



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

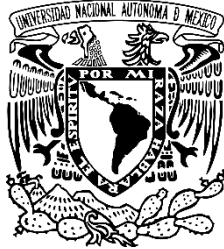
FACULTAD DE  
INGENIERÍA

MI. Marco Antonio Martínez  
Quintana

Estructura de Datos y  
Algoritmos

Asignatura:

Estructura de Datos y Algoritmos 1



Actividad Miércoles

#1 Repaso: Fundamentos de programación

Alumna

Citlali Cuahtepitzi Cuatlapantzi



Fecha

(25/febrero/2021)



## Repaso: Fundamentos de la programación

En la materia de Fundamentos de programación, como su nombre indica, fue una introducción encaminada a dar a los alumnos bases sobre lo que la tecnología en este ámbito es capaz de hacer, problemas, optimización, uso de herramientas etc., que se fueron desarrollando a lo largo la clase teórica en conjunto con el laboratorio y su respectivas prácticas. Las prácticas fueron sustanciales en este periodo pues los ejercicios planteados reunían un tema completo con sus variantes posibles, a grandes rasgos se describirá en forma de resumen lo realizado en cada práctica con un total de 13 agregando comentarios.

### Práctica 1

La práctica uno tuvo como objetivo adentrarnos en el uso de herramientas de software disponibles en internet que nos permiten a nosotros como alumnos realizar actividades académicas de una manera óptima, con mayor facilidad, calidad y organización. Entre las actividades realizadas encontramos el almacenamiento en línea y búsquedas de información avanzadas; el almacenamiento en línea sin duda tiene gran funcionalidad al hacer trabajos de cualquier tipo en un medio electrónico, pues permite gran disponibilidad en acceder a un documento en cualquier sitio sin necesidad del uso de memoria de nuestro dispositivo e incluso que dicho documento sea compartido a diversas cuentas simultáneamente haciendo que trabajar en equipo y a distancia sea una tarea más fácil o menos tediosa; las búsquedas avanzadas fueron un extra ventajoso a la hora de realizar una investigación haciendo de este un proceso más fácil, rápido y de requerirlo, específico. La práctica me permitió encontrar nuevos software que se adaptan a mí, considerar los distintos aspectos que nos ofrecen, evaluando sus ventajas y desventajas, e incluso utilizar distintos de estos.

### Práctica 2

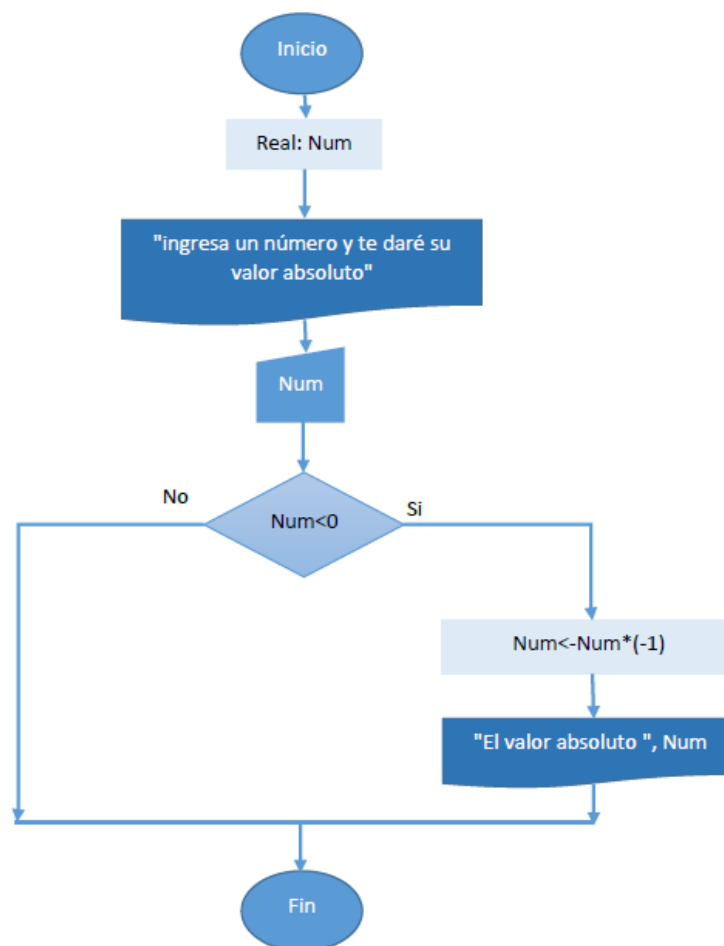
Fue planteada la importancia de un sistema operativo de una computadora, señalando este como el conjunto de órdenes y programas que controlan los procesos básicos de una computadora y permiten el funcionamiento de otros programas entre sus principales funciones. Se presentó un sistema operativo, GNU/Linux que se encuentra en línea, y con ello a partir de ejercicios aprendimos a utilizar algunos comandos básicos que nos permitieran escribir, movernos, guardar archivos y salir del editor. Nos fue introducido un nuevo concepto y presentado de manera práctica para entender mejor a que se refiere.

