

# TAREA TEMA 10 PRO

17/02/2023

Autor: Derimán Tejera Fumero

## Actividades de comprobación

### 10.1. Una excepción en Java:

- a) Se produce cuando un disco está defectuoso.
- b) Es un valor único de una variable.
- c) Se arroja al sistema cuando se produce una condición anómala durante la ejecución de un programa.
- d) Tiene lugar cuando un código es sintácticamente incorrecto.

### 10.2. Una excepción comprobada es:

- a) Una excepción que hemos reparado.
- b) Una excepción que no detiene la ejecución del programa.
- c) Una excepción previsible, que el propio compilador nos obliga a gestionar.
- d) Una excepción muy conocida.

### 10.3. Cuando llegamos al final de un flujo de entrada de tipo `FileReader`, el método `read()`:

- a) Muestra el mensaje: End of Ríe
- b) Devuelve `null`.
- c) Produce una excepción `EOFException`
- d) Devuelve -1.

### 10.4. La palabra reservada `finally`:

- a) Termina la ejecución de un programa.
- b) Termina la ejecución de un método, forzando el `return`.
- c) En una estructura `try-catch`, fuerza la ejecución de su bloque antes de que se ejecute una sentencia `return` e independientemente de si se produce o no una excepción.
- d) Indica el final de un método.

### 10.5. Un flujo de tipo `BufferedReader`:

- a) Crea un archivo de texto con búfer.
- b) Solo sirve para leer cadenas de caracteres.
- c) Nos permite acceder a archivos binarios.
- d) Accede a un archivo de texto para lectura con búfer.

### 10.6. La Clase `Scanner`:

- a) Solo permite leer texto de cualquier flujo de texto.
- b) Permite digitalizar imágenes.
- c) Permite leer y analizar texto de cualquier flujo de entrada de texto.
- d) Solo nos permite leer de la consola.

### 10.7. Para cambiar de línea al escribir en el flujo salida de tipo `BufferedWriter` debemos ejecutar:

- a) `salida.write("\n")`
- b) `salida.write("\r\n")`
- c) `salida.write("newLine")`
- d) `salida.newLine()`

**10.8. Nos tenemos que asegurar de que todos los flujos abiertos deben cerrarse antes de que termine la aplicación...**

- a) Porque se quedarían abiertos hasta que se apague el ordenador.
- b) Porque otra aplicación podría alterarlos.
- c) Porque se deben liberar los recursos asociados, como los archivos. Además, podrían quedar caracteres del búfer sin escribir.
- d) Porque se pueden borrar datos de un archivo.

**10.9. Los flujos se cierran:**

- a) Con el método `close()`.
- b) Apagando el ordenador.
- c) Abortando el programa.
- d) Con el método `cerrar ()`.

**10.10. Apertura de flujos con recursos:**

- a) Consiste en abrir flujos asociados con varios archivos a la vez.
- b) Es abrir archivos recurriendo a una tabla.
- c) Es una nueva forma de abrir flujos en Java, que permite prescindir del cierre explícito de los archivos y del método `close()`.
- d) Consiste en abrir flujos sin peligro de que se produzcan excepciones.

## ACTIVIDADES DE APLICACIÓN

**10.11. Escribe un programa que solicite al usuario el nombre de un fichero de texto y muestre su contenido en pantalla. Si no se proporciona ningún nombre de fichero, la aplicación utilizará por defecto prueba, txt**

```

18  * @param args the command line arguments
19  */
20  public static void main(String[] args) throws Exception {
21      String texto = "";
22      BufferedReader in = null;
23      Scanner sc = new Scanner(System.in);
24
25      System.out.println("Este programa permite acceder al contenido de un fichero de texto con el nombre dado por el usuario.");
26
27      System.out.print("¿Cuál es el nombre del fichero que deseas abrir? (INCLUIR extensión): ");
28      String nombreFichero;
29      nombreFichero = sc.nextLine();
30      System.out.println("=====");
31
32      File archivo = new File(nombreFichero);
33
34      if (archivo.exists()) {
35          System.out.println("El archivo [" + nombreFichero + "] se ha encontrado con éxito.");
36      } else if (nombreFichero == "") {
37          nombreFichero = "prueba" + ".txt";
38          System.out.println("No se ha proporcionado ningún nombre de archivo, se abrirá el archivo por defecto: " + nombreFichero);
39      } else {
40          System.out.println("No se ha encontrado ningún archivo con ese nombre.");
41          System.out.println("Finalizando programa... ");
42          System.exit(0);
43      }
44
45      try {
46          in = new BufferedReader(new FileReader(nombreFichero));
47
48          int c = in.read();
49          while (c != -1) {
50              texto = texto + (char) c;
51              c = in.read();
52          }
53      } finally {
54          in.close();
55      }
56      System.out.println(texto);
57  }
58
59  //Autor: Derimán Tejera Fumero
60  /*
61  *
62  * Escribe un programa que solicite al usuario el nombre de un fichero de texto y muestre su contenido en pantalla.
63  * Si no se proporciona ningún nombre de fichero, la aplicación utilizará por defecto prueba, txt
64  */

