



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor: GARCIA MORALES KARINA ING.

Asignatura: FUNDAMENTOS DE PROGRAMACION

Grupo: 1121

No de Práctica(s): 6

Integrante(s): JOSE DANIEL CALLEJAS SANDOVAL

*No. de Equipo de
cómputo empleado:* 36

Semestre: 1

Fecha de entrega: 3-10-2018

Observaciones:

CALIFICACIÓN: _____

Guía práctica de estudio 06: Entorno de C (editores, compilación y ejecución)

OBJETIVOS: Conocer y usar los ambientes y herramientas para el desarrollo y ejecución de programas en Lenguaje C, como editores y compiladores en diversos sistemas operativos.

DESARROLLO: Es el lenguaje que comprendemos nosotros y es de alto nivel.

De código fuente pasa al compilador y se genera el programa ejecutable (extensión .obj O .exe)

No es lo mismo un editor de texto que un procesador de texto. El primero edita un texto plano que puede tener muchas utilidades como guardar una configuración (solo es texto-bloc de notas).

Un procesador de texto permite dar formato al texto, a la hoja donde está escrito, incrustar imágenes, etc., su salida puede ser un archivo de texto plano que contiene etiquetas que señalan el formato que se le dio al texto o algo un poco más complejo.

Un compilador, pasa el programa a código binario y además valida que nuestra sintaxis sea valida

GCC Conjunto de compilador

Creamos el archivo, lo compilamos y después lo ejecutamos.

Editor Visual Interface de GNU/Linux (VI)

El editor vi (visual interface) es el editor más común en cualquier distribución de sistemas operativos con núcleo basado en UNIX. Está disponible en línea de comandos y si el sistema operativo tiene entorno gráfico se puede acceder a él desde la terminal.

VI es un editor que puede resultar difícil de usar en un inicio. Aunque existen editores más intuitivos en su uso, en muchas ocasiones VI es el único disponible.

Para iniciar VI, debe taclearse desde la línea de comandos:

```
vi nombre_archivo[.ext]
```

Donde “nombre_archivo” puede ser el nombre del archivo a editar o el nombre de un archivo nuevo que se creará con VI. Es válido incluir la ruta donde se localiza o localizará el archivo. Existen más métodos de apertura para usuarios más avanzados. VI tiene tres modos de operación.

Modo comando

Es el modo por defecto de VI cuando se abre. Las teclas presionadas ejecutan diversas acciones predeterminadas y no se puede editar el texto libremente. Los comandos son sensitivos a las mayúsculas y a las minúsculas. Algunos ejemplos son:

↑ o k mueve el cursor hacia arriba.

↓ o j mueve el cursor hacia abajo.

← o h mueve el cursor hacia la izquierda.

→ o l mueve el cursor hacia la derecha.

1G lleva el cursor al comienzo de la primera línea.

G lleva el cursor al comienzo de la última línea.

x borra el carácter marcado por el cursor.

dd borra o corta la línea donde está el cursor.

ndd donde n es la cantidad de líneas que se borrarán o cortarán después del cursor.

D borra o corta desde la posición de cursor hasta el final de la línea.

dw borra o corta desde la posición del cursor hasta el final de una palabra.

yy copia la línea donde está el cursor.

p pega un contenido copiado o borrado.

u deshace el último cambio.

Modo de última línea

Se puede acceder a él desde el modo de última línea. Es muy similar al modo comando, pero los comandos no tendrán efecto hasta que se presiona la tecla Enter además de que se visualizará el comando en la última línea del editor. Es posible cancelar el comando con la tecla Esc. Los comandos de última línea se caracterizan porque inician con /, ? o :. Algunos ejemplos son:

/texto donde la cadena texto será buscada hacia delante de donde se encuentra el cursor.

?texto donde la cadena texto será buscada hacia atrás de donde se encuentra el cursor.

:q para salir de VI sin haber editado el texto desde la última vez que se guardó.

:q! para salir de VI sin guardar los cambios.

:w para guardar los cambios sin salir de VI.

:w archivo para realizar la orden “guardar como”, siendo archivo el nombre donde se guardará el documento.

:wq guarda los cambios y sale de VI.

Modo insertar

Este modo permite insertar texto. Las teclas presionadas ya no harán una acción como en el modo comando sino será el contenido que formará el texto del documento. Se puede desplazar con las flechas del teclado y borrar con la tecla de retroceso o de suprimir.

Para ingresar al modo insertar existen varios comandos:

i pasa al modo insertar poniendo el texto a la izquierda del cursor.

a pasa al modo insertar poniendo el texto a la derecha del cursor.

A pasa al modo insertar colocando el texto al final de la línea donde el cursor se encuentra.

