

ARCHIVO DE

CONFIGURACIÓN

Y CANTIDAD

**MARIO JIMÉNEZ MARSET**

**ÍNDICE**

[1. ENUNCIADO - OBJETIVOS 3](#_Toc115428335)

[2. PROGRAMA (SOLUCIÓN Y PASOS) 3](#_Toc115428336)

# ENUNCIADO - OBJETIVOS

En esta práctica se pedía crear un programa en Java con una serie de especificaciones, las cuáles llevaban a un resultado concreto dentro del archivo que se pedía descargar.

Este archivo se llama “inventario.conf”; la extensión .conf se refiere a los archivos de configuración que contienen todos los parámetros y configuraciones necesarios para el correcto funcionamiento de varios procesos de un sistema operativo, servidor o aplicaciones específicas. Normalmente, el contenido de los archivos.conf se almacena generalmente en el estándar de codificación ASCII.

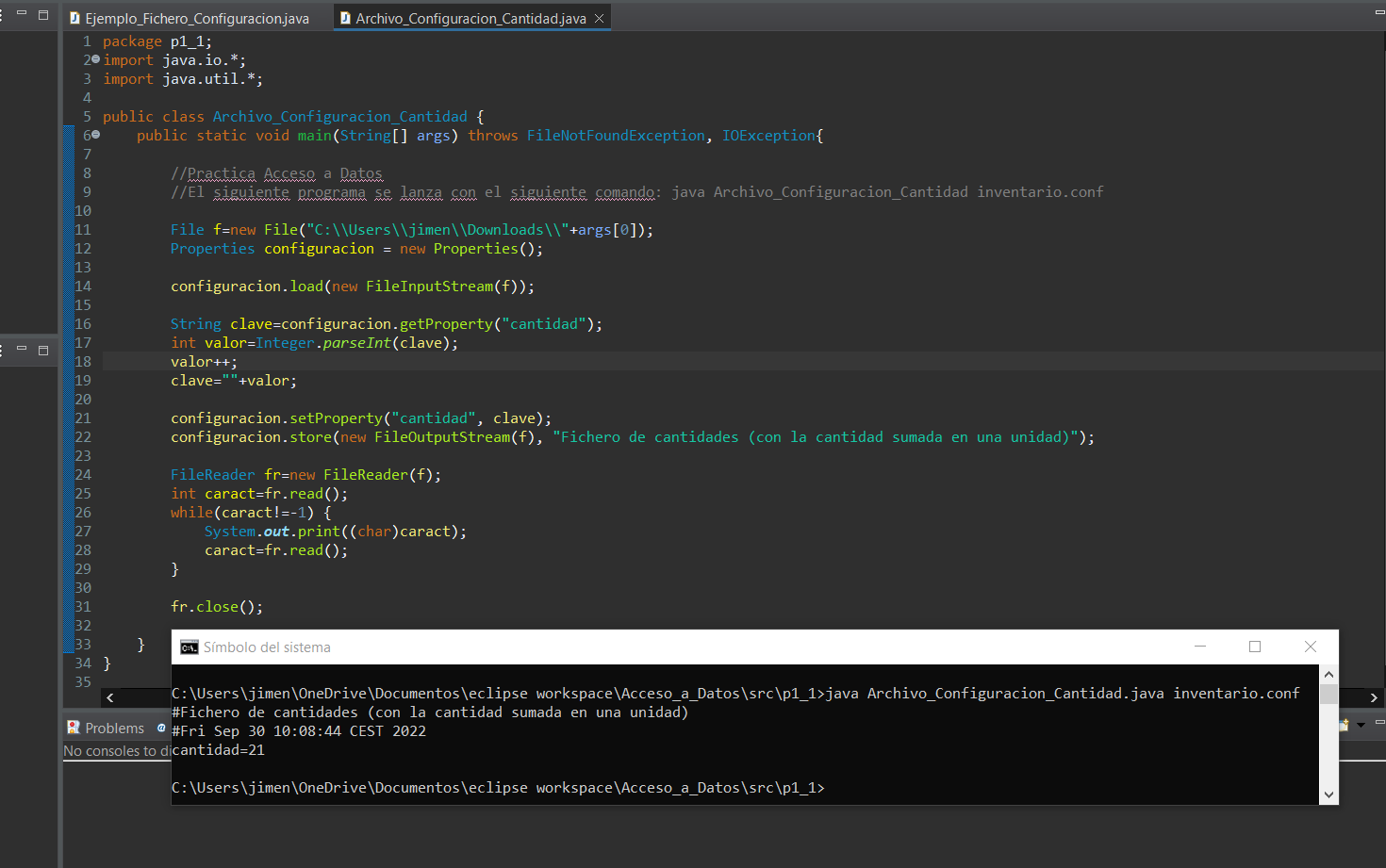
El programa se lanzará con el siguiente comando:

java Archivo\_Configuracion\_Cantidad.java inventario.conf

Se lee el archivo de configuración y se recupera la variable cantidad, a la cual se le suma una unidad. Luego, se guarda el archivo de configuración con esta variable ya cambiada.

