



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE INGENIERIA



ASIGNATURA: ESTRUCTURA DE DATOS Y ALGORITMOS

ACTIVIDAD #1

NOMBRE DEL ALUMNO: RIVERA VARGAS DONOVAN JOSHEP

Repaso de lo que aprendí en la asignatura de Fundamentos de Programación.

FECHA: 25/02/2021



REPASO DE LO APRENDIDO EN MI CURSO DE FUNDAMENTOS DE PROGRAMACION

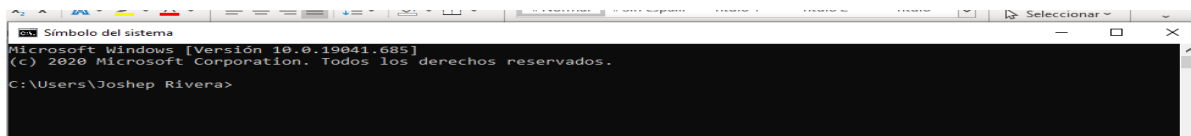
Cuando inicié el curso de fundamentos de programación no tenía una idea concreta de que se trataría, y aunque hay que destacar que no tengo aprendidos el 100 % del temario, mi aprendizaje de ese curso lo consideraría a un 50%, al realizar este repaso reforzare estos conocimientos y anexare algunos aprendizajes.

Lo primero que aprendí fue el código binario y como funcionaba, después fue descubrir y poder ocupar herramientas de software que hay en internet que me funcionarían para el manejo de repositorios de manera mas avanzada.

Anexo foto del software que aprendí a ocupar el cual es GITHUB



Posteriormente aprendí algo muy importante que para ser honesto no acabo de comprenderlo al 100%, es el "SISTEMA OPERATIVO DE UNA COMPUTADORA GNU/LINUX". En este tema aprendí a navegar en un Sistema operativo y el manejo de archivos dentro de él.



Después vi algo muy importante para la programación, el cual es el ciclo de vida del software, que es la base para poder dar solución a un problema por medio de los algoritmos que fue lo que aprendí posterior a esto.



Figura 1: Ciclo de vida del software.

Al elaborar un algoritmo tenía que pasar por un proceso el cual se llama prueba de escritorio el cual verifica que nuestro programa funcione y que los 3 módulos estén haciendo su función.

Resultados del análisis del problema

- Con qué datos se cuenta, cuáles son necesarios como valores de entrada, qué restricciones deben considerarse, cómo debe ser la salida para que el problema se resuelva.

Contrucción del algoritmo

- Se refiere a la descripción detallada de los pasos que deben seguirse para resolver el problema.

Verificación del algoritmo

- Consiste en el seguimiento del mismo, empleando datos que son representativos del problema que se desea resolver (esto se conoce como prueba de escritorio).

Módulo de Entrada

- Representa los datos que requieren para resolver el problema. Estos datos se pueden solicitar al usuario, leer de un archivo, consultar de una base de datos, etc.

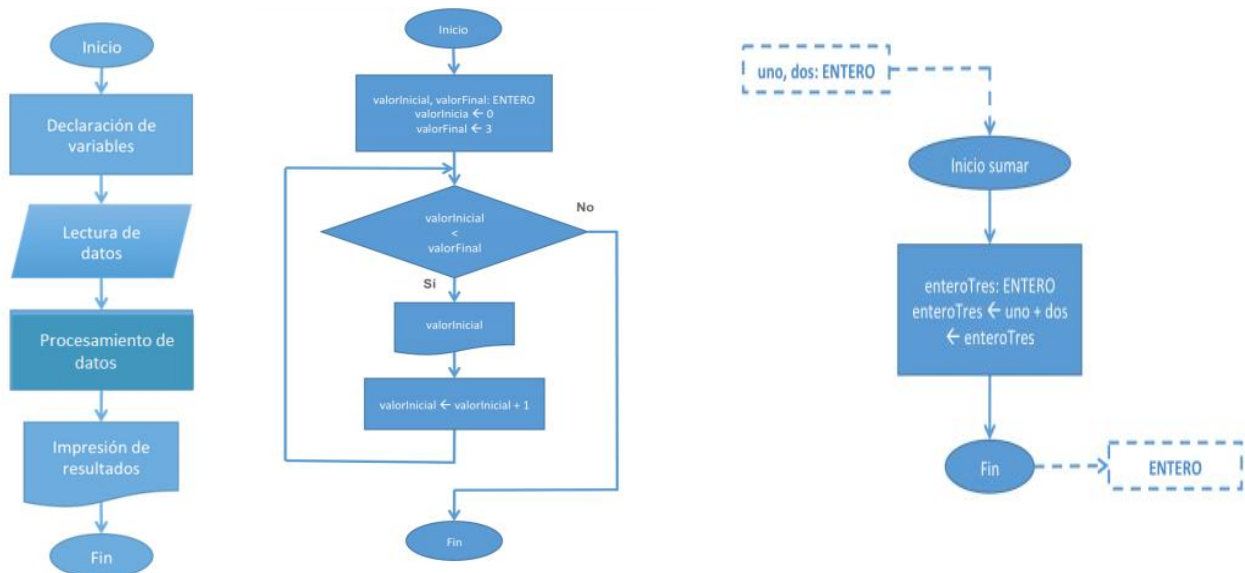
Módulo de Procesamiento

- Representa las operaciones necesarias para obtener un resultado a partir de los datos de entrada.