I pasa al modo insertar colocando el texto al principio de la línea donde el cursor se encuentra.

O coloca una línea arriba de la línea seleccionada por el cursor y pasa al modo insertar.

o coloca una línea debajo de la seleccionada por el cursor y pasa al modo insertar.

Para salir del modo insertar debe presionarse la tecla Esc. Para verificar que se encuentra en modo insertar es se puede ver -- insertar -- en la última línea del editor.

GNU NANO

Es un editor de texto disponible para sistemas operativos basados en UNIX en línea de comandos. Se puede acceder en un entorno gráfico desde la aplicación de terminal.

Este editor es mucho más intuitivo que VI, aunque menos potente. No es necesario saber cómo se utiliza sino proporciona una interfaz que describe los comandos básicos.

NANO es un editor clon de otro editor llamado PICO.

Para iniciar NANO, debe taclearse desde la línea de comandos:

```
nano nombre_archivo[.ext]
```

Donde “nombre_archivo” puede ser el nombre del archivo a editar o el nombre de un archivo nuevo.

Una vez en el editor, en la parte inferior se pueden observar los comandos básicos. Si se presiona la tecla F1 es posible visualizar la ayuda con la lista de todos comandos que existen.

Los atajos de teclado pueden corresponder a:

^ que es la tecla Ctrl.

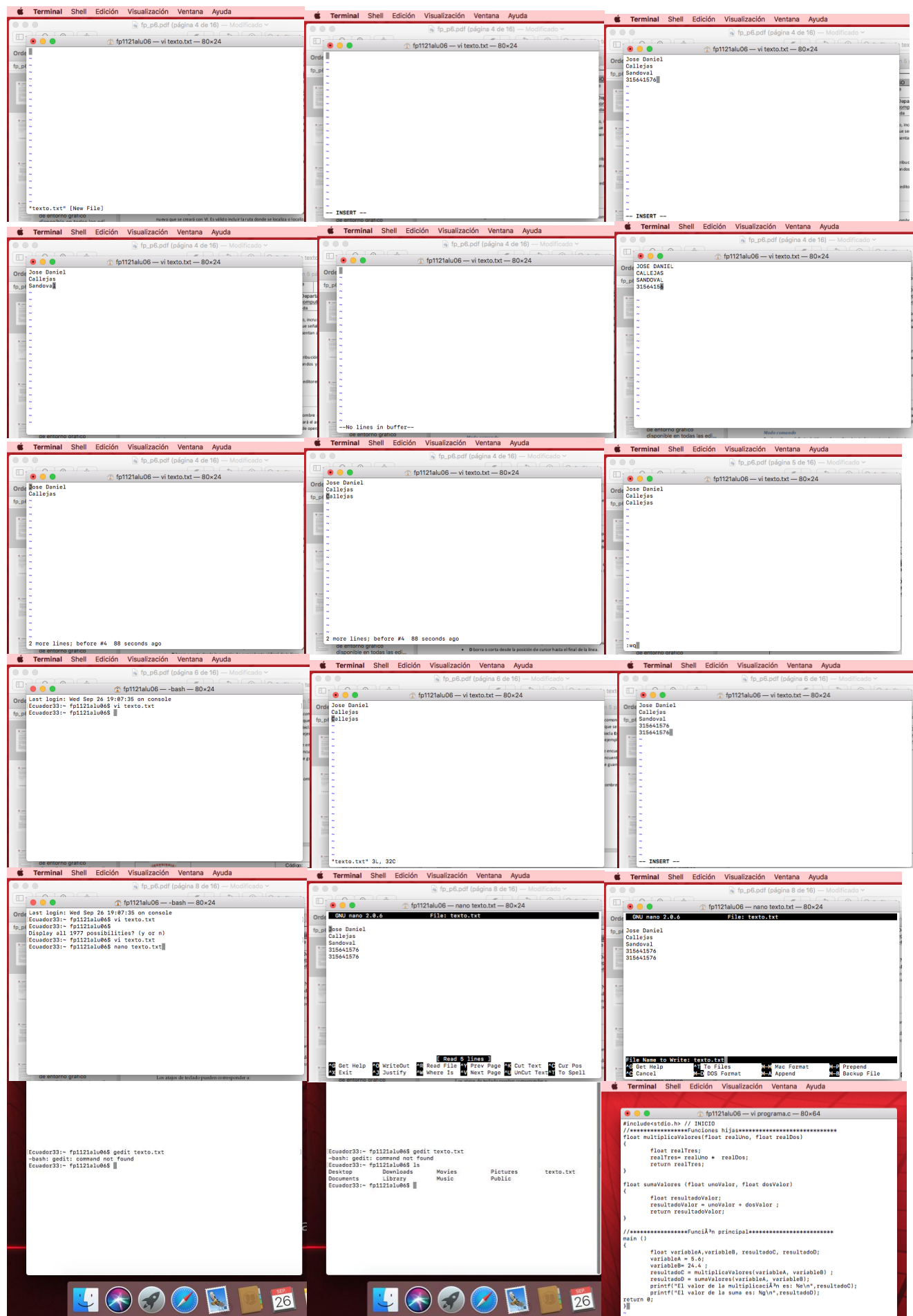
M- que es la tecla Esc o bien Alt.