# PROGRAMA (SOLUCIÓN Y PASOS)

En esta imagen se muestra el código del programa con la solución ya puesta en el cmd.

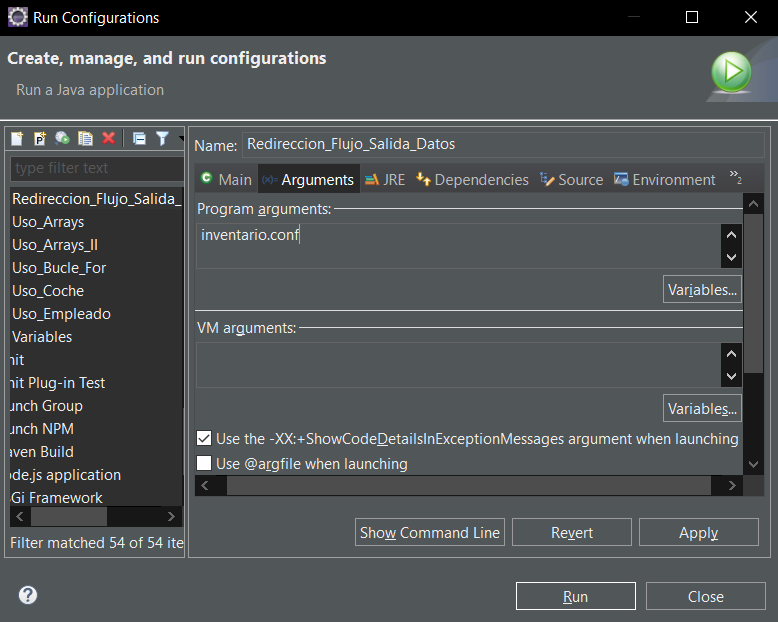


Se crea una clase con el nombre “Archivo\_Configuracion\_Cantidad”, dentro del cual hay un main, donde se harán todas las líneas de código.

**File f=new File("C:\\Users\\jimen\\Downloads\\"+args[0]);**

Se crea un fichero “f”, de la clase File, donde se especifica la ubicación del fichero descargado “inventario.conf” con su ruta y, además, este se especifica en args[0]: dentro de Eclipse, se puede especificar qué argumentos pasar al programa.

Para ello, se ha entrado en “Run Configurations”, a la parte de arguments. Ahí, se especifica el nombre del archivo “inventario.conf”, por lo que args[0] es igual a “inventario.conf”.



**Properties configuracion = new Properties();**

Después, se crea un objeto de la clase Properties, el cual será muy importante a la hora de desarrollar el programa.

**configuracion.load(new FileInputStream(f));**

Se usa el objeto Properties para llamar al método load, el cual lee el nuevo FileInputStream creado, dentro del cual se pasa el fichero “f” anteriormente creado, donde se pasaba la ruta de “inventario.conf”.

**String clave=configuracion.getProperty("cantidad");**

**int valor=Integer.parseInt(clave);**

**valor++;**

**clave=""+valor;**

Se declara un string clave, donde se pasa, con el objeto Properties, el método getProperty, el cual permite buscar la property especificada dentro de este método.

Luego, se declara un int valor, el cual convierte el string anterior en int, con el objetivo de sumarle una unidad a la cantidad.

Finalmente, se dice que clave es igual a valor más una cadena string vacía.

**configuracion.setProperty("cantidad", clave);**

**configuracion.store(new FileOutputStream(f), "Fichero de cantidades (con la cantidad sumada en una unidad)");**

Con el método setProperty, se fija el valor de la cantidad, que es igual a la clave.

Con el método store, se crea un FileOutputStream, donde se pasa el fichero f y un comentario informativo. Este FileOutputStream guarda los cambios realizados en el fichero, mientras que el FileInputStream simplemente lee.

**FileReader fr=new FileReader(f);**

**int caract=fr.read();**

**while(caract!=-1) {**

**System.out.print((char)caract);**

**caract=fr.read();**

**}**

**fr.close();**

Por último, se ha realizado un FileReader con su bucle while para leer el archivo inventario.conf y ver los cambios realizados.

Se ha realizado esta solución a partir del ejemplo dado en clase, donde se utilizaban estos métodos de la clase Properties, la cual es el eje de lo dado hasta ahora en el tema.