Módulo de Salida

- Permite mostrar los resultados obtenidos a partir del módulo de procesamiento de datos. Los resultados pueden mostrarse en diversos sitios: en la pantalla, en un archivo, en una base de datos, etc.

Pasamos a la primera construcción de un programa de una manera sencilla pero funcional lo cual es “DIAGRAMA DE FLUJO” este diagrama se basa en darle una función y especificación a una figura geométrica la cual va servir como guía para seccionar las partes de un programa



El diagrama de flujo es la base del pseudocódigo y este es la base de cualquier lenguaje de programación, ya que en este lenguaje creado por humanos para humanos te permite entender la sintaxis del software y los algoritmos, así mismo facilita el desarrollo de programas complejos para después pasarlos a un lenguaje formal

La sintaxis del pseudocódigo es algo compleja al principio ya que aprendes términos como los que son: declarar una variable, iniciar o terminar un programa, declarar un contador, un arreglo, operadores aritméticos, lectura, escritura, reglase de mayúsculas o minúsculas, sangría, notación de camello, etc.

También vimos lo que son **“ESTRUCTURAS DE FLUJO”** que son las que nos permiten la ejecución y repetición de una serie de instrucciones y existen 3 estructuras de control: secuencial, condicional y repetitivas o iterativas.

También vi **“Estructuras de control condicionales (o selectivas)”** estas nos permiten evaluar una premisa como verdadera o falsa bajo una condición.

“Estructuras de control iterativas o repetitivas” fue otra de las que vimos la cual nos permite ejecutar una serie de instrucciones mientras se cumpla la expresión lógica. “Existen dos tipos de expresiones cíclicas MIENTRAS y HACER- MIENTRAS.”

También aprendimos lo que son las **“FUNCIONES”** cuando tenemos un problema que dependa de una solución de software muy extensa, acudimos a este término el cual consiste en dividir el programa en funciones, las cuales son secciones del programa, la “función está constituida por un identificador de función (nombre), de cero a n parámetros de entrada y un valor de retorno”

Pasamos a lo último que vi en el semestre pero bastante extenso para ser honesto lo cual es el primer lenguaje de programación visto en clase el cual fue **“C”**, empezar con este lenguaje se me hace una manera muy compleja, necesaria y correcta de empezar a programar ya que es un lenguaje que se basa de varias reglas que están estructuradas por su sintaxis

Lo primero que vi fue los editores, compilación y ejecución. Aprendí que los editores necesitan ser escritos en un formato de texto para que se pueda ejecutar en un compilador y la computadora lo pueda ejecutar, existen varios como ejecutar desde nuestra terminal GNU/LINUX o directamente en un software especial para compilar y ejecutar.

Teniendo en cuenta esto pudimos empezar a estudiar la sintaxis del lenguaje C elaborando programas muy sencillos subiendo el nivel poco a poco.

Para cualquier código en C se tienen que agregar librerías y tener una estructura la cual es la siguiente (ya que C está estructurado por funciones).

```
#include <stdio.h>

main() {
    // Este código compila y ejecuta
    /* pero no muestra salida alguna
    debido a que un comentario
    ya sea por línea o por bloque */
    // no es tomado en cuenta al momento
    // de compilar el programa,
    /* sólo sirve como documentación en el */
    /* código fuente
    */
}
```

En esta imagen vemos una estructura y sintaxis del inicio de un programa con una librería y como único contenido comentarios

Tipo	Bits	Valor Mínimo	Valor Máximo	Tipo de dato	Especificador de formato
<i>float</i>	32	3.4 E-38	3.4 E38	<i>Entero</i>	%d, %i, %ld, %li, %o, %x
<i>double</i>	64	1.7 E-308	1.7 E308	<i>Flotante</i>	%f, %lf, %e, %g
<i>long double</i>	80	3.4 E-4932	3.4 E4932	<i>Carácter</i>	%c, %d, %i, %o, %x
				<i>Cadena de caracteres</i>	%s

En las imágenes de arriba podemos observar los tipos de variables(las cuales tienen que llevar un identificador) y datos que existen en lenguaje c , en el programa de abajo podemos ver un pequeño programa que ya incluye sintaxis de como imprimir una variable, operadores aritméticos, estructura de control selectiva, etc.

```
#include<stdio.h>

/* Este programa valida si el número a es mayor al número b.
*/

int main (){
    int a, b;
    a = 3;
    b = 2;

    if (a > b) {
        printf("\ta (%d) es mayor a b (%d).\n",a,b);
    }
    printf("\t\tEl programa sigue su flujo.\n");
    return 0;
}
```

Para finalizar este informe puedo mencionar que vimos muchas características de lenguaje c diría que el 65% o mas de todas las básicas e intermedias, no aplicamos todas pero vimos ejemplos de la mayoría, las cuales fueron; estructuras de repetición, depuración de programas, arreglos unidimensionales y multidimensionales, funciones en c, finalmente lectura y escritura de datos.

Esas fueron a grandes rasgos los conocimientos que obtuve en mi curso de fundamentos de programación, el cual reitero fue un curso muy bueno pero de todo lo mencionado en este escrito puedo decir que aprendí de un 50 a un 65 por ciento.