Compiladores

Una vez codificado un programa en C en algún editor de texto, éste debe ser leído por un programa que produzca un archivo ejecutable. A este programa se le conoce como compilador y depende totalmente del hardware de la computadora y el sistema operativo que corre sobre ella.

Recordando, un programa en C es universal, por lo que cada una de las instrucciones que lo conforman debe poder entenderlas muchos de los equipos en el mercado, aunque su naturaleza sea distinta. Por ello, un compilador depende del equipo, porque es un traductor que transforma

PRACTICA DURANTE LA CLASE:



```
Ecuador33~ fp121al068 gedit texto.txt
-bash: gedit: command not found
Ecuador33~ fp121al068 ls
Desktop  Downloads  Movies  Pictures  texto.txt
Documents Library Music Public
Ecuador33~ fp121al068 vi programa.c
Ecuador33~ fp121al068 gcc programa.c -o programa.out
programa.c:18:1: warning: type specifier missing, defaults to 'int'
      [-Wimplicit-int]
main ()
^
1 warning generated.
Ecuador33~ fp121al068
```

```
fp121al068 - vi programa.c - 80x64
#include<stdio.h> // INICIO
//=====funciones hijas=====
float multiplicarValores(float resUno, float resDos)
{
    float realRes;
    realRes= resUno * resDos;
    return realRes;
}

float sumaValores (float unoValor, float dosValor)
{
    float resultadoValor;
    resultadoValor = unoValor + dosValor ;
    return resultadoValor;
}

//=====FunciÃn principal=====
int main ()
{
    float variableA,variableB, resultadoC, resultadoD;
    variableA = 5.6;
    variableB= 24.4 ;
    resultadoC = multiplicarValores(variableA, variableB) ;
    resultadoD = sumaValores(variableA, variableB);
    printf("El valor de la multiplicaciÃn es: %f\n",resultadoC);
    printf("El valor de la suma es: %f\n",resultadoD);

    return 0;
}
```

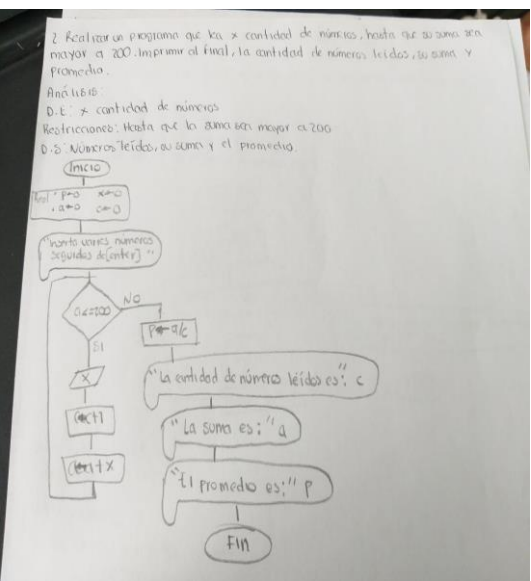
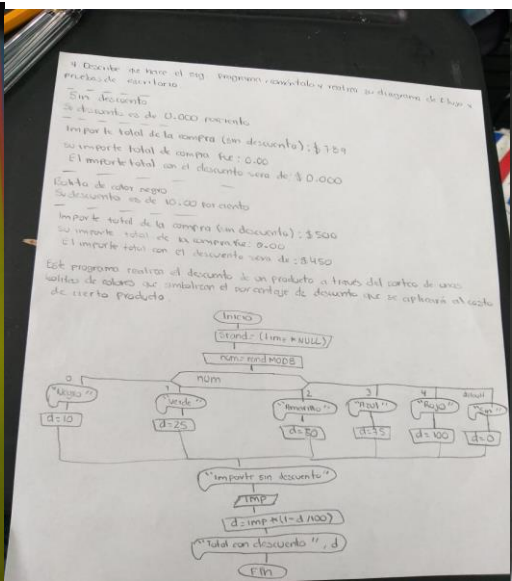
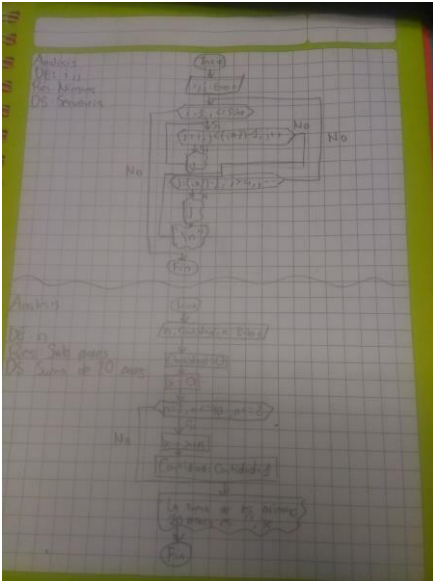
```
Ecuador33~ fp121al068 gedit texto.txt
-bash: gedit: command not found
Ecuador33~ fp121al068 ls
Desktop  Downloads  Movies  Pictures  texto.txt
Documents Library Music Public
Ecuador33~ fp121al068 vi programa.c
Ecuador33~ fp121al068 gcc programa.c -o programa.out
programa.c:18:1: warning: type specifier missing, defaults to 'int'
      [-Wimplicit-int]
main ()
^
1 warning generated.
Ecuador33~ fp121al068 vi programa.c
Ecuador33~ fp121al068 gcc programa.c -o programa.out
Ecuador33~ fp121al068
```

```
Ecuador33~ fp121al068 gedit texto.txt
-bash: gedit: command not found
Ecuador33~ fp121al068 ls
Desktop  Downloads  Movies  Pictures  texto.txt
Documents Library Music Public
Ecuador33~ fp121al068 vi programa.c
Ecuador33~ fp121al068 gcc programa.c -o programa.out
programa.c:18:1: warning: type specifier missing, defaults to 'int'
      [-Wimplicit-int]
main ()
^
1 warning generated.
Ecuador33~ fp121al068 vi programa.c
Ecuador33~ fp121al068 gcc programa.c -o programa.out
Ecuador33~ fp121al068 vi programa.c
Ecuador33~ fp121al068 ls
Desktop  Library  Movies  Pictures  programa.out  texto.txt
Documents Library Music Public
Downloads Music Public programa.c
Ecuador33~ fp121al068
```

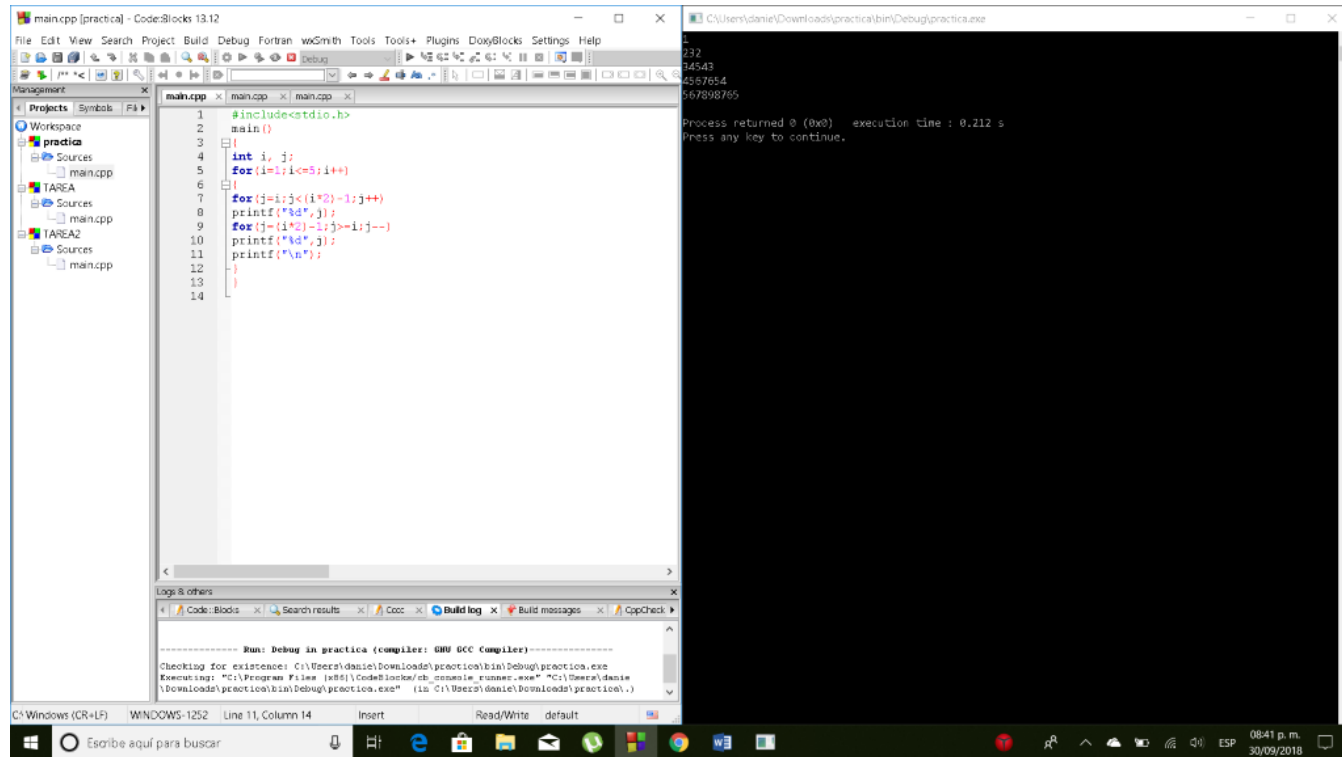
```
Ecuador33~ fp121al068 gedit texto.txt
-bash: gedit: command not found
Ecuador33~ fp121al068 ls
Desktop  Downloads  Movies  Pictures  texto.txt
Documents Library Music Public
Ecuador33~ fp121al068 vi programa.c
Ecuador33~ fp121al068 gcc programa.c -o programa.out
programa.c:18:1: warning: type specifier missing, defaults to 'int'
      [-Wimplicit-int]
main ()
^
1 warning generated.
Ecuador33~ fp121al068 vi programa.c
Ecuador33~ fp121al068 gcc programa.c -o programa.out
Ecuador33~ fp121al068 vi programa.c
Ecuador33~ fp121al068 ls
Desktop  Library  Movies  Pictures  programa.out  texto.txt
Documents Library Music Public
Downloads Music Public programa.c
Ecuador33~ fp121al068 ./programa.out
El valor de la multiplicaciÃn es: 1.366400e+02
El valor de la suma es: 30
Ecuador33~ fp121al068
```