```

```

Output - Actividad 10.11 (run) X
run:
Este programa permite acceder al contenido de un fichero de texto con el nombre dado por el usuario.
¿Cuál es el nombre del fichero que deseas abrir? (INCLUIR extensión): fichero1.txt
=====
El archivo [fichero1.txt] se ha encontrado con éxito.
Contenido fichero1.txt:

      *,,,,,*,*,*
      '0,,,,0,,,,0
      _||_||_||_||_
      '*,,{,,,,,,},*
      0,,{\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/},'0
      _||_{_____}"_____}_||_
{\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/}
{,,,,,,}
{\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/}
{_____}"_____}
      Felicidades,
      has abierto fichero1.txt

Si estás leyendo esto, es que has abierto correctamente el archivo.
BUILD SUCCESSFUL (total time: 11 seconds)
|

```



```

14
15
16 public class Actividad1012 {
17
18     /**
19      * @param args the command line arguments
20      */
21     public static void main(String[] args) {
22
23         System.out.println("Este programa guarda un nombre y una edad introducidas por el usuario en un fichero datos.txt.");
24         Scanner sc = new Scanner(System.in);
25
26         System.out.print("Nombre: ");
27         String nombre;
28         nombre = sc.nextLine();
29         System.out.print("Edad: ");
30         String edad;
31         edad = sc.nextLine();
32         System.out.println("=====");
33
34         BufferedWriter out = null;
35         try {
36             out = new BufferedWriter(new FileWriter("datos.txt", true));
37
38             out.write(nombre, 0, nombre.length());
39             out.write(" ");
40             out.write(edad, 0, edad.length());
41             out.newLine();
42
43         } catch (IOException ex) {
44             System.out.println(ex.getMessage());
45             System.out.println("Ha ocurrido un error.");
46
47         } finally {
48             if (out != null) {
49                 try {
50                     out.close();
51                 } catch (IOException ex) {
52                     System.out.println(ex);
53                 }
54             }
55         }
56     }
57 }
58 //Autor: Derimán Tejera Fumero
59 /**
60  * Diseria una aplicación que pida al usuario su nombre y edad. Estos datos deben guardarse en el fichero datos.txt
61  * Si este fichero existe, deben añadirse al final en una nueva línea, y en caso de no existir, debe crearse.
62  */

```

```

Output - Actividad10.12 (run)
run:
Este programa guarda un nombre y una edad introducidas por el usuario en un fichero datos.txt.
Nombre: Eduardo Rex
Edad: 35
=====
BUILD SUCCESSFUL (total time: 35 seconds)

```

datos.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Napoleón Bonaparte 40  
Julio César 35  
Oscar Wilde 36  
Miguel De Cervantes 75  
asdfs 56  
Edu 23  
Eduardo Rex 35

Línea 1, columna 1 100% Windows (CRLF) UTF-8

**10.13. Implementa un programa que lea dos listas de números enteros no ordenados de sendos archivos con un número por línea, los reúna en una lista única y los guarde en orden creciente en un tercer archivo, de nuevo uno por línea.**

```
71         if (izquierda < derecha) {
72             int indice = dividir(tabla, izquierda, derecha);
73             ordenar(tabla, izquierda, indice - 1);
74             ordenar(tabla, indice + 1, derecha);
75         }
76     }
77 }
78
79 private static int dividir(int tabla[], int izquierda, int derecha) {
80     int indicePivote = tabla[derecha];
81     int y = izquierda - 1;
82     for (int x = izquierda; x <= derecha - 1; x++) {
83
84         if (tabla[x] < indicePivote) {
85             y++;
86             intercambiar(tabla, y, x);
87         }
88     }
89     intercambiar(tabla, y + 1, derecha);
90
91     return y + 1;
92 }
93
94 private static void intercambiar(int tabla[], int x, int y) {
95     int temp = tabla[x];
96     tabla[x] = tabla[y];
97     tabla[y] = temp;
98 }
99
100 private static void escribirNumerosArchivo(String nombreArchivo, int tabla[]) {
101     try (BufferedWriter writer = new BufferedWriter(new FileWriter(nombreArchivo))) {
102         for (int num : tabla) {
103             writer.write(num + "\n");
104         }
105         System.out.println("Escritos los números en el fichero " + nombreArchivo + "[COMPLETADO]");
106     } catch (IOException e) {
107         e.printStackTrace();
108         System.out.println("Ha ocurrido un error.");
109     }
110 }
111 }
112
113 //Autor: Derimán Tejeras Fumero
114 /*
115 Implementa un programa que lea dos listas de números enteros no ordenados de sendos archivos con un número por línea,
116 los reúna en una lista única y los guarde en orden creciente en un tercer archivo, de nuevo uno por línea.
117 */
118
```

```
Output - Actividad10.13 (run) X
run:
Leídos los números de ficheros numeros1.txt[COMPLETADO]
Leídos los números de ficheros numeros2.txt[COMPLETADO]
Fusionando tablas [COMPLETADO]
Este programa lee el contenido de dos ficheros (numeros1.txt y numeros2.txt) los números que contienen y los ordena en un tercer fichero (numerosOrdenados.txt).
Escritos los números en el fichero numerosOrdenados.txt[COMPLETADO]
Finalizando programa.
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
numeros1.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
54
20
12
45
7
49
9
1
2
8
4
7
85
15
546
125
45
121
2547
```

```
numeros2.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
58
45
8
12
789
45
124
12
46
4
5
3
2
1
```

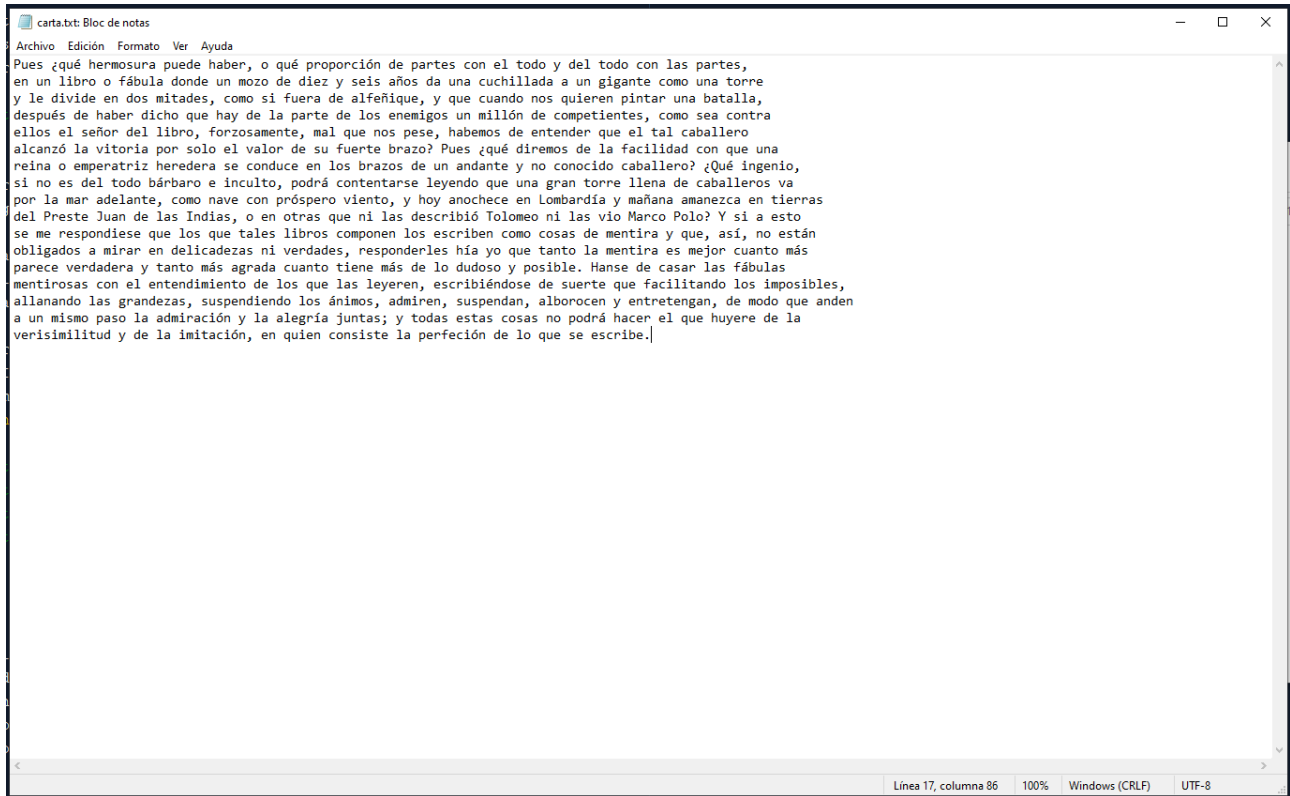


**10.14. Escribe un programa que lea un fichero de texto llamado carta.txt. Tenemos que contar los caracteres, las líneas y las palabras. Para simplificar supondremos que cada palabra está separada de otra por un único espacio en blanco o por un cambio de línea.**

```
18  *param args the command line arguments
19  */
20  public static void main(String[] args) {
21      String fichero = "carta.txt";
22      int caracteres = 0;
23      int lineas = 0;
24      int palabras = 0;
25
26      System.out.println("Este programa contabiliza los caracteres, líneas y palabras de un texto contenido en carta.txt.");
27
28      try {
29          BufferedReader lector = new BufferedReader(new FileReader(fichero));
30          String linea;
31          while ((linea = lector.readLine()) != null) {
32              caracteres += linea.length();
33              lineas++;
34              palabras += contarPalabras(linea);
35          }
36          lector.close();
37      } catch (IOException e) {
38          System.out.println("Error al leer el archivo " + fichero + ": " + e.getMessage());
39          return;
40      }
41      System.out.println("Datos del archivo [" + fichero + "]:");
42      System.out.println("1. Caracteres: " + caracteres);
43      System.out.println("2. Líneas: " + lineas);
44      System.out.println("3. Palabras: " + palabras);
45
46  }
47
48  public static int contarPalabras(String linea) {
49      String palabras[] = linea.split("[ \\n]");
50      int contador = 0;
51      for (String palabra : palabras) {
52          if (!palabra.equals("")) {
53              contador++;
54          }
55      }
56      return contador;
57  }
58
59  }
60  //Autor: Derimán Tejera Fumero.
61  /*
62  Escribe un programa que lea un fichero de texto llamado carta.txt.
63  Tenemos que contar los caracteres, las líneas y las palabras. Para
64  simplificar supondremos que cada palabra está separada de otra por
65  un único espacio en blanco o por un cambio de línea.
66  */
```



```
Output - Actividad10.14 (run) X
run:
Este programa contabiliza los caracteres, líneas y palabras de un texto contenido en carta.txt.
Datos del archivo [carta.txt]
1. Caracteres: 1766
2. Líneas: 17
3. Palabras: 320
BUILD SUCCESSFUL (total time: 0 seconds)
