muestra la estructura de package que existe a la fecha **29/09/2016**



Descripción de packages:

**com.mch.actividades**: En este package van las clases que cumplen varias funciones o que llaman varias tareas Ej: Una clase que invoque un procedimiento almacenado.

**com.mch.bean**: En este package van las clases que cumpla ciertas normas con los nombres de sus propiedades y métodos, estas normas son:

* Debe tener un constructor sin parámetros
* Debe tener todos los atributos con el modificador de acceso prívate
* Cada atributo debe tener un método Set y Get
* Esta clase debe ser Serializable

**com.mch.configuraciones**: En este package van las clases que se encargan de configurar alguna acción Ej: conexión a una base de datos, configuración de un pool de instancias o de conexiones etc.

**com.mch.excepciones**: En este package van las clases que personalizan la excepciones que pueda generar algún código.

**com.mch.main**: En este package va una única clase, esta clase es la raíz principal del proyecto, acá es donde se mapean las clases principales de las tareas que contenga el proyecto, en este caso se deberían mapear las clases que están en el package **PROCESOS**.

**com.mch.procesos:** En este package van las clases que son la tarea que se quiere ejecutar, es un conjunto de actividades que se ejecutan para realizar la tarea global que se quiere hacer.

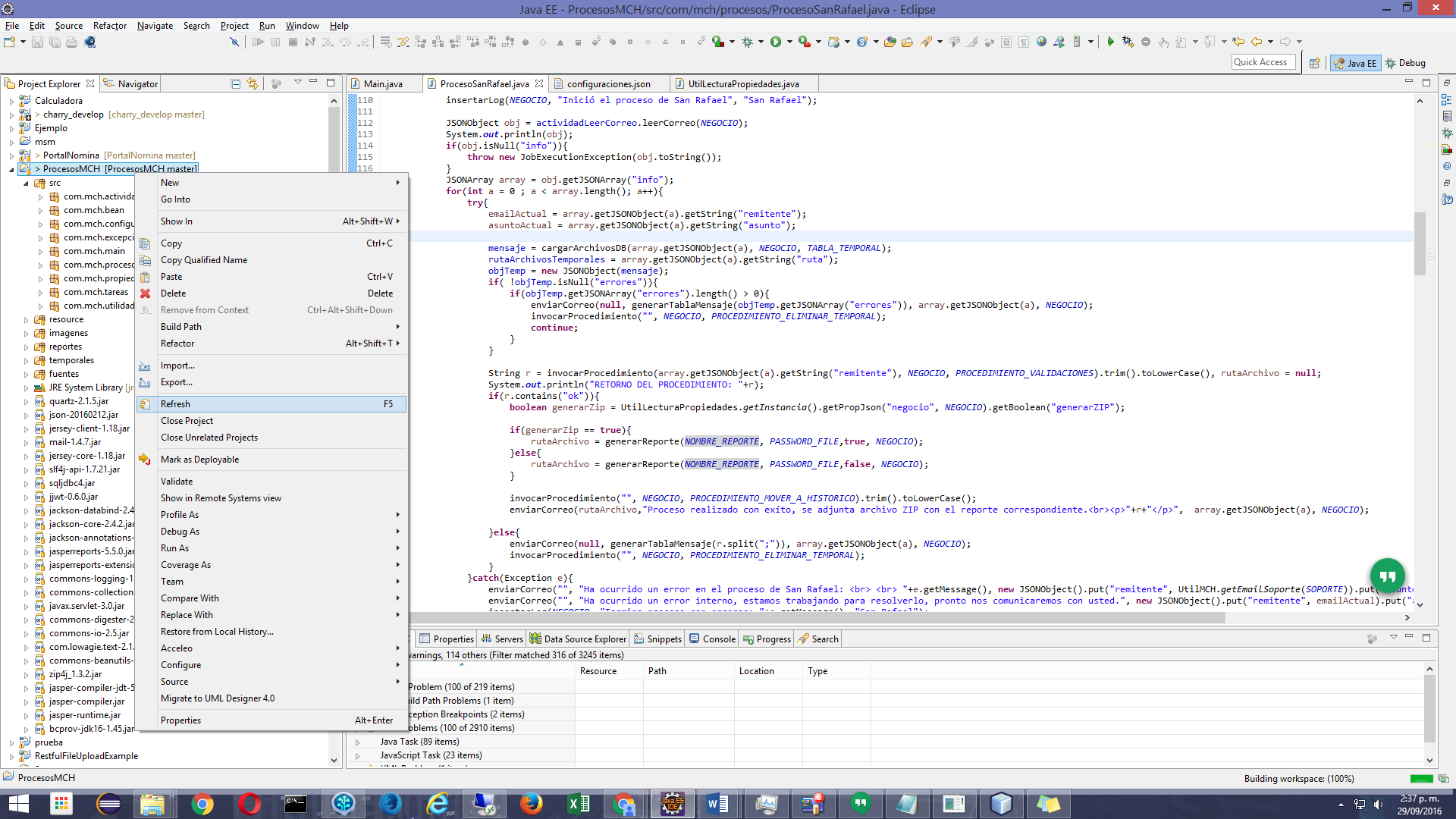
**com.mch.propiedades.servicios**: En este package van las clases que son un Bean pero este vean tiene los atributos que espera recibir el servicio que se va a invocar.

**com.mch.tareas**: En este package van las clases que cumplen una función en especifica Ej: Convertir un File a InputStream.

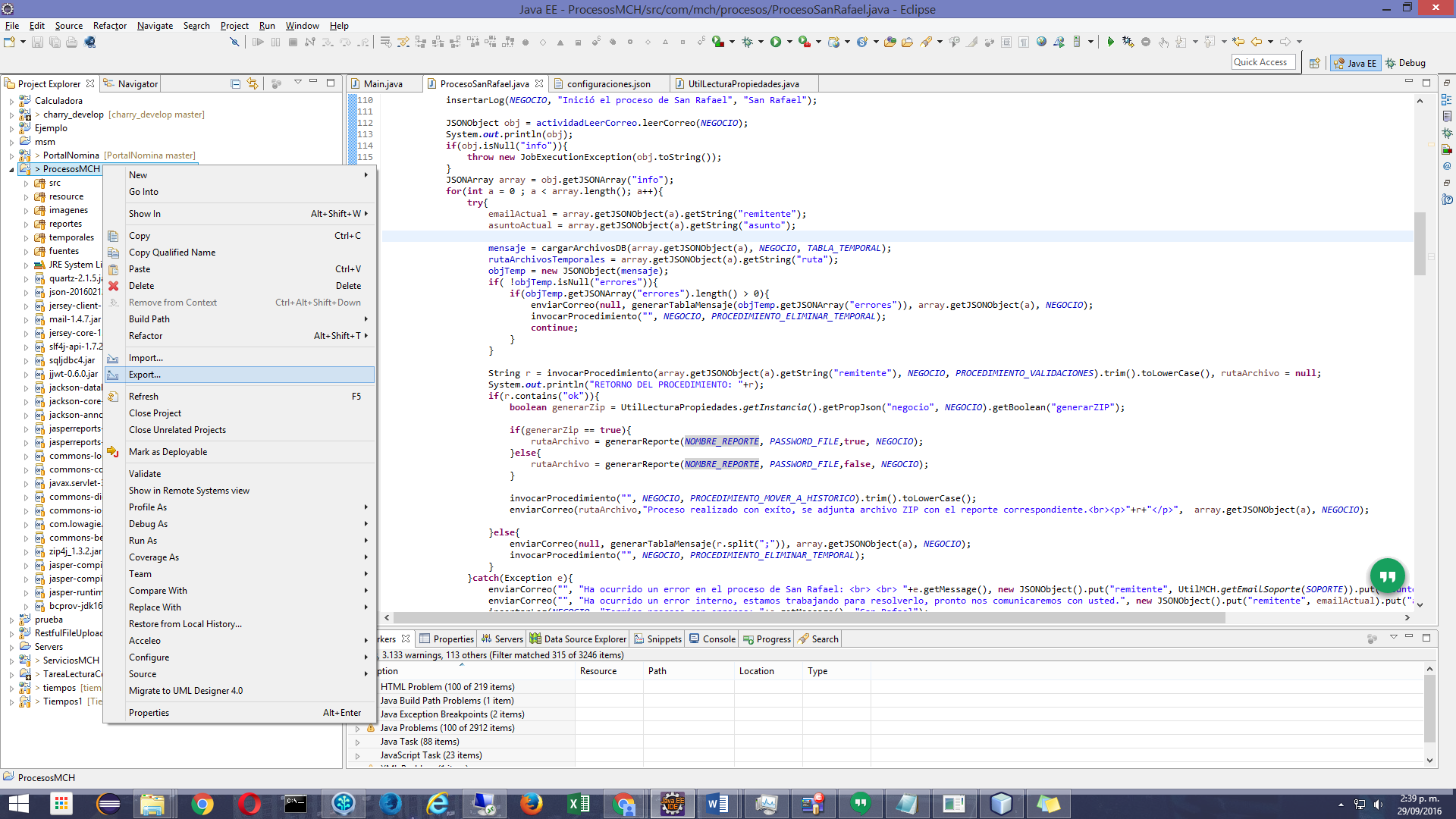
**com.mch.utilidades**: En este package van las clases que cumplen una tarea en especifica pero no se relacionan directamente con una tarea, actividad o proceso.

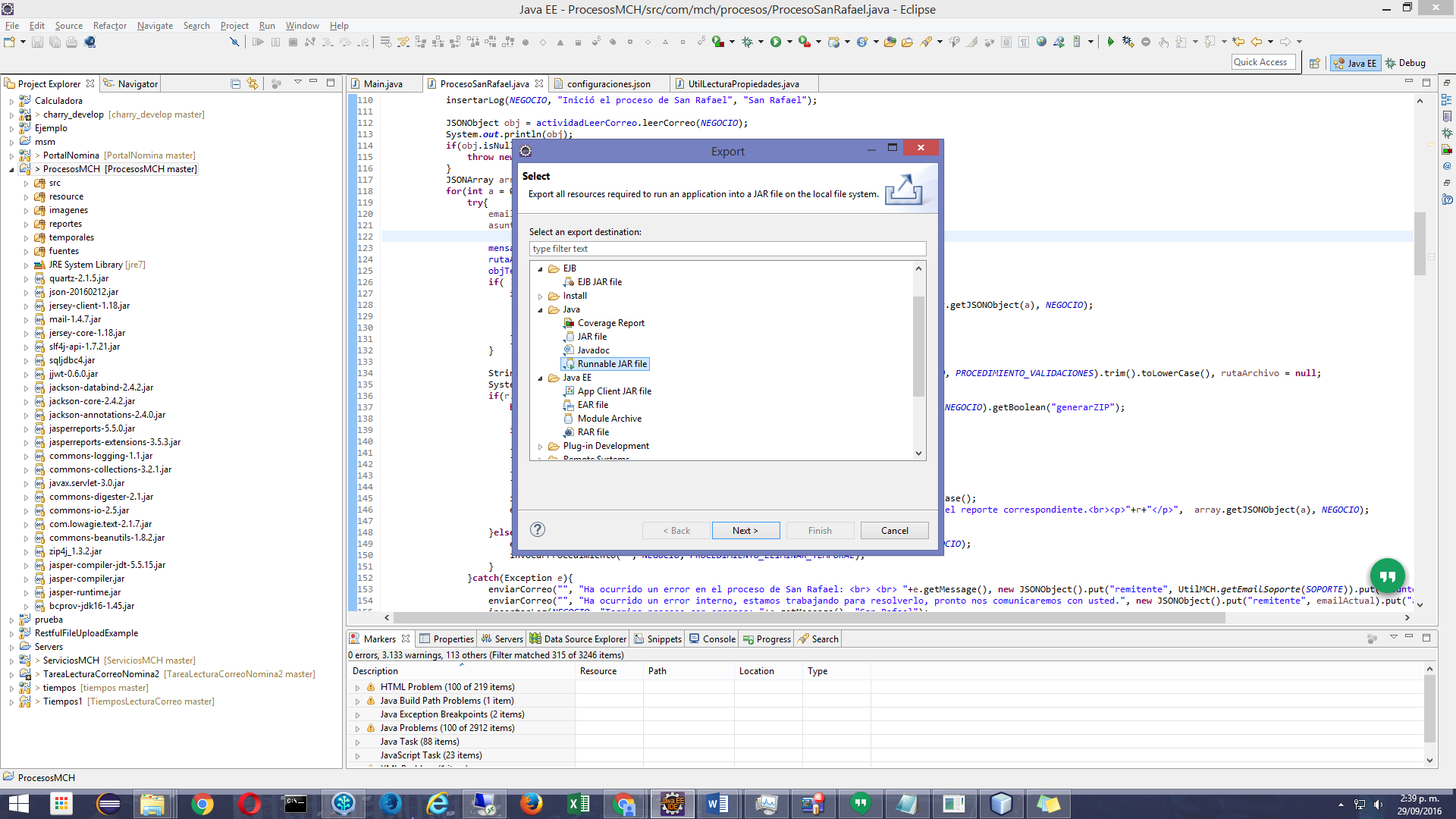
**Como exportar el proyecto de ProcesosMCH a un Jar desde eclipse (luna)**

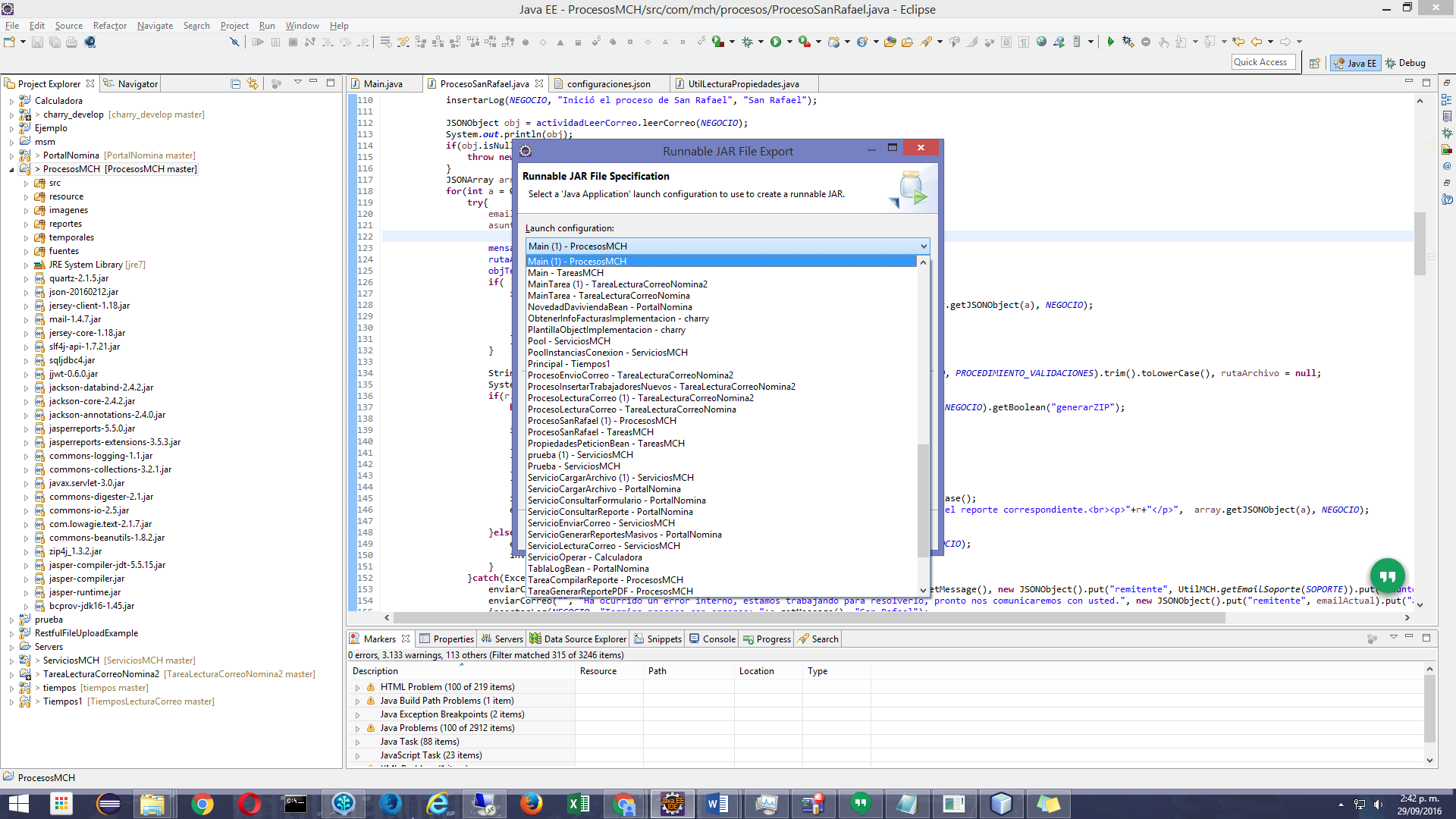
1. Refrescar el proyecto



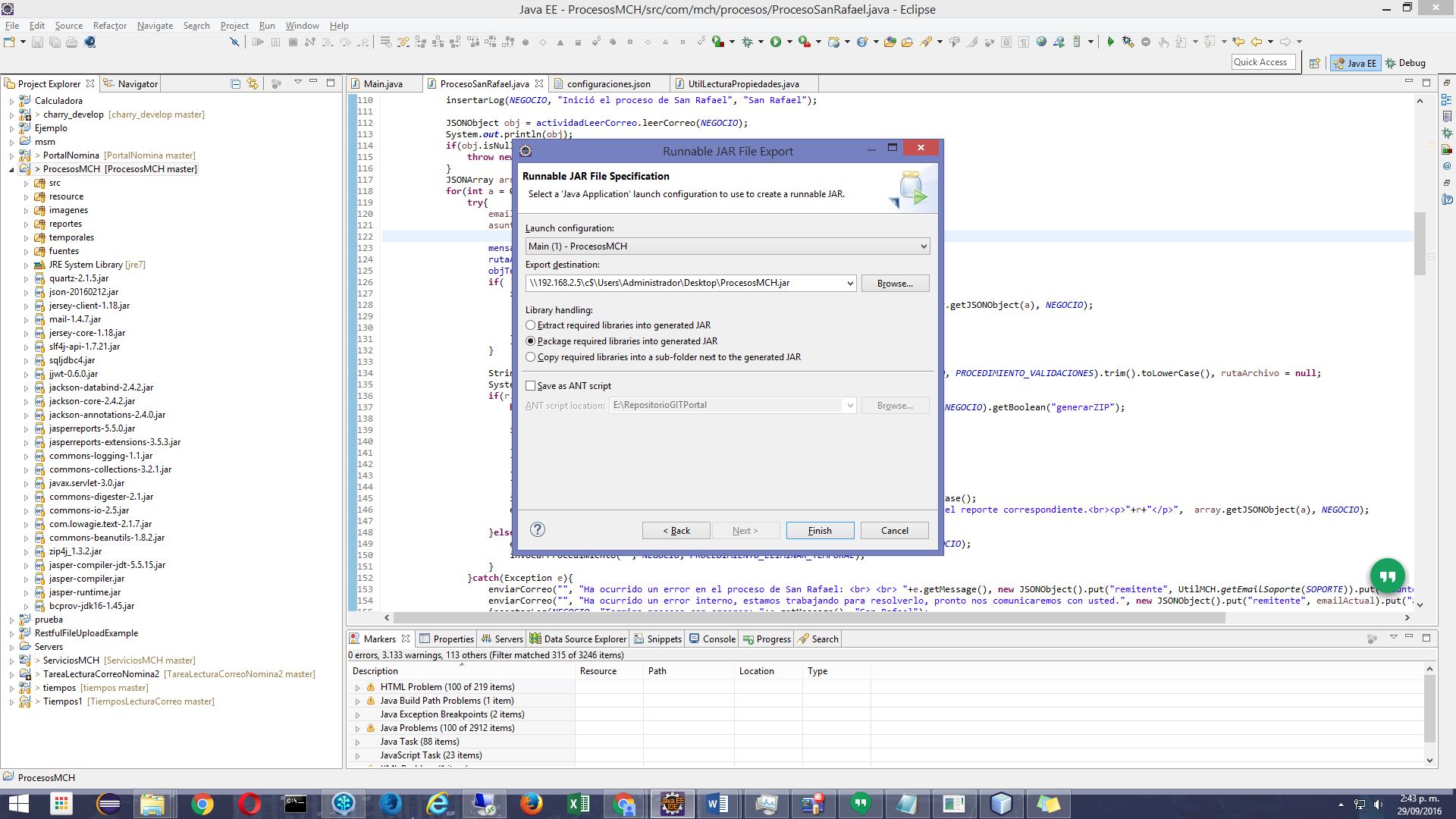
1. Clic derecho sobre el proyecto, seleccionar **Export**



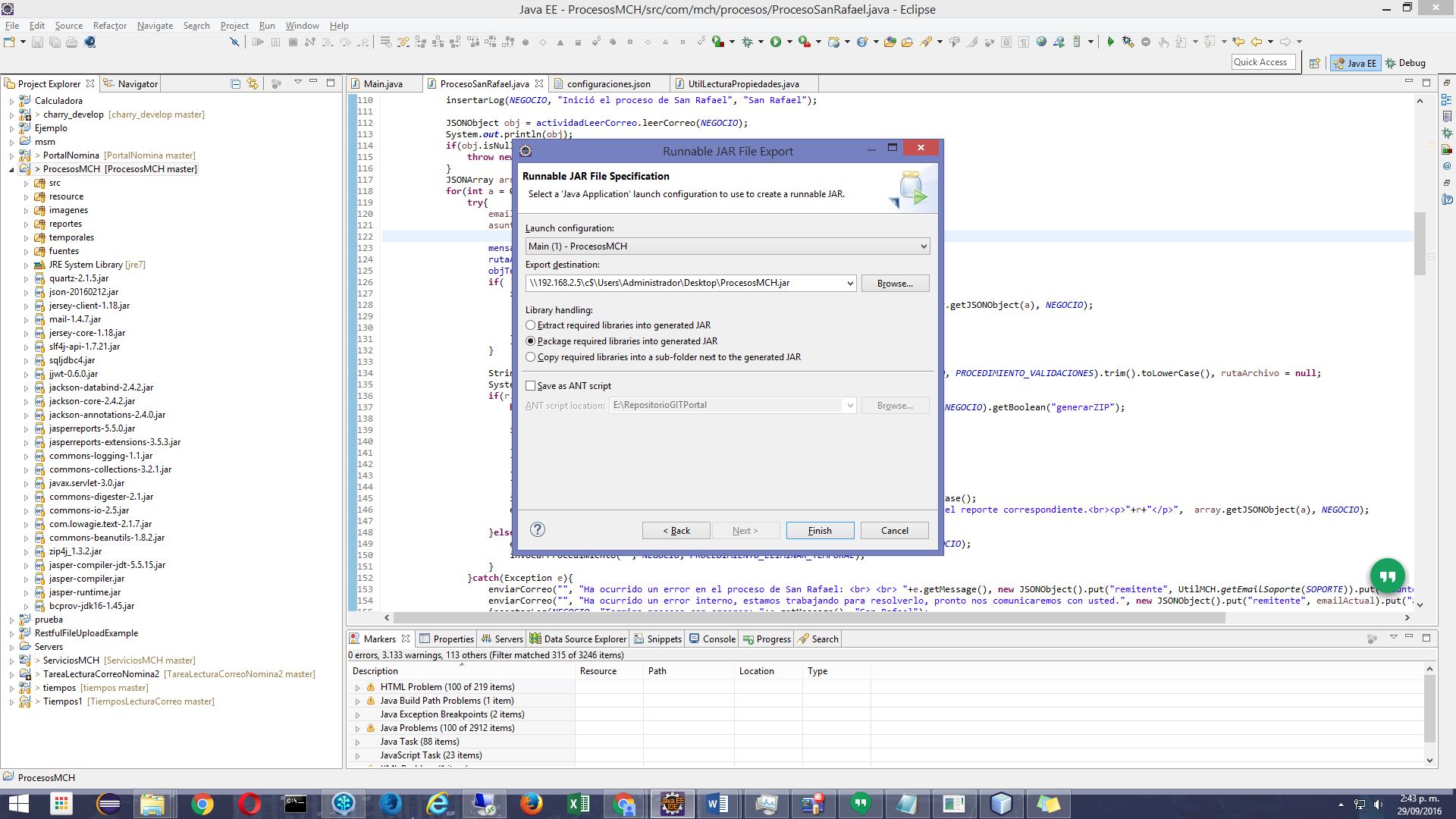
1. Seleccionar la opción:  
   