Ejercicios de tarea:

DIAGRAMAS



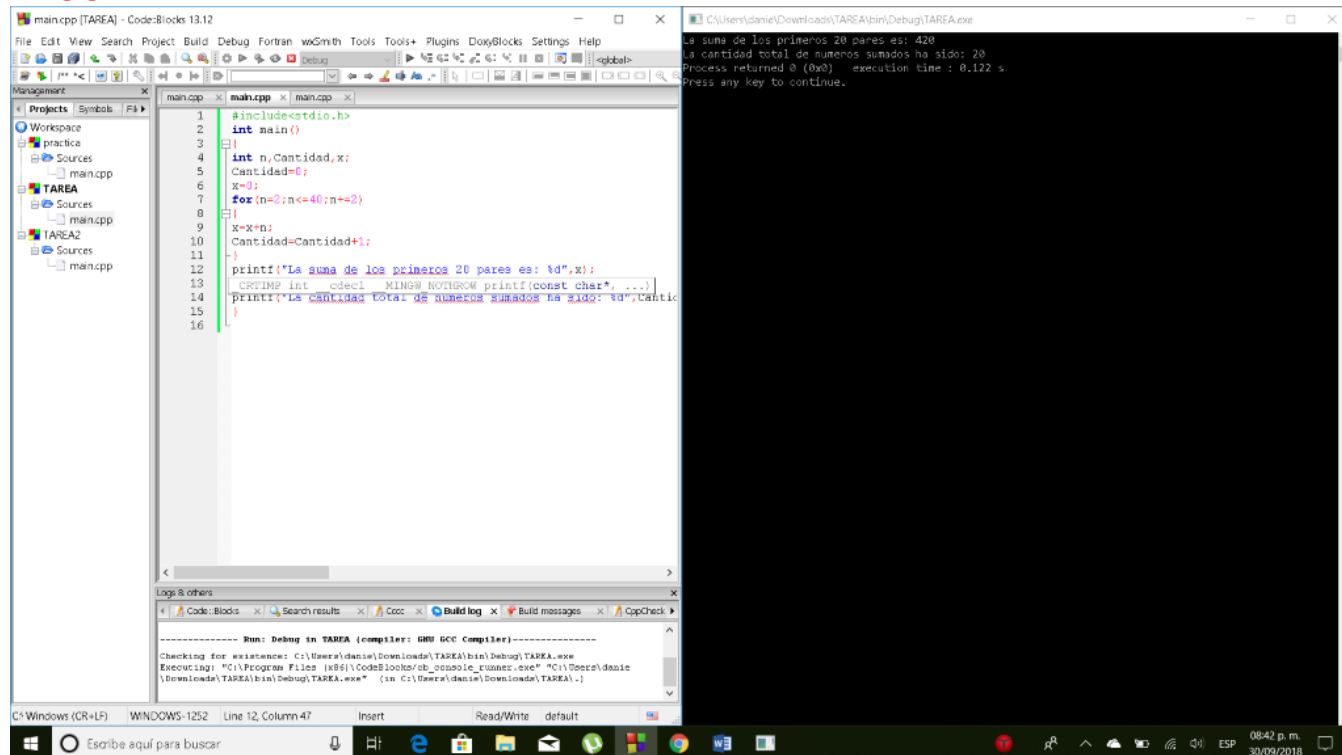
PROGRAMA 1



```
main.cpp [practica] - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects: Workspace, practica, Sources, main.cpp, TAREA, Sources, main.cpp, TAREA2, Sources, main.cpp
main.cpp x1 main.cpp x2 main.cpp x3
1 #include<stdio.h>
2 main()
3 {
4     int i, j;
5     for(i=1; i<=5; i++)
6     {
7         for(j=i; j<=(i*2)-1; j++)
8             printf("%d", j);
9         for(j=(i*2)-1; j>=i; j--)
10             printf("%d", j);
11         printf("\n");
12     }
13 }
14

Log: 8 others
Code::Blocks Search results Coc Build log Build messages CppCheck
Run: Debug in practica (compiler: GNU GCC Compiler)
Checking for existence: C:\Users\daniel\Downloads\practica\bin\Debug\practica.exe
Executing: "C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe" "C:\Users\daniel\Downloads\practica\bin\Debug\practica.exe" (in C:\Users\daniel\Downloads\practica\bin\Debug\practica.exe)
Process returned 0 (0x0) execution time : 0.212 s
Press any key to continue.
```

