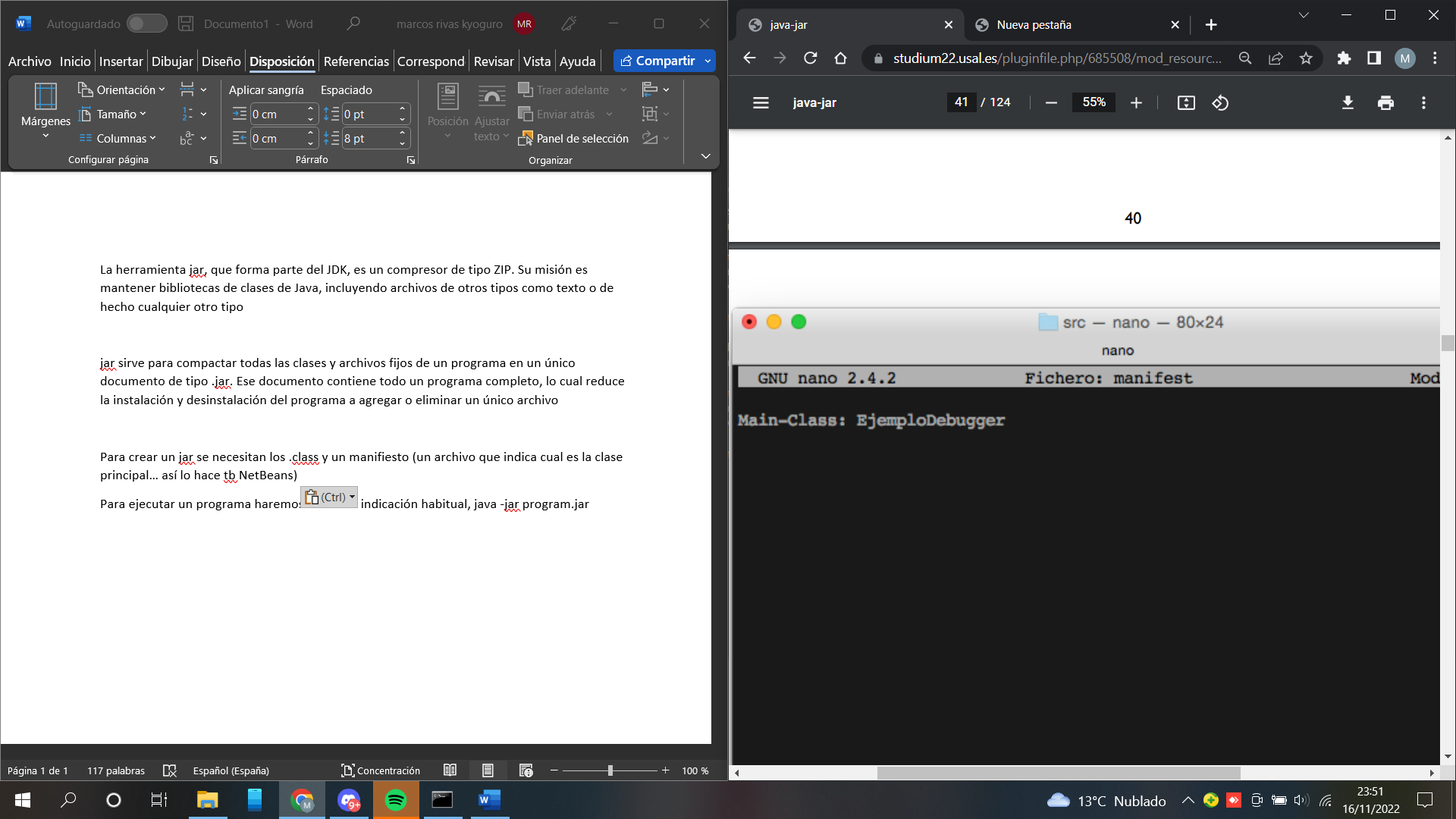
# LA HERRAMIENTA JAR

La herramienta jar, que forma parte del JDK, es un compresor de tipo ZIP. Su misión es mantener bibliotecas de clases de Java, incluyendo archivos de otros tipos como texto. Sirve para compactar todas las clases y archivos fijos de un programa en un único documento de tipo .jar. Ese documento contiene todo un programa completo, lo cual reduce la instalación y desinstalación del programa a agregar o eliminar un único archivo

Para crear un jar se necesitan los .class y un manifiesto (un archivo que indica cual es la clase principal)

Si el jar se crea mediante NetBeans, entonces se hallará en la carpeta dist del proyecto. Si el jar se crea mediante la línea de órdenes (una orden jar, un makefile) entonces aparecerá en el directorio de trabajo. También es posible indicar el directorio en que se quiere almacenar el jar, anteponiendo una ruta al nombre del archivo

