



ugr

Universidad
de **Granada**

PRÁCTICA FLEX



DECSAI

Universidad de Granada

INTRODUCCIÓN	3
DESCRIPCIÓN DEL PROBLEMA	3
DESCRIPCIÓN DE USO DE LA HERRAMIENTA	3
Ejemplo:	3
DESCRIPCIÓN DEL CÓDIGO	6
SELECCIÓN DE DECLARACIONES	6
SELECCIÓN DE REGLAS	6
SELECCIÓN DE PROCEDIMIENTOS	7
DESCRIPCIÓN DE USO DE LA HERRAMIENTA	8

INTRODUCCIÓN

Flex es utilizado en la primera etapa de elaboración de un compilador, un intérprete, emulador o cualquier otro tipo de herramienta que necesite procesar un fichero de entrada para poder cumplir su misión. Otra utilidad de flex consiste en ser una herramienta que nos proporciona la posibilidad de ejecutar acciones tras la localización de cadenas de entrada que emparejan con expresiones regulares. En esta práctica nos centraremos en esta segunda utilidad de flex.

Se le pide al estudiante la realización de un trabajo práctico de procesamiento de un fichero que involucre el uso de flex para localizar ciertas cadenas en el fichero y ejecutar una acción correspondiente con cada una de ellas. Por ejemplo, localizar y borrar direcciones de un correo electrónico en una página web, cambiar de color ciertas palabras de un fichero html, etc. El alumno deberá plantear su propio trabajo práctico de procesamiento de ficheros usando flex.

DESCRIPCIÓN DEL PROBLEMA

En esta práctica, como previamente hemos descrito, debemos presentar un problema y dedicarle una solución utilizando flex. Para este caso práctico he escogido la realización de un programa capaz de calcular las notas medias de los alumnos asociados a un profesor. Dando como salida la nota final, si está o no aprobado y el porcentaje de suspensos y aprobados.

DESCRIPCIÓN DE USO DE LA HERRAMIENTA

Para hacer uso del programa será necesario aparte del código del programa, un “.txt” donde se almacenarán los datos de los alumnos (Nombre alumno) y las notas correspondientes a cada uno. Se calculará una media aritmética.

Ejemplo:

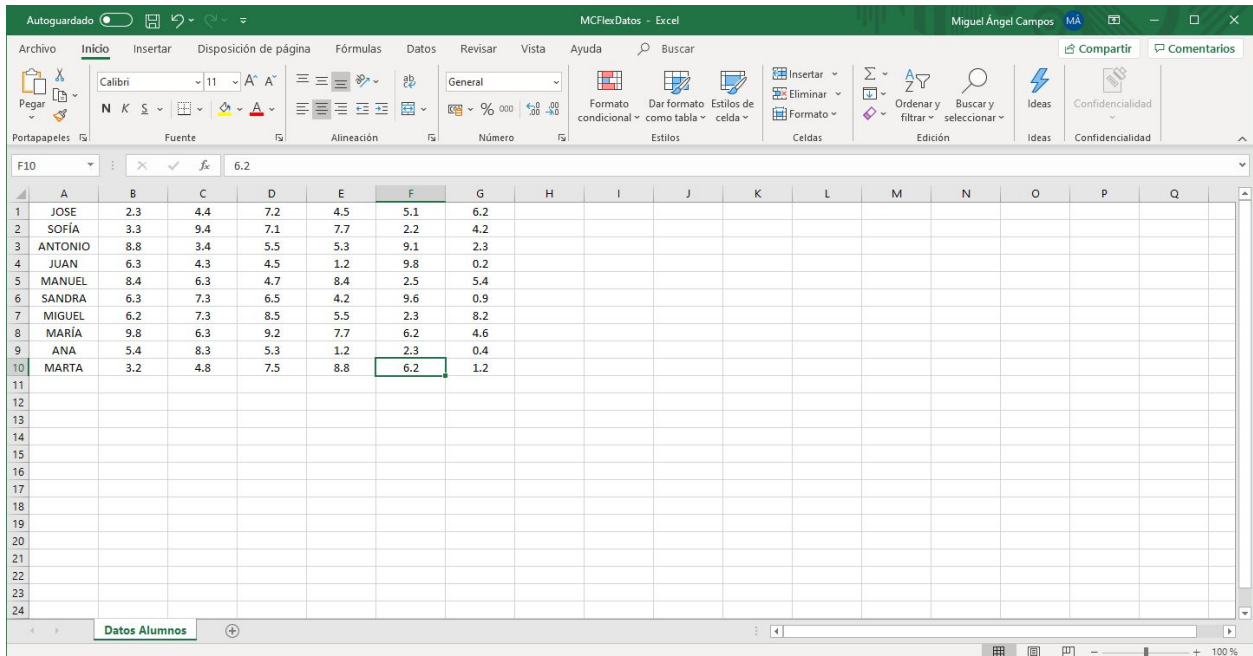
JOSE	2.3	9.5	4.1	5.6	1.2
------	-----	-----	-----	-----	-----

NOTA: las separaciones entre cada elemento se realizarán mediante tabulador.

La tabla anterior muestra el formato que seguirán los datos referentes a las notas de los alumnos

. Para representar un conjunto de datos de prueba para darle como entrada a nuestro programa que hará uso de flex, utilizaré una herramienta de hojas de cálculo por comodidad y simpleza para exportar los datos a formato “.txt” de manera sencilla una vez los hayamos definido fácilmente.

En este caso he hecho uso de la herramienta de hojas de cálculo Excel.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	JOSE	2.3	4.4	7.2	4.5	5.1	6.2										
2	SOFA	3.3	9.4	7.1	7.7	2.2	4.2										
3	ANTONIO	8.8	3.4	5.5	5.3	9.1	2.3										
4	JUAN	6.3	4.3	4.5	1.2	9.8	0.2										
5	MANUEL	8.4	6.3	4.7	8.4	2.5	5.4										
6	SANDRA	6.3	7.3	6.5	4.2	9.6	0.9										
7	MIGUEL	6.2	7.3	8.5	5.5	2.3	8.2										
8	MARIA	9.8	6.3	9.2	7.7	6.2	4.6										
9	ANA	5.4	8.3	5.3	1.2	2.3	0.4										
10	MARTA	3.2	4.8	7.5	8.8	6.2	1.2										
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	

Figura 1. Hoja de cálculo excel con los datos de entrada para nuestro programa.

Una vez definido el conjunto de datos de prueba, exportamos estos datos a formato “.txt” para poder ser interpretado por flex.

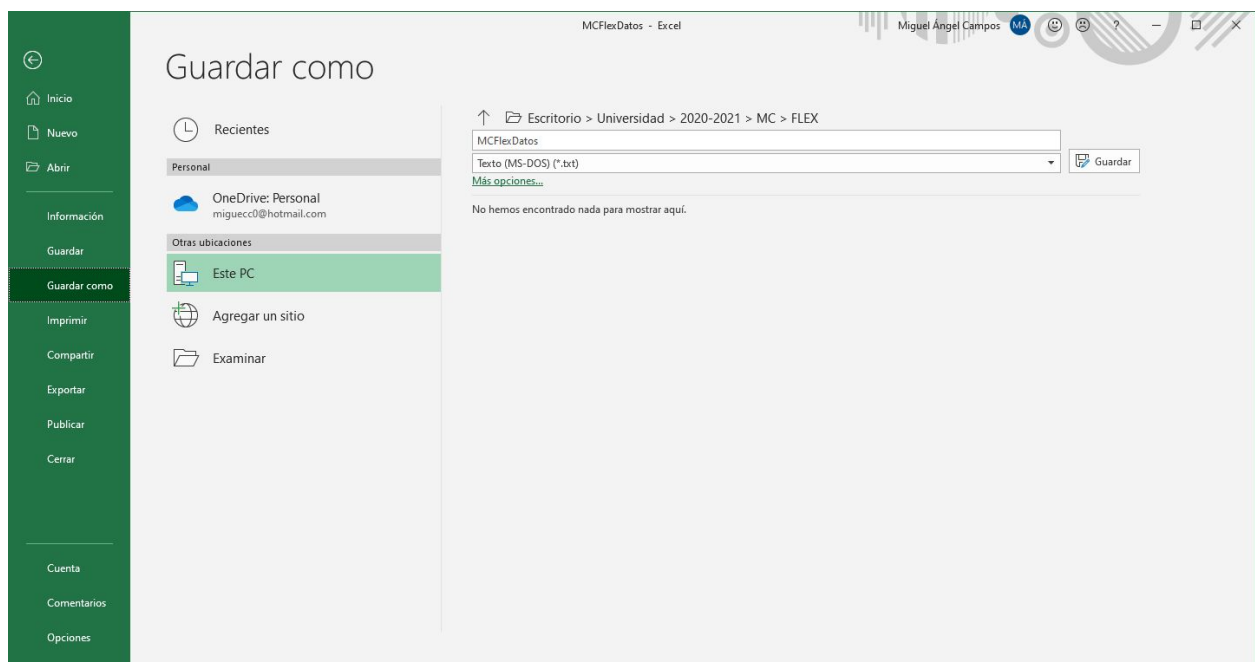
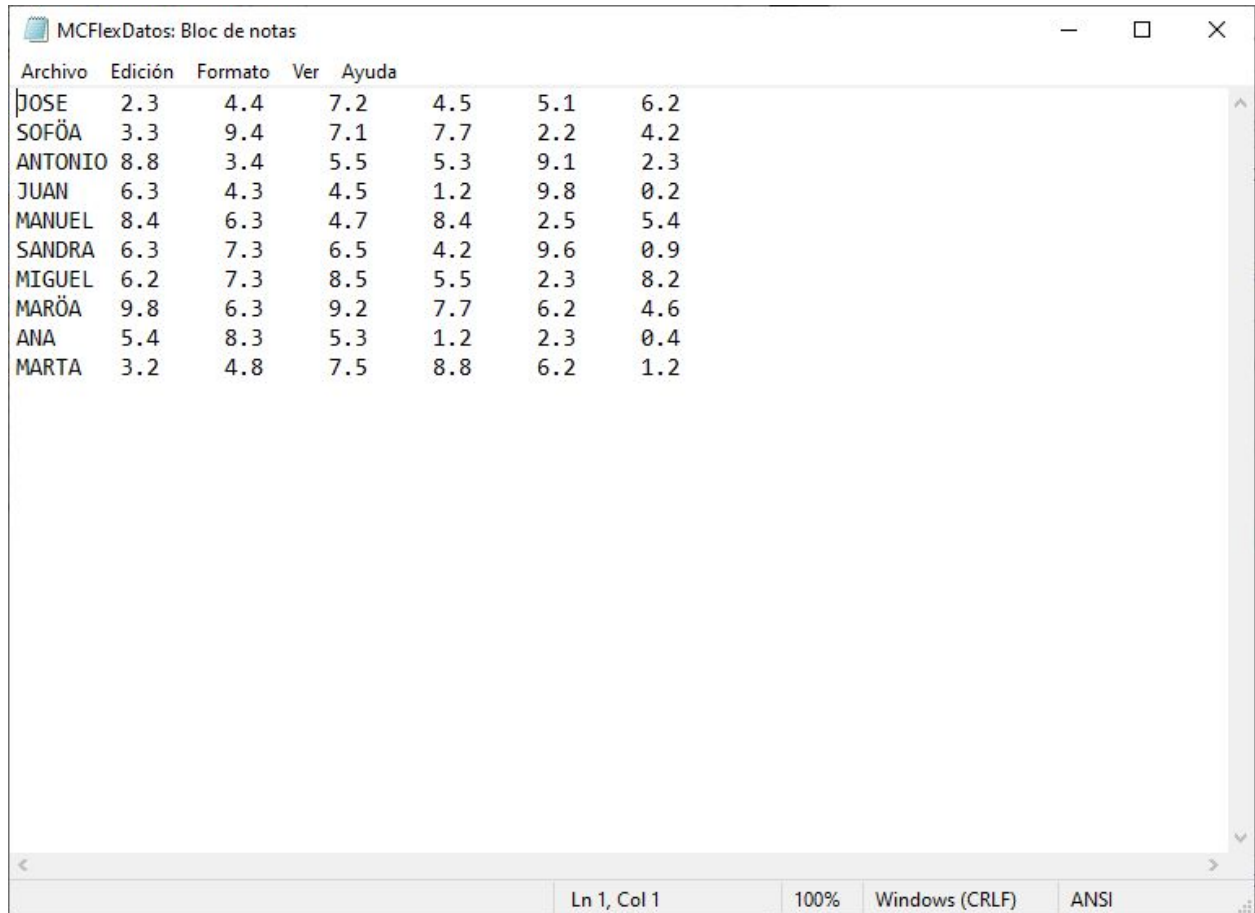


Figura 2. Exportamos los datos a formato “.txt”.



Archivo	Edición	Formato	Ver	Ayuda		
JOSE	2.3	4.4	7.2	4.5	5.1	6.2
SOFÖA	3.3	9.4	7.1	7.7	2.2	4.2
ANTONIO	8.8	3.4	5.5	5.3	9.1	2.3
JUAN	6.3	4.3	4.5	1.2	9.8	0.2
MANUEL	8.4	6.3	4.7	8.4	2.5	5.4
SANDRA	6.3	7.3	6.5	4.2	9.6	0.9
MIGUEL	6.2	7.3	8.5	5.5	2.3	8.2
MARÖA	9.8	6.3	9.2	7.7	6.2	4.6
ANA	5.4	8.3	5.3	1.2	2.3	0.4
MARTA	3.2	4.8	7.5	8.8	6.2	1.2

Figura 3. Datos en formato “.txt” con tabulaciones.

DESCRIPCIÓN DEL CÓDIGO

Línea 3 → `%option noyywrap` - sirve para leer un único archivo de entrada, y acabar el programa una vez leído.

```
1 %option noyywrap
```

Figura 4. `%option noyywrap`.

SELECCIÓN DE DECLARACIONES

- **Líneas 6-7** → Declaración de las librerías necesarias.
- **Línea 11** → Declaramos un tipo fichero.
- **Líneas 13-17** → Declaramos las variables necesarias para nuestro programa.

```

3 /*---- Selección de Declaraciones ----*/
4
5 %{
6 #include <iostream>
7 #include <fstream>
8
9 using namespace std;
10
11 ifstream fichero;
12
13 float nota, media = 0.0;
14 int filas, numero_alumno_aprobado, numero_alumno_suspenso = 0;
15 bool aprobado;
16
17 string cadena;
18 %}

```

Figura 5. Selección de declaraciones.

SELECCIÓN DE REGLAS

- **Líneas 24-26** → Leemos los nombres. Esta función va guardando de carácter en carácter los nombres de cada alumno.

- **Líneas 28-32** → Leemos las notas. Esta función va guardando de número en número las notas de cada alumno. Como pueden ser decimales, añadimos el "." y se va almacenando en la variable media que usaremos más adelante. Así mismo, añadimos por cada nota en una variable "filas" para luego poder calcular correctamente la nota media.

- **Líneas 35-52** → Se indica con un "\n" (Final de línea). Esto indica al programa que han acabado las notas del alumno del que se está calculando la media. Por tanto, para obtener la nota media, dividimos el total acumulado del alumno (media) entre el total de notas (fila). Mostramos el nombre, si está aprobado o no y su nota final. Si la nota es mayor o igual que 5.0 también incrementamos en uno la variable "numero_alumnos_aprobados" para luego calcular el rendimiento del profesor, y "numero_alumnos_suspensos" en caso de ser inferior a 5.0.

```

20 /*---- Selección de Reglas ----*/
21
22 %%
23
24 [a-zA-Z] {
25     cadena+=yytext;
26 }
27
28 [0-9]+\.[0-9]* {
29     nota=atof(yytext);
30     media+=nota;
31     filas++;
32 }
33
34 \n {
35     media = media/filas;
36     if(media>=5.0)
37         aprobado = true;
38     else
39         aprobado = false;
40
41     if(aprobado){
42         cout << "El/La alumn@ " << cadena << " está aprobad@ con una media de " << media << endl;
43         numero_alumno_aprobado++;
44     }else{
45         cout << "El/La alumn@ " << cadena << " está suspens@ con una media de " << media << endl;
46         numero_alumno_suspenso++;
47     }
48
49     filas=0;
50     cadena="";
51     media=0.0;
52 }
53
54 %%

```

Figura 6. Selección de reglas.

SELECCIÓN DE PROCEDIMIENTOS

En el main del programa, leemos el archivo. Hay dos casuísticas de error en el mismo:

- Error al leer el fichero.
- Error al introducir los parámetros a la hora de su ejecución.

Si se ha realizado correctamente la lectura activamos el proceso del programa. Como salida este muestra el promedio de aprobados y suspensos.

```

56 /*---- Selección de Procedimientos ----*/
57
58 int main(int argc, char*argv[]){
59     if(argc==2){
60         fichero.open(argv[1]);
61         if(!fichero){
62             cerr << "Error en la lectura del archivo" << endl;
63             exit(1);
64         }
65     }else{
66         cout << "Formato: ./medias<Nombre-fichero>"<< endl;
67         exit(1);
68     }
69
70     nota=0.0;
71     filas=0.0;
72
73     yyFlexLexer flujo(&fichero,0);
74     flujo.yylex();
75
76     float paprobados = (numero_alumno_aprobado*100)/(numero_alumno_aprobado+numero_alumno_suspenso);
77     float psuspensos = (numero_alumno_suspenso*100)/(numero_alumno_aprobado+numero_alumno_suspenso);
78
79     cout << "\n\nEl rendimiento del profesor es: " << endl;
80     cout << "    - Aprobados => " << paprobados << "%" << endl;
81     cout << "    - Suspensos => " << psuspensos << "%" << endl;
82
83     return 0;
84 }

```

Figura 7. Selección de procedimientos.

DESCRIPCIÓN DE USO DE LA HERRAMIENTA

Para ejecutar el programa debemos utilizar el makefile que se encuentra alojado en la carpeta **FLEX**. Este makefile genera el archivo **plantilla.c** a partir del archivo **plantilla.l** que es tratado por flex interpretando las reglas léxicas que he planteado. Una vez tiene este archivo **plantilla.c** g++ lo compila y genera el programa **medias**.

Si queremos ejecutar una prueba del programa **medias** sobre los datos previamente definidos en la hoja de cálculo, simplemente debemos ejecutar “make ejemplo”, este comando ejecutará la línea 11 del makefile “./medias Datos/notas.txt”.

Por otra parte si queremos hacer una limpieza de los archivos que make nos genera, tenemos un make clean.

En caso de que se quiera ejecutar el programa **medias** con otro “.txt” que aloje datos diferentes tenemos dos opciones, ejecutar el comando “./medias archivo.txt” desde terminal directamente o automatizarlo de la misma manera desde el makefile.

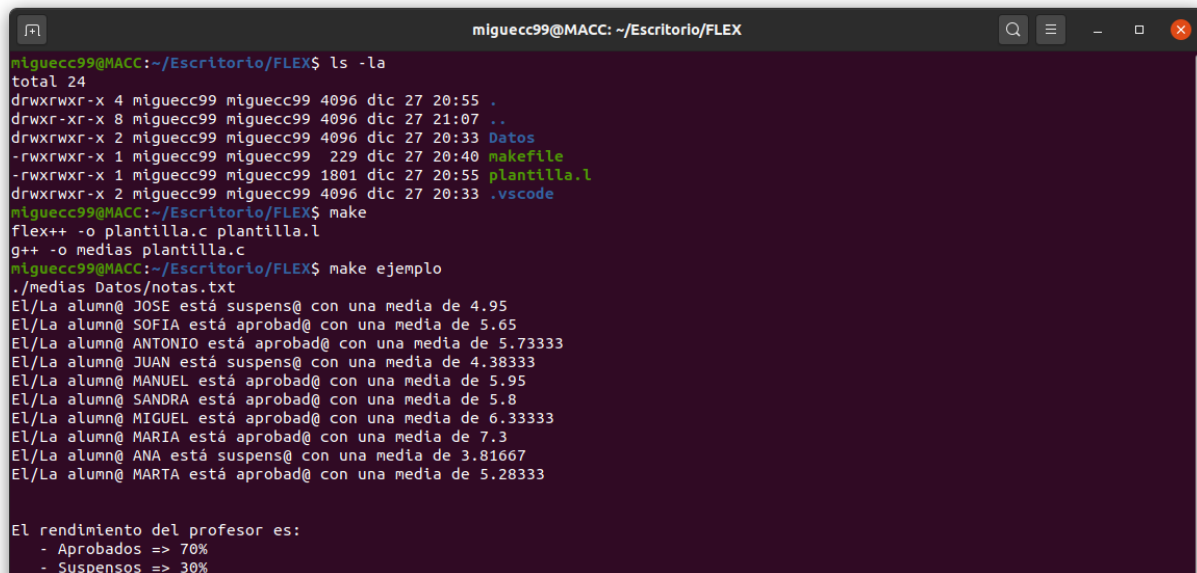

```

1 COMP=g++
2
3 medias: plantilla.c
4     $(COMP) -o $@ $^$(LIBS)
5
6 plantilla.c: plantilla.l
7     flex++ -o $@ $^
8
9 # Ejecutamos make ejemplo para probar un ejemplo
10 ejemplo:
11     ./medias Datos/notas.txt
12 clean:
13     rm -f medias plantilla.c

```

Figura 4. Makefile.

Una ejecución de ejemplo del programa con los datos generados previamente tendría una salida como esta:



```

miguecc99@MACC:~/Escritorio/FLEX$ ls -la
total 24
drwxrwxr-x 4 miguecc99 miguecc99 4096 dic 27 20:55 .
drwxr-xr-x 8 miguecc99 miguecc99 4096 dic 27 21:07 ..
drwxrwxr-x 2 miguecc99 miguecc99 4096 dic 27 20:33 Datos
-rwxrwxr-x 1 miguecc99 miguecc99 229 dic 27 20:40 makefile
-rwxrwxr-x 1 miguecc99 miguecc99 1801 dic 27 20:55 plantilla.l
drwxrwxr-x 2 miguecc99 miguecc99 4096 dic 27 20:33 .vscode
miguecc99@MACC:~/Escritorio/FLEX$ make
flex++ -o plantilla.c plantilla.l
g++ -o medias plantilla.c
miguecc99@MACC:~/Escritorio/FLEX$ make ejemplo
./medias Datos/notas.txt
El/La alumn@ JOSE está suspens@ con una media de 4.95
El/La alumn@ SOFIA está aprobad@ con una media de 5.65
El/La alumn@ ANTONIO está aprobad@ con una media de 5.73333
El/La alumn@ JUAN está suspens@ con una media de 4.38333
El/La alumn@ MANUEL está aprobad@ con una media de 5.95
El/La alumn@ SANDRA está aprobad@ con una media de 5.8
El/La alumn@ MIGUEL está aprobad@ con una media de 6.33333
El/La alumn@ MARIA está aprobad@ con una media de 7.3
El/La alumn@ ANA está suspens@ con una media de 3.81667
El/La alumn@ MARTA está aprobad@ con una media de 5.28333

El rendimiento del profesor es:
- Aprobados => 70%
- Suspensos => 30%

```

Figura 5. Ejecución del programa `media` con los datos de ejemplo `notas.txt`.