



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: ALEJANDRO PIMENTEL

Asignatura: FUNDAMENTOS DE PROGRAMACIÓN

Grupo: 3

No de Práctica(s): 7

Integrante(s): CARLOS SOTELO LEYVA

*No. de Equipo de
cómputo empleado:* 35

No. de Lista o Brigada: 47

Semestre: 2020-1

Fecha de entrega: 06/10/2019

Observaciones: Tarde entrega.

CALIFICACIÓN: 8

Si se deseara imprimir los resultados de multiplicar un número fijo por otro que adopta valores entre 0 y 9, la forma normal de programar esto sería crear una constante para el primer número y un par de variables para el segundo y para el resultado del producto. Una variable, en realidad, no es más que un nombre para identificar una (o varias) posiciones de memoria donde el programa guarda los distintos valores de una misma entidad. Un programa debe definir todas las variables que utilizará antes de comenzar a usarlas, a fin de indicarle al compilador de que tipo serán y, por tanto, cuánta memoria debe destinar para albergar a cada una de ellas.

Así, una variable es un lugar donde se puede almacenar temporalmente un dato. En C las variables tienen un nombre que las identifica, y sirve para hacer referencia a ellas. También tienen un tipo, que es el tipo de datos que puede almacenar. El valor de las variables es, como su propio nombre indica, variable. Podemos alterar su valor en cualquier punto del programa.

OBJETIVO:

Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

TIPOS DE VARIABLES:

- Short int
- Unsigned short int
- Unsigned int
- Int
- Long int
- Unsigned long int
- Long long int
- Unsigned long long int

Estos tipos de variables tienen ciertos bytes que ocupan y también tienen cierto rango de números para trabajar.

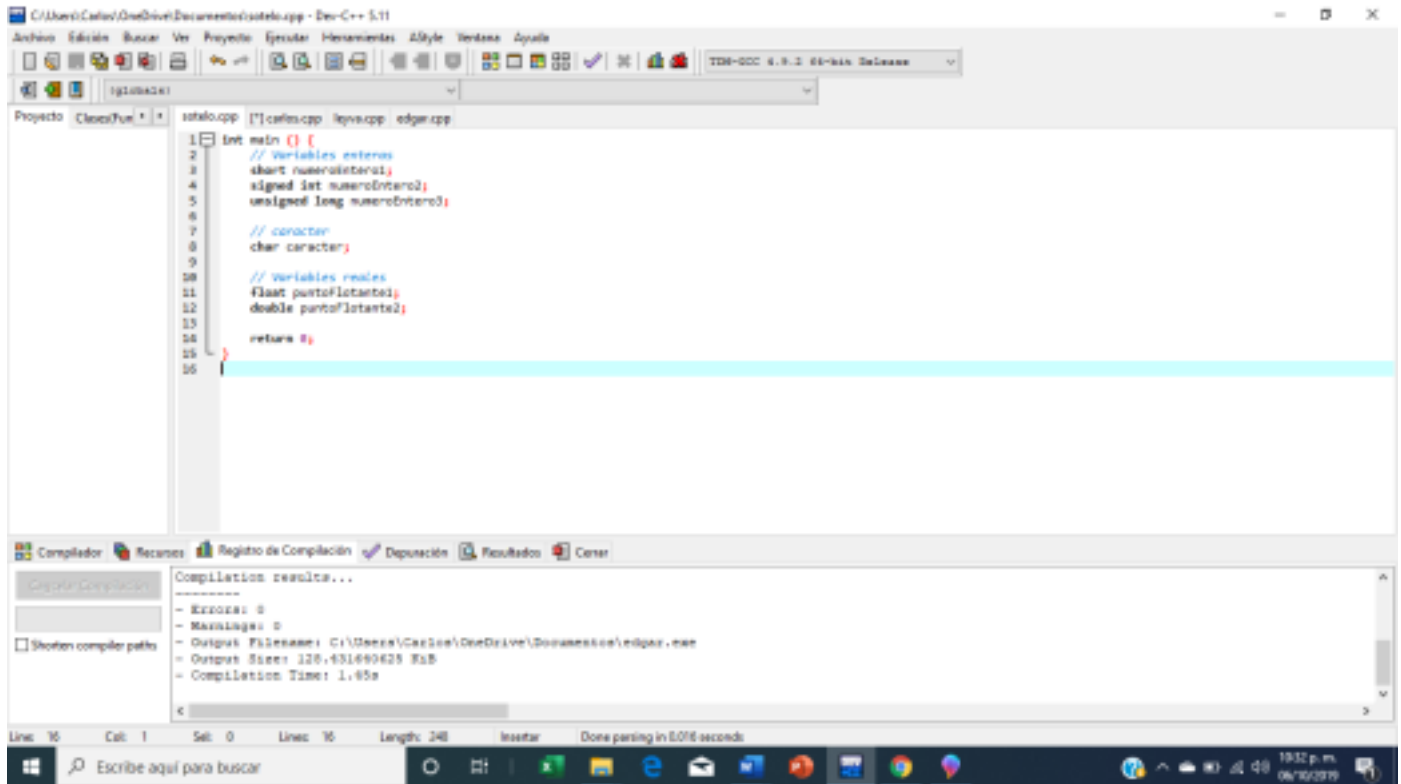
Tipos de variables:

Para los reales, se tienen también diferentes tipos de variables que asignan más bits para tener mayor rango y mayor precisión. Las variables reales siempre poseen signo.

TIPO	BITS	VALOR MÍNIMO	VALOR MÁXIMO
Float	32	3.4 E-38	3.4 E38

Double	64	1.7 E-308	1.7 E308
Long double	80	3.4 E-4932	3.4 E4932

Tipos de variables (actividad 1):



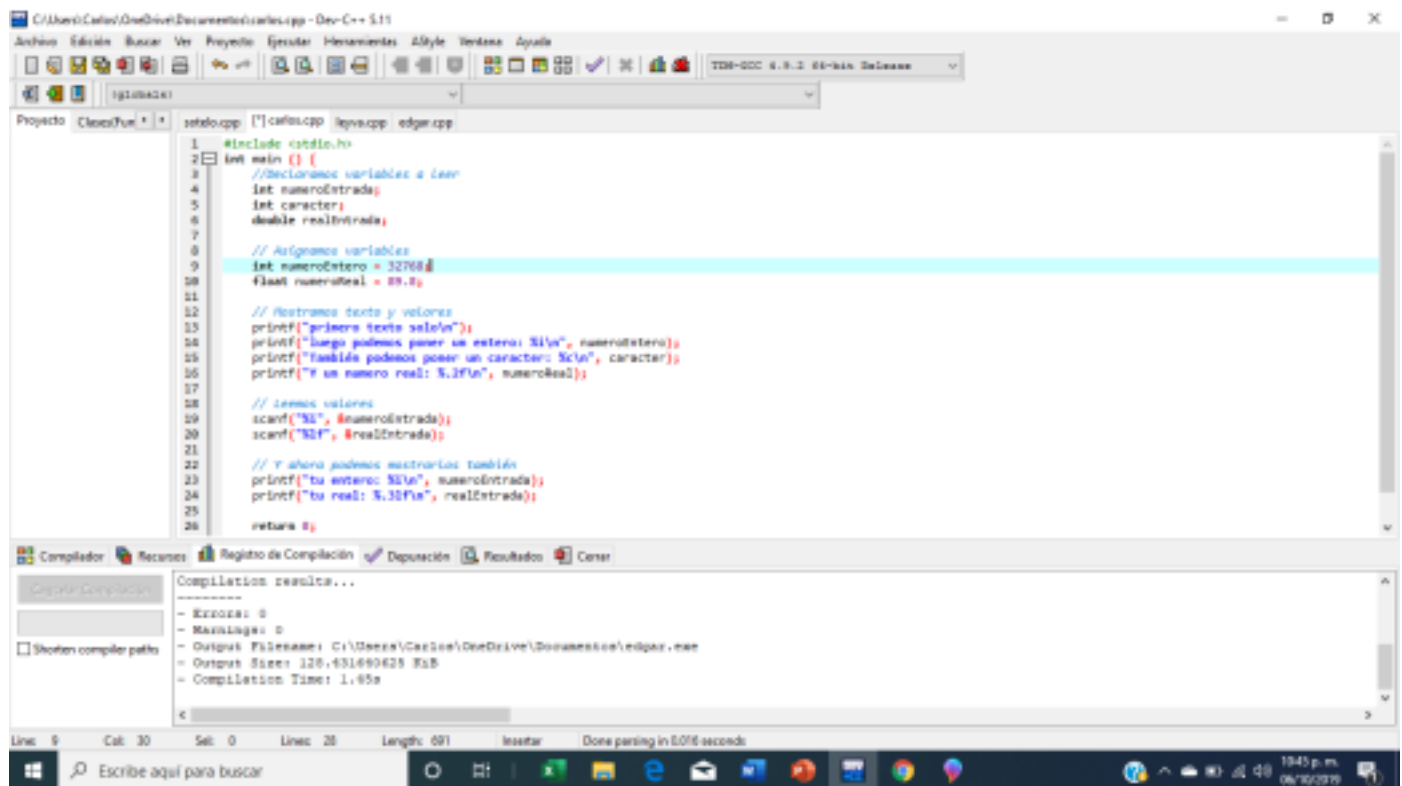
Como se puede ver en la captura, en la parte de abajo ahí mismo aparece la terminal que tiene la aplicación que se descargó, nos dice que lo que se escribió no muestra ningún error.

MOSTRAR Y LEER:

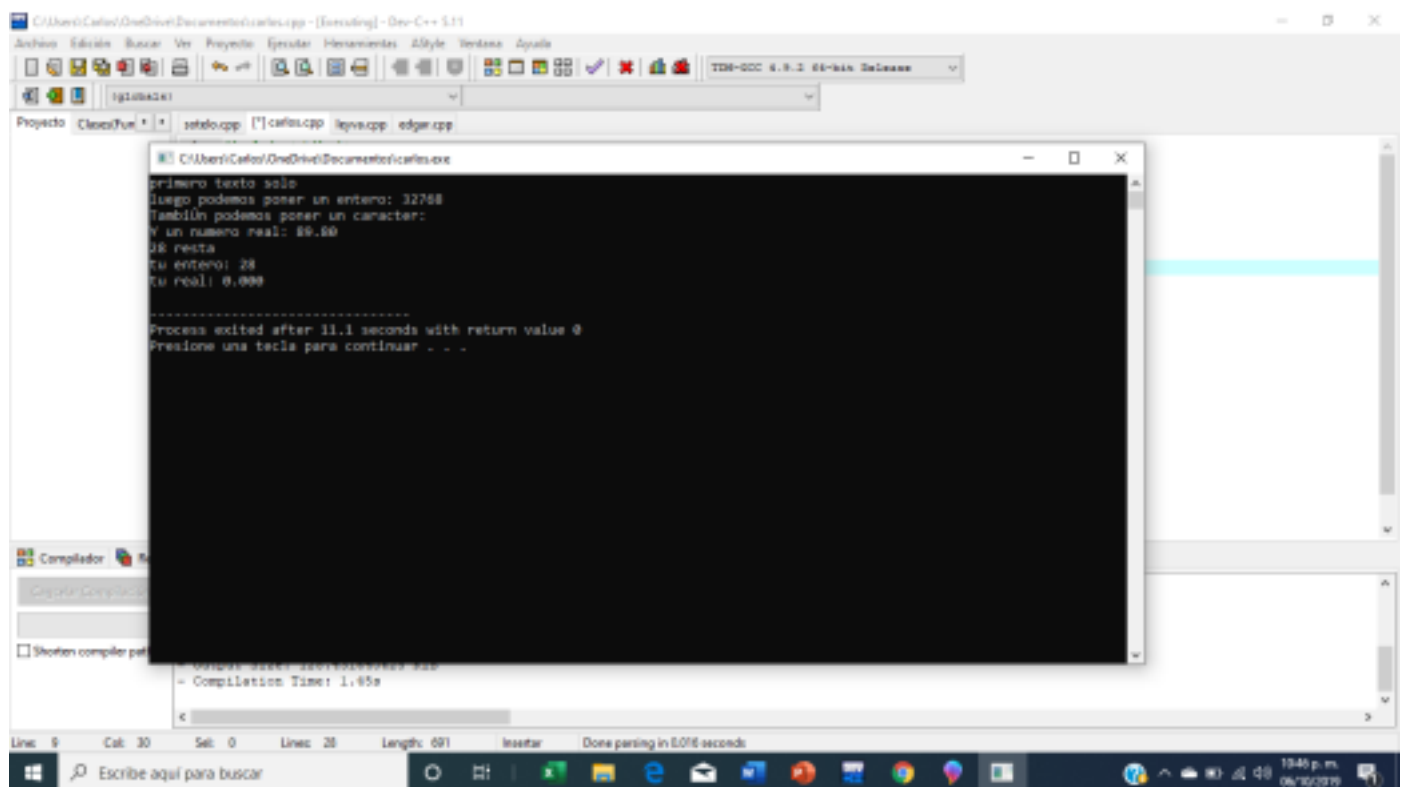
TIPO DE DATO:	ESPECIFICADOR DE FORMATO:
ENTERO	%d, %i, %ld, %li, %o, %x
FLOTANTE	%f, %lf, %e, %g
CARÁCTER	%c, %d, %i, %o, %x
CADENA DE CARACTERES	%s

MOSTRAR Y LEER (Actividad 2):

En es parte desarrollamos igual el código por asi decirlo que esta en la práctica:



Y ahora la ejecutamos:



OPERADORES:

A continuación veremos algunos operadores básicos:

OPERADOR	OPERACIÓN	USO	RESULTADO
+	Suma	125.78 + 62.5	188.28
-	Resta	65.3 - 32.33	32.97
*	Multiplicación	8.27 * 7	57.75
/	División	15 / 4	3.75
%	Módulo	4 % 2	0

OPERADORES (actividad 3):

Primero lo compilamos:

```

1  #include <stdio.h>
2  int main() {
3
4      int dos, tres, cuatro, cinco;
5      double resultado;
6
7      dos = 2;
8      tres = 3;
9      cuatro = 4;
10     cinco = 5;
11
12     resultado = cinco/dos;
13     printf("5 / 2 = %.15f\n", resultado);
14
15     resultado = (double)cinco/dos;
16     printf("5 / 2 = %.15f\n", resultado);
17
18     return 0;
19 }
20

```

Compilation results...

```

- Errors: 0
- Messages: 0
- Output Filename: C:\Users\Cecilio\Documents\edgar.exe
- Output Size: 128,431,699,628 KB
- Compilation Time: 1.45s

```

Y ahora lo ejecutamos:

```

1 #include <stdio.h>
2 int main() {
3     5 / 2 = 2.0000000000
4     5 / 2 = 2.5000000000
5 }

```

Process exited after 0.03167 seconds with return value 0
Presione una tecla para continuar . . .

COMPARACIONES:

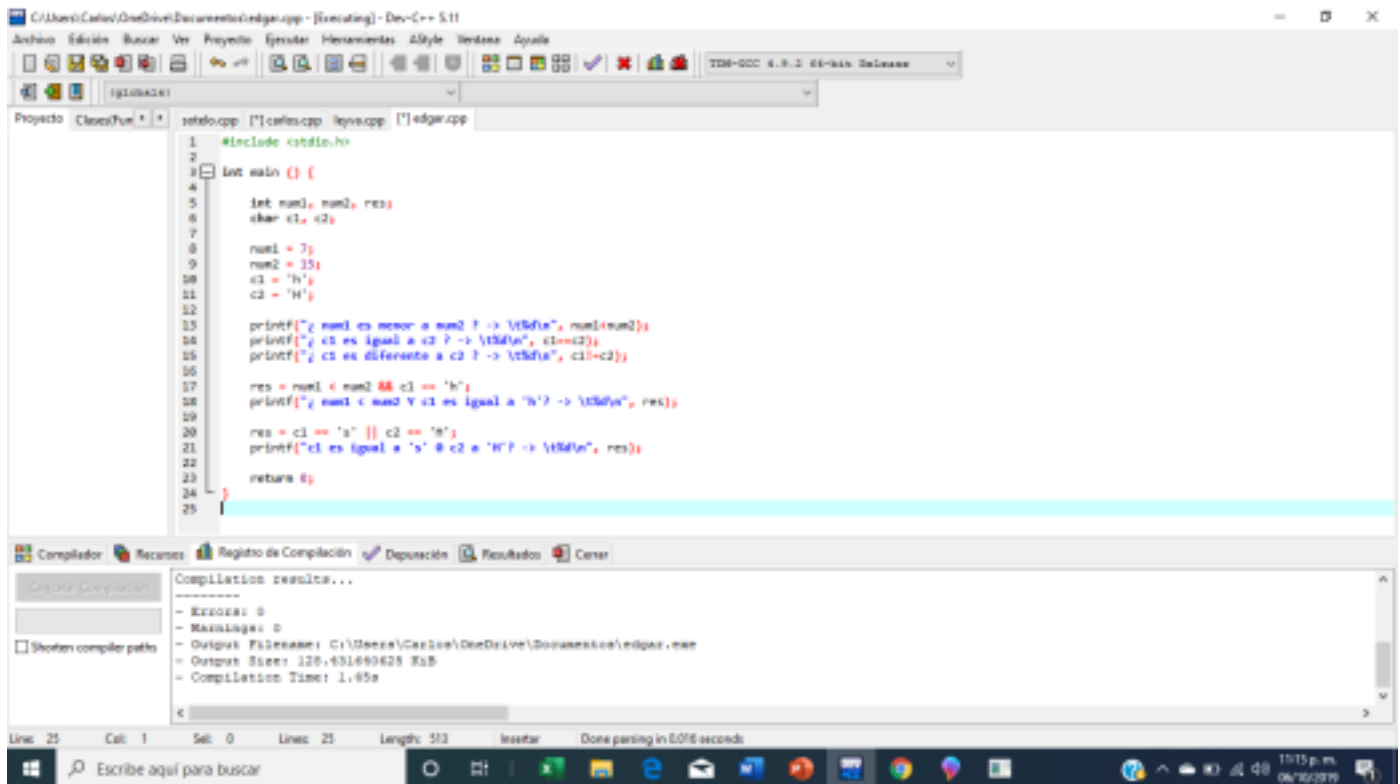
OPERADOR	OPERACIÓN	USO	RESULTADO
==	Igual que	'h' == 'H'	Falso
!=	Diferente a	'a' != 'b'	Verdadero
<	Menor que	7<15	Verdadero
>	Mayor que	11>22	Falso
<=	Menor o igual	15<=22	Verdadero
>=	Mayo o igual	20>=35	Falso

OPERADORES LÓGICOS:

OPERADOR	OPERACIÓN
!	No
&&	Y
	O

OPERADORES LÓGICOS (actividad 4):

Primero se compilan para ver si no hay ningún error:

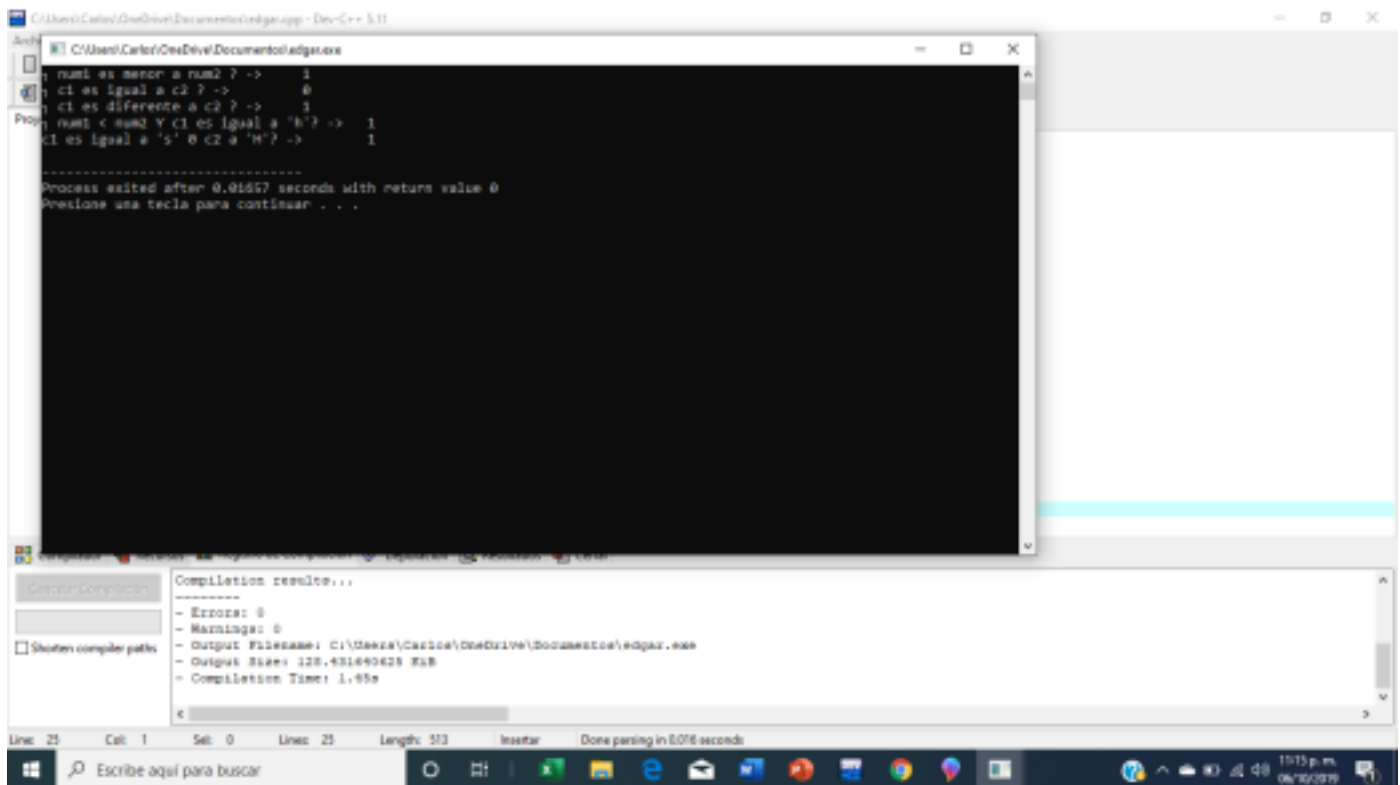


```
1 #include <stdio.h>
2
3 int main () {
4
5     int num1, num2, res;
6     char c1, c2;
7
8     num1 = 7;
9     num2 = 15;
10    c1 = 'h';
11    c2 = 'H';
12
13    printf("%d num1 es menor a num2 ? -> %d\n", num1 < num2);
14    printf("%d c1 es igual a c2 ? -> %d\n", c1 == c2);
15    printf("%d c1 es diferente a c2 ? -> %d\n", c1 != c2);
16
17    res = num1 < num2 && c1 == 'h';
18    printf("%d num1 < num2 y c1 es igual a 'h' ? -> %d\n", res);
19
20    res = c1 == 'a' || c2 == 'H';
21    printf("%d c1 es igual a 'a' o c2 a 'H' ? -> %d\n", res);
22
23    return 0;
24 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Carlos\OneDrive\Documents\edgar.exe
- Output Size: 128,431,690,628 B
- Compilation Time: 1.65s

Y ahora se ejecuta:



```
num1 es menor a num2 ? -> 0
c1 es igual a c2 ? -> 0
c1 es diferente a c2 ? -> 1
num1 < num2 y c1 es igual a 'h' ? -> 1
c1 es igual a 'a' o c2 a 'H' ? -> 1

Process exited after 0.01657 seconds with return value 0
Presione una tecla para continuar . . .
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Carlos\OneDrive\Documents\edgar.exe
- Output Size: 128,431,690,628 B
- Compilation Time: 1.65s

REPORTE DE PRÁCTICA:

En esta práctica como Podemos observar cada vez más nos vamos adentrando a los que es la programación en c, en esta práctica vimos principales operadores lógicos, algunos símbolos para poder comparar, operadores como lo son suma, resta, multiplicación, division y módulos, también vimos los diferentes tipos de datos

como son el entero, double, carácter y las cadenas de los caracteres, los diferentes tipos de variable que Podemos emplear para este tipo de programación, algunos de ellos son: int, long long int, long int, short int, etc. Todos estos tipos de variables al final siempre llevan la palabra int (entero).

Esto es como lo básico que Debemos de saber para poder efectuar con precision las compilaciones y ejecuciones de programas a future, por decirlo así que el programa corra dependiendo de lo que nos pidan a nosotros, eso es todo profe.