

# PROGRAMACION 1

ING. OTTO PARRA GONZALEZ

JUNIO DEL 2019



# ARCHIVOS

PROGRAMACION 1



# INTRODUCCION

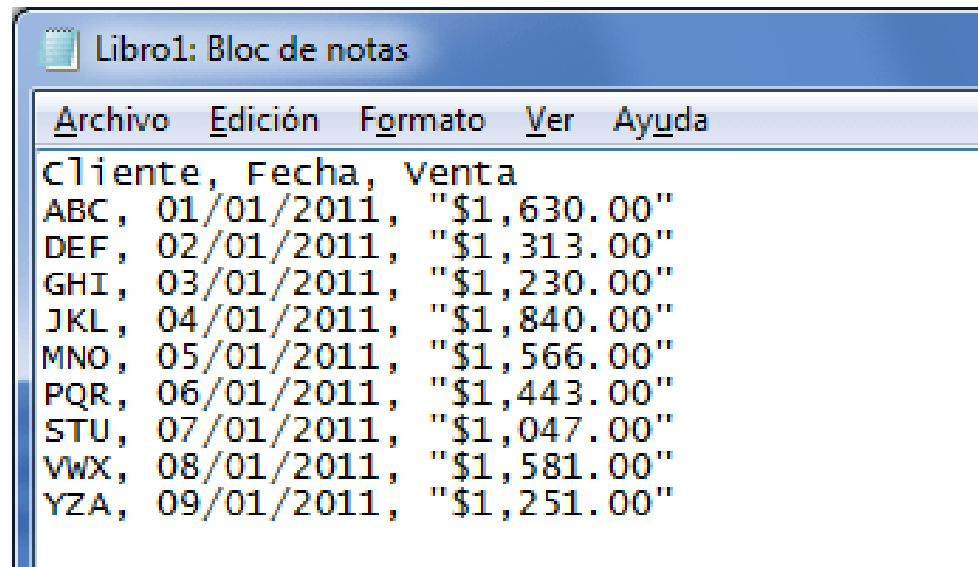
- Los datos de los problemas que se han estudiado hasta ahora en el curso han residido en la memoria principal del computador.
- Sin embargo, las grandes cantidades de datos se almacenan normalmente en un dispositivo de memoria secundaria. Estas colecciones de datos se conocen como archivos.
- La importancia del almacenamiento de dicha información es obvia: envío de información a otros usuarios, posponer el trabajo varios días o semanas sin tener que introducir manualmente los datos de nuevo, acceso a información almacenada en sistemas remotos, etc. Incluso para desarrollos de software de relativamente corta longitud resulta relevante la gestión de datos, por ahorrar una cantidad de tiempo considerable.

# TIPOS DE ARCHIVOS

- Básicamente hay dos tipos de archivos: de texto y binarios.
- Los archivos de texto son aquellos que una persona puede leerlos sin ningún procesamiento previo, mediante el uso de un editor simple de texto.
- Los archivos binarios son los que no son fáciles de leer por parte de una persona, debido a que contiene caracteres “extraños” para el usuario. Por ejemplo, un archivo binario es uno que contiene información y se lo ha grabado mediante un procesador de palabras como Microsoft Word.

# ARCHIVOS DE TEXTO

- Un archivo de texto es una secuencia de caracteres organizadas en líneas terminadas por un carácter de nueva línea.
- En estos archivos se pueden almacenar canciones, fuentes de programas, base de datos simples, etc.
- Los archivos de texto se caracterizan por ser planos, es decir, todas las letras tienen el mismo formato y no hay palabras subrayadas, en negrita, o letras de distinto tamaño o ancho.



# ARCHIVO BINARIO

- Un archivo binario es una secuencia de bytes que tienen una correspondencia uno a uno con un dispositivo externo. Así que no tendrá lugar ninguna traducción de caracteres.
- Además, el número de bytes escritos (leídos) será el mismo que los encontrados en el dispositivo externo.
- Ejemplos de estos archivos son Fotografías, imágenes, texto con formatos, archivos ejecutables (aplicaciones), etc.

```
Ghex: index.html
Archivo  Editar  Ver  Ventanas  Ayuda
00000000  C 21 44 4F 43 54 59 50 45 20 48 54 4D 4C 20 50 <!DOCTYPE HTML P
00000010  55 42 4C 49 43 20 22 2D 2F 2F 57 33 43 2F 2F 44 UBLIC "-//W3C//D
00000020  54 44 20 48 54 4D 4C 20 34 2E 30 20 54 72 61 6E TD HTML 4.0 Tran
00000030  73 69 74 69 6F 6E 61 6C 2F 2F 45 4E 22 20 22 68 sitional//EN" "h
00000040  74 74 70 3A 2F 2F 77 77 77 2E 77 33 2E 6F 72 67 ttp://www.w3.org
00000050  2F 54 52 2F 52 45 43 2D 68 74 6D 6C 34 30 2F 6C /TR/REC-html40/1
00000060  6F 6F 73 65 2E 64 74 64 22 3E 0D 0A 3C 48 54 4D oose.dtd">..<HTM
00000070  4C 3E 3C 48 45 41 44 3E 0D 0A 0D 0A 3C 21 2D 2D L><HEAD>...<!--
00000080  20 4D 45 54 41 20 49 4E 46 4F 52 4D 41 54 49 4F META INFORMATIO
00000090  4E 20 42 45 47 49 4E 20 2D 2D 3E 0D 0A 0D 0A 3C N BEGIN -->....<
000000A0  54 49 54 4C 45 3E 53 6F 66 74 77 61 72 65 20 47 TITLE>Software G
000000B0  6C 6F 62 61 6C 69 7A 61 74 69 6F 6E 20 2D 20 53 localization - S
000000C0  75 6E 20 5B 54 4D 5D 20 47 6C 6F 62 61 6C 20 41 un [TM] Global A
000000D0  70 70 6C 69 63 61 74 69 6F 6E 20 44 65 76 65 6C pplication Devel
000000E0  6F 70 65 72 20 43 6F 72 6E 65 72 3C 2F 54 49 54 oper Corner</TIT
000000F0  4C 45 3E 0D 0A 3C 4D 45 54 41 20 68 74 74 70 2D LE>..<META http-
00000100  65 71 75 69 76 3D 22 43 6F 6E 74 65 6E 74 2D 54 equiv="Content-T
00000110  79 70 65 22 20 63 6F 6E 74 65 6E 74 3D 22 74 65 ype" content="te
00000120  78 74 2F 68 74 6D 6C 3B 63 68 61 72 73 65 74 3D xt/html; charset=
00000130  75 74 66 2D 38 22 3E 0D 0A 3C 4D 45 54 41 20 4E utf-8">..<META N
00000140  41 4D 45 3D 22 64 65 73 63 72 69 70 74 69 6F 6E AME="description"
```



# FUNCIONES EN C PARA ARCHIVOS

PROGRAMACION 1



# INTRODUCCION

- Un archivo de texto es un conjunto de bytes secuenciales terminados con el carácter especial EOF. En los archivos de texto, estos bytes son interpretados como caracteres.
- Toda la información que se almacena en un archivo de texto son caracteres. Esto quiere decir, entre otras cosas, que toda la información almacenada en un archivo de texto puede ser visualizada por un procesador de textos o un editor cualquiera, de la misma manera que se visualiza la información de un archivo que contiene un programa fuente de C.

|     |     |     |     |       |     |
|-----|-----|-----|-----|-------|-----|
| 'H' | 'O' | 'L' | 'A' | ..... | EOF |
|-----|-----|-----|-----|-------|-----|



# FUNCIONES EN C PARA ABRIR/CERRAR UN ARCHIVO

- Los archivos, sean de texto o binarios, es necesario abrirlos antes de extraer o introducir datos. Y no sólo eso, después de llevar a cabo todas las operaciones de lectura/escritura sobre ellos es obligatorio cerrarlos, de lo contrario puede perderse toda la información contenida en ellos.
- La primera operación que habrá que realizar siempre que se desee trabajar con un archivo será la operación de apertura del mismo. La especificación de la función de apertura es:

```
FILE *fopen(char *pathname, char *type)
```

- Donde:
  - pathname indica el nombre del archivo y la ruta de acceso.
  - type indica el modo de apertura del archivo

