



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

~ Facultad de Ingeniería ~

Estructura de Datos y Algoritmos I

Actividad Asíncrona 1 miércoles

“Redacción aprendizaje de Fundamentos de
Programación”

Nombre: Raudales Palma Leonardo de Jesús

Fecha: 24/02/2021



¿QUÉ APRENDÍ EN LA ASIGNATURA DE FUNDAMENTOS DE PROGRAMACIÓN?

Hablando sobre lo que aprendí en la asignatura que le antecede a esta creo que debo comenzar por la idea básica de qué era programar, pues yo al ser nuevo en el tema creía que lo único que se tenía que hacer era sentarse y empezar a crear códigos con ideas que fueran saliendo de tu mente para así poder conseguir la aplicación de conocimientos que esperas, pero estaba muy equivocado, recuerdo que desde las primeras clases se nos enseñó que programar es todo un proceso bien estructurado que tiene un inicio y un fin. Por ejemplo, aprendí la estructura a rasgos básicos de idear un proyecto de programación que iba desde pensar la simple idea, luego crear un algoritmo, este término ya lo conocía, pero creo que no sabía bien que significaba y solo lo había escuchado en distintas conversaciones de aplicaciones populares y sus algoritmos que a veces eran burlados por todo tipo de atentados virtuales o algo por el estilo. Pero en la clase de Fundamentos de Programación aprendí que es más bien como una receta o un tipo de instructivo y para mi sorpresa vi que no se aplicaba solo a la programación sino a distintas actividades más, como las instrucciones para hacer un simple dibujo o para decir a alguien como crear un objeto, recuerdo que una parte que se me complicó fue el tema de datos de entrada y datos de salida, donde con el ejemplo de una cajita a la cual le metías datos o materiales te devolvía un resultado, al principio no lo entendía muy bien pero conforme fue avanzando la clase creo que lo fui entendiendo mejor. También recuerdo mucho la primera práctica en la que teníamos que reproducir los comandos que se pueden utilizar en la búsqueda de Google para así conseguir búsquedas avanzadas, recuerdo comando como in title o las variantes de Google para cada distinta búsqueda, por ejemplo, Google scholar que tenía la función de brindar resultados relacionados a búsquedas académicas y proporcionaba artículos, libros y herramientas más enfocadas al conocimiento científico, o Google imágenes que eran un buscador especializado en solamente ese rubro, podías buscar cualquier tipo de palabra e inmediatamente te salía imágenes o páginas con contenido pictográfico para que pudieras conseguir toda clase de ilustraciones. Se me había pasado que también conocí más acerca de la historia del Computo ya que tuvimos que investigar sobre eso, desde los inicios que comenzó como una idea de generar operaciones y como las realizaba de manera muy básica con una capacidad limitada y conforme fue pasando el tiempo comenzó a avanzar más y más hasta que realizaba operaciones que no se comparaban al inicio, luego pues vino la implementación de tubos al vacío y comenzaron a nacer las computadoras potentes, también recuerdo que conocí la historia de Alan Turing quién fue el que desarrollo la computadora para poder descifrar el código enigma que era una pieza clave para vencer a los alemanes en la guerra, con esto pues comprendí un poco mejor la importancia que tiene el avance de la tecnología orientada a la programación y que si consigues la capacidad de programar puedes incluso crear aplicaciones, programas o dispositivos que sean no solo de utilidad si no una idea literalmente materializada y que funciona conforme a lo que tú quieras, es decir, tienes el poder de decirle a la computadora o al dispositivo que quieras que hacer, generas un código como si

te estuvieras comunicando con ella o más bien como si le dieras instrucciones de que hacer en distintas situaciones para que así genere un resultado y este resultado es la funcionalidad del proyecto. Siguiendo con el esquema de las prácticas lo siguiente que aprendí creo fue lo que es un diagrama de flujo, que este es la representación gráfica de un algoritmo, yo había visto anteriormente en muchas ocasiones que varias tareas a lo largo de mi trayecto académico requerían que se elaborarán en forma de diagrama de flujo, pero nunca entendí bien como era este tipo de diagrama, en la práctica venían cosas como el tipo de figura o conector que representaba cada alternativa al camino del diagrama, por ejemplo condiciones de si o no y el resultado de estas, también aprendí que se tiene que expresar un inicio y un fin dentro del diagrama para saber cuál es el principio de todo ese proceso y exactamente donde termina. Lo siguiente que vimos fue la otra cara de la moneda, la representación escrita de un algoritmo o lo que se llama el pseudocódigo, debo reconocer que este me costó entenderlo más que los temas anteriores, el pseudocódigo es como una vista de lo que después se convertirá en código y de hecho para eso sirve, como una premisa de lo que después ya será lo que vamos a pasar en código de Lenguaje C o de otro lenguaje de programación, en este se usan distintas “funciones” o palabras claves como por ejemplo para representar las escritura en pantalla usas **ESCRIBIR** mientras que si quieres reproducir algo en la pantalla usas **IMPRIMIR** y más o menos cosas por el estilo. Antes cuando mencioné lo de Lenguaje C recordé que también revisamos lo que es el sistema operativo Linux y el software libre en general, la diferencia entre un software que no es libre como es el caso de Windows en el que pues primero que nada no posees poder total sobre él, es decir, tienes limitaciones en su uso, no puedes modificar en su totalidad el sistema operativo, tiene un costo dado que cuenta con una licencia que debe ser pagada para su uso además de que varias de las aplicaciones o programas relacionadas con el sistema también pueden ser de costo, y sobretodo un punto importante es que no se permite la participación del usuario ni uso de herramientas que tienen que ver en cuanto a la programación como el código para el desarrollo y mejora del mismo sistema operativo, cosa que no ocurre con un SO de software libre, el cual es totalmente gratuito, posees control sobre él, vas a poder modificar o ajustar distintas cosas que en el sistema operativo de paga no podrías hacer y la participación de usuarios con la finalidad de que así el proyecto mejore para toda la comunidad en general, también se te brindan los códigos referentes que se usaron para así fomentar al desarrollo siendo utilizados por los usuarios.

Siguiendo con el programa de estudio me parece que, entonces ya conociendo algoritmo, diagrama de flujo y pseudocódigo ya empezamos a adentrarnos de lleno en lo básico de Lenguaje C, lo primero que hicimos fue instalar el compilador y configurarlo en caso de que no corriera a la primera, también me parece que en la segunda práctica usamos un servidor de la escuela que era de Linux y trabajamos un poco en él, vimos los comandos básicos como `list` y `cd`, no recuerdo si eran los mismos de la terminal pero si usábamos `cd` para movernos de una carpeta a otra, también existía una carpeta raíz a la cual podías llegar por el comando de 3 puntos o comandos por ejemplo `cp` que se utilizaba para copiar carpetas, en fin, una vez estuvo la instalación del entorno en C completa pues comenzamos a usar el procesador de texto que

ocupamos durante todo el semestre que fue NotePad++, la verdad creo que la mayoría de mis compañeros sabrían ya para ese momento que era un procesador de textos pero en mi caso fue algo nuevo, es una herramienta en la que básicamente escribes todo el código de lo que vas a programar, ya sea en Lenguaje C o cualquier otro lenguaje, también posee sangrías, colores que te ayudan a distinguir entre distintas acciones y el autocompