



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Karina García Morales

Asignatura: Fundamentos de la Programación

Grupo: 20

No. de práctica(s): 06

Integrante(s): Martínez Ordoñez Diego Tonatiah

No. de lista o brigada: 30

Semestre: 2023-1

Fecha de entrega: 08 / noviembre / 2022

Observaciones:

CALIFICACIÓN: _____

Práctica 06: Entorno y fundamentos del lenguaje C

- Objetivo:

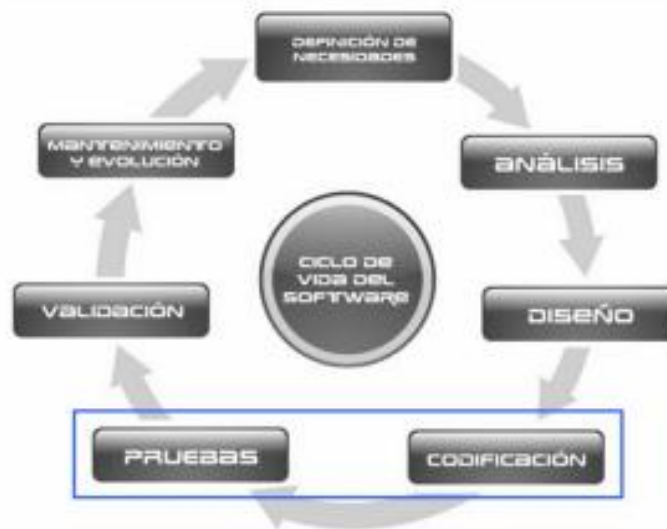
El alumno elaborará programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

- Desarrollo:

Una vez que un problema dado ha sido analizado (se identifican los datos de entrada y la salida deseada), que se ha diseñado un algoritmo que lo resuelva de manera eficiente (procesamiento de datos), y que se ha representado el algoritmo de manera gráfica o escrita (diagrama de flujo o pseudocódigo) se puede proceder a la etapa de codificación.

La codificación se puede realizar en cualquier lenguaje de programación estructurada. En este curso se aprenderá el uso del lenguaje de programación C.

Dentro del ciclo de vida del software, la implementación de un algoritmo se encuentra en la etapa de codificación del problema.



- Entorno de C:

El lenguaje C es uno de los mayores lenguajes usados para poder expresar instrucciones que la computadora podrá realizar, teniendo una relevancia muy importante por las funciones que te puede aportar.

Teniendo un punto fuerte, este es que tiene un gran potencial por la manera en que te puede ayudar a combinar características de lenguajes de alto nivel, con unos de bajo. Teniendo como impacto la creación de variantes que permiten programar miles de dispositivos electrónicos en el mundo, con los compiladores que le corresponden a cada uno.

Para poder elaborar un Programa en C, es sumamente importante recordar es que tienes que describir cada una de las instrucciones, bajo las condiciones que tiene este tipo de lenguaje para poder ser procesado correctamente en un compilador.

Tiene que estar bien escrito de acuerdo a lo que quieras obtener, ya que, al introducir las instrucciones tienes que tener cuidado para poder obtener como salida el programa ejecutable.

- Editores:

Este tipo de programa debe ser escrito en un editor de textos para que después pueda generar un programa ejecutable después para pasar a ser compilado.