PROGRAMA 2



```
main.cpp [TAREA] - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects: Workspace, practica, Sources, main.cpp, TAREA, Sources, main.cpp, TAREA2, Sources, main.cpp
main.cpp x1 main.cpp x2 main.cpp x3
1 #include<stdio.h>
2 int main()
3 {
4     int n, Cantidad, x;
5     Cantidad=0;
6     x=0;
7     for(n=2; n<=40; n+=2)
8     {
9         x=x+n;
10        Cantidad=Cantidad+1;
11    }
12    printf("La suma de los primeros 20 pares es: %d", x);
13    CRTIMP int _cdecl MINGW_NOTHROW printf(const char*, ...)
14    printf("La cantidad total de numeros sumados ha sido: %d", Cantidad);
15 }
16

Log: 8 others
Code::Blocks Search results Coc Build log Build messages CppCheck
Run: Debug in TAREA (compiler: GNU GCC Compiler)
Checking for existence: C:\Users\daniel\Downloads\TAREA\bin\Debug\TAREA.exe
Executing: "C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe" "C:\Users\daniel\Downloads\TAREA\bin\Debug\TAREA.exe" (in C:\Users\daniel\Downloads\TAREA\bin\Debug\TAREA.exe)
Process returned 0 (0x0) execution time : 0.122 s
Press any key to continue.
```

The image displays a Windows desktop with two Code::Blocks IDE windows open. The left window, titled 'main.cpp [TAREA2] - Code::Blocks 13.12', shows the source code of a C++ program. The code includes `<stdio.h>`, defines `main()`, and declares variables `num`, `sum`, `cantidad`, `prom`, and `sum`. It uses a `while` loop to sum numbers until `sum >= 200`. The program prints the sum, the number of numbers summed, and the average. The right window, titled 'C:\Users\danie\Downloads\TAREA2\bin\Debug\TAREA2.exe', shows the program's execution output. The output matches the code's logic, showing a sum of 190, 2 numbers summed, and an average of 95.000000. The process returned 0 (0x0) and took 3.606 seconds to execute. The bottom status bar of the IDE shows 'Line 20, Column 11'.

```
#include<stdio.h>
int main ()
{
    int num,sum,cantidad;
    float prom;
    sum=0;
    cantidad=0;
    while (sum<=200)
    {
        printf("Dame un numero entero a sumar:");
        scanf("%d",&num);
        sum=num+sum;
        cantidad=cantidad+1;
    }
    sum=sum-num;
    printf("La suma sin sobrepasar el 200 fue: %d",sum);
    cantidad=cantidad-1;
    printf("\nLa cantidad de numeros sumados fue: %d",cantidad);
    prom=sum/cantidad;
    printf("\nEl promedio fue: %f",prom);
}
```

Dame un numero entero a sumar:90
Dame un numero entero a sumar:100
Dame un numero entero a sumar:20
La suma sin sobrepasar el 200 fue: 190
La cantidad de numeros sumados fue: 2
El promedio fue: 95.000000
Process returned 0 (0x0) execution time : 3.606 s
Press any key to continue.

Run: Debug in TAREA2 (compiler: GNU GCC Compiler)
Checking for runtime: C:\Users\danie\Downloads\TAREA2\bin\Debug\TAREA2.exe
Executing: "C:\Program Files (x86)\CodeBlocks\cb_gnu_gcc_compiler.exe" "C:\Users\danie\Downloads\TAREA2\bin\Debug\TAREA2.exe" (in C:\Users\danie\Downloads\TAREA2\)

Windows (CR-IF) WINDOWS-1252 Line 20, Column 11 Insert Read/Write default

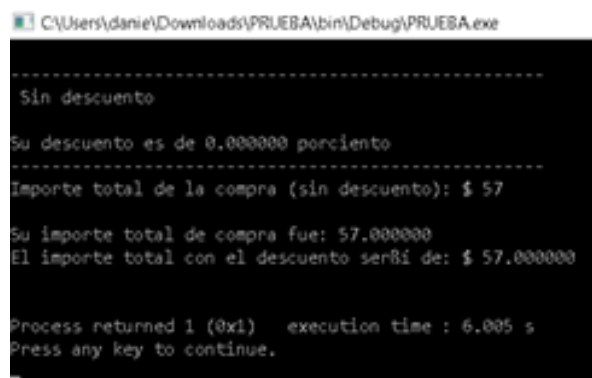
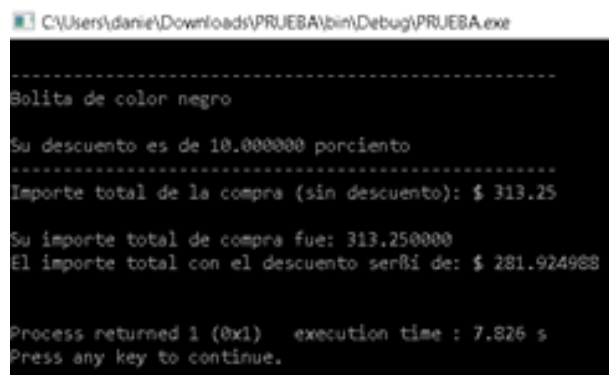
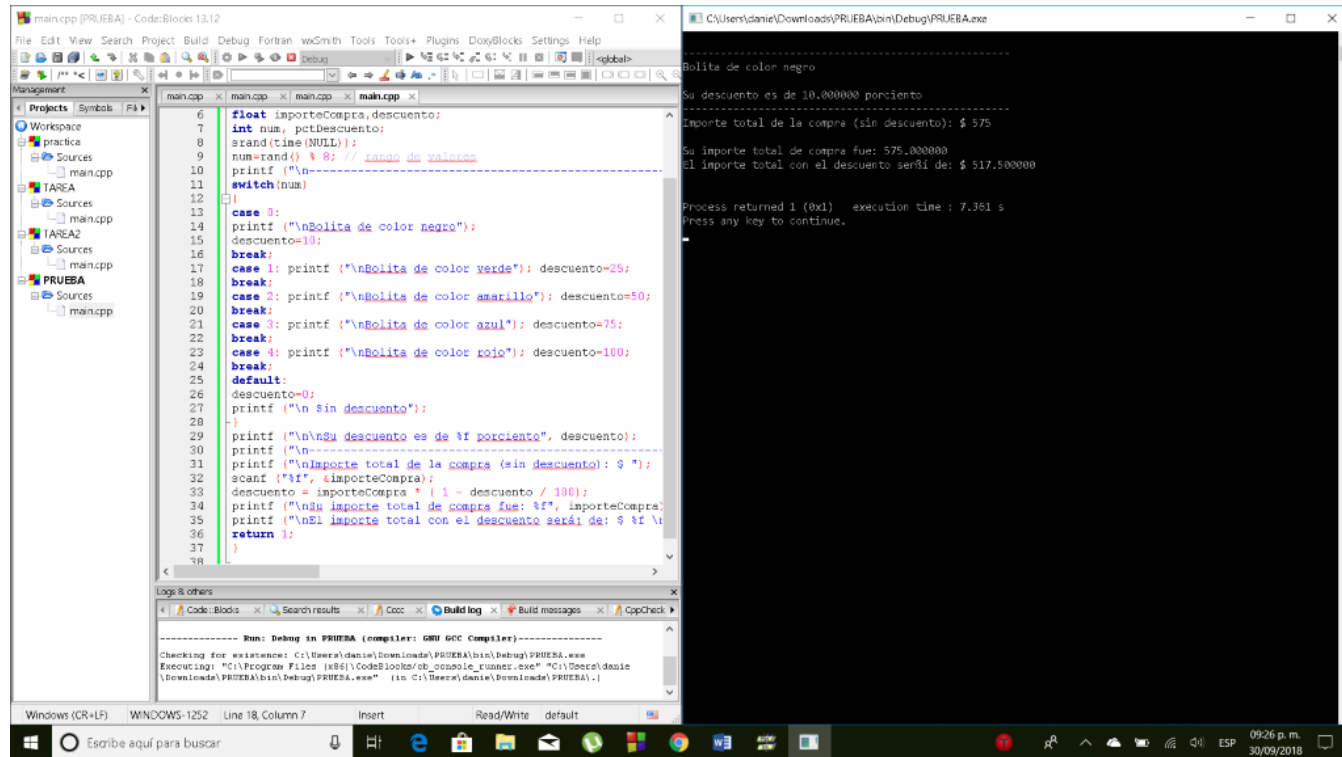
Escribe aquí para buscar