2. Seleccionar la clase main del proyecto, si no aparece en la lista, abrir la clase y ejecutarla para que eclipse la reconozca como clase principal del proyecto.



1. Seleccionar la ruta en donde se va a crear el jar



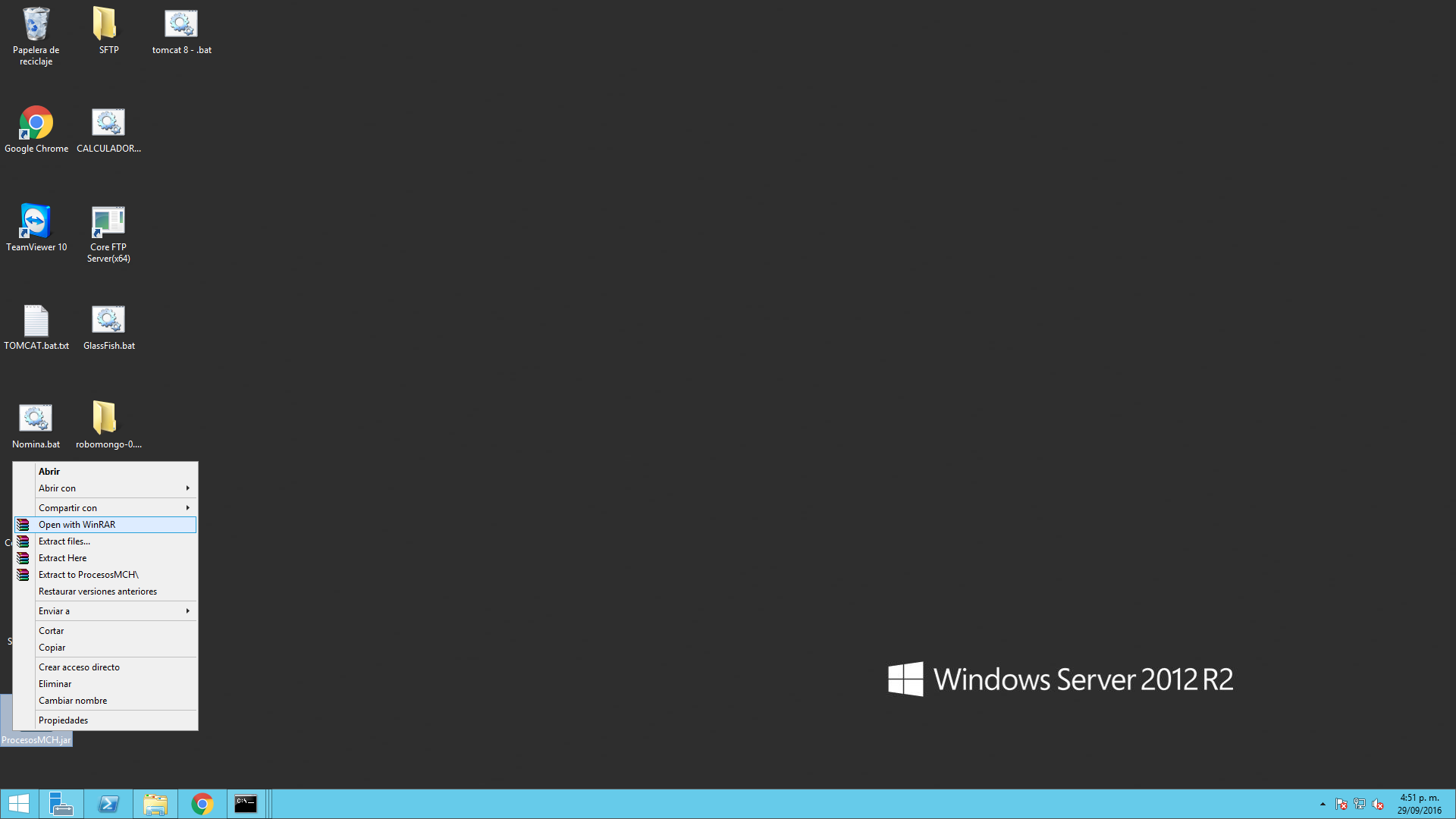
1. **IMPORTANTE:** hay que seleccionar la segunda opción en el ítem Library handling.