# DESCRIPTOR DE UN ARCHIVO

- Los archivos se referencian dentro del programa mediante su nombre, pero esto se hace únicamente en la apertura del archivo.
- A partir del momento en que el archivo ha sido abierto, éste se manipulará dentro del programa a través de un descriptor de fichero, un puntero de un tipo especial llamado **FILE** que devuelve la función **fopen** y que está definido en el archivo de cabecera **<stdio.h>**

# MODOS DE APERTURA

- Los modos de acceso a un archivo son:

| Modo | Significado  |
|------|--|
| r    | Abre un archivo de texto para lectura.                     |
| w    | Crea un archivo de texto para escritura.                   |
| a    | Abre un archivo de texto para añadir.                      |
| rb   | Abre un archivo binario para lectura.                      |
| wb   | Crea un archivo binario para escritura.                    |
| ab   | Abre un archivo binario para añadir.                       |
| r+   | Abre un archivo de texto para lectura / escritura.         |
| w+   | Crea un archivo de texto para lectura / escritura.         |
| a+   | Añade o crea un archivo de texto para lectura / escritura. |
| r+b  | Abre un archivo binario para lectura / escritura.          |
| w+b  | Crea un archivo binario para lectura / escritura.          |
| a+b  | Añade o crea un archivo binario para lectura / escritura.  |

# FUNCIONES EN C

- Algunas de las funciones que se disponen en el lenguaje C para el manejo de archivos son:

| Nombre    | Función   |
|-----------|---|
| fopen()   | Abre un archivo.  |
| fclose()  | Cierra un archivo.  |
| fgets()   | Lee una cadena de un archivo.   |
| fputs()   | Escribe una cadena en un archivo                                      |
| fseek()   | Busca un byte específico de un archivo.                               |
| fprintf() | Escribe una salida con formato en el archivo.                         |
| fscanf()  | Lee una entrada con formato desde el archivo.                         |
| feof()    | Devuelve cierto si se llega al final del archivo.                     |
| ferror()  | Devuelve cierto si se produce un error.                               |
| rewind()  | Coloca el localizador de posición del archivo al principio del mismo. |
| remove()  | Borra un archivo.   |
| fflush()  | Vacía un archivo.   |

# EJERCICIOS EN C

PROGRAMACION 1

## EJEMPLO 1

- Crea un programa que vaya leyendo las frases que el usuario teclea y las guarde en un archivo de texto llamado "frases.txt". Terminará cuando la frase introducida sea "fin" (esa frase no deberá guardarse en el archivo):

```
Project [Project Explorer] [*] archivo1.c
1  #include <stdio.h>
2  #include <string.h>
3  |
4  int main()
5  {
6      FILE* ptFichero;
7      char fin[]="fin";
8      char frase[60];
9
10     ptFichero = fopen("frases.txt", "wt");
11     printf(" PROGRAMA para ESCRIBIR FRASES.\nCuando quiera salir,"
12           "escriba la palabra fin.\n\n");
13     do
14     {
15         puts("\nEscriba una FRASE:\n(o fin). \n");
16         gets(frase);
17         if (strcmp(frase, fin) == 0)
18             break;
19         fprintf(ptFichero, "%s\n", frase);
20     }
21     while (strcmp(frase, fin) != 0);
22
23     fclose(ptFichero);
24     return 0;
25 }
```

# EJEMPLO 1

- El resultado de ejecutar este programa es:

```
C:\Users\ottoparra\Documents\cursos_Ingenieria_2016-2017\Programacion_1\archiv...
PROGRAMA para ESCRIBIR FRASES.
Cuando quiera salir, escriba la palabra fin.

Escriba una FRASE:
(o fin).

Hola mundo, que tal?

Escriba una FRASE:
(o fin).

Que hora es?

Escriba una FRASE:
(o fin).

Hoy no hay clases.

Escriba una FRASE:
(o fin).

fin

-----
Process exited after 29.7 seconds with return value 0
Presione una tecla para continuar . . . _
```

```
frases: Bloc de notas
Archivo Edición Formato Ver Ayuda
Hola mundo, que tal?
Que hora es?
Hoy no hay clases.
```

## EJERCICIO 2

- Un programa que pida al usuario que teclee frases, y las almacene en el archivo "masfrases.txt". Acabará cuando el usuario pulse la tecla "ENTER" sin teclear nada. Después deberá mostrar el contenido del archivo:

```
Project 1 archivo1.c [*] Untitled2
1  #include <stdio.h>
2  #include <string.h>
3
4  int main()
5  {
6      FILE* ficheroU;
7      char frase[61];
8      int i=0;
9
10     ficheroU = fopen("masfrases.txt", "wt");
11     printf(" PROGRAMA para ESCRIBIR y almacenar FRASES.\n"
12           "Cuando quiera salir, simplemente pulse \"Intro\".\n\n");
13     do
14     {
15         if (i == 0)
16             puts("\nEscriba una FRASE:\n(o pulse \"Intro\"). \n");
17         else
18             puts("\nEscriba otra FRASE:\n(o pulse \"Intro\"). \n");
19         gets(frase);
20         fprintf(ficheroU, "%s\n", frase);
21         i++;
22     }
23     while (strcmp(frase, "") != 0);
24     printf("He aqui lo que escribio:\n\n");
25     fclose(ficheroU);
```



## EJERCICIO 2

- Ahora a mostrar lo que se ha grabado en el archivo:

```
Project  archivo1.c  [*] Untitled2
26
27 ficheroU = fopen("frases.txt", "rt");
28 do
29 {
30     fgets(frase, 60, ficheroU);
31     puts(frase);
32 }
33 while (!feof(ficheroU));
34 getchar();
35 printf("...Hasta luego!");
36 getchar();
37 fclose(ficheroU);
38
39 return 0;
40 }
41
```

```
masfrases: Bloc de notas
Archivo Edición Formato Ver Ayuda
que tal todo, como vas?
hay examen hoy?
```

```
C:\Users\otoparra\Documents\cursos_Ingenieria_2016-2017\Programacion_1\otroArchivo.exe
Cuando quiera salir, simplemente pulse "Intro".

Escriba una FRASE:
(o pulse "Intro").

que tal todo, como vas?

Escriba otra FRASE:
(o pulse "Intro").

hay examen hoy?

Escriba otra FRASE:
(o pulse "Intro").

He aqui lo que escribio:
que tal todo, como vas?
hay examen hoy?

...Hasta luego!

-----
Process exited after 91.08 seconds with return value 0
Presione una tecla para continuar . . .
```