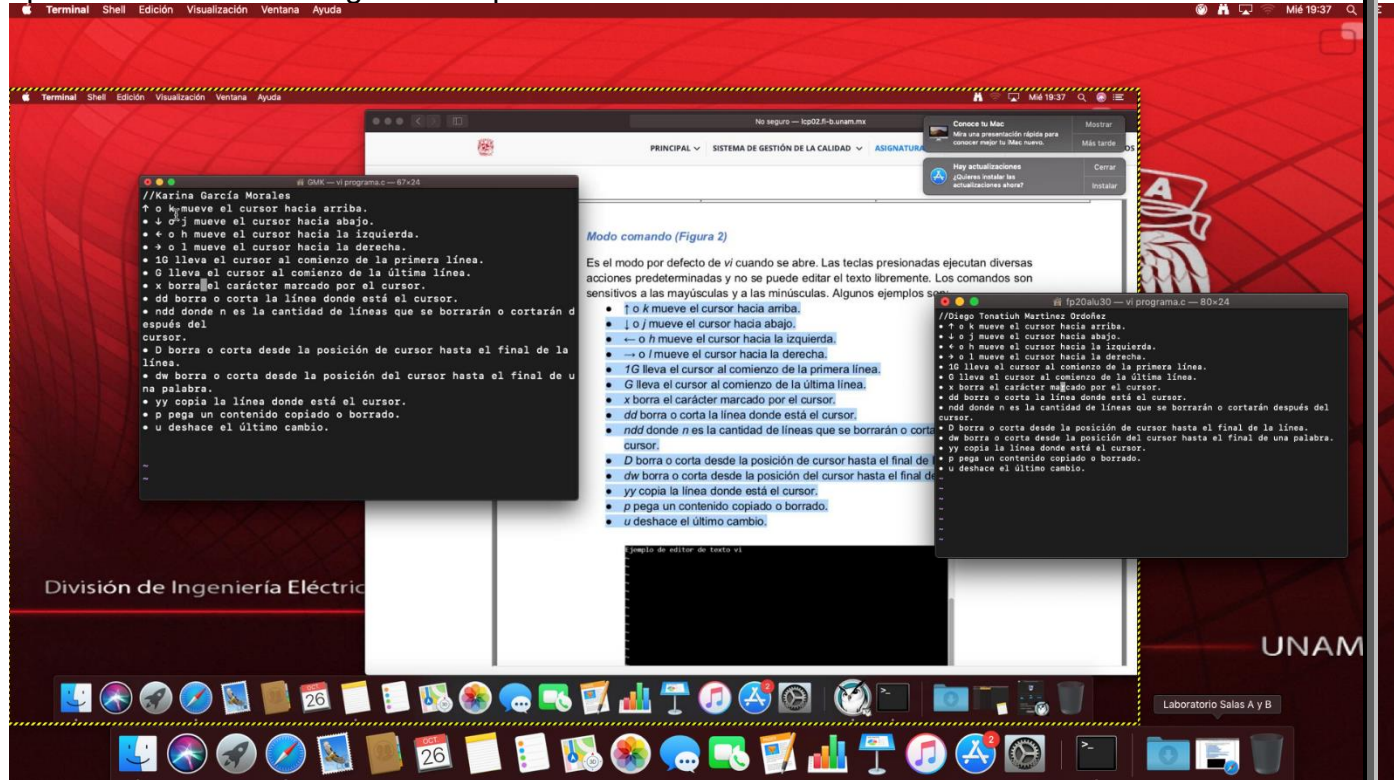
De la mano van el editor de textos y compilador con el sistema operativo y si es que esta posee o no la interfaz gráfica, para poder saber si es que se podrá compilar por ser lenguaje

C.

Los editores pueden tener diferentes funciones como guardar una configuración, tener escrito un programa, etc. También te permite dar formato al texto y poder editarlo. Teniendo como resultado el texto simple que pudimos realizar o algo más complejo.

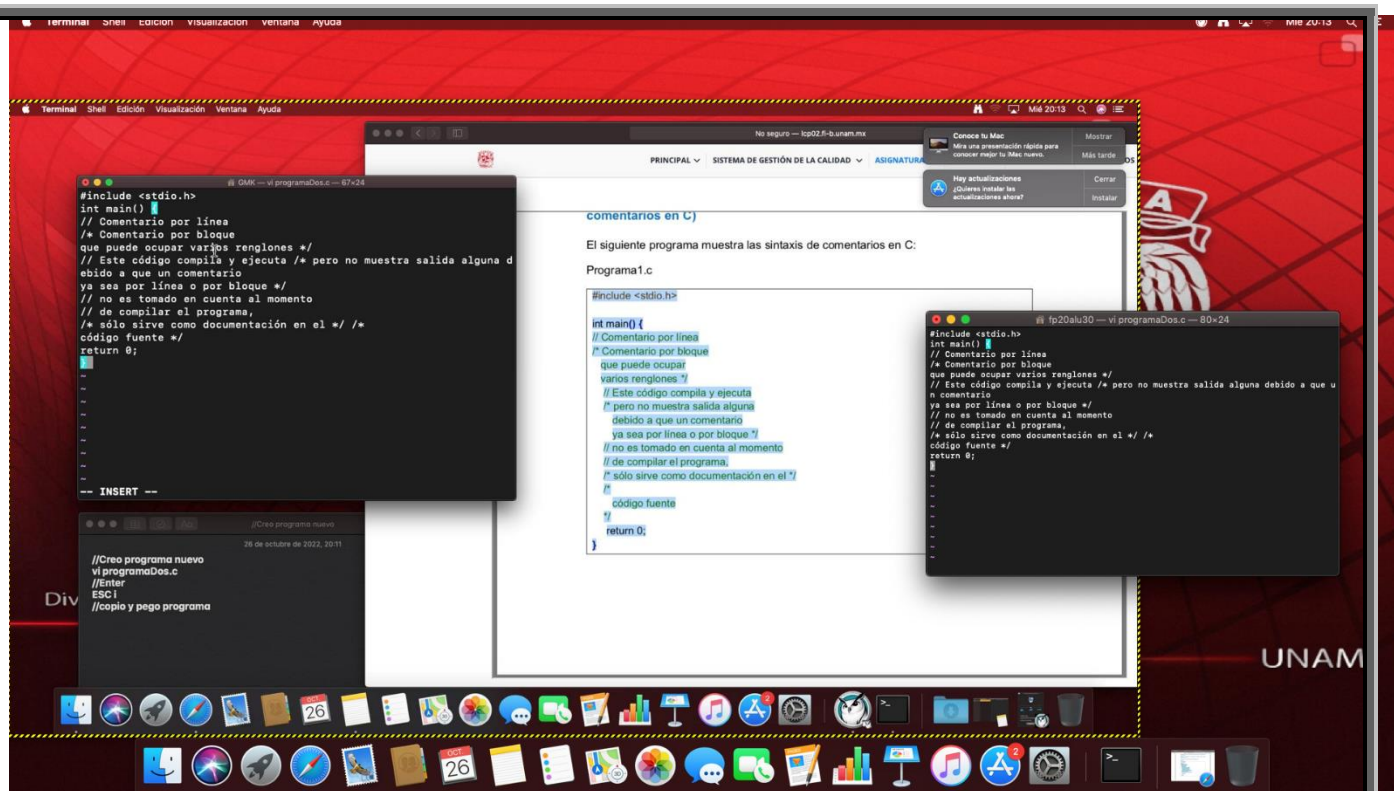
- Editor Visual Interface de GNU/Linux (vi).

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.



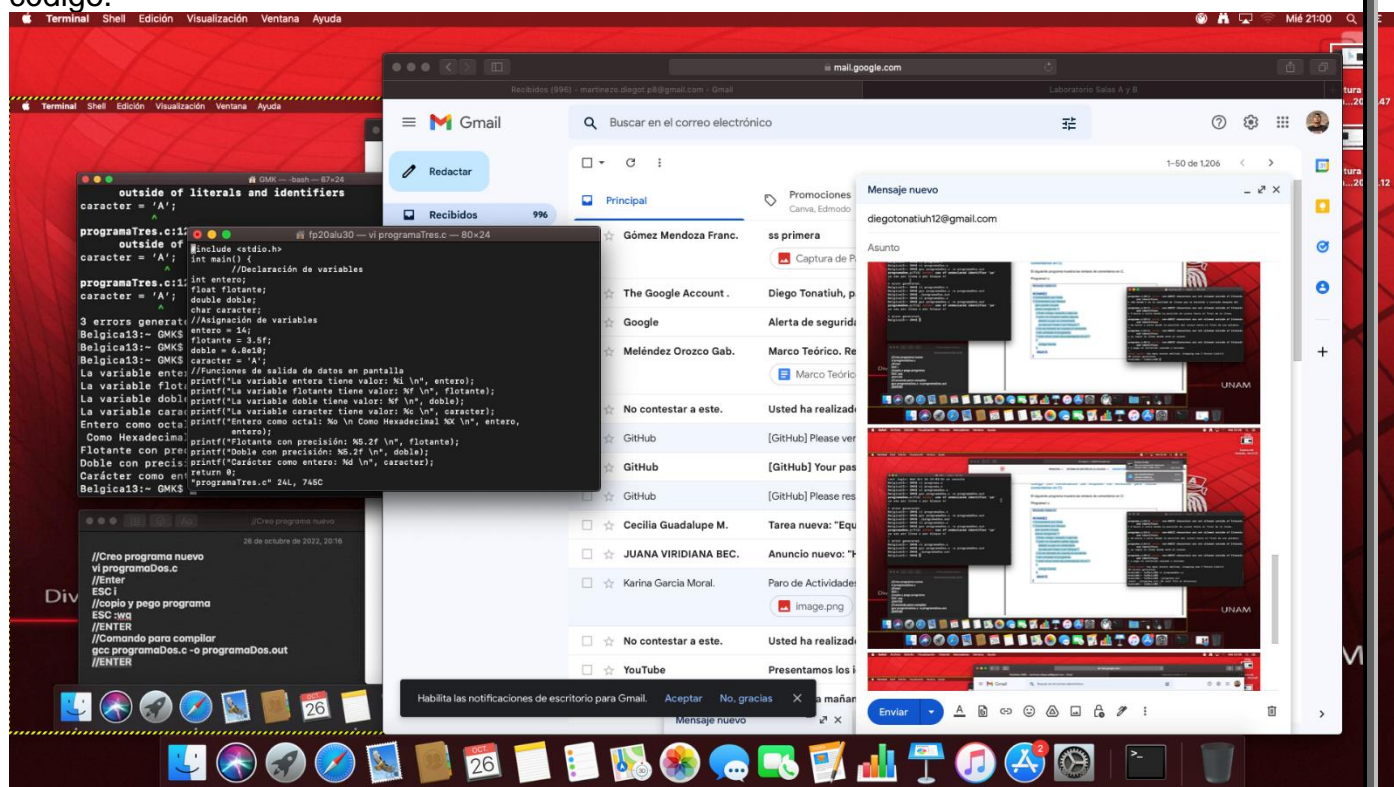
- Comentarios.

Es una buena práctica en cualquier lenguaje de programación realizar comentarios para documentar el programa. En C existen dos tipos de comentarios: el comentario por línea y el comentario por bloque.

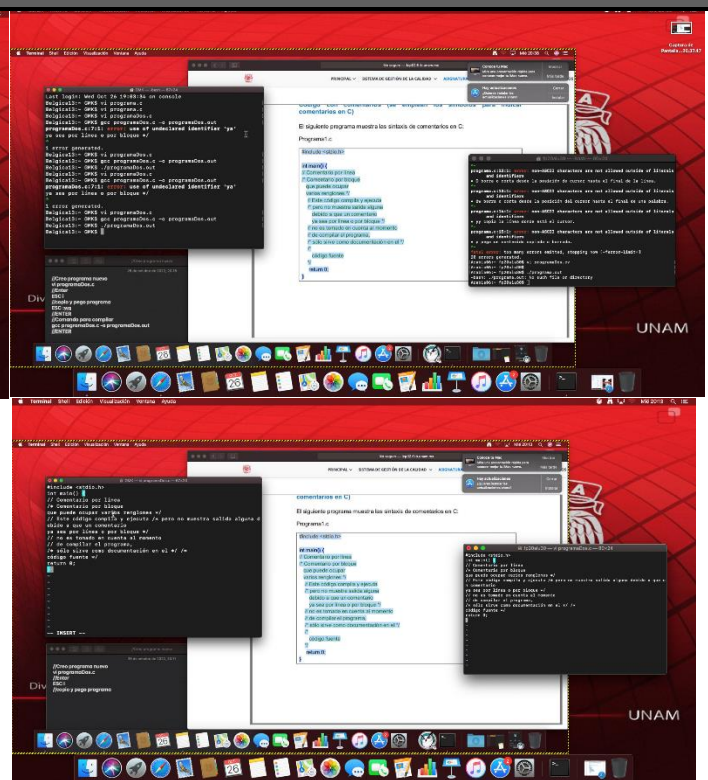
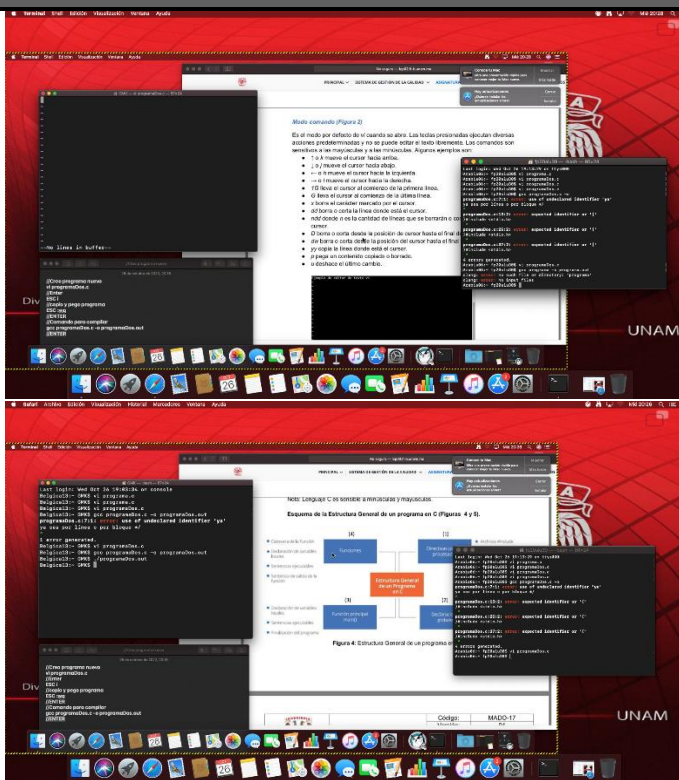


- Código declaración de variables (se declaran variables y se asignan datos):

Aquí ya tenemos un ejemplo en el que ya estamos declarando nuestras variables y al mismo tiempo le estamos asignando valores para poder realizar la ejecución y compilar nuestro código.



- Aquí adjunto las demás capturas que tome en la práctica:



- Tarea:

1. Investigar cual es el dato que se encuentra por default en en lenguaje C(signed o unsigned):

Declaración: switch (expresión) { case n1:sentencia1: break; default: sentenciaN; }

2. Indicar que sucede cuando en una variable tipo carácter se emplea el formato %d, %i, %o, %x

Se emplea para declarar variables y de esta manera para saber que tipo de variable sea, por ejemplo: ENTERO, que se usarían %d y %i, para poder insertar datos con estas características.

3. Mencionar las características con las que debe crearse una variable.

Debemos darle un nombre único para poder identificarla y el tipo de variable que será, dándole un valor para poder asignarla e identificarla en nuestro programa.