```



**10.15. En el archivo números, txt disponemos de una serie de números (uno por cada línea). Diseña un programa que procese el fichero y nos muestre el menor y el mayor.**

```

5 package actividad10.pkg15;
6
7 import java.io.BufferedReader;
8 import java.io.FileReader;
9 import java.io.IOException;
10
11 /**
12  *
13  * @author Usuario
14  */
15 public class Actividad1015 {
16
17     /**
18      * @param args the command line arguments
19      */
20     public static void main(String[] args) {
21
22         String fichero = "numeros.txt";
23         int numeroMenor = Integer.MIN_VALUE;
24         int numeroMayor = Integer.MIN_VALUE;
25
26         try {
27             System.out.println("Este programa muestra el menor y el mayor de los números contenidos en el fichero numeros.txt.");
28             BufferedReader lector = new BufferedReader(new FileReader(fichero));
29             String linea;
30             while ((linea = lector.readLine()) != null) {
31                 int numero = Integer.parseInt(linea.trim());
32                 if (numero < numeroMenor) {
33                     numeroMenor = numero;
34                 }
35                 if (numero > numeroMayor) {
36                     numeroMayor = numero;
37                 }
38             }
39             lector.close();
40             System.out.println("El menor número del fichero es: " + numeroMenor);
41             System.out.println("El mayor número del fichero es: " + numeroMayor);
42         } catch (IOException e) {
43             System.out.println("Error al leer el fichero " + fichero + ": " + e.getMessage());
44         }
45     }
46 }
47
48 //Autor: Derimán Tejera Fumero.
49 /*
50 En el archivo números, txt disponemos de una serie de números (uno por cada línea).
51 Diseña un programa que procese el fichero y nos muestre el menor y el mayor.
52 */

```

```

Output - actividad10.15 (run) X
run:
Este programa muestra el menor y el mayor de los números contenidos en el fichero numeros.txt.
El menor número del fichero es: -80
El mayor número del fichero es: 1000000
BUILD SUCCESSFUL (total time: 0 seconds)

```

numeros.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

```

5
4
2
4
5
-8
788
45
123
128
78
954
56
12
0
789
45
12156
545
1000000
50
45
-80
-78
89