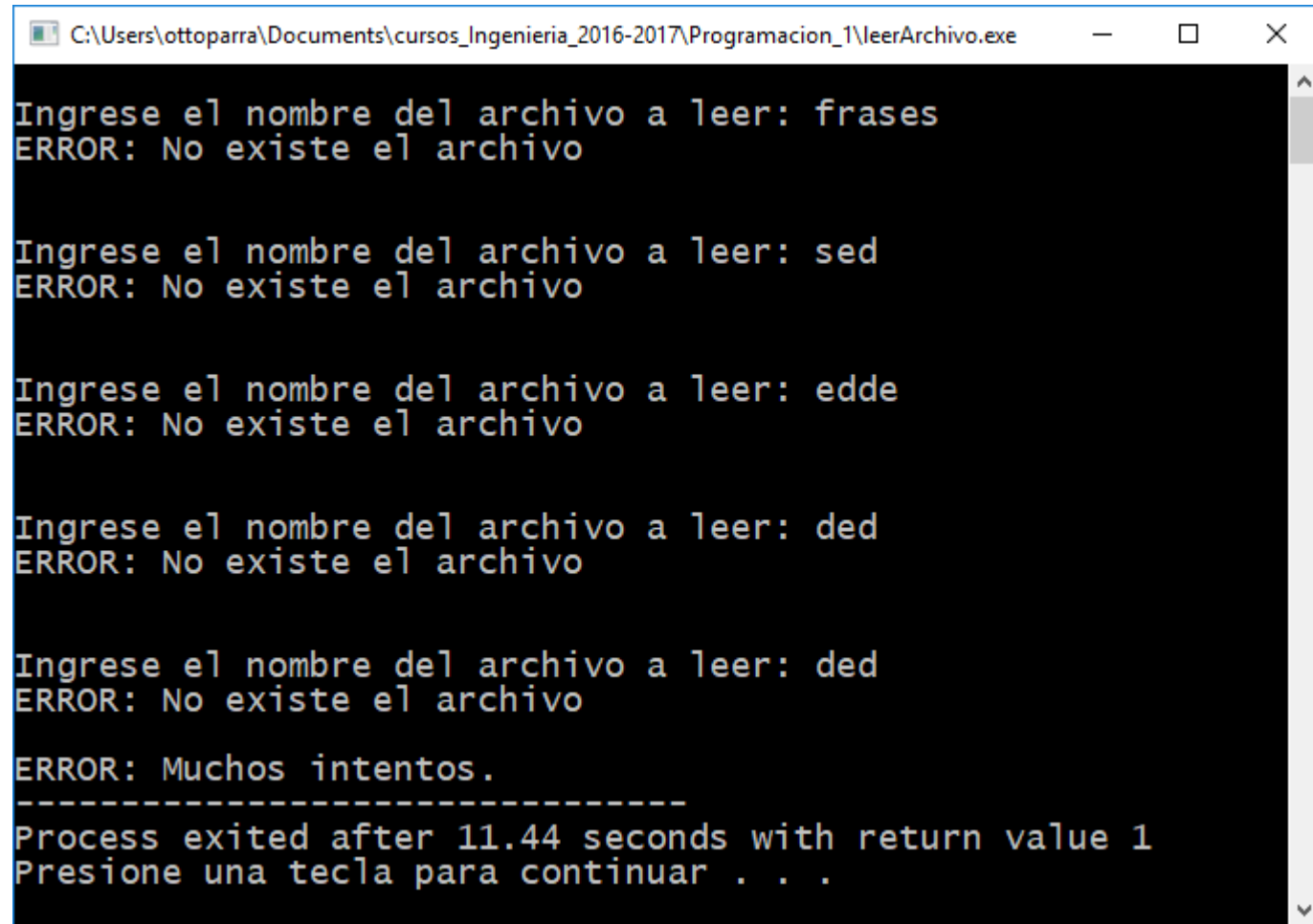
## EJERCICIO 3

- Un programa que pregunte un nombre de archivo y muestre en pantalla el contenido de ese archivo, haciendo una pausa después de cada 25 líneas, para que dé tiempo a leerlo.
- Cuando el usuario pulse ENTER, se mostrarán las siguientes 25 líneas, y así hasta que termine el archivo.

```
archivo1.c  otroArchivo.c  [*] leerArchivo.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      FILE *archivo;
7      char linea[100], nombre[40];
8      int i=0;
9      do
10     {
11         printf("\nIngrese el nombre del archivo a leer: ");
12         gets(nombre);
13
14         archivo = fopen(nombre, "rt");
15
16         if (archivo == NULL)
17         {
18             printf("ERROR: No existe el archivo\n\n");
19             i++;
20             if (i == 5){
21                 printf("ERROR: Muchos intentos.");
22                 exit(1);
23             }
24         }
25     }
26     while (archivo == NULL);
27     while (!feof(archivo))
28     {
29         for (i=0; i<25; i++){
30             fgets(linea, 100, archivo);
31             if (!feof(archivo))
32             {
33                 puts(linea);
34             }
35         }
36         getchar();
37     }
38     fclose(archivo);
39     return 0;
40 }
41
```

