

## Práctica 2. Ficheros de texto.

Esta práctica tiene como objetivo aprender cómo se trabaja con ficheros de texto utilizando el lenguaje C++.

### Recordatorio.

En primer lugar hay que destacar que, para trabajar con ficheros, siempre debemos incluir el fichero de cabecera `fstream`:

```
#include <fstream>
```

Lo primero que debemos hacer para trabajar con ficheros en C++, es declarar variables de tipo fichero, es decir, ficheros lógicos. La declaración dependerá de si queremos trabajar con **ficheros de entrada** o **ficheros de salida**.

```
// Declaración de un fichero de entrada
ifstream <fichLogico>;
// Declaración de un fichero de salida
ofstream <fichLogico>;
```

A continuación debemos crear una conexión lógica entre fichero lógico y el fichero físico para lo cual se utiliza la siguiente instrucción:

```
<fichLogico>.open(<nombreFicheroFisico>,modo);
```

En la anterior instrucción hay que tener en cuenta que:

- El nombre del fichero físico puede venir dado en una **variable**, o puede ser **constante**, en este último caso se escribirá entre **dobles comillas**. Notar también que se puede trabajar con rutas relativas o absolutas. En caso de querer dar la ruta absoluta deberemos utilizar `\\`, por ejemplo: `"C:\\\\DATOS.DAT"`.
- Los modos disponibles son: `ios::in` (para leer del fichero), `ios::out` (para escribir en el fichero, se sobrescribe), `ios::app` (para añadir datos al final del fichero), e `ios::binary` (para trabajar con ficheros binarios).

Una vez que se ha establecido la conexión lógica entre el fichero lógico y el físico, podemos escribir y leer datos del fichero:

```
// Escribir datos en un fichero
<fichLogico> << <expresión>;
// Leer datos de un fichero
<fichLogico> >> <nombreVariable>;
```

En caso de que queramos comprobar si se ha llegado al final del fichero, utilizaremos el método:

```
<fichLogico>.eof();
```

**Recordar** que para utilizar el método anterior es necesario haber leído anteriormente del fichero.

Por último, una vez que se ha terminado de trabajar con el fichero se debe cerrar la conexión utilizando la instrucción:

```
<fichLogico>.close();
```

## **Ejercicios.**

1. Diseña un subprograma que cree un fichero de texto solicitando enteros positivos por teclado (termina con la introducción de un número menor o igual que 0). De los enteros leídos sólo se almacenarán en el fichero los que sean pares y, al escribirlos en el fichero, se separarán por saltos de línea. El nombre del fichero debe ser un parámetro del subprograma.
2. Construye un subprograma que muestre por pantalla el contenido de un fichero de texto cuyos datos son enteros. El nombre del fichero debe ser un parámetro del subprograma.
3. Construye un subprograma que dados dos ficheros de texto cuyos datos sean enteros añada, al final del primer fichero, todos los elementos del segundo fichero.
4. Construye un subprograma que, a partir de dos ficheros de texto cuyos datos son enteros y que están ordenados crecientemente, cree un nuevo fichero de texto, también ordenado, y que contenga TODOS los datos de los dos ficheros de partida.
5. Construye subprogramas para cada uno de los cinco primeros apartados del Ejercicio 10 de la Hoja 2 (ejercicios de ficheros).