```

Línea 1, columna 1 100% Windows (CRLF) UTF-8

**10.16. Un libro de firmas es útil para recoger los nombres de todas las personas que han pasado por un determinado lugar. Crea una aplicación que permita mostrar el libro de firmas e insertar un nuevo nombre (comprobando que no se encuentre repetido). Llamaremos al fichero firmas.txt.**

```
Output - Actividad10.16BIS (run) X
run:
MENU DE OPCIONES3
1. Mostrar firmas
2. Añadir firma
3. Salir
Seleccione una opción:1

Las firmas almacenadas son:
[1] Paco Ruiz
[2] Jose Luis García
[3] Topo Topo
[4] Reno Renardo
[5] Pablo Fernández
[6] Juana de Arco
[7] Mozart
[8] asdfa
[9] Olaf Stuth
[10] asdfdsa
[11] Gato con botas
[12] Gustavo Maalaf
[13] Anibal Barca
[14] Papa XIII

93     for (int i = 0; i < MAX_FIRMAS; i++) {
94         if (firmas[i] != null) {
95             in.write(firmas[i]);
96             in.newLine();
97         }
98     }
99     } catch (IOException e) {
100         System.out.println("Error al guardar el fichero de firmas: " + e.getMessage());
101     }
102 }
103
104 private static void mostrarFirmas() {
105     System.out.println("");
106     System.out.println("");
107     System.out.println("Las firmas almacenadas son: ");
108     for (int i = 0; i < MAX_FIRMAS; i++) {
109         if (firmas[i] != null) {
110             System.out.println("[ " + (i + 1) + " ] " + firmas[i]);
111         }
112     }
113     System.out.println("");
114     System.out.println("");
115 }
116
117 private static void agregarFirma(String nombre) {
118     for (int i = 0; i < MAX_FIRMAS; i++) {
119         if (firmas[i] == null) {
120             firmas[i] = nombre;
121             break;
122         }
123     }
124 }
125
126 private static int buscarFirma(String nombre) {
127     for (int i = 0; i < MAX_FIRMAS; i++) {
128         if (firmas[i] != null && firmas[i].equalsIgnoreCase(nombre)) {
129             return i;
130         }
131     }
132     return -1;
133 }
134 }
135 //Autor: Derián Tejera Fumero.
136 /*
137 Un libro de firmas es útil para recoger los nombres de todas las personas que
138 han pasado por un determinado lugar. Crea una aplicación que permita mostrar
139 el libro de firmas e insertar un nuevo nombre (comprobando que no se encuentre repetido).
140 Llamaremos al fichero firmas.txt.
141 */
```

```
MENU DE OPCIONES3
1. Mostrar firmas
2. Añadir firma
3. Salir
Seleccione una opción:1
```

Las firmas almacenadas son:

```
[1] Paco Ruiz
[2] Jose Luis García
[3] Topo Topo
[4] Reno Renardo
[5] Pablo Fernández
[6] Juana de Arco
[7] Mozart
[8] asdfa
[9] Olaf Stuth
[10] asdfdsa
[11] Gato con botas
[12] Gustavo Maalaf
[13] Anibal Barca
[14] Papa XIII
[15] Andy Smith
```

```
MENU DE OPCIONES3
1. Mostrar firmas
2. Añadir firma
3. Salir
Seleccione una opción:2
Introduzca un nombre:Andy Smith
El nombre ya está en la lista.
```

```
MENU DE OPCIONES3
1. Mostrar firmas
2. Añadir firma
3. Salir
Seleccione una opción:1
```

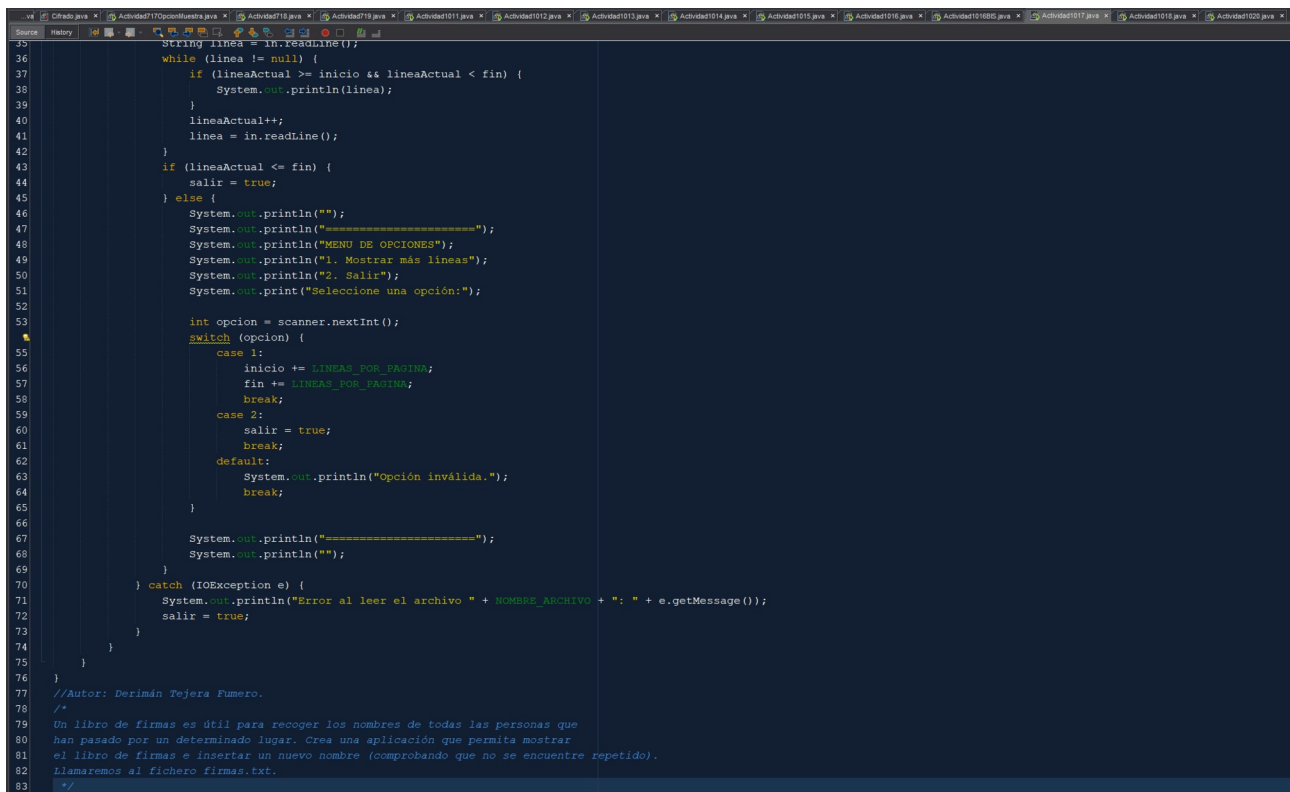
Las firmas almacenadas son:

```
[1] Paco Ruiz
[2] Jose Luis García
[3] Topo Topo
[4] Reno Renardo
[5] Pablo Fernández
[6] Juana de Arco
[7] Mozart
[8] asdfa
[9] Olaf Stuth
[10] asdfdsa
[11] Gato con botas
[12] Gustavo Maalaf
[13] Anibal Barca
[14] Papa XIII
[15] Andy Smith
```



