

**Universidad Nacional Autónoma de México**



**Facultad de Ingeniería**



**Estructura de Datos y Algoritmos I**

**Actividad 1 (miércoles)**

**Herrera Alcántara Emilio Ramsés**

**02/03/2021**

## Lo que aprendí en Fundamentos de Programación

En esta tarea voy a explicar un poco de lo que aprendí a lo largo del curso “Fundamentos de Programación”, lo explicaré en orden cronológico de cómo se presentaron los temas.



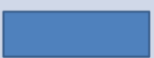
Lo primero con lo que trabajamos fue la introducción a los algoritmos, en el que se nos explicó que un algoritmo es un método para resolver problemas mediante una serie de pasos precisos, definidos y finitos. Además, es una serie de operaciones detalladas que se pueden formular de muchas formas siempre cuidando no generar ambigüedad.

Recuerdo que los algoritmos pueden ser de dos tipos: cualitativos (se describen los pasos usando palabras) y cuantitativos (se describen los pasos usando números).




Trabajamos un par de clases con algoritmos sumamente sencillos, tales como: preparar un agua de limón o una guía de camino a la escuela. Todo esto con el fin de comprender la utilidad de estos.

Un día, el profesor nos presentó los diagramas de flujo, herramienta que ya conocía con anterioridad, pero que nunca la había explotado como tal.

Recordé que los diagramas de flujo son los mismos algoritmos presentados de una manera gráfica por distintos símbolos, que son los siguientes:

Símbolo	Descripción
	Indica el inicio y el final del diagrama de flujo.
	Indica la entrada y salida de datos.
	Símbolo de proceso. Indica la asignación de un valor y/o la ejecución de una operación aritmética.

Símbolo	Descripción
	Símbolo de decisión. Indica la realización de una comparación de valores.
	Se utiliza para representar los llamados a los subprogramas o sobprocesos
	Conector dentro de página. Representa la continuidad del diagrama dentro de la misma página.

Símbolo	Descripción
	Conector fuera de página. Representa la continuidad del diagrama en otra página.
	Indica la salida de información por impresora
	Líneas de flujo o dirección. Indican la Secuencia en que se realizan las operaciones

También, el profesor nos brindó algunas recomendaciones para trabajar con los diagramas, esas recomendaciones las mostraré a continuación:

- Se deben usar solamente líneas de flujos horizontales y/o verticales.
- Se deben usar conectores solo cuando sea necesario.
- No deben quedar líneas de flujo sin conectar.
- Se deben trazar los símbolos de manera que se puedan leer de arriba hacia abajo y de izquierda a derecha.
- Todo texto ubicado dentro de un símbolo deberá ser escrito claramente.

Al principio, todas estas recomendaciones y la idea de trabajar con diagramas de flujo, me parecía poco útil, pero no fue hasta el proyecto final en dónde comprendí la gran utilidad es estos.

Nuestro primer acercamiento con la programación se presentó después de saber todo lo básico sobre algoritmos, la herramienta que utilizamos fue nada más y nada menos que PSEINT, un programa el cuál se enfoca en darnos herramientas parecidas al lenguaje "C" pero utilizando palabras en español y no poniendo tanta atención a signos de vital importancia para otros lenguajes.

Cómo primer ejercicio realizamos el famoso e irremplazable "hola mundo", ejercicio que nos daría una pequeña idea de cómo funciona PSEINT, ver el compilador, ejecutar el programa, entre otras cosas.

Seguimos practicando, haciendo pequeños programas con una estructura cualitativa en los cuales dábamos instrucciones al usuario sobre cómo hacer un agua de limón o una guía hacia la facultad, solo usando la herramienta de mostrar en pantalla.

Poco a poco estos programas se fueron complicando y empezamos a trabajar con las operaciones básicas, recuerdo el primer programa que hicimos el cuál trataba sobre sacar el área de un cuadrado usando las primeras operaciones.

Llegamos hasta el punto en el que conocimos a las palabras reservadas: Inicio, fin, leer, escribir, si, sino, fin si, para, fin, para, mientras que, fin mientras que, repita, hasta, regresar.

Inconscientemente, clase tras clase teníamos un mejor orden en nuestros pequeños programas y aumentábamos el uso de nuevas palabras. Poníamos condiciones o sacábamos promedios de n calificaciones.

Seguimos un par de semanas con PSEINT hasta conocer todas las bases y presentar una serie de 50 ejercicios con diagrama de flujo, posteriormente llegó el examen y hasta la siguiente clase, el profesor nos sorprendió con la noticia de que trabajaríamos con el lenguaje "C".