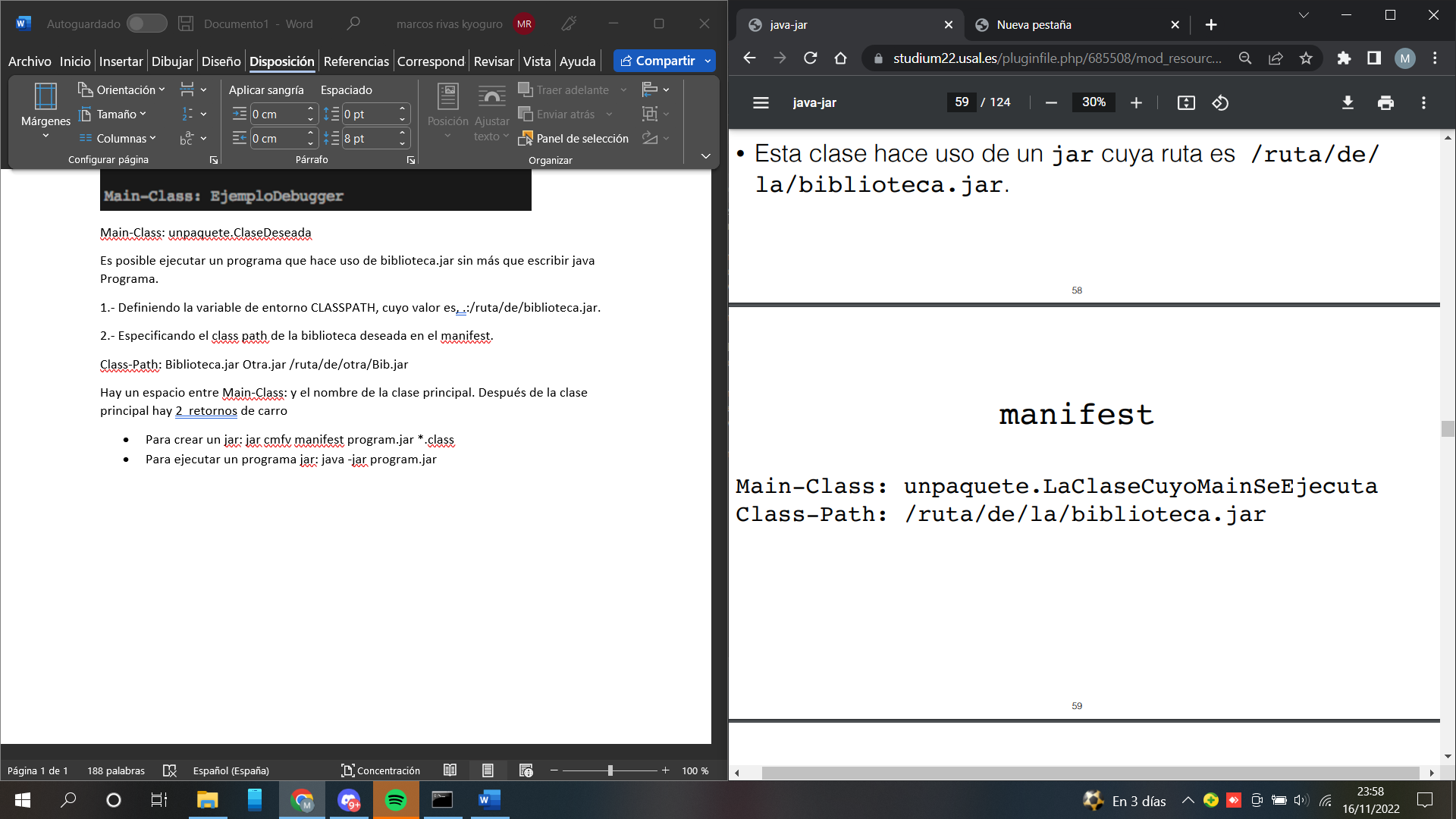
 Main-Class: unpaquete.ClaseDeseada

Es posible ejecutar un programa que hace uso de biblioteca.jar sin más que escribir java Programa.

1.- Definiendo la variable de entorno CLASSPATH, cuyo valor es, .:/ruta/de/biblioteca.jar.

2.- Especificando el class path de la biblioteca deseada en el manifest.

Class-Path: Biblioteca.jar Otra.jar /ruta/de/otra/Bib.jar



Hay un espacio entre Main-Class: y el nombre de la clase principal. Después de la clase principal hay 2 retornos de carro.

# OPERACIONES CON JARS

* *Para crear un jar:* jar cmfv manifest programa.jar \*.class
* *Para ejecutar un jar:* java -jar programa.jar
* *Para ver el contenido de un jar:* jar tf programa.jar
* *Para extraer el contenido de un jar:* jar xf program.jar
* *Para añadir un índice a un jar:* jar i programa.jar
* *Para actualizar un jar:* jar uf programa.jar archivo(s).java
* *Para eliminar el contenido de un jar:* zip -d programa.jar archivo\_no\_deseado.java

La construcción de un jar es un proceso complicado cuando el número de archivos de código fuente va creciendo. Una forma de automatizar este proceso es la utilizada por NetBeans, Apache Ant. La aplicación Ant está escrita en Java, y utiliza archivos de texto (llamados buildfiles) para construir un jar que contiene todas las clases y archivos de la aplicación. Normalmente se llama build.xml, está escrito en XML, fácil de analizar desde Java. XML resulta cómodo para su parsing por parte de una máquina, y engorroso a la hora de escribirlo por parte de una persona

# MAKE

Una forma alternativa es utilizar la herramienta make. Hace uso de un archivo llamado makefile (equivalente a un buildfile) y otro llamado manifest (cuya información, el nombre de la clase principal, aparecía también en el buildfile). Las herramientas ant y make precisan la misma información, para crear un jar. El archive makefile no utiliza XML, sino texto normal, y posee tres partes obligatorias: Relación de dependencias / Regla de construcción del jar / Reglas de construcción de los .class . Y opcionalmente, reglas para limpieza posterior (clean:).

Los makefile y manifest se encuentran en la carpeta src. Cuando se escribe make, se crea el jar . Una vez creado el jar, se ejecuta mediante java -jar program.jar