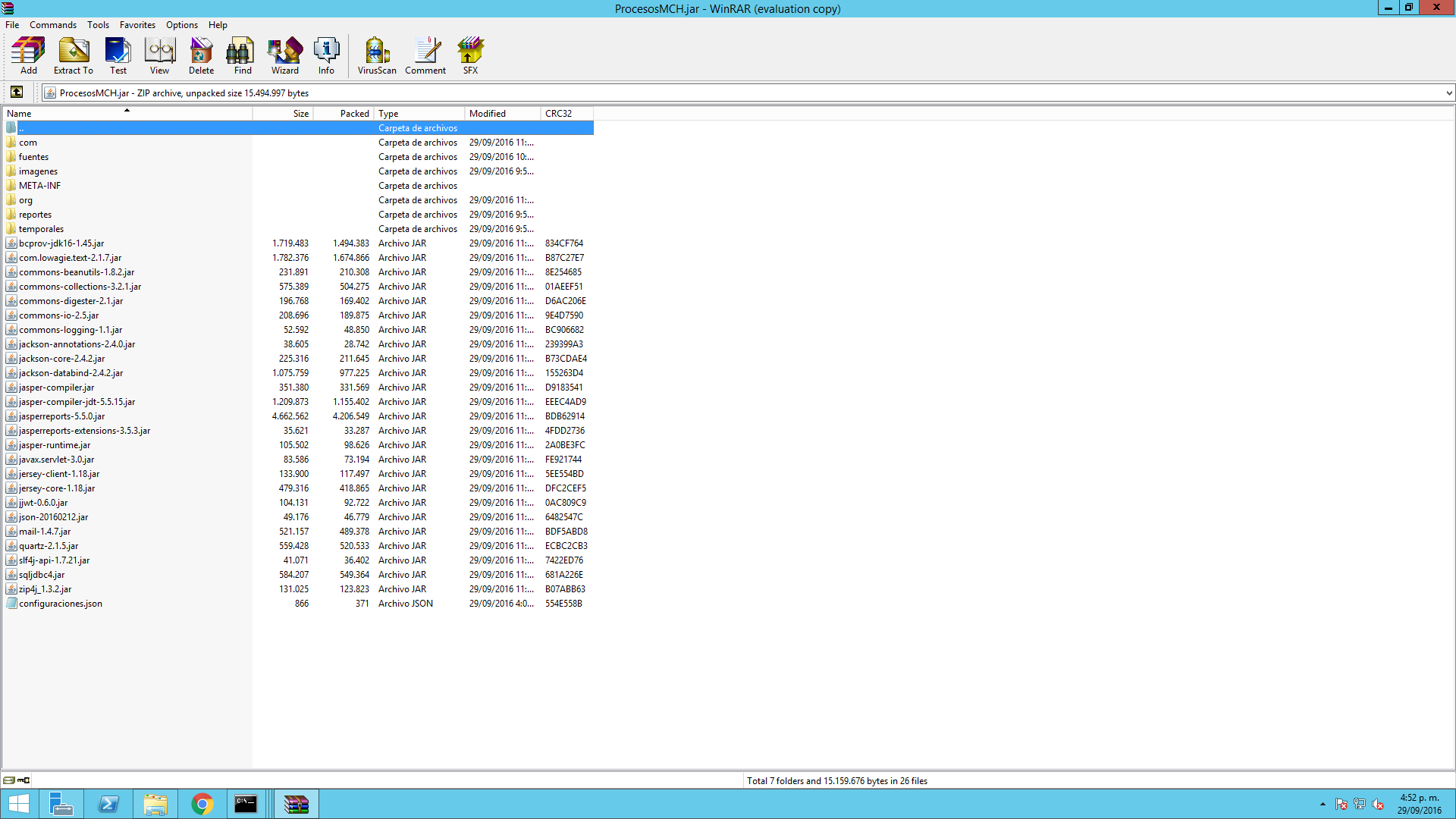


1. Y por último **Finish**

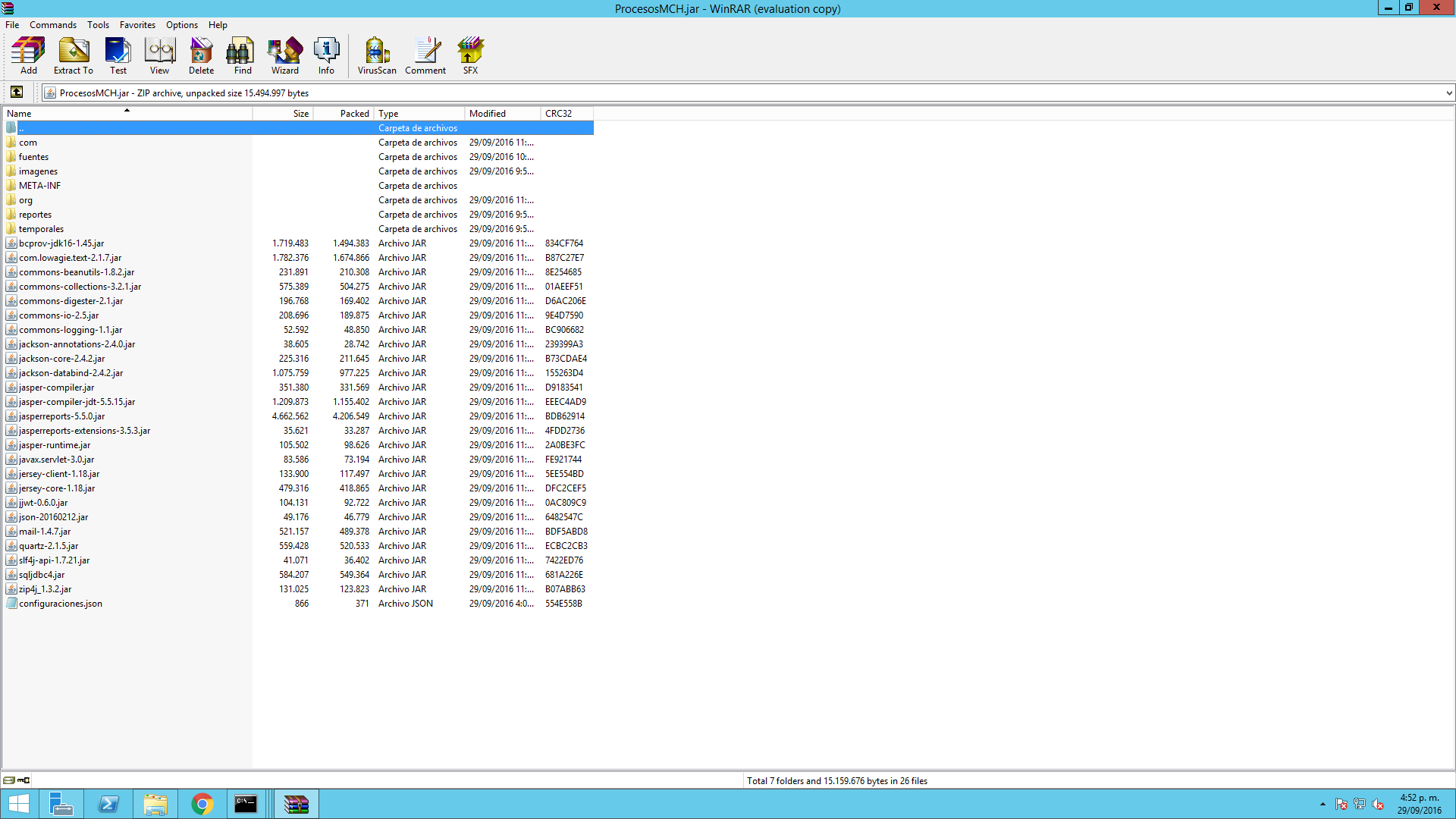
Con estos pasos se exporta el proyecto con las carpetas y librerías necesarias para el correcto funcionamiento.

**Paso a producción de un archivo Jar**

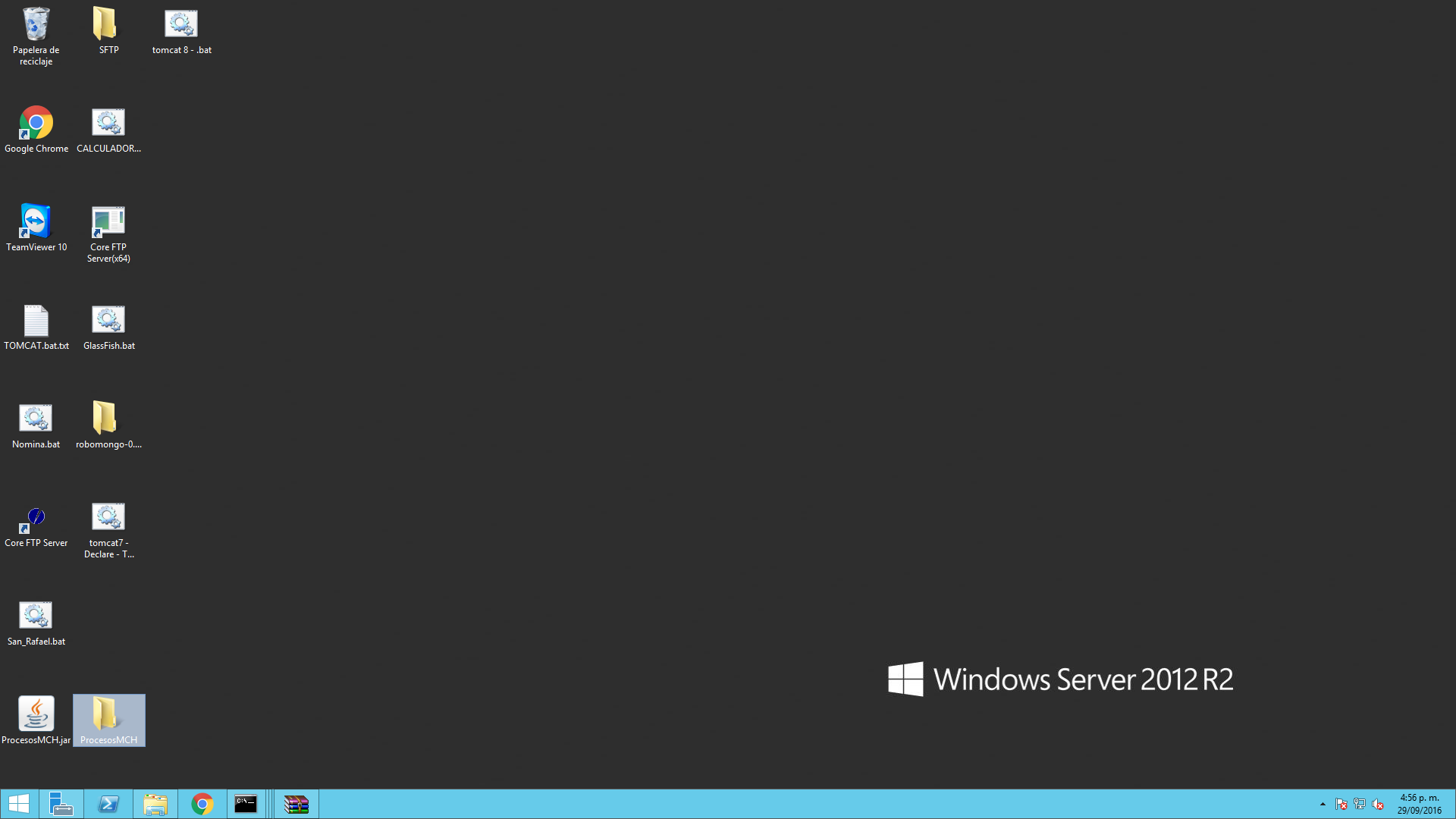
Una vez tengamos el Jar, procedemos a abrirlo con WinRar y se va a ver algo así:   


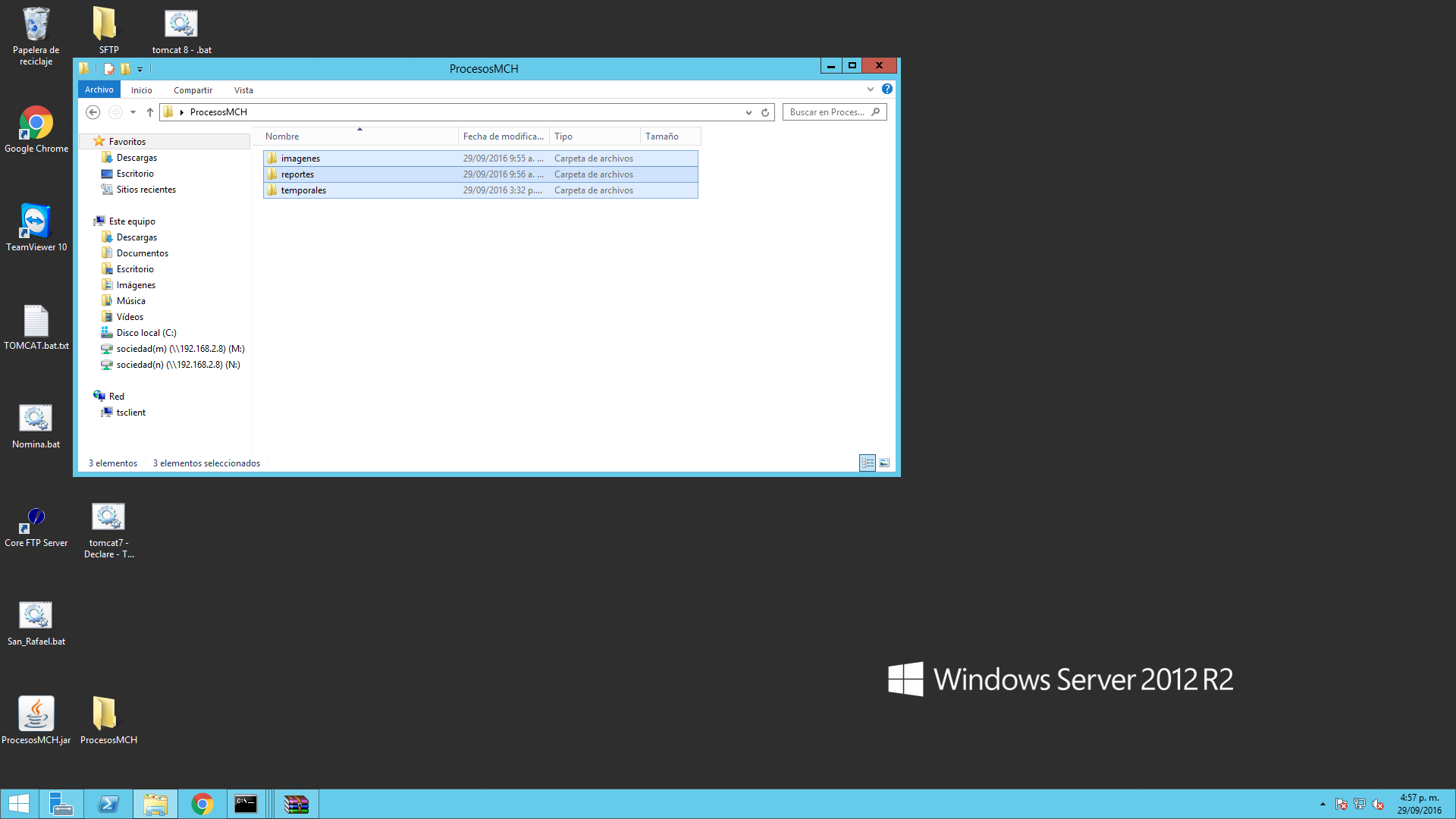


Ahora se copian las siguientes carpetas:

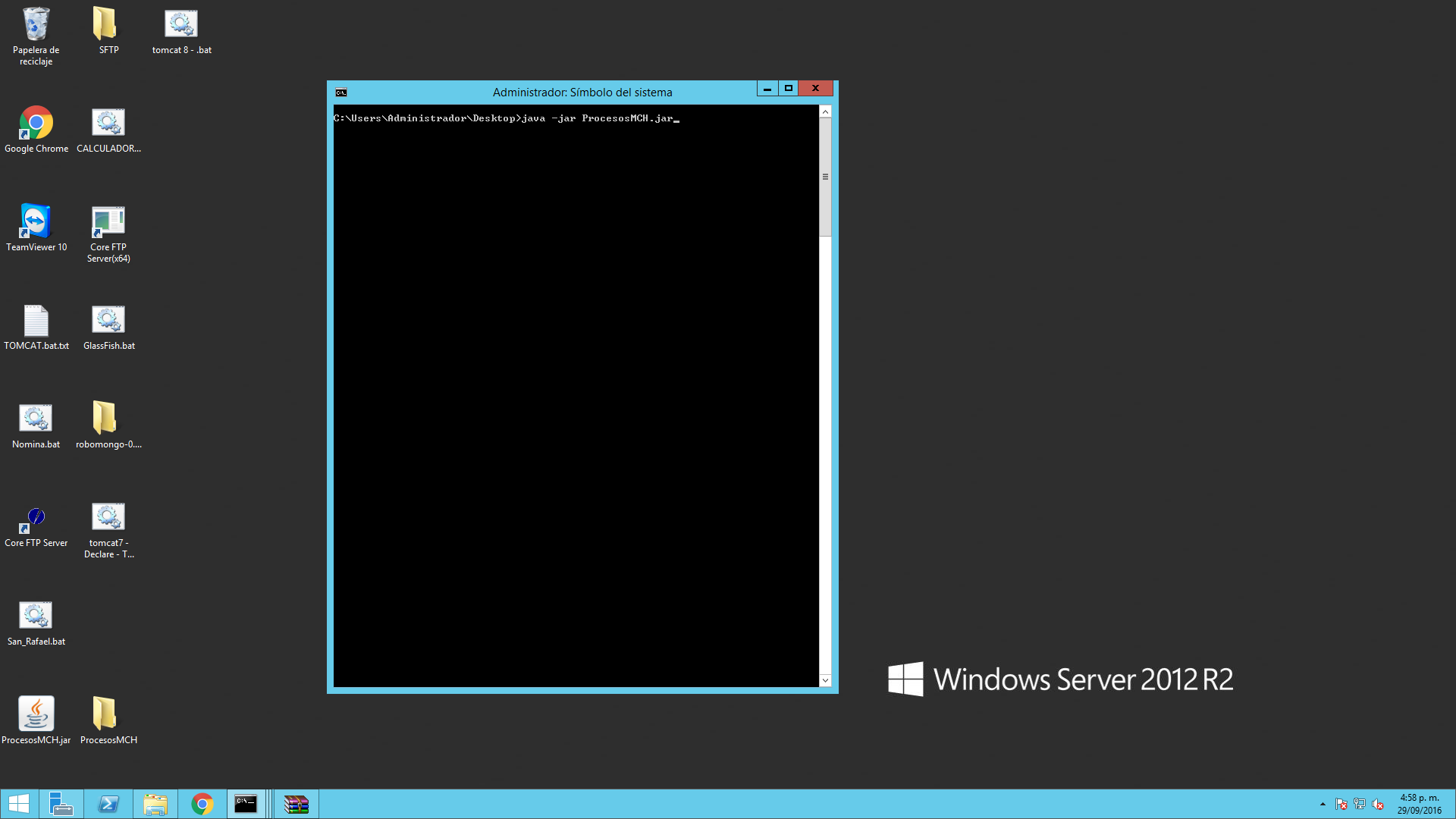


Después se crea una carpeta con el mismo nombre del proyecto, debe estar ubicada en el mismo lugar en donde está el Jar, en este caso el Jar está en el escritorio y la carpeta también debe estar en el escritorio, y después pegamos las carpetas que copiamos.



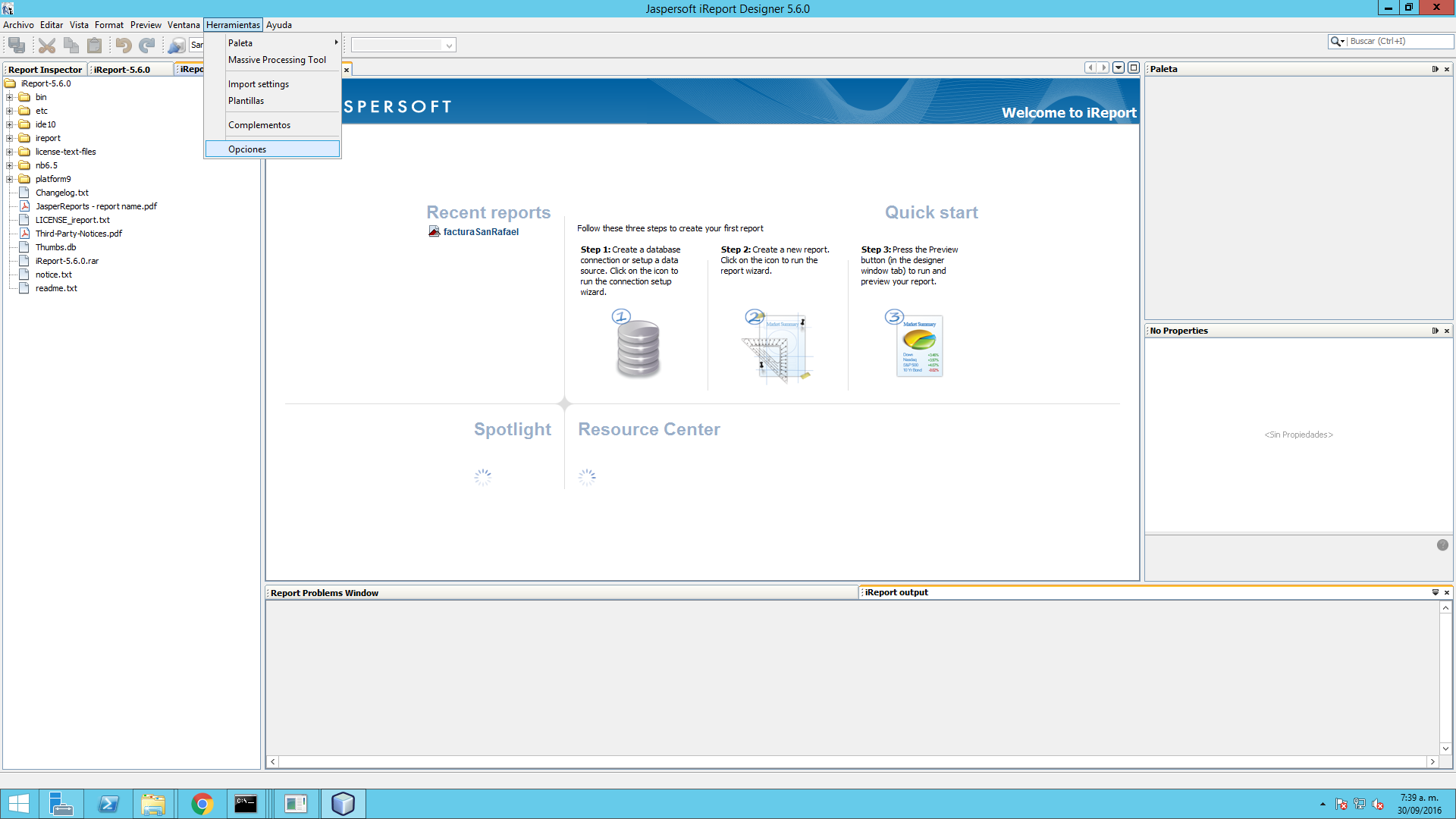


Y ejecutamos el Jar desde **CMD** para el caso de Windows:

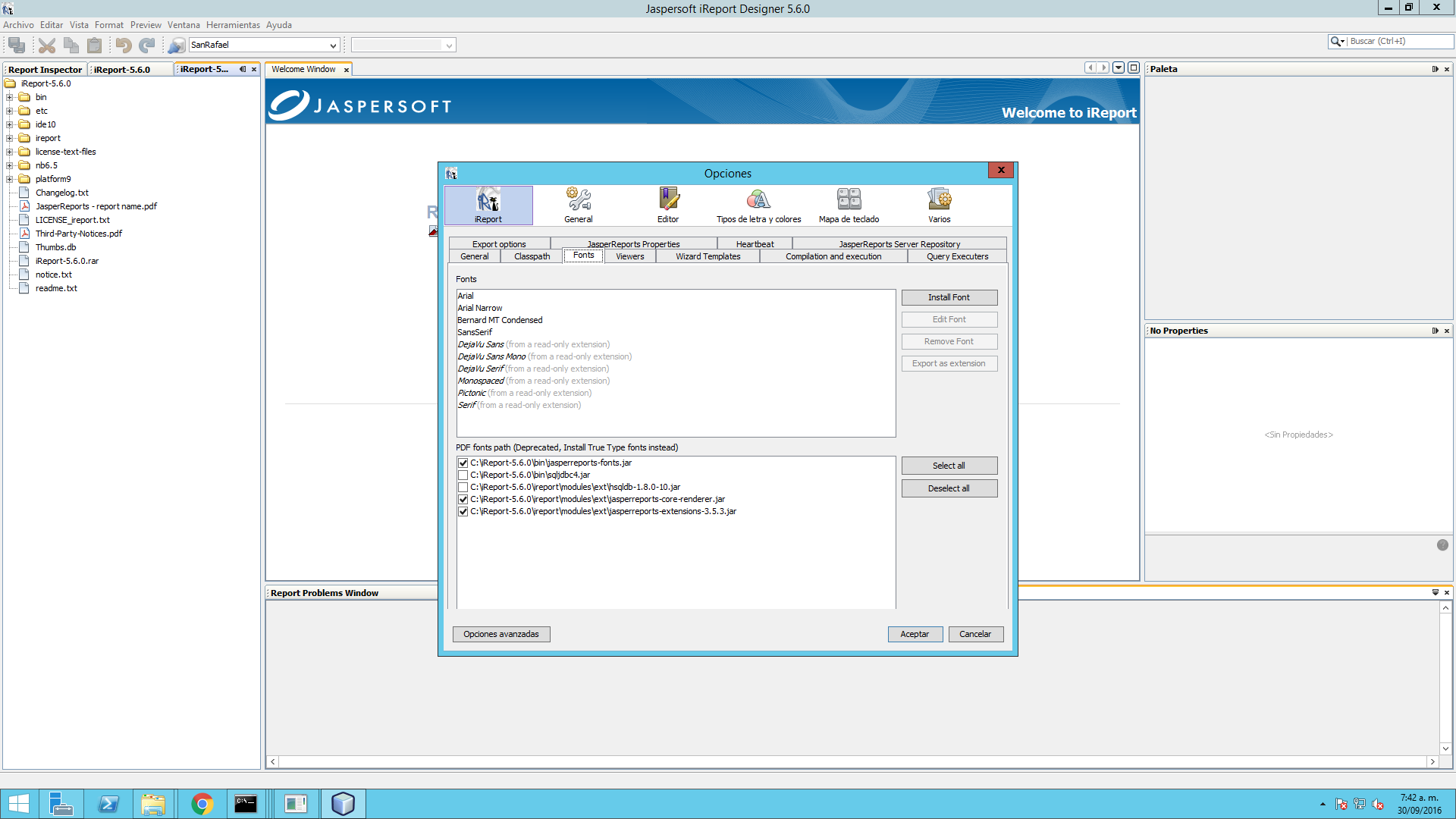


Si al ejecutar el Jar se muestra un error al compilar algún reporte, hay que importar las fuentes necesarias, esto se hace de la siguiente manera:

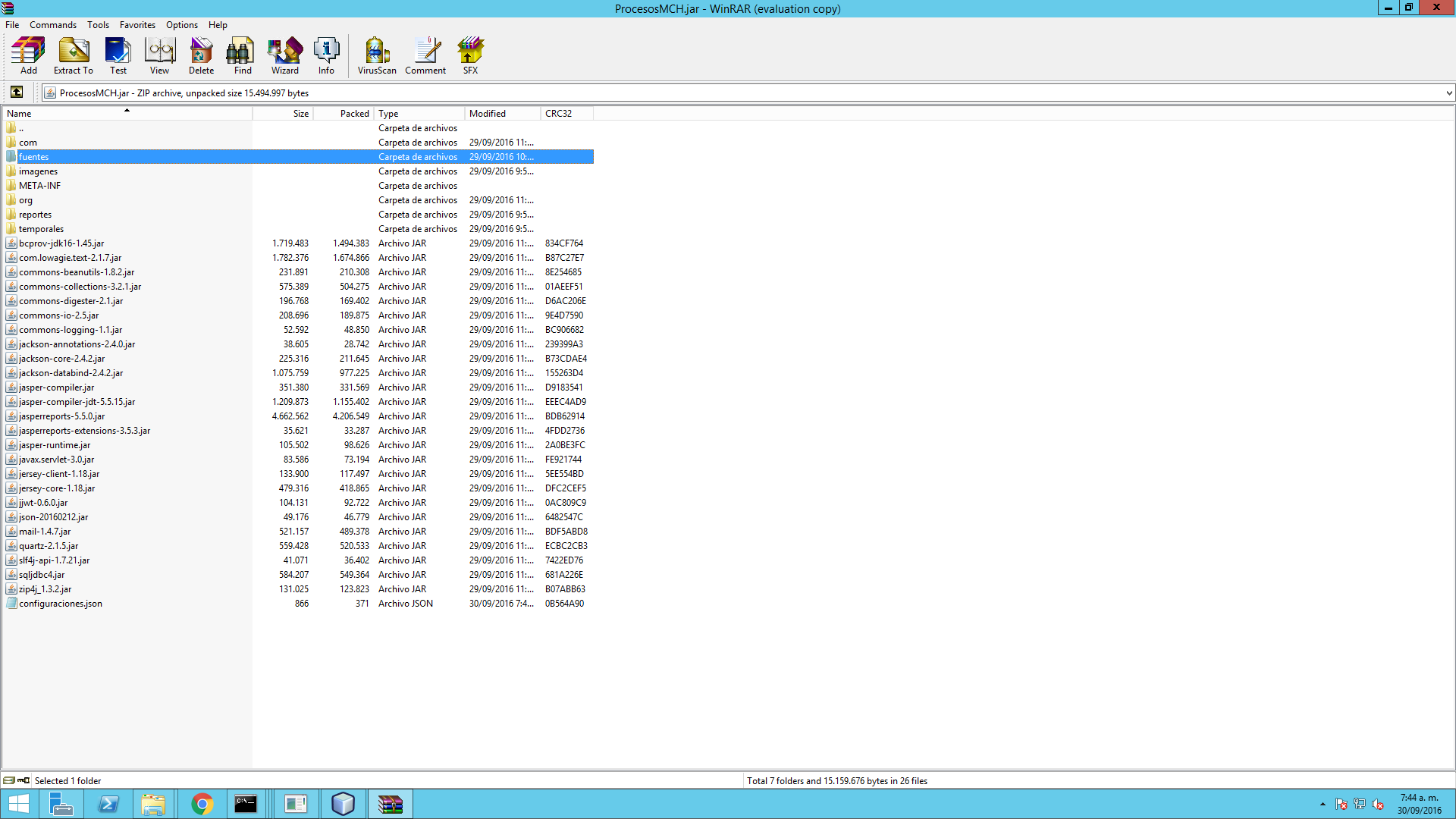
1. Primero hay que abrir el irreport 5.6.0, debe estar en el mismo servidor en donde está el Jar.
2. Clic en opciones:

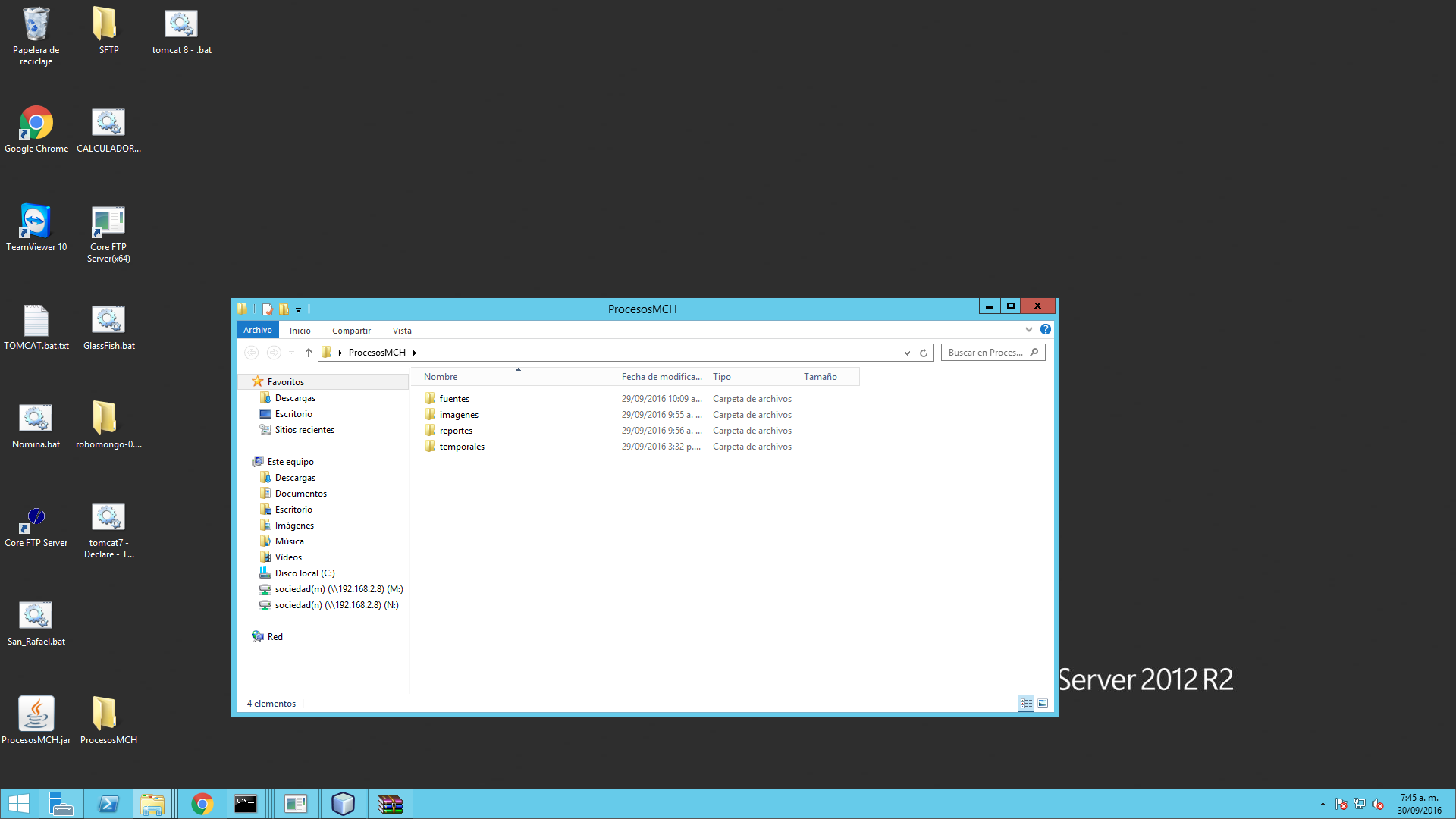


1. Clic en fonts, después clic en Install Font

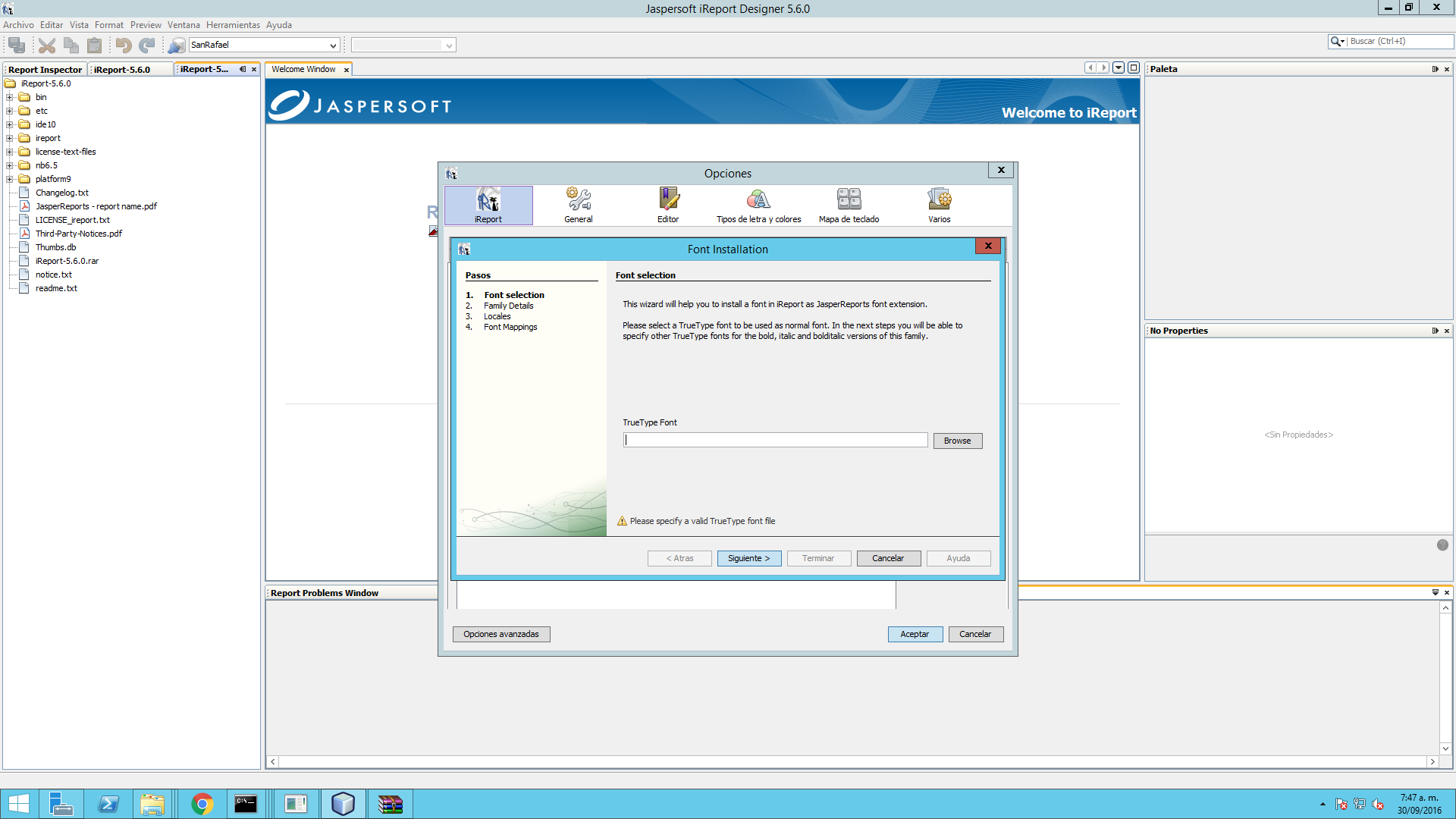


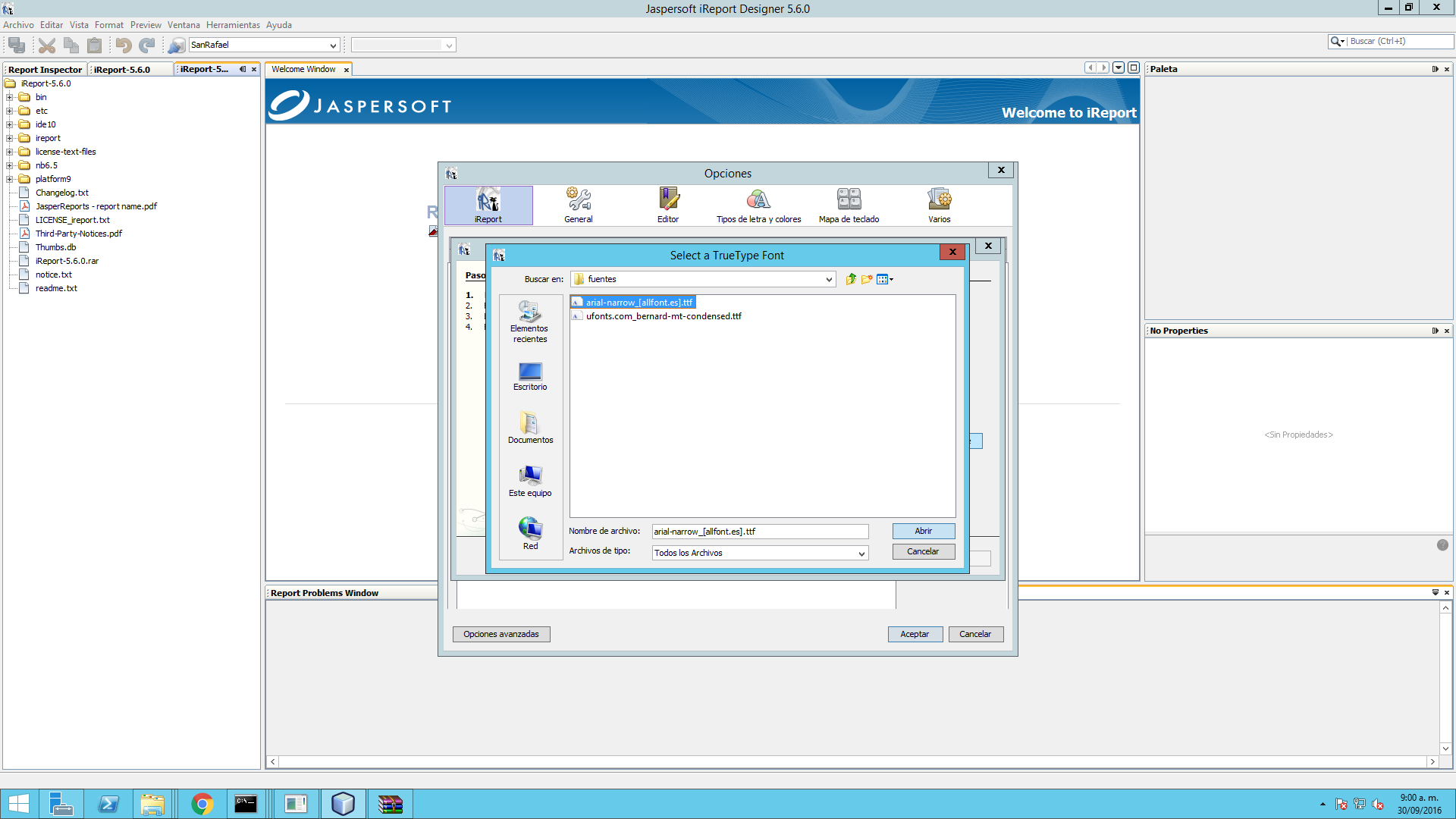
1. Después se copia la carpeta fuentes que se encuentra en el Jar y se pega en la carpeta del proyecto

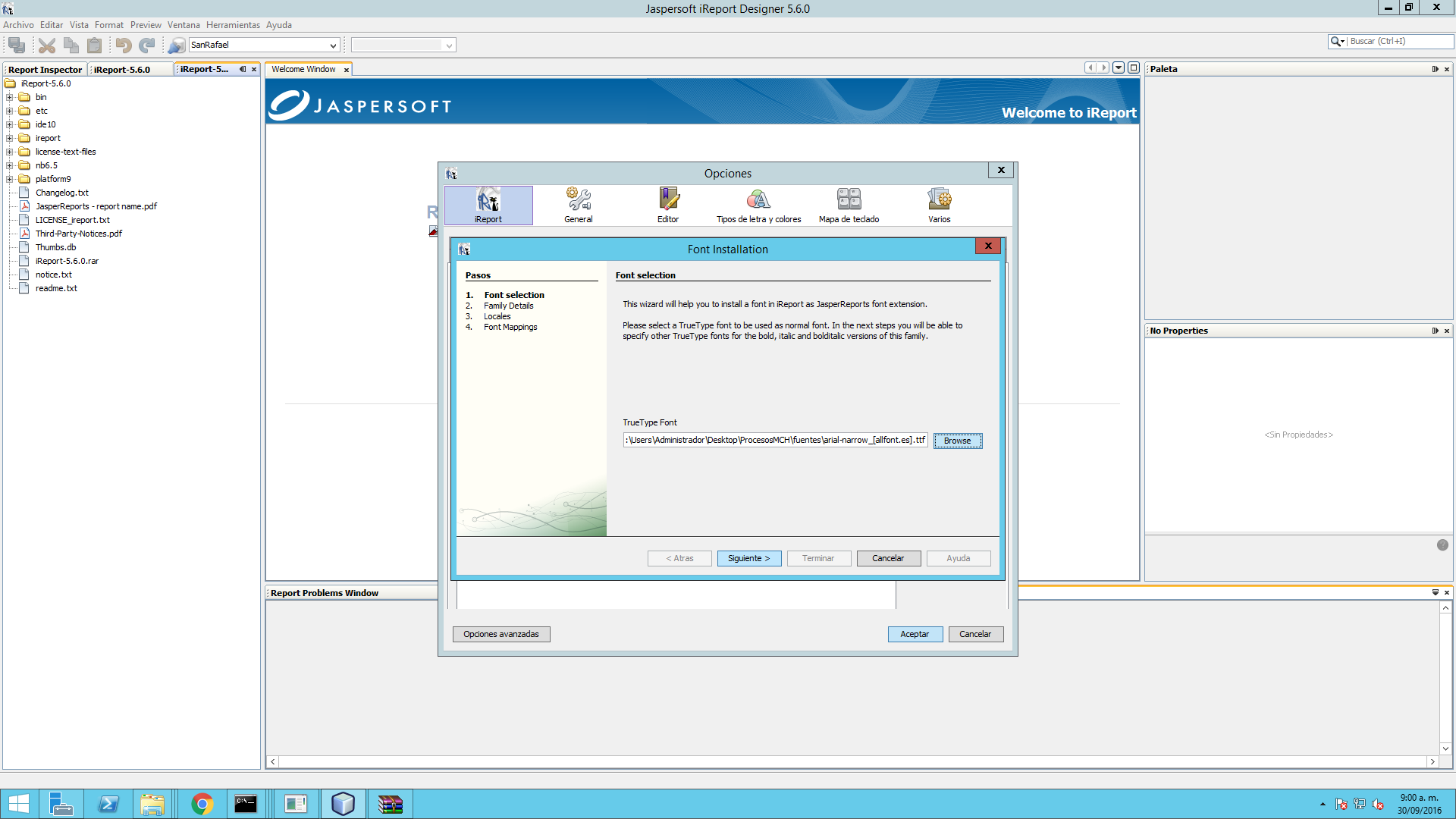




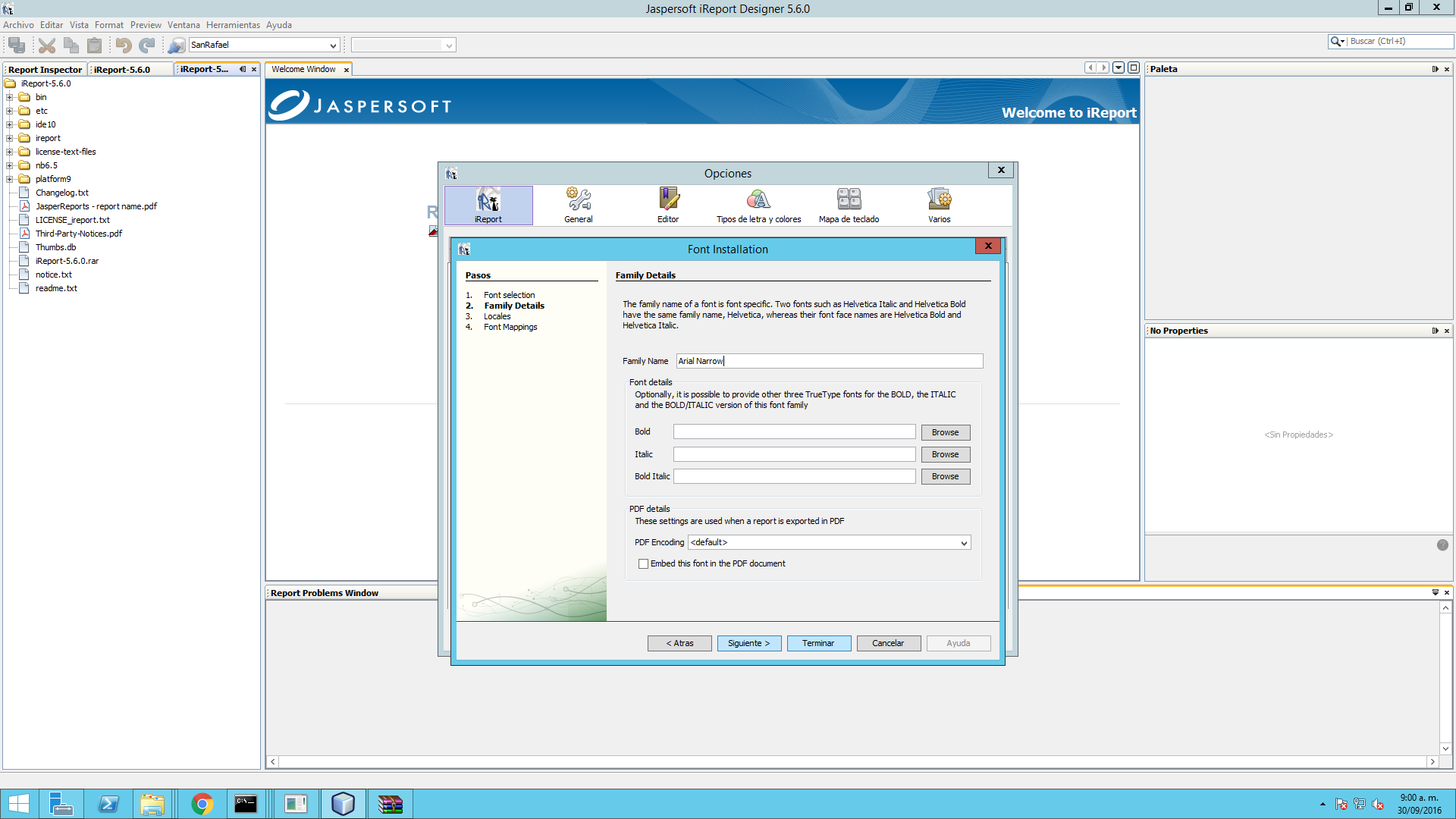
1. Ahora hay que importar las fuentes que están en la carpeta