## EJERCICIO 3

- La ejecución del programa da el siguiente resultado:



```
C:\Users\ottoparra\Documents\cursos_Ingenieria_2016-2017\Programacion_1\leerArchivo.exe

Ingrese el nombre del archivo a leer: frases
ERROR: No existe el archivo

Ingrese el nombre del archivo a leer: sed
ERROR: No existe el archivo

Ingrese el nombre del archivo a leer: edde
ERROR: No existe el archivo

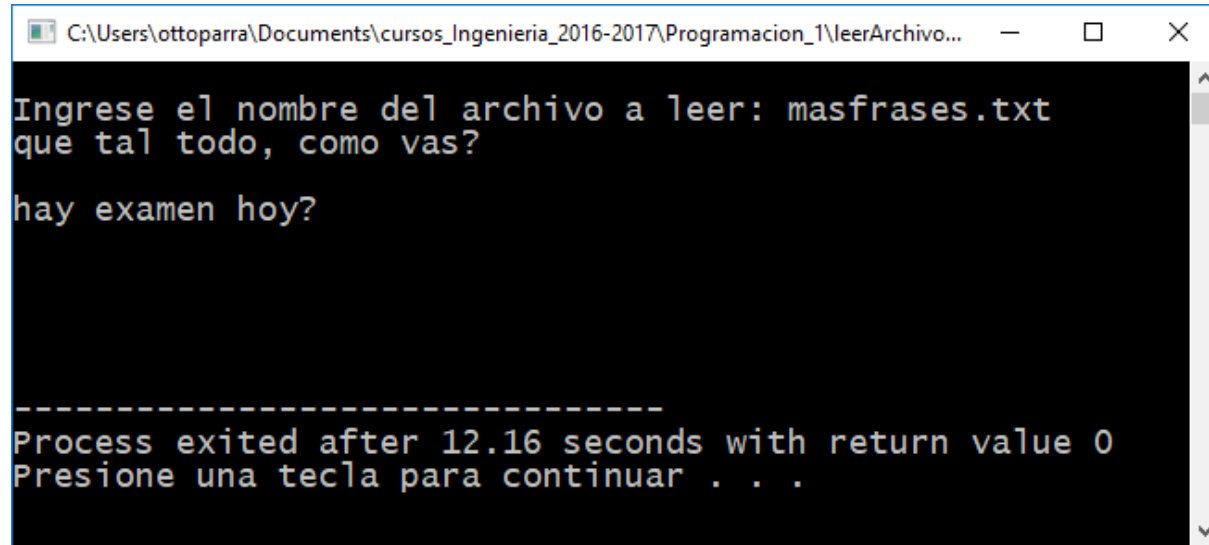
Ingrese el nombre del archivo a leer: ded
ERROR: No existe el archivo

Ingrese el nombre del archivo a leer: ded
ERROR: No existe el archivo

ERROR: Muchos intentos.
-----
Process exited after 11.44 seconds with return value 1
Presione una tecla para continuar . . .
```

## EJERCICIO 3

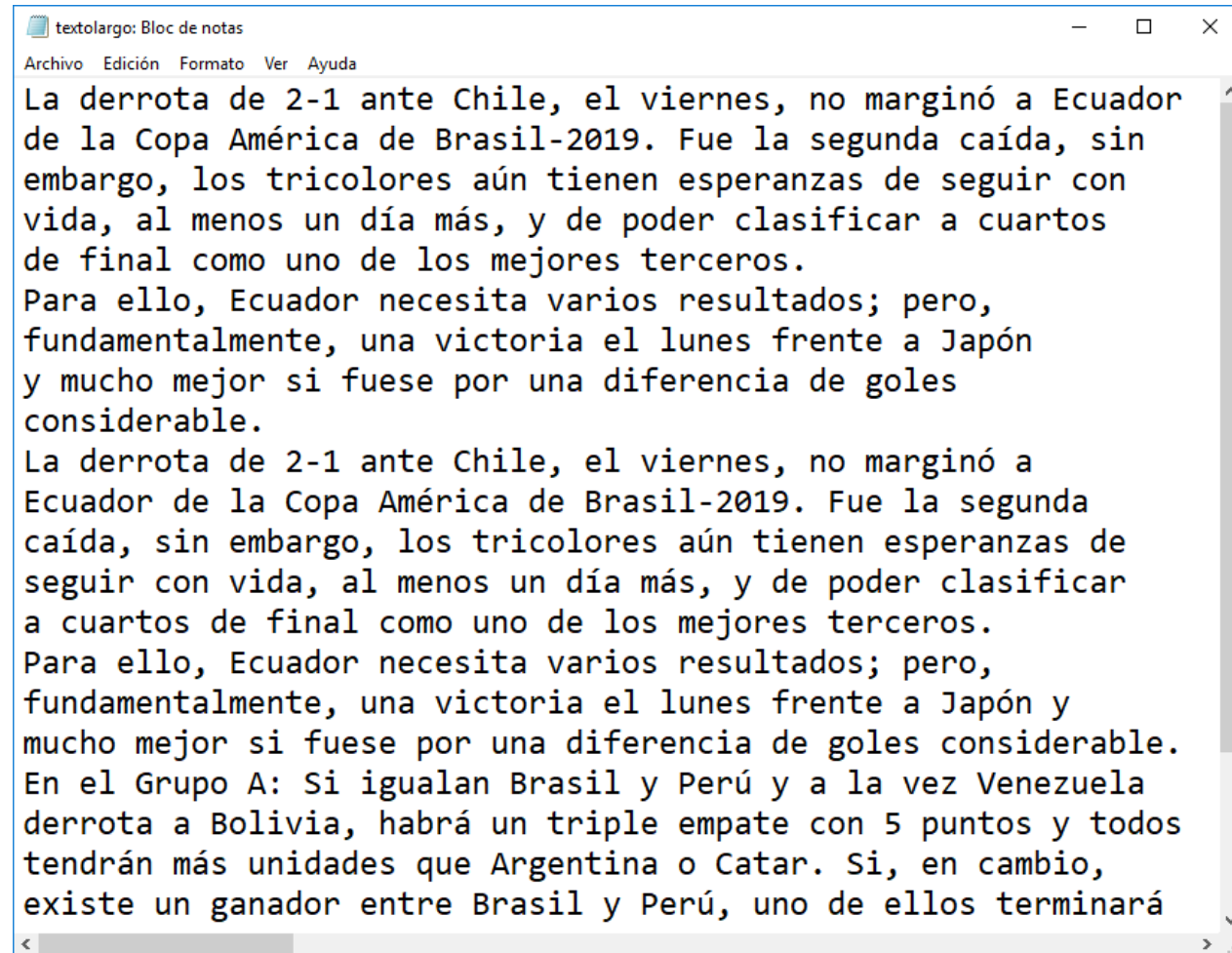
- Al dar un nombre de archivo que si existe:



```
C:\Users\ottoparra\Documents\cursos_Ingenieria_2016-2017\Programacion_1\leerArchivo...  
Ingrese el nombre del archivo a leer: masfrases.txt  
que tal todo, como vas?  
hay examen hoy?  
  
-----  
Process exited after 12.16 seconds with return value 0  
Presione una tecla para continuar . . .
```

## EJERCICIO 3

- Un archivo de texto con más de 25 líneas (se muestra una parte de él):



# EJERCICIO 3

- La primera parte:

```
C:\Users\ottoparra\Documents\cursos_Ingenieria_2016-2017\Programacion_1\LeerArchivo.exe

Ingrese el nombre del archivo a leer: textlargo.txt
ERROR: No existe el archivo

Ingrese el nombre del archivo a leer: textolargo.txt
La derrota de 2-1 ante Chile, el viernes, no marginó a Ecuador
de la Copa Am rica de Brasil-2019. Fue la segunda ca da, sin
embargo, los tricolores a n tienen esperanzas de seguir con
vida, al menos un d a m s, y de poder clasificar a cuartos
de final como uno de los mejores terceros.
Para ello, Ecuador necesita varios resultados; pero,
fundamentalmente, una victoria el lunes frente a Jap n
y mucho mejor si fuese por una diferencia de goles
considerable.

La derrota de 2-1 ante Chile, el viernes, no marginó a
Ecuador de la Copa Am rica de Brasil-2019. Fue la segunda
ca da, sin embargo, los tricolores a n tienen esperanzas de
seguir con vida, al menos un d a m s, y de poder clasificar
a cuartos de final como uno de los mejores terceros.
Para ello, Ecuador necesita varios resultados; pero,
fundamentalmente, una victoria el lunes frente a Jap n y
```

```
C:\Users\ottoparra\Documents\cursos_Ingenieria_2016-2017\Programacion_1\LeerArchivo.exe

mucho mejor si fuese por una diferencia de goles considerable.
En el Grupo A: Si igualan Brasil y Per  y a la vez Venezuela
derrota a Bolivia, habr  un triple empate con 5 puntos y todos
tendr n m s unidades que Argentina o Catar. Si, en cambio,
existe un ganador entre Brasil y Per , uno de ellos terminar 
con 4 unidades. Y pesar  el gol diferencia.

El camino no es f cil, pues Ecuador no tiene puntos y su
diferencia de goles es de -5, producto de la paliza que le

-----
Process exited after 65.5 seconds with return value 0
Presione una tecla para continuar . . .
```

# REFERENCIAS

- Ejercicios tomados de:
- <http://www.aprendeaprogramar.com/mod/forum/discuss.php?d=664>