**10.17.** En Linux disponemos del comando `more`, al que se le pasa un fichero y lo muestra poco a poco: cada 24 líneas. Implementa un programa que funcione de forma similar.

**10.18.** Escribe la función `Integer[] leerEnteros(String texto)`, al que se le pasa una cadena y devuelve una tabla con todos los enteros que aparecen en ella.



```
Output - Actividad10.17 (run) X
run:
Wolfgang Amadeus Mozart
(Extracto de biografíA en Wikipedia: https://es.wikipedia.org/wiki/Wolfgang_Amadeus_Mozart)

Johannes Chrysostomus Wolfgangus Theophilus Mozarta? (Salzburgo, 27 de enero de 1756-Viena, 5 de diciembre de 1791), mUs conocido como Wolfgang Amadeus Mozart, fue un compositor, pianista, director de orquesta y profesor del antiguo Arzobispado de Salzburgo (anteriormente parte del Sacro Imperio Romano Germánico, actualmente parte de Austria), maestro del Clasicismo, considerado como uno de los mUsicos mUs influyentes y destacados de la historia.

La obra mozartiana abarca todos los gUneros musicales de su Epoca e incluye mUs de seiscientas creaciones, en su mayorUa reconocidas como obras maestras de la mUsica sinfÓnica, concertante, de cÓmara, para fortepiano, operÓtica y coral, logrando una popularidad y difusiÓn internacional.

En su niDez mUs temprana en Salzburgo, Mozart mostrÓ una capacidad prodigiosa en el dominio de instrumentos de teclado y del violÓn. Con tan solo cinco aDos ya componíA obras musicales y sus interpretaciones eran del aprecio de la aristocracia y realza europea. A los diecisiete aDos fue contratado como mUsico en la corte de Salzburgo, pero su inquietud le llevÓ a viajar en busca de una mejor posiciÓn, siempre componiendo de forma prolÓfica. Durante su visita a Viena en 1781, tras ser despedido de su puesto en la corte, decidiÓ instalarse en esta ciudad, donde alcanzÓ la fama que mantuvo el resto de su vida, a pesar de pasar por situaciones financieras difÍciles. En sus aDos finales, compuso muchas de sus sinfonÍas, conciertos y Óperas mUs conocidas, asÍ como su RÓquiem. Las circunstancias de su temprana muerte han sido objeto de numerosas especulaciones y elevadas a la categorÍa de mito.

SegÚn crÍticos de mUsica como Nicholas Till, Mozart siempre aprendíA vorazmente de otros mUsicos y desarrollÓ un esplendor y una madurez de estilo que abarcÓ desde la luz y la elegancia, a la oscuridad y la pasiÓn Ótodo bien fundado por una visiÓn de la humanidad Óredimida por el arte, perdonada y reconciliada con la naturaleza y lo absolutoÓÓ.1? Su influencia en toda la mUsica occidental posterior es profunda; Ludwig van Beethoven escribiÓ sus primeras composiciones a la sombra de Mozart, de quien Joseph Haydn escribiÓ que Óla posteridad no verÍ tal talento otra vez en cien aDosÓ.

|-----|
|BiografÍa|

=====
MENU DE OPCIONES
1. Mostrar mUs lÓneas
2. Salir
Seleccione una opciÓn:
```

```
Output - Actividad10.17 (run) X
La obra mozartiana abarca todos los gUneros musicales de su Epoca e incluye mUs de seiscientas creaciones, en su mayorUa reconocidas como obras maestras de la mUsica sinfÓnica, concertante, de cÓmara, para fortepiano, operÓtica y coral, logrando una popularidad y difusiÓn internacional.