Empezamos con lo más básico de la teoría que fue el cómo comentar una línea usando la doble "//" o el cómo comentar varias líneas usando "/\*" para abrir y "\*/" para cerrar el comentario.

La siguiente clase comenzamos con la explicación de la estructura de los programas en C, que era bastante parecida a la de PSEINT, esta estructura es la siguiente:

```
/*inclusión de bibliotecas, definición de constantes y */
```

```
/*macros, definición de prototipos de funciones*/
```

```
main() {
```

```
/*cuerpo de la función main
```

```
/*En esta función escribiremos el código principal de nuestro
```

```
programa*/
```

```
}
```

```
/*Definición de otras funciones*/
```

Nuevamente trabajamos con las palabras reservadas pero ahora con sus verdaderos nombres y definiendo lo que hacía cada uno, funcionan con una gran similitud a los vistos en PSEINT por no decir que igual, algunos ejemplos de estas palabras son: const, double, float, int, short, struct, break, continue, else, for, long, switch, void, case, default, register, sizeof, char, do, if, return while, entre otros.

Repasamos la declaración de variables, aunque aquí nos encontramos con ciertas restricciones cómo son:

- Cada variable es de un tipo de datos determinado.
- Una variable puede almacenar diferentes valores en distintas partes del programa.
- Debe comenzar con una letra o el carácter guión bajo (\_).
- El resto puede contener letras, números o \_
- Ejemplos de variables válidas: numero, \_color, identificador\_1

Trabajamos con expresiones que representan una unidad de datos simple, tal como un número o carácter. También puede estar formados por identificadores y operadores:

- $a + b$
- $\text{num1} * \text{num2}$
- $A + (b - e) + 2 * x - (a + b + 3)$

Vi sentencias de asignación y la gran utilidad de definir un valor para una variable al momento de declararlo. Este es uno de los primeros ejercicios ya utilizando las asignaciones:

```
#include <stdio.h>
main()
{
    int a = 1;
    float b = 4.0;
    int c, d;
    char letra;
    c = 10;
    letra = 'a';
    d = a + c;
    printf("a = %d \n", a);
    printf("b = %f \n", b);
    printf("c = %d \n", c);
    printf("d = %d \n", d);
    printf("La letra es %c \n", letra);
}
```

Otros temas que vimos, pero no tuvieron mucha “importancia” a lo largo del curso fueron: “define, operadores aritméticos, conversión de tipos, jerarquía de los operadores, operadores de incremento y decremento, etc.”.

Uno de los temas que me parecieron más útiles fueron el uso de los operadores racionales y lógicos, ya que además de ser símbolos que ya los conocía desde hace mucho, fueron la puerta a programas más grandes y complicados.

Estudiamos durante un par de clases las sentencias de control que son de 3 tipos:

- Secuencia. Ejecución sucesiva de dos o más operaciones.
- Selección. Se realiza una u otra operación dependiendo de una condición.
- Iteración. Repetición de una operación mientras se cumpla una condición.

Retomamos el uso tanto del if como del switch, ya con ejemplos más completos y analizamos la estructura desde una perspectiva gráfica con los diagramas.

Finalmente cumplimos con otros temas más complejos del temario cómo fueron: Los apuntadores, las estructuras y el manejo de archivos, todo esto con el fin de tener las herramientas suficientes para realizar nuestro proyecto final.

El proyecto final fue un gran reto, sería por equipos y es el primer gran programa que realizamos.

Teníamos una rubrica con la cuál el profesor podía tener un mejor control y así asignarnos una calificación justa.

El proyecto consistía en hacer el videojuego Deal or No Deal en el lenguaje “C”, los principales problemas que tuvimos fue que no teníamos dominados los temas más recientes que vimos y eso era parte fundamental para presentar un buen proyecto.

Fueron un par de semanas solo para repasar manejo de archivos, apuntadores, cómo insertar música en el programa, registros, entre muchas otras cosas.

Decidimos entre el equipo dividir el programa en 4 partes y cada quién trabajaría en una de esas partes, para posteriormente unirlas y corregir errores. Las 4 partes fueron:

- Registro de usuarios, estadísticas y total de juegos
- Menú del juego y toda la parte gráfica
- Desarrollo del juego
- Diseño de maletines y colores para el juego.

Finalmente yo diseñé la parte de los registros, estadísticas y el total de juegos, fue un proyecto que me pareció sumamente interesante y un buen reto en algunos momentos.