

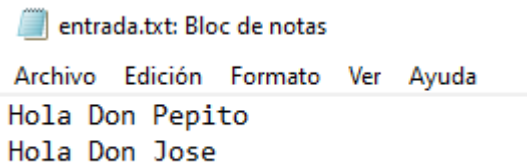
EJERCICIO 116 – ejer116.java

Desarrolla un programa que, dado cualquier fichero en la entrada, escriba por pantalla estadísticas básicas del mismo.

En concreto, escribirá número de palabras, número de caracteres totales del texto (sin contar los espacios) y la longitud media de una palabra medida en número de caracteres.

Supondremos que las palabras solo estan separadas por espacios simples, sin puntos, comas, ni signos de puntuación.

Por ejemplo, para el siguiente fichero de texto:



```
entrada.txt: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
Hola Don Pepito
Hola Don Jose
```

El programa devolvería las siguientes estadísticas:

```
<terminated> ejer116 [Java Application] C:\Program Files\JAVA\openjdk-12.0.1_
Numero de palabras: 6
Numero de caracteres: 24
Longitud media de una palabra:4.0
```

EJERCICIO 117 – ejer117.java

Crea un fichero de texto con el nombre y contenido que tu quieras. Ahora crea una aplicación que lea este fichero de texto y muestre su contenido por pantalla sin espacios.

En el ejemplo anterior, el programa devolvería:

```
<terminated> ejer117 [Java Application] C:\Program Files\JAVA\openjdk-12.0.1_
HolaDonPepito
HolaDonJose
```

EJERCICIO 118 – ejer118.java

Crea un fichero de texto con el nombre y contenido que tu quieras. Ahora crea una aplicación que lea este fichero de texto y muestre por pantalla el mismo texto pero variando entre mayúsculas y minúsculas.

Supondremos que las palabras solo estan separadas por espacios simples, sin puntos, comas, ni signos de puntuación.

En el ejemplo anterior, el programa devolvería:

```
<terminated> ejer118 [Java Application] C:\Program Files\JAVA\openjdk-12.0.1_
hOLA dON pEPITO
hOLA dON jOSE
```

EJERCICIO 119 – ejer119.java

Construir un programa que permita buscar palabras en un fichero de texto. La palabra debe aparecer exactamente igual, coincidiendo mayúsculas y minúsculas. Se debe mostrar el número de línea y su contenido, para cada línea que contenga la palabra buscada. Si no existe en todo el fichero, también se debe informar de esta circunstancia.

Para el fichero siguiente:

```
entrada.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Hola Don Pepito
Hola Don Jose
Paso usted por mi casa por su casa yo pase
Vio usted a mi abuela a su abuela yo la vi
Adios Don Pepito
Adios Don Jose|
```

Si quisiéramos buscar una palabra que no existe...

```
<terminated> ejer119 [Java Application] C:\Program Files\JAVA\openjdk-12.0.1_
¿Qué palabra quieres buscar en el fichero de entrada?
don
La palabra don no se encuentra en el fichero.
```

Y si la palabra sí que existe...

```
<terminated> ejer119 [Java Application] C:\Program Files\JAVA\openjdk-12.0.1_
¿Qué palabra quieres buscar en el fichero de entrada?
Pepito
Encontrado en la línea: 1
Hola Don Pepito
Encontrado en la línea: 5
Adios Don Pepito
```

EJERCICIO 120 – ejer120.java

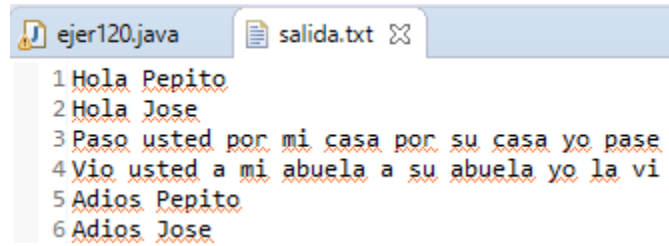
Desarrolla un programa que permita eliminar todas las ocurrencias de una palabra dada en un fichero de texto. El resultado se escribirá en un fichero de salida.

Supondremos que las palabras solo estan separadas por espacios simples, sin puntos, comas, ni signos de puntuación.

Por ejemplo, si el usuario introduce la siguiente información:

```
<terminated> ejer120 [Java Application] C:\Program Files\JAVA\openjdk-12.0.1_v
Introduzca palabra a borrar del fichero de entrada:
Don
```

El resultado sería, con el fichero del ejercicio anterior:



The screenshot shows a Java IDE window with two tabs: 'ejer120.java' and 'salida.txt'. The 'salida.txt' tab is active, displaying the output of the program. The output consists of six lines of text, each preceded by a line number from 1 to 6. The text is as follows:

```
1 Hola Pepito
2 Hola Jose
3 Paso usted por mi casa por su casa yo pase
4 Vio usted a mi abuela a su abuela yo la vi
5 Adios Pepito
6 Adios Jose
```