4. ¿Cuál es la diferencia entre variable estática y constante?

La variable estática se le asigna un valor y este lo va mantener fijo sin importar lo que pase; y una variable constante será aquella que puede tener una cantidad susceptible de diferentes valores.

5. Menciona en que momento empleas los dos tipos de diferentes (< > !=).

Los empleas cuando algún valor es mayor que otro o es menor, de igual manera se pueden utilizar para saber que son distintos o son iguales; o en otro caso para saber que no son iguales o este dato es diferente al que deseamos.

6. Crea un programa en el que declares 4 variables haciendo uso de las reglas signed/unsigned, las cuatro variables deben ser solicitadas al usuario(se emplea scanf) y deben mostrarse en pantalla (emplear printf).

```

using namespace std;

int main()
{
    signed int numero1, numero2;
    unsigned int numero3, numero4;

    printf("intriduca el 1er numero");
    scanf("%d", &numero1);

    printf("intriduca el 2do numero");
    scanf("%d", &numero2);

    printf("intrudusca el 3er numero");
    scanf("%d", &numero3);

    printf("introduzca el 4to numero");
    scanf("%d", &numero4);

    printf("su 1er numero es: %d", numero1);
    printf("su 2do numero es: %d", numero2);
    printf("su 3er numero es: %d", numero3);
    printf("su 4to numero es: %d", numero4);

    return 0;
}

```

7. Crea un programa que le solicite su edad al usuario, leer el datos(emplear scanf) y mostrarlo en pantalla.

```

Online C++ Compiler.
Code, Compile, Run and Debug C++ program
Write your code in this editor and press "Run" button

*****

#include <iostream>

using namespace std;

int main()
{
    int edad;
    printf("introduzca su edad:");
    scanf("%d", &edad);

    printf("tu tienes %d años", edad);

    return 0;
}

```

8. Revisar y colocar cuando se emplea MOD y cuando se emplea % (pseudocódigo y codificación) agrega un ejemplo de su uso.

MOD se usa para probar si dos números son divisibles entre ellos de manera exacta o aislar el resto de un calculo de división. E

Ejemplo: 7 MOD 3 devolverá 1, ya que al realizar la división entera de 7 entre 3 se obtiene 2 como conciente y 1 como resto.

?: retorna el remanente de la división del operando izquierdo por el operando derecho.

Ejemplo: $5\%2=1$

9. Comparación entre Editor de Texto y Procesador de Texto(Realizar una tabla comparativa).

Editor de texto	Procesador de textos.
Son programas que te permiten crear y modificar texto sin formato, en donde no se pueden agregar imágenes.	Permite creación, edición, modificación y procesamiento de documentos de texto con formato, donde puedes agregar imágenes.

10. Indica los comandos utilizados para compilar y para ejecutar un programa en iOS o Linux.

gcc programa.c -o y ./programa

11. Compilación y prueba del programa antes mostrado en DEV C++ u otro IDE en sistema operativo Windows.

No es posible ya que son distintos.

12. Genera un programa que solicite dos variables enteras al usuario y realice las 4 operaciones básicas, compila y ejecuta el programa utilizando terminal y los comandos indicados para cada instrucción.

- Conclusiones:

Puedo concluir que estamos aprendiendo nuevas formas de poder adaptar datos a nuevas circunstancias para poder realizar trabajos de todo aspecto y que podrían presentarse a lo largo de nuestra vida, como lo puede ser en el ámbito social, laboral, académico, etc. Estamos diferenciando que existen diferentes programas que comparten similitudes, pero no son para desarrollar las mismas cosas, como lo fue el editor y el procesador de textos, se pueden llegar complementar entre ambos pero no son iguales. Son herramientas que nos apoyan en nuestra estancia académica. Esta vez ya estamos aprendiendo a desarrollar cualquier tipo de datos y el como puede ser interpretada.

- Referencias:

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.