En su niDez mUs temprana en Salzburgo, Mozart mostrÓ una capacidad prodigiosa en el dominio de instrumentos de teclado y del violÓn. Con tan solo cinco aDos ya componíA obras musicales y sus interpretaciones eran del aprecio de la aristocracia y realza europea. A los diecisiete aDos fue contratado como mUsico en la corte de Salzburgo, pero su inquietud le llevÓ a viajar en busca de una mejor posiciÓn, siempre componiendo de forma prolÓfica. Durante su visita a Viena en 1781, tras ser despedido de su puesto en la corte, decidiÓ instalarse en esta ciudad, donde alcanzÓ la fama que mantuvo el resto de su vida, a pesar de pasar por situaciones financieras difÍciles. En sus aDos finales, compuso muchas de sus sinfonÍas, conciertos y Óperas mUs conocidas, asÍ como su RÓquiem. Las circunstancias de su temprana muerte han sido objeto de numerosas especulaciones y elevadas a la categorÍa de mito.

SegÚn crÍticos de mUsica como Nicholas Till, Mozart siempre aprendíA vorazmente de otros mUsicos y desarrollÓ un esplendor y una madurez de estilo que abarcÓ desde la luz y la elegancia, a la oscuridad y la pasiÓn Ótodo bien fundado por una visiÓn de la humanidad Óredimida por el arte, perdonada y reconciliada con la naturaleza y lo absolutoÓÓ.1? Su influencia en toda la mUsica occidental posterior es profunda; Ludwig van Beethoven escribiÓ sus primeras composiciones a la sombra de Mozart, de quien Joseph Haydn escribiÓ que Óla posteridad no verÍ tal talento otra vez en cien aDosÓ.

|-----|
|BiografÍa|

=====
MENU DE OPCIONES
1. Mostrar mUs lÓneas
2. Salir
Seleccione una opciÓn:1
=====

|-----|

Familia e infancia
*****

Wolfgang Amadeus Mozart naciÓ el 27 de enero de 1756 en Salzburgo, en la actual Austria, que en esa Epoca era un arzobispado independiente del Sacro Imperio Romano Germánico. Fue el Óltimo hijo de Leopold Mozart, mUsico al servicio del prÍncipe arzobispo de Salzburgo. Leopold era el segundo maestro de capilla en la corte del arzobispo aunque fue un experimentado profesor. Su madre se llamaba Anna Maria Pertl. Debido a la altÍsima mortalidad infantil en la Europa de la Epoca, de los siete hijos que tuvo el matrimonio solo sobrevivieron Maria Anna, apodada cariÓsamente Nannerl, y Wolfgang Amadeus. Fue bautizado en la catedral de San Ruperto el dÍa despuÍs de su nacimiento con los nombres de Joannes Chrysostomus Wolfgangus Theophilus Mozart; a lo largo de su vida firmarÍa con diversas variaciones sobre su nombre original, siendo una de las mUs recurrentes ÓWolfgang AmadÓ MozartÓ.3?

La casa natal de Mozart se encuentra en la Getseidegasse de la ciudad de Salzburgo. Se trata de una casa que actualmente cuenta con una gran cantidad de objetos de la Epoca e instrumentos que pertenecieron a Mozart durante su niDez. Es uno de los lugares mUs visitados de Salzburgo y una especie de santuario para mUsicos y aficionados a la mUsica de todo el mundo.4?

Leopold componíA y daba clases de mUsica. El aDo del nacimiento de Wolfgang publicÓ un exitoso tratado para la interpretaciÓn del violÓn titulado Versuch einer grÚndlichen Violinschule. DespuÍs del nacimiento de Wolfgang abandonÓ todo, salvo las tareas propias de su cargo, para dedicarse de manera exclusiva a la formaciÓn de su hijo. Fue exigente como padre y como profesor y en todo momento estuvo al tanto de la formaciÓn de Wolfgang, para guiarlo como hombre y como artista.

Nannerl y Wolfgang Amadeus mostraron desde muy pequeÑos facultades para la mUsica. Nannerl comenzÓ a recibir clases de teclado con su padre cuando tenÍa siete aDos, y su hermano, cuatro aDos y medio menor que ella, la miraba evidentemente fascinado. ADos despuÍs de la muerte de su hermano, ella rememorÓ:

A menudo pasaba mucho tiempo en el teclado, eligiendo terceras, que a Ól siempre le sorprendÍan y mostraba con placer que el sonido le gustaba. [...] En su cuarto

=====
MENU DE OPCIONES
1. Mostrar mUs lÓneas
2. Salir
Seleccione una opciÓn:
```