### Práctica 3

Se introdujo el concepto algoritmo como un método para resolver un problema mediante una serie de pasos definidos, precisos y finitos que en su forma más general está constituido por tres partes, entrada, proceso y salida. Un algoritmo puede ser expresado por pseudocódigo y/o diagramas de flujo (DFD) por tanto elaboramos algoritmos correspondientes a dos Origami, por parte de los diagramas de flujo aprendimos los símbolos correspondientes a cada proceso que se pueden exhibir, como se usan y representan. Encontré importante la necesidad de realizar un algoritmo correctamente puesto al querer solucionar un problema, un error en este, supondrá siempre un error en nuestro resultado final, pero también se aprecia que los problemas siempre tienen distintas soluciones, por lo que algoritmos diferentes o con variaciones pueden llegar a un mismo resultado siempre y cuando esté hecho correctamente, ese es el fin.

### Práctica 4

A lo largo de esta práctica elaboramos diagramas de flujo que daban solución a distintos problemas utilizando los conocimientos previos de lo visto en la práctica tres manejando la estructura de control condicional y de control iterativa. Para reforzar las funciones que tienen los símbolos de DFD fue de gran utilidad pues practicando se logró adquirir cierta destreza en su elaboración.



## Práctica 5

Con ayuda del programa PSEINT comenzamos a elaborar pseudocódigos que representaran soluciones algorítmicas con las mismas condiciones de la práctica anterior, usando estructuras de control de flujo condicional e iterativo, con ello empezamos a aprender a manejar un programa y las herramientas que nos puede ofrecer.

```
1  Algoritmo Valor_Absoluto
2      //este pseudocódigo devuelve el valor absoluto de un número
3      //Entradas:Un número
4      //Salidas:Valor absoluto
5      //Restricciones:Ninguna
6      Definir Num Como Real
7      Escribir "ingresa un numero y te dare su valor absoluto"
8      Leer Num
9      Si Num<0 Entonces
10         Num←Num*(-1)
11     Fin Si
12     Escribir "El valor absoluto ",Num
13 FinAlgoritmo
14 |
```

## Práctica 6

Después de usar pseint aprovechamos otro editor presente en el laboratorio, DevC++ y retomando GNU/Linux, también de Gcc, que lo encontramos en la página <https://bellard.org/jslinux/> y continuamos elaborando programas, con esto nos pudimos percatar de la existencia de diversos editores disponibles para la elaboración de cierto código, las diferencias o semejanzas que existen entre ellos a la hora de compilar y ejecutar. Al final se entiende que se pueden ocupar distintos editores e incluso entre ellos algunos que estén en línea, con diferentes características, más todos tienen el mismo fin, resolver un problema, y no está de más saber aprovechar y manejar aquello que nos esté disponible.

## Práctica 7

Elaboramos programas en lenguaje C utilizando instrucciones de control de tipo secuencia, realizando 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. Alternamos el uso de DevC++ y Gcc en la elaboración de estos ejercicios haciendo un aporte extra a la práctica anterior.

## Práctica 8

Seguimos elaborando programas en lenguaje C , esta vez avanzando con las diferentes estructuras de selección que existen, if, if - else, switch y ternaria (o condicional) para dar solución a problemas básicos indicando las estradas, salidas y restricciones, haciendo uso nuevamente de DevC++ y Gcc alternadamente en la realización de todos los ejercicios.

```
/*Realiza diferentes tipos de conversiones de monedas*/
//Entradas:cantidad inicial de Dolar,Pesos,Euros
//Salidas:conversion de Dolar,Pesos,Euros
//Restricciones: Ninguna
#include <stdio.h>
#define p printf
#define s scanf
int main()
{
    float dolar1,peso1,euro1, dolar2,peso2,euro2;
    int opc;
    p("\n\n\tBienvenido");
    p("\n\n\t(1)DOLAR A PESO\n\t(2)PESO A DOLAR\n\t(3)PESO A EURO\n\t(4)EURO A PESO ");
    p("\n\n\tElija la operacion del menu segun el numero que corresponda");
    s("%d",&opc);
    switch(opc)
    {
        case 1:
            p("\n\t\tConversion DOLAR A PESO");
            p("\n\t\tIngrese la cantidad a convertir" );
            s("%f",&dolar1);
            peso2=dolar1*20.05;
            p("\n\t\t%f dolares son igual a %f pesos ", dolar1,peso2 );
            break;
        case 2:
            p("\n\t\tConversion PESO A DOLAR");
            p("\n\t\tIngrese la cantidad a convertir" );
            s("%f",&peso1);
            dolar2=peso1/20.05;
            p("\n\t\t%f pesos son igual a %f dolares ", peso1,dolar2 );
            break;
        case 3:
            p("\n\t\tConversion PESO A EURO");
            p("\n\t\tIngrese la cantidad a convertir" );
            s("%f",&peso1);
            euro2=peso1/24.16;
            p("\n\t\t%f pesos son igual a %f euros ", peso1, euro2 );
            break;
        case 4:
            p("\n\t\tConversion EURO A PESO");
            p("\n\t\tIngrese la cantidad a convertir" );
            s("%f",&euro1);
            peso2=euro1*24.16;
            p("\n\t\t%f euros son igual a %f pesos ", euro1,peso2 );
            break;
        default:
            p("\n\n\t El numero no corresponde al menu");
    }
    getch();
}
```

## Práctica 9

Fue un seguimiento pues continuamos con estructuras de repetición haciendo ejercicios levemente más complejos, es esta ocasión sin hacer uso ya de Gcc.

```
//Cajero automatico
#include<stdio.h>
#include<math.h>
#define p printf
#define s scanf
#include<conio.h>
int main ()
{
    int opc;
    float saldo, retiro, deposito;
    p("\n\n\t\t\tCAJERO AUTOMATICO");
    p("\n\n\tIngrese el saldo inicial");
    s("%f",&saldo);
    do
    {
        p("\n\n\t\t (1)Consultar saldo");
        p("\n\n\t\t (2)Retirar efectivo");
        p("\n\n\t\t (3)Realizar deposito");
        p("\n\n\t\t (4)SALIR");
        s("%d",&opc);
        switch (opc)
        {
            case 1:
                p("\n\n\t\tSu saldo es de: %f",saldo);
                break;
            case 2:
                p("\n\n\t\tIngrese monto a retirar");
                s("%f",&retiro);
                if(retiro<saldo)
                {
                    saldo=saldo-retiro;
                    p("\n\n\t\tSu nuevo saldo es de: %f",saldo);
                }
                else
                {
                    p("\n\n\t\tNo cuenta con suficiente saldo");
                }
                break;
            case 3:
                p("\n\n\t\tIngrese monto a depositar");
                s("%f",&deposito);
                saldo=saldo+deposito;
                p("\n\n\t\tSu nuevo saldo es de: %f",saldo);
                break;
            case 4:
                p("\n\n\t\tGRACIAS.\n\n\t\tVuelva pronto.");
                break;
        }
    }
    while (opc!=4);
}
```

## Práctica 10

Al realizar programas en DevC++ siempre nos encontramos con diversos problemas para que nuestro código lograra compilarse o ejecutarse, arreglar estos detalles consistía en un análisis tardado por nuestra parte, fue en esta práctica donde aprendimos sobre depuración de programas y las técnicas básicas para hacerlo en programas en c, que nos permitía revisar de manera precisa el flujo de la ejecución y el valor de las variables, permitiéndonos encontrar los errores que pudimos cometer de manera directa. A pesar de no dominar bien la depuración sin duda es una herramienta que permite localizar los errores, sobre todo en códigos largos donde analizar todo el código por uno mismo requiere de mucho tiempo.

## Práctica 11-12-13

Iniciamos un nuevo tema en cuanto a la programación en c, “Arreglos unidimensionales y multidimensionales”, se comenzaron a elaborar códigos sencillos para capturar la finalidad que tenían, agrupar datos de un mismo tipo de distintas formas que nos permitían resolver otro tipo de problemas que con las anteriores estructuras se habrían hecho complicados y poco óptimos, fuimos avanzando con problemas más complejos, en la práctica doce utilizamos funciones para elaborar códigos que representaba distintas ventajas al separar operaciones de código, dando una presentación limpia y entendible a cualquier usuario, finalmente en la práctica 13 con el tema de “Archivos” se entendió de las ventajas que se nos daba al utilizarla, pues nos permitía el almacenamiento de los datos que ingresáramos en la ejecución de nuestros programas de ser necesario, pues a veces es necesario seguir trabajando con esos datos para analizarlos después o realizar algún otro tipo de proceso, este fue el tema en que profundizamos menos, pues vimos conceptos básicos y realización de ejercicios básicos, aquí tengo que añadir que tanto este tema, como el de la práctica once y doce, no son mi fuerte pues considero que en ejercicios complejos presentaría dificultad para resolverlos.

Este fue mi curso de fundamentos de la programación, se aprecia que las prácticas van reuniendo los conocimientos de las que le hayan antecedido, lo mismo con el proyecto final, la materia a pesar de haber sido aplicada a la resolución de ejercicios quizás no cotidianos, puede aplicarse en los que suceden de la vida real añadiendo que fue un curso con herramientas básicas, que hay más, mejores y más profesionales, en general y a pesar de ciertas complicaciones, disfruté la materia junto con esta idea de que todos los problemas tienen solución, y si lo puedes imaginar, lo puedes programar.