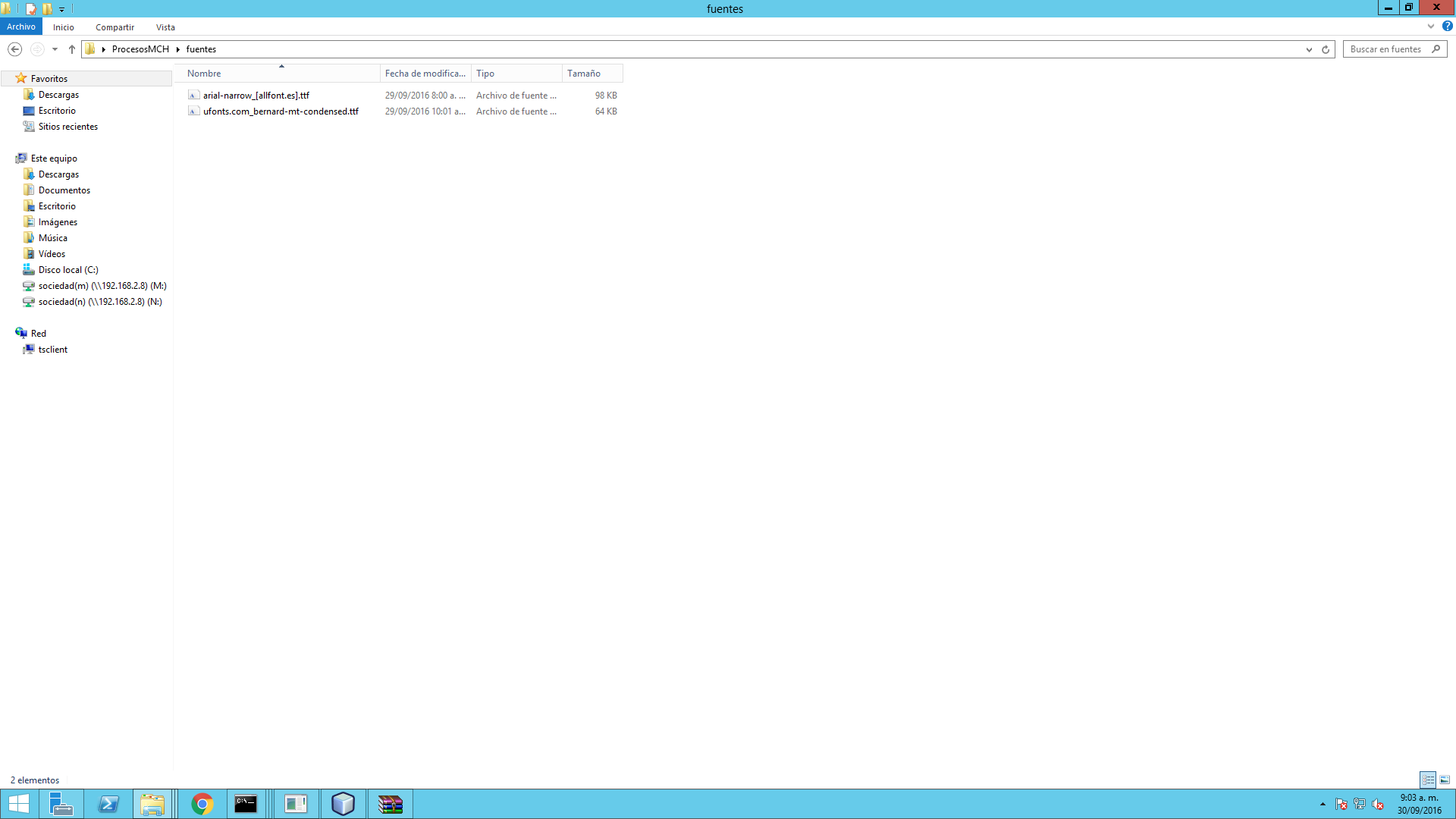




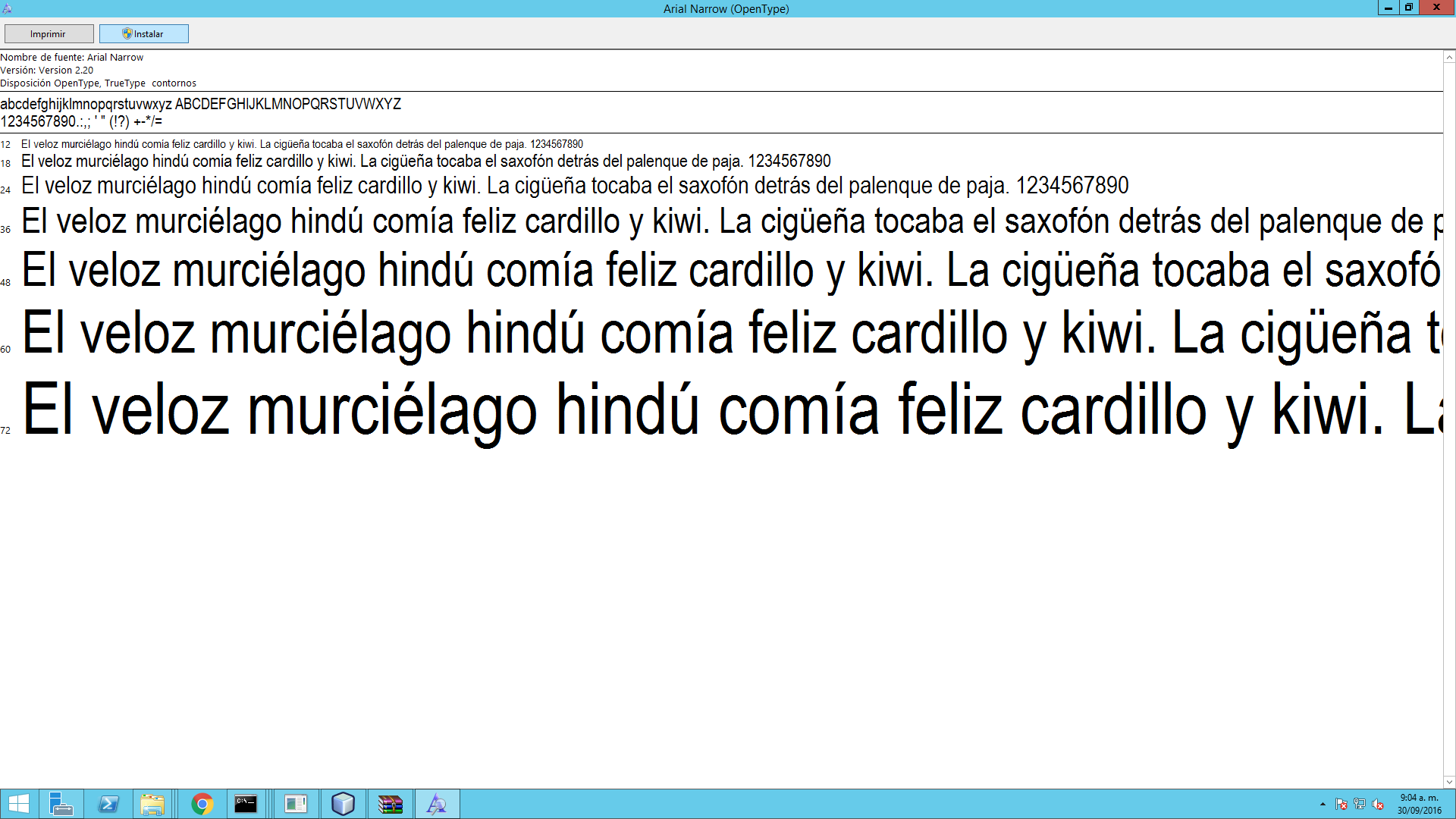
Y terminar.



Ahora hay que volver a la carpeta de fuentes

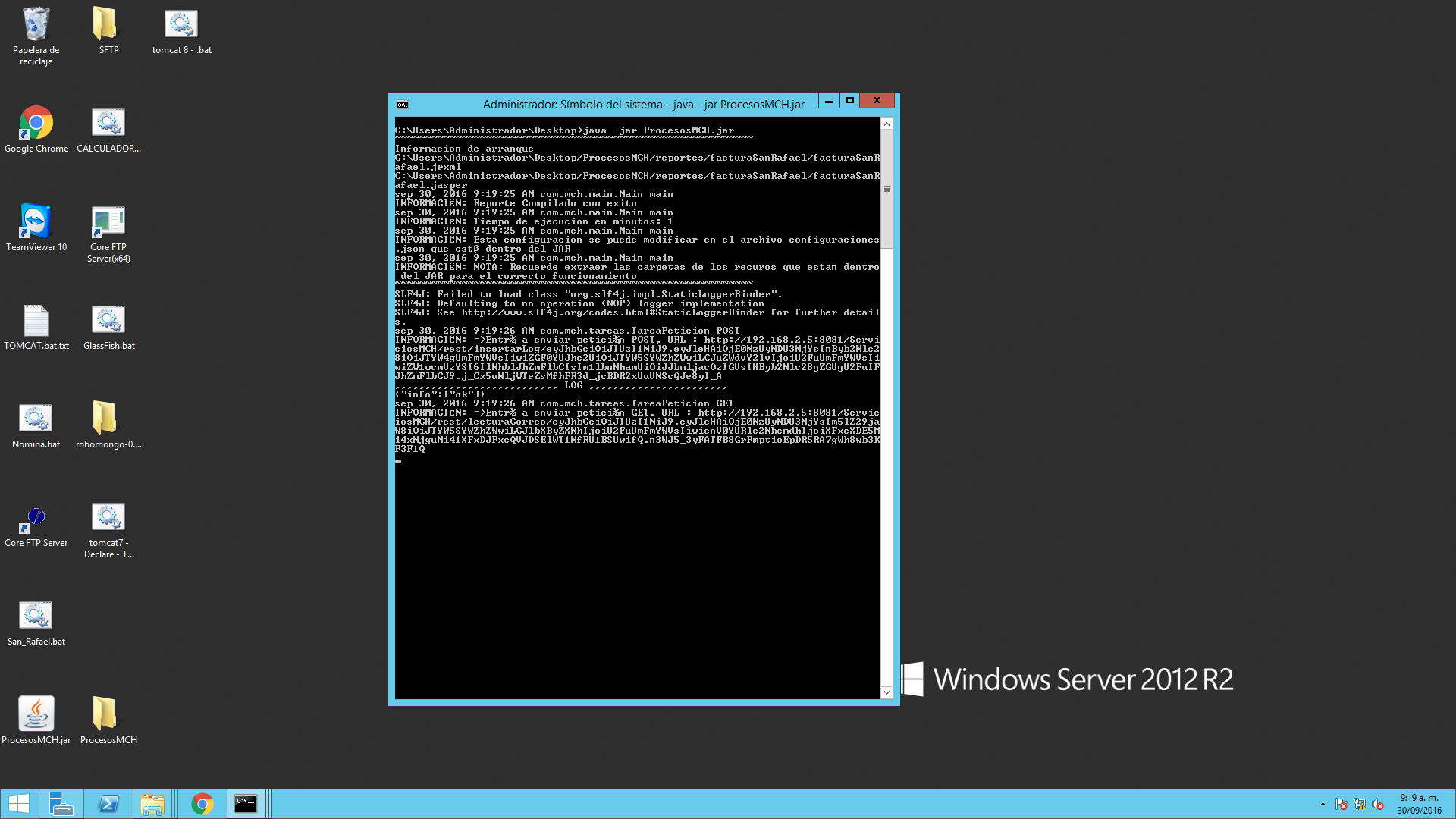


Después se instalan cada una de las fuentes que hay en esta carpeta, esto se hace haciendo doble clic sobre la fuente y después instalar:



Después se vuelve a ejecutar el Jar y debería iniciar con normalidad, si llega a mostrar nuevamente el error pero con otra fuente, hay que descargar la fuente e instalarla con los pasos ya mencionados.

Al iniciar la tarea, se muestra la siguiente información de arranque:



Esta información indica que la tarea inició correctamente.