**10.18. Escribe la función Integer[] leerEnteros(String texto), al que se le pasa una cadena y devuelve una tabla con todos los enteros que aparecen en ella.**

```
34         if (!Character.isDigit(c)) {
35             if (c == ',' || c == '.') {
36                 esEntero = false;
37                 break;
38             }
39             if (c != ' ') {
40                 esEntero = false;
41                 break;
42             }
43         }
44     }
45     if (esEntero) {
46         contador++;
47     }
48 }
49
50 int enteros[] = new int[contador];
51 contador = 0;
52 for (int j = 0; j < palabras.length; j++) {
53     String palabra = palabras[j];
54     if (palabra.length() > 0 && (Character.isDigit(palabra.charAt(0)) || (palabra.charAt(0) == '-' && palabra.length() > 1 && Character.isDigit(palabra.charAt(1))))) {
55         boolean esEntero = true;
56         for (int i = 1; i < palabra.length(); i++) {
57             char c = palabra.charAt(i);
58             if (!Character.isDigit(c)) {
59                 if (c == ',' || c == '.') {
60                     esEntero = false;
61                     break;
62                 }
63                 if (c != ' ') {
64                     esEntero = false;
65                     break;
66                 }
67             }
68         }
69         if (esEntero) {
70             enteros[contador] = Integer.parseInt(palabra);
71             contador++;
72         }
73     }
74 }
75 return enteros;
76 }
77
78 //Autor: Derimin Tejera Fumero.
79 /*
80 Escribe la función Integer[] leerEnteros(String texto), al que se le pasa una cadena y devuelve una tabla con todos los enteros que aparecen en ella.
81 */
```

```
Output - Actividad10.18 (run) X
run:
Los números enteros encontrados en la cadena de texto son: [24, 2082, 22, -5]
BUILD SUCCESSFUL (total time: 0 seconds)
```

La cadena de texto es: String texto = "Hoy 24 de Marzo de 2082 la temperatura es de unos agradables 22 grados Celcius, o lo que es lo mismo, 71,6 grados Fahrenheit o 295.15 grados Kelvin. En contraste, ayer hizo -5 grados Celcius.";

**10.20. Algunos sistemas operativos disponen de la orden comp. que compara dos archivos y nos dice si son iguales o distintos. Diseña esta orden de forma que. además, nos diga en qué línea y carácter se encuentra la primera diferencia. Utiliza los ficheros texto1.txt y texto2.txt**

```
37 }
38 }
39
40 if (contenidoArchivo1.length != contenidoArchivo2.length) {
41     System.out.println("Los ficheros tienen diferente cantidad de lineas.");
42 } else {
43     System.out.println("Los ficheros son idénticos.");
44 }
45
46 } catch (IOException e) {
47     System.out.println("Error al leer los ficheros.");
48     e.printStackTrace();
49 }
50
51 }
52
53 private static String leerFichero (String nombreFichero) [] throws IOException {
54     BufferedReader lector = new BufferedReader(new FileReader(nombreFichero));
55     String linea;
56     String contenido[] = new String[0];
57
58     while ((linea = lector.readLine()) != null) {
59         String nuevoContenido[] = new String[contenido.length + 1];
60         System.arraycopy(contenido, 0, nuevoContenido, 0, contenido.length);
61         nuevoContenido[contenido.length] = linea;
62         contenido = nuevoContenido;
63     }
64
65     lector.close();
66     return contenido;
67 }
68
69 private static int buscarPrimeraDiferencia(String linea1, String linea2) {
70     int longitud = Math.min(linea1.length(), linea2.length());
71
72     for (int i = 0; i < longitud; i++) {
73         if (linea1.charAt(i) != linea2.charAt(i)) {
74             return i;
75         }
76     }
77
78     return longitud;
79 }
80
81 //Autor: Derimán Tejera Fumero.
82 /*
83 Algunos sistemas operativos disponen de la orden comp. que compara dos archivos y nos dice si son iguales
84 o distintos. Diseña esta orden de forma que, además, nos diga en qué línea y carácter se encuentra la
85 primera diferencia. Utiliza los ficheros texto1.txt y texto2.txt
86 */
```

```
run:
Este programa muestra por pantalla la línea y número de carácter donde difieren 2 textos dados en los ficheros texto1.txt y texto2.txt.
La primera diferencia está en la línea [3], carácter [36].
BUILD SUCCESSFUL (total time: 0 seconds)
```

texto1.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Wolfgang Amadeus Mozart

Johannes Chrysostomus Wolfgangus Theophilus Mozarta (Salzburgo, 27 de enero de 1756-Viena, 5 de diciembre de 1791), más conocido como Wolfgang Amadeus Mozart, fue un compositor, pianista, director de orquesta y profesor del antiguo Arzobispado de Salzburgo (anteriormente parte del Sacro Imperio Romano Germánico, actualmente parte de Austria), maestro del Clasicismo, considerado como uno de los músicos más influyentes y destacados de la historia.

Línea 1, columna 1 100% Windows (CRLF) UTF-8