09:00 p. m.
30/09/2018

```
C:\Users\danie\Downloads\TAREA2\bin\Debug\TAREA2.exe
Dame un numero entero a sumar20
Dame un numero entero a sumar15
Dame un numero entero a sumar60
Dame un numero entero a sumar10
Dame un numero entero a sumar0
Dame un numero entero a sumar1
Dame un numero entero a sumar80
Dame un numero entero a sumar100
La suma sin sobrepasar el 200 fue: 186
La cantidad de numeros sumados fue: 7
El promedio fue: 26.000000
Process returned 0 (0x0)   execution time : 12.124 s
Press any key to continue.
```

```
C:\Users\danie\Downloads\TAREA2\bin\Debug\TAREA2.exe
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
Dame un numero entero a sumari0
La suma sin sobrepasar el 200 fue: 200
La cantidad de numeros sumados fue: 20
El promedio fue: 10.000000
Process returned 0 (0x0)   execution time : 15.233 s
Press any key to continue.
```


PROGRAMA 4



PROGRAMAS EN LA TERMINAL

```
Bosnia18:~ fp1121alu06$ vi TAR.c
Bosnia18:~ fp1121alu06$ gcc TAR.c -o TAR.out
Bosnia18:~ fp1121alu06$ ./TAR.out
La suma de los primeros 20 pares es: 420
La cantidad total de numeros sumados ha sido: 200Bosnia18:~ fp1121alu06$ vi TAR2.c
Bosnia18:~ fp1121alu06$ gcc TAR2.c -o TAR2.out
Bosnia18:~ fp1121alu06$ ./TAR2.out
Dame un numero entero a sumar150
Dame un numero entero a sumar130
La suma sin sobrepasar el 200 fue: 150
La cantidad de numeros sumados fue: 1
El promedio fue: 150.000000Bosnia18:~ fp1121alu06$ vi TAR3.c
Bosnia18:~ fp1121alu06$ gcc TAR3.c -o TAR3.out
Bosnia18:~ fp1121alu06$ ./TAR3.out
1
232
34543
4567654
567890765
Bosnia18:~ fp1121alu06$
```

Conclusiones: La práctica, me adentro más a todo lo que engloba el lenguaje C (editores) y puede aprender de manera mas significativa, que función realiza cada elemento.

La practica tuvo un ritmo muy rápido, pero los ejercicios eran muy sencillos de seguir y de realizar, por lo que uno iba aprendiendo, conforme lo iba haciendo.

Los ejercicios de tarea tuvieron un nivel moderado, con lo que pude hacer uso de las diversas estructuras vistas, y comencé a programar de mejor manera y recordando toda la estructura necesaria.

Bibliografía

Dr. Pedro Alberto Enríquez Palma. Editor VI. Consulta: septiembre de 2016. Disponible en: <http://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

Francisconi Hugo Adrian. Nano. Consulta: septiembre de 2016. Disponible en: <http://francisconi.org/linux/comandos/nano>

G2 Crowd. ATOM vs. Notepad++. Consulta: septiembre de 2016. Disponible en: <https://www.g2crowd.com/compare/atom-vs-notepad>

Gerald Pfeifer (GCC team). GCC, the GNU Compiler Collection. Consulta: septiembre de 2015. Disponible en: <https://gcc.gnu.org>

MinGW.org. MinGW - Minimalist GNU for Windows. Consulta: septiembre de 2015. Disponible en: <http://www.mingw.org>

White-Hat Hacking. Uso de gcc bajo Linux. Consulta: septiembre de 2016. Disponible en: <https://whitehathacking.wordpress.com/2010/10/31/uso-de-gcc-bajo-linux/>

Willus.org. Win32/64 C/C++ Compilers Page. Consulta: septiembre de 2016. Disponible en: <http://www.willus.com/ccomp.shtml>

Fabrice Bellard. Tiny C Compiler. Consulta: septiembre de 2015. Disponible en: <http://bellard.org/tcc/>