	
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación

Profesor: Saavedraa Hernández Honorato

Asignatura: Fundamentos de programación

Grupo: 01

No de Práctica(s): 8“Estructuras de selección”

Integrante(s): Luis Salinas Ludwig

Semestre: 2018-1

Fecha de entrega: 23/10/2017

Observaciones:

CALIFICACIÓN: _____

salas A y B

Objetivo

Elaborar programas en lenguaje C que incluyan las estructuras de selección if, if-else, switch y ternaria (o condicional) para la resolución de problemas básicos.

Desarrollo

Se comenzó por crear un nuevo proyecto en CodeBlocks para poder realizar las actividades de la práctica.

Lo primero a ver fue la Estructura de control selectiva if.

Dónde se evalúan las expresiones lógicas y, si se cumplen (si la condición es verdadera) se ejecuta las instrucciones que contiene el bloque el cual pertenece a estructura de control selectiva. De lo contrario cuando este no se cumple, continúa con el flujo normal del programa.

La sintaxis para la estructura de control selectiva if es la siguiente:

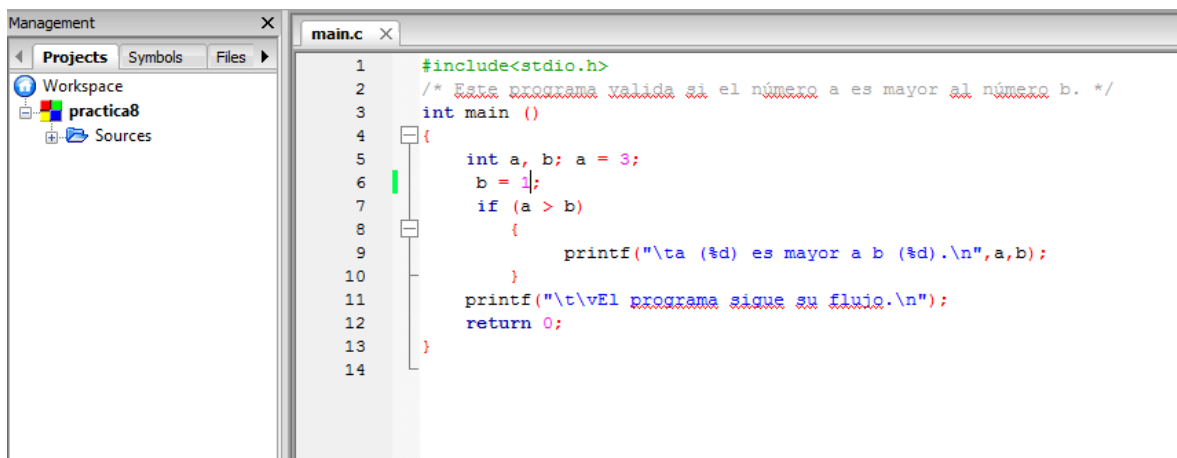
```
if (expresión_lógica)
```

```
{
```

```
// bloque de código a ejecutar
```

```
}
```

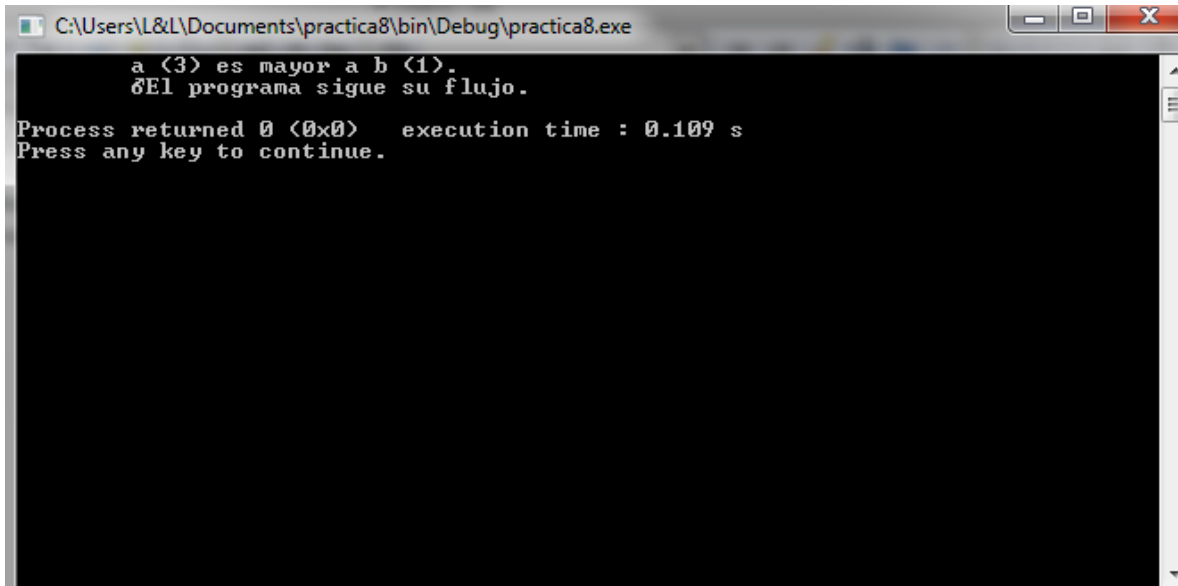
Las llaves no son necesarias, pero estas nos ayudan a comprender mejor nuestro programa y el proceso al cual es sometido y de esta manera entendemos mejor la estructura de control selectiva. Para entender mejor lo anterior se corrió un programa en CodeBlocks.



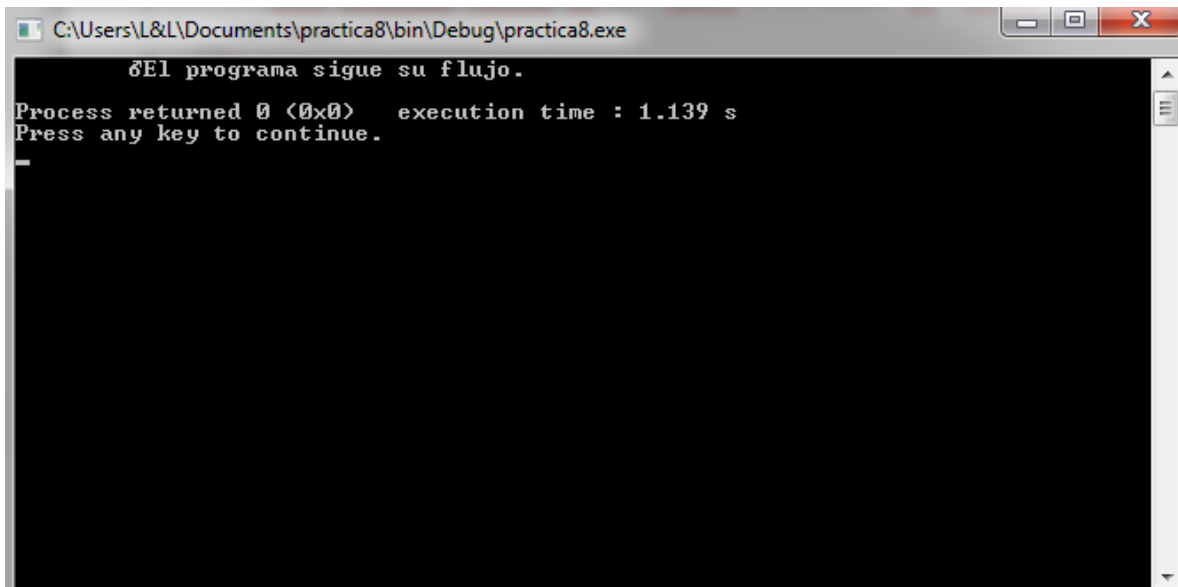
```
1  #include<stdio.h>
2  /* Este programa valida si el número a es mayor al número b. */
3  int main ()
4  {
5      int a, b; a = 3;
6      b = 1;
7      if (a > b)
8      {
9          printf("\ta (%d) es mayor a b (%d).\n",a,b);
10     }
11     printf("\t\vEl programa sigue su flujo.\n");
12     return 0;
13 }
14
```

Que al correrlo lo que realizara será el proceso del if el cual es que si cumple la condición este realizara las acciones que le pertenecen a la acción del if, cuyo

resultado será imprimir lo que se ordeno o del caso contrario nos imprimirá en pantalla que el proceso sigue su flujo.



```
C:\Users\L&L\Documents\practica8\bin\Debug\practica8.exe
a (3) es mayor a b (1).
El programa sigue su flujo.
Process returned 0 (0x0) execution time : 0.109 s
Press any key to continue.
```



```
C:\Users\L&L\Documents\practica8\bin\Debug\practica8.exe
El programa sigue su flujo.
Process returned 0 (0x0) execution time : 1.139 s
Press any key to continue.
```

Posteriormente se realizaron más ejemplos para entender un poco mas cual es el proceso que realiza la estructura de control selectiva if que anteriormente ya se había abordado en pseudocódigo en SI.

Continuamos con la estructura de control selectiva if-else.

Esta es muy parecida al if pero esta estructura evalúa la expresión lógica y si la condición se cumple es decir que sea verdadera esta ejecutara las instrucciones del bloque perteneciente al if, mientras que si es falsa ahora se ejecutara el bloque de código reservado para 'else'. Hay que mencionar que dentro de nuestro

programa es posible agregar varias estructuras if-else, en otras palabras que dentro de una estructura if-else podemos tener mas estructuras if-else.

Su sintaxis es la siguiente:

```
if (expresión_lógica)
```

```
{
```

```
// bloque de código a ejecutar
```

```
// si la condición es verdadera
```

```
}
```

```
else
```

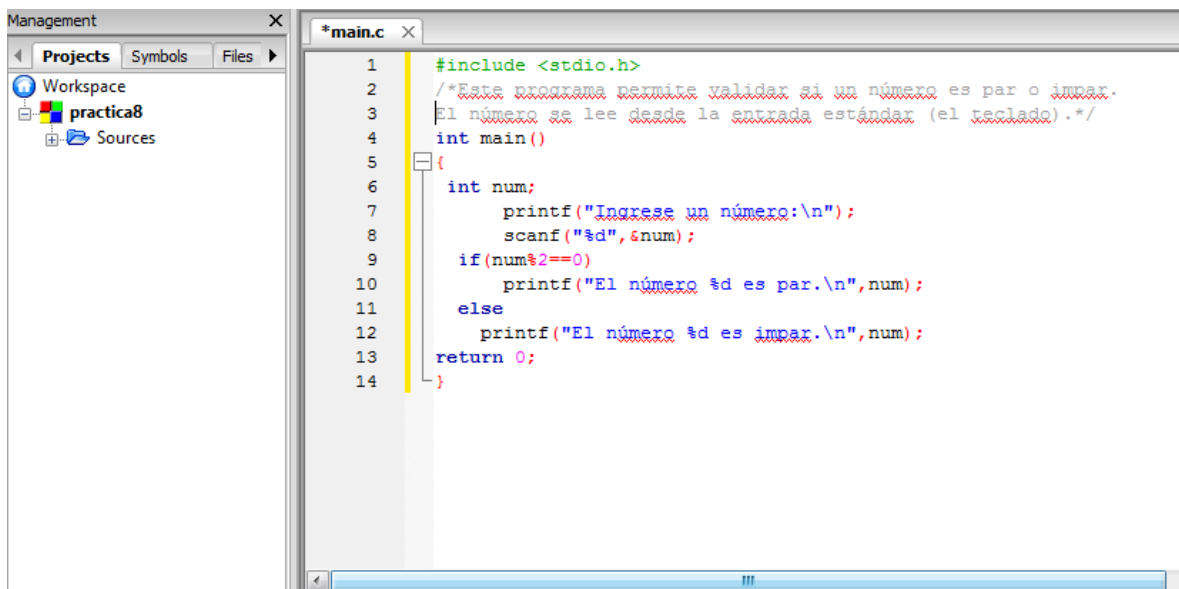
```
{
```

```
// bloque de código a ejecutar
```

```
// si la condición es falsa
```

