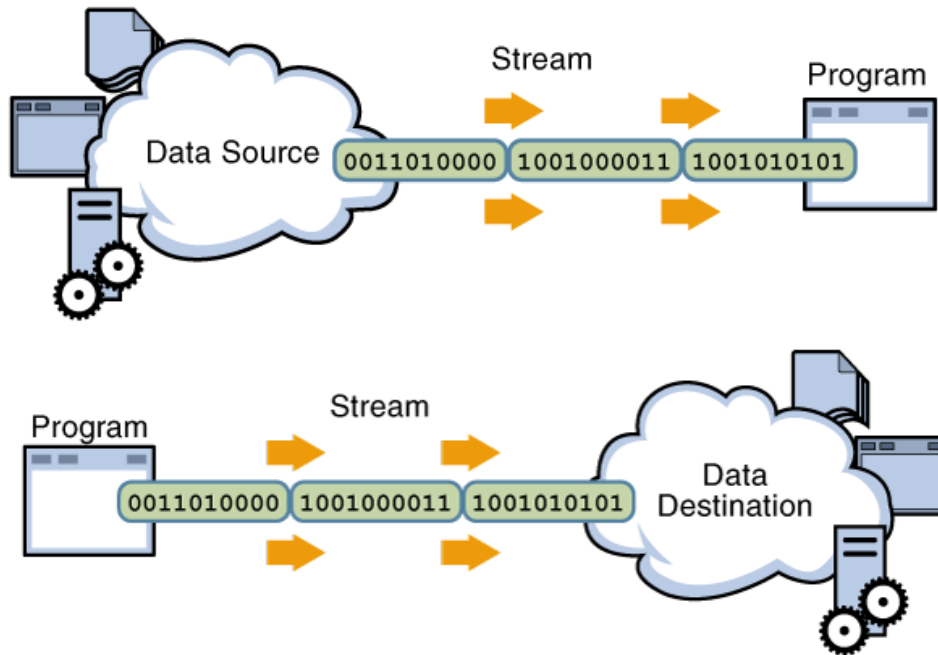


Ejercicios 2 de la UD06



1. **Ejercicios**
2. **Fuentes de información**

1. Ejercicios

1. (`CuentaLineas`) Escribe un programa que, sin utilizar la clase `Scanner`, muestre el número de líneas que contiene un fichero de texto. El nombre del fichero se solicitará al usuario al comienzo de la ejecución.
2. (`CuentaPalabras`) Escribe un programa que, sin utilizar la clase `Scanner`, muestre el número de palabras que contiene un fichero de texto. El nombre del fichero se solicitará al usuario al comienzo de la ejecución.

Sugerencia: Lee el fichero, línea a línea y utiliza la clase `StringTokenizer` o bien el método `split` de la clase `String` para averiguar el nº de palabras.

3. (`Censura`) Escribir un programa que sustituya por otras, ciertas palabras de un fichero de texto. Para ello, se desarrollará y llamará al método `void aplicaCensura(String entrada, String censura, String salida)`, que lee de un fichero de entrada y mediante un fichero de censura, crea el correspondiente fichero modificado. Por ejemplo:

Fichero de entrada:

```
1 | En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho
   | tiempo que vivía un hidalgo de los de lanza en astillero
```

Fichero de censura:

```
1 | lugar sitio
2 | quiero debo
3 | hidalgo noble
```

Fichero de salida:

```
1 | En un sitio de la Mancha, de cuyo nombre no debo acordarme, no ha mucho
   | tiempo que vivía un noble de los de lanza en astillero
```

Valora la posibilidad de cargar el fichero de censura en un mapa o par clave, valor.

4. (`Concatenar1`) Escribe un programa que dados dos ficheros de texto `f1` y `f2` confeccione un tercer fichero `f3` cuyo contenido sea el de `f1` y a continuación el de `f2`.
5. (`Concatenar2`) Escribe un programa que dados dos ficheros de texto `f1` y `f2`, añada al final de `f1` el contenido de `f2`. Es decir, como el ejercicio anterior, pero sin producir un nuevo fichero.
6. (`Iguales`) Escribir un programa que compruebe si el contenido de dos ficheros es idéntico. Puesto que no sabemos de qué tipo de ficheros se trata, (de texto, binarios, ...) habrá que hacer una comparación byte por byte.
7. Escribe los siguientes métodos y programa:
 - Método `void generar()` que genere 20 números aleatorios enteros entre 1 y 100 y los muestre por pantalla.
 - Método `void media()` que lea de teclado 20 números enteros y calcule su media.

- Programa que, modificando la entrada y la salida estándar, llame a `generar()` para que los datos se graben en un fichero y a continuación llame a `media()` de manera que se tomen los datos del fichero generado.
8. (Notas) Escribir un programa que almacene en un fichero binario (`notas.dat`) las notas de 20 alumnos. El programa tendrá el siguiente funcionamiento:
- En el fichero se guardarán como máximo 20 notas, pero se pueden guardar menos. El proceso de introducción de notas (y en consecuencia, el programa) finalizará cuando el usuario introduzca una nota no válida (menor que cero o mayor que 10).
 - Si, al comenzar la ejecución, el fichero ya contiene notas, se indicará al usuario cuántas faltan por añadir y las notas que introduzca el usuario se añadirán a continuación de las que hay.
 - Si, al comenzar la ejecución, el fichero ya contiene 20 notas, se le preguntará al usuario si desea sobrescribirlas. En caso afirmativo las notas que introduzca sustituirán a las que hay y en caso negativo el fichero no se modificará.

2. Fuentes de información

- [Wikipedia](#)
- [Programación \(Grado Superior\) - Juan Carlos Moreno Pérez \(Ed. Ra-ma\)](#)
- Apuntes IES Henri Matisse (Javi García Jimenez?)
- Apuntes AulaCampus
- [Apuntes José Luis Comesaña](#)
- [Apuntes IOC Programació bàsica \(Joan Arnedo Moreno\)](#)
- [Apuntes IOC Programació Orientada a Objectes \(Joan Arnedo Moreno\)](#)
- [Apuntes Lionel](#)