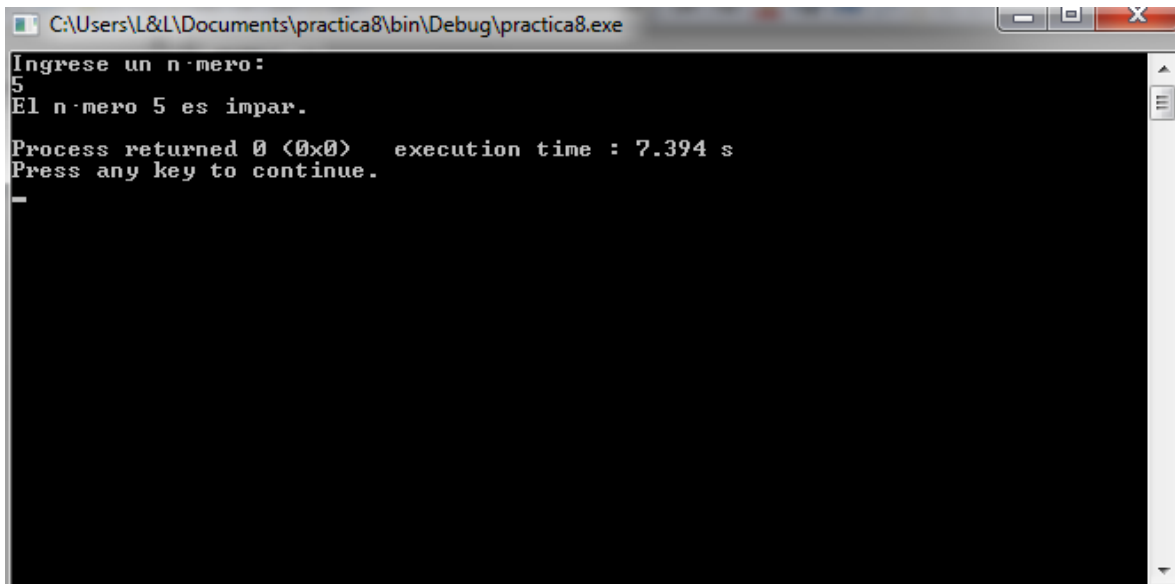
```
}
```

Un ejemplo que se realizo es el siguiente:

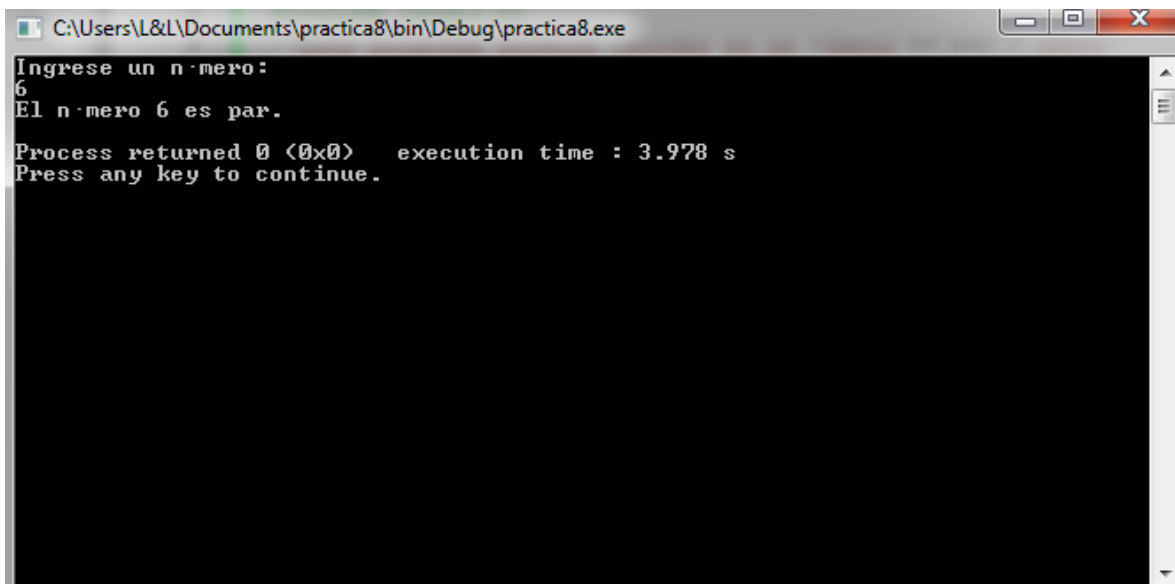
The image shows a screenshot of a code editor window titled "Management". On the left, there is a sidebar with "Projects", "Symbols", and "Files" tabs. Under "Projects", there is a workspace named "practica8" with a sub-folder "Sources". The main editor area shows a file named "main.c" with the following C code:

```
1  #include <stdio.h>
2  /*Este programa permite validar si un número es par o impar.
3  El número se lee desde la entrada estándar (el teclado).*/
4  int main()
5  {
6      int num;
7      printf("Ingrese un número:\n");
8      scanf("%d", &num);
9      if(num%2==0)
10         printf("El número %d es par.\n", num);
11     else
12         printf("El número %d es impar.\n", num);
13     return 0;
14 }
```

El cual dependiendo lo que se cumpliera nos imprimiría lo siguiente, pero ello dependería del número que le asignáramos para que el programa realizara su trabajo:



```
C:\Users\L&L\Documents\practica8\bin\Debug\practica8.exe
Ingrese un n-mero:
5
El n-mero 5 es impar.
Process returned 0 (0x0) execution time : 7.394 s
Press any key to continue.
_
```



```
C:\Users\L&L\Documents\practica8\bin\Debug\practica8.exe
Ingrese un n-mero:
6
El n-mero 6 es par.
Process returned 0 (0x0) execution time : 3.978 s
Press any key to continue.
_
```

Se continuaron viendo más ejemplos parecidos y un poco más desarrollados que el que se expuso anteriormente.

Continuamos con la estructura de control selectiva switch-case

Esta estructura evalúa la evaluaría que se encuentre dentro del paréntesis para después de la palabra reservada switch y la compara con valores constantes que tiene cada caso. Los tipos de datos que se evalúan aquí son enteros, caracteres y enumeraciones, donde al final de cada caso esta ejecuta la instrucción break, si es omitido el break se ejecutaría el siguiente cas, indicándonos que el código a ejecutar termino.

Si este no coincide dentro de algún caso entonces se ejecuta el bloque por defecto (default). La sintaxis es la siguiente.

switch (opción_a_evaluar)

{

case valor1:

/* Código a ejecutar*/

break;

case valor2:

/* Código a ejecutar*/

break;

case valorN:

/* Código a ejecutar*/

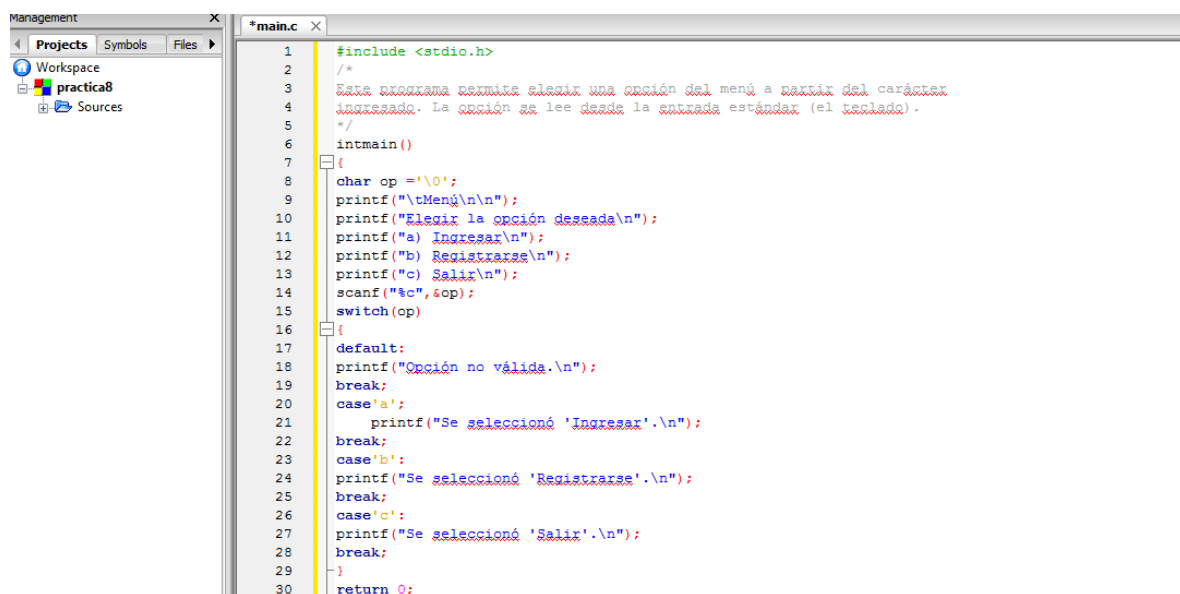
break;

default;

/* Código a ejecutar*/

}

EL ejemplo que se abordo fue el siguiente:



The screenshot shows a code editor with a file named `*main.c`. The code is in C and implements a menu system using a switch statement. The menu options are 'Ingresar', 'Registrarse', and 'Salir'. The code includes standard headers, prints the menu, reads user input, and executes the corresponding action based on the input character. The switch statement has a default case for invalid options and specific cases for 'a', 'b', and 'c'.

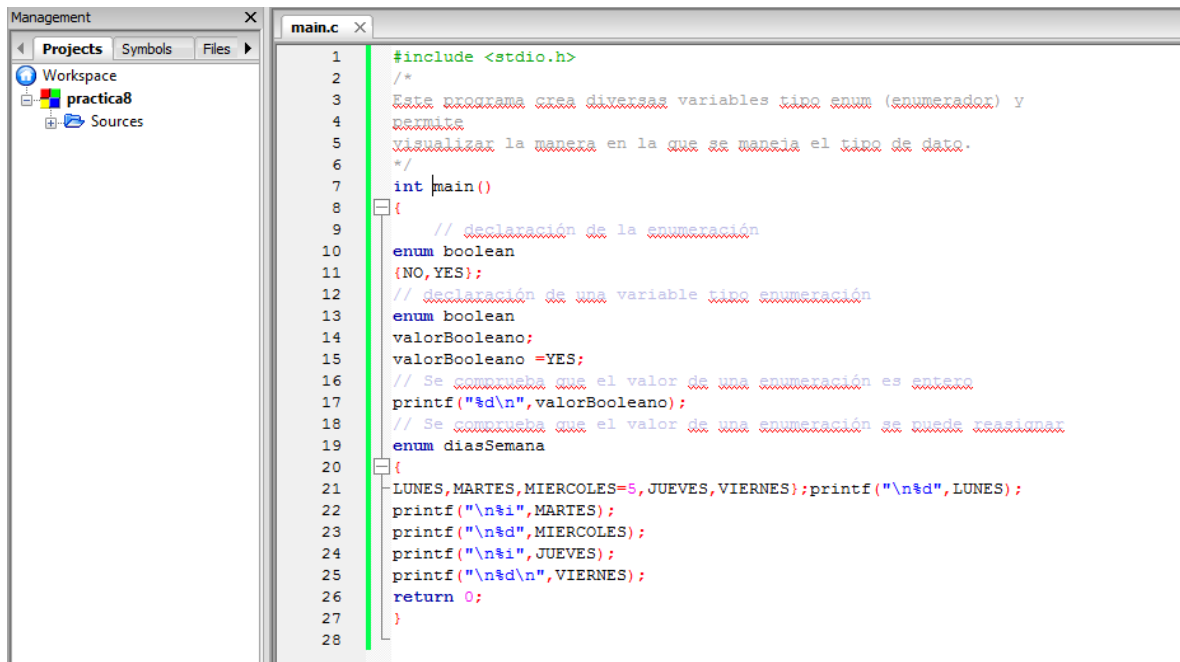
```
1  #include <stdio.h>
2  /*
3  Este programa permite elegir una opción del menú a partir del carácter
4  ingresado. La opción se lee desde la entrada estándar (el teclado).
5  */
6  int main()
7  {
8      char op = '\0';
9      printf("\tMenú\n\n");
10     printf("Elegir la opción deseada\n");
11     printf("a) Ingresar\n");
12     printf("b) Registrarse\n");
13     printf("c) Salir\n");
14     scanf("%c", &op);
15     switch(op)
16     {
17     default:
18         printf("Opción no válida.\n");
19         break;
20     case 'a':
21         printf("Se seleccionó 'Ingresar'.\n");
22         break;
23     case 'b':
24         printf("Se seleccionó 'Registrarse'.\n");
25         break;
26     case 'c':
27         printf("Se seleccionó 'Salir'.\n");
28         break;
29     }
30     return 0;
```

Continuamos por la Enumeración

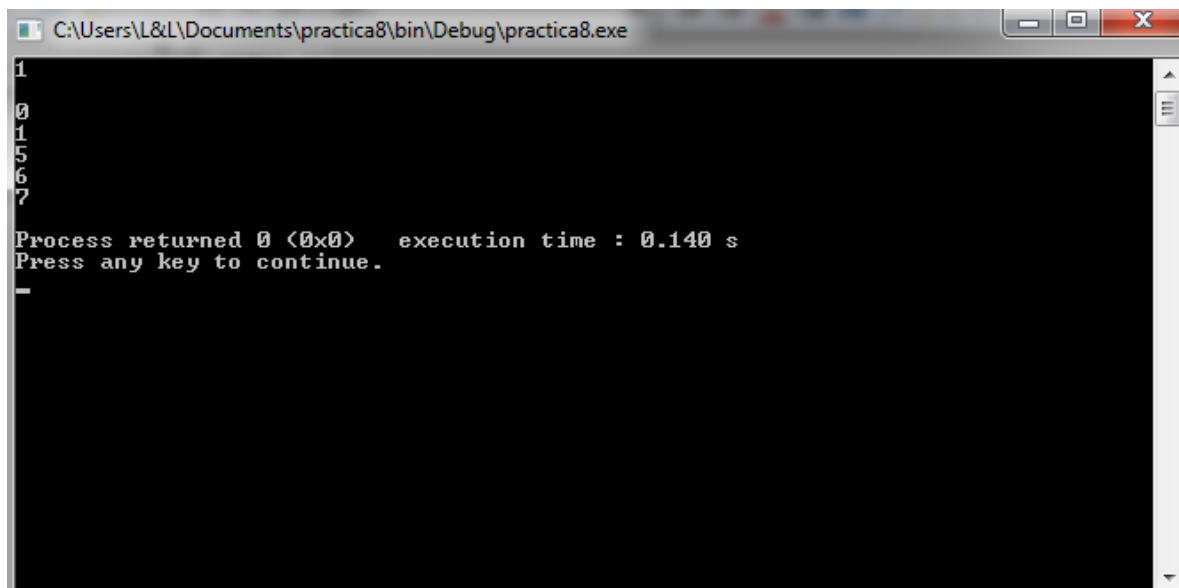
Este tipo de dato constante conocido como enumeración, donde la variable enumerador se puede crear de la siguiente forma:

```
enum identificador {VALOR1, VALOR2, ... , VALORN};
```

Para que podamos realizar lo anterior necesitamos utilizar la palabra reservada enum seguida del identificador. Se vio el siguiente ejemplo:



```
1  #include <stdio.h>
2  /*
3   * Este programa crea diversas variables tipo enum (enumerador) y
4   * permite
5   * visualizar la manera en la que se maneja el tipo de dato.
6   */
7  int main()
8  {
9      // declaración de la enumeración
10     enum boolean
11     {NO, YES};
12     // declaración de una variable tipo enumeración
13     enum boolean
14     valorBooleano;
15     valorBooleano = YES;
16     // Se comprueba que el valor de una enumeración es entero
17     printf("%d\n", valorBooleano);
18     // Se comprueba que el valor de una enumeración se pueda asignar
19     enum diasSemana
20     {
21     LUNES, MARTES, MIERCOLES=5, JUEVES, VIERNES};
22     printf("%i\n", MARTES);
23     printf("%d\n", MIERCOLES);
24     printf("%i\n", JUEVES);
25     printf("%d\n", VIERNES);
26     return 0;
27 }
28
```



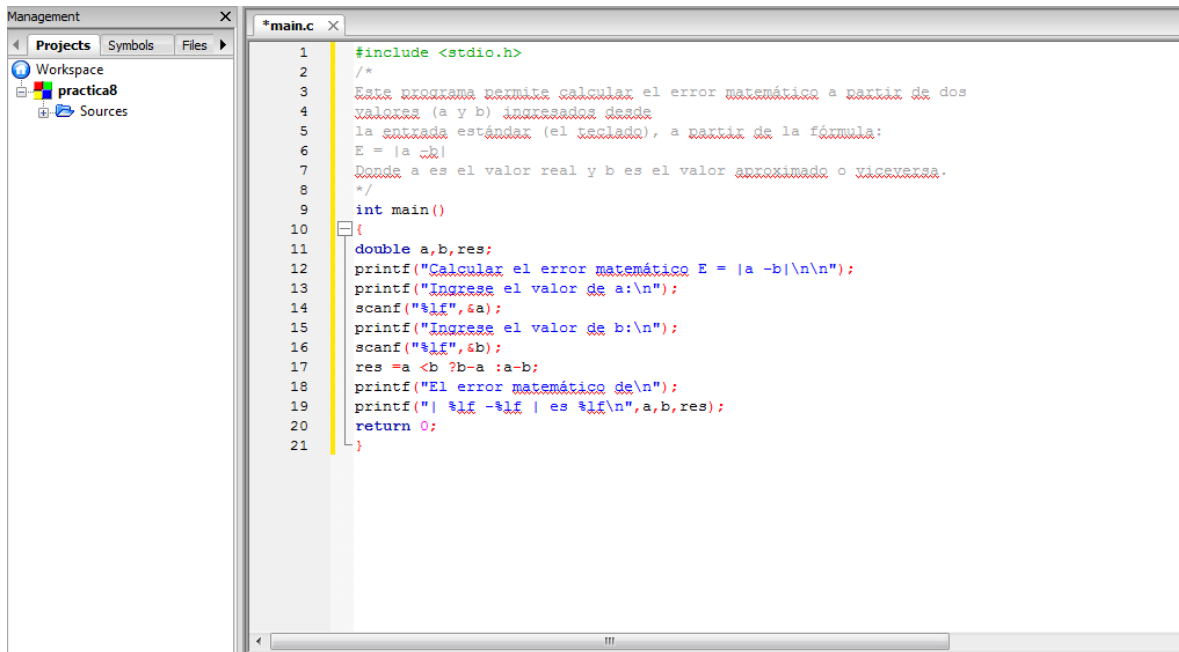
```
C:\Users\L&L\Documents\practica8\bin\Debug\practica8.exe
1
0
1
5
6
2
Process returned 0 (0x0)   execution time : 0.140 s
Press any key to continue.
-
```

Por último se vio muy general la estructura de control selectiva condicional

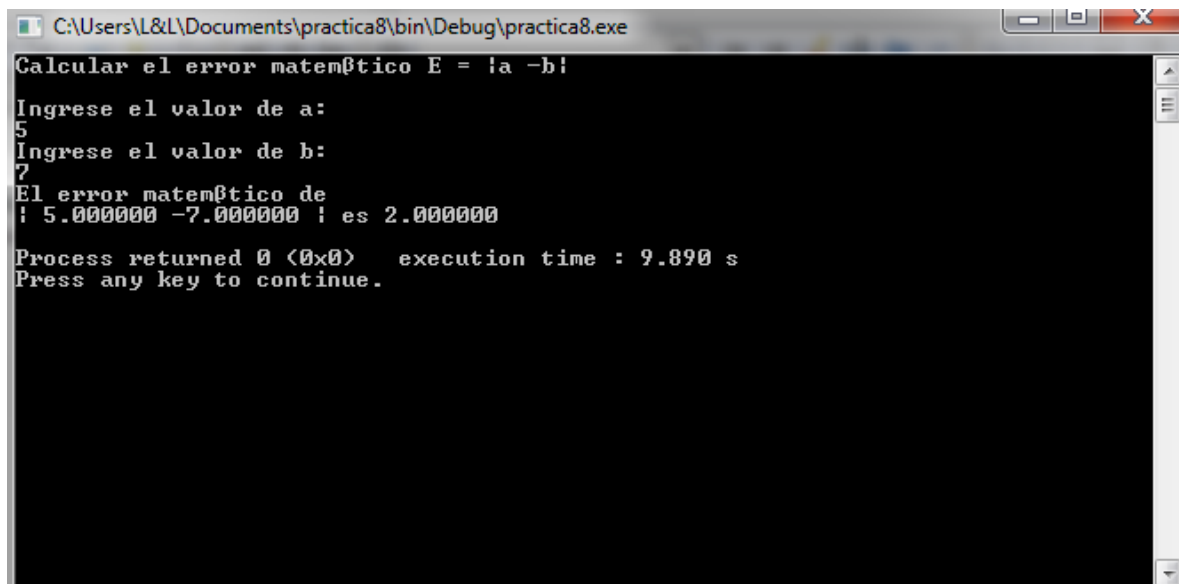
Esta también llamado operador ternario nos permite realizar una comparación rápida. La sintaxis es la siguiente:

Condición? SiSeCumple: SiNoSeCumple.

El ejemplo fue el siguiente:



```
1  #include <stdio.h>
2  /*
3   Esta programa permite calcular el error matemático a partir de dos
4   valores (a y b) ingresados desde
5   la entrada estándar (el teclado), a partir de la fórmula:
6   E = |a - b|
7   Donde a es el valor real y b es el valor aproximado o viceversa.
8   */
9  int main()
10 {
11     double a,b,res;
12     printf("Calcular el error matemático E = |a -b|\n\n");
13     printf("Ingrese el valor de a:\n");
14     scanf("%lf",&a);
15     printf("Ingrese el valor de b:\n");
16     scanf("%lf",&b);
17     res = a < b ? b-a : a-b;
18     printf("El error matemático de\n");
19     printf("| %lf -%lf | es %lf\n",a,b,res);
20     return 0;
21 }
```



```
C:\Users\L&L\Documents\practica8\bin\Debug\practica8.exe
Calcular el error matemático E = |a -b|
Ingrese el valor de a:
5
Ingrese el valor de b:
7
El error matemático de
| 5.000000 -7.000000 | es 2.000000
Process returned 0 (0x0)   execution time : 9.890 s
Press any key to continue.
```


Conclusión

Nuestras estructuras de control selectivas seguirán un ciclo y este terminara dependiendo de lo que tengamos en las condiciones, es algo muy evidente que observamos a un poco de detalle en los ejemplos que se abordaron. Todo lo que se realizo a lo largo de esta práctica se baso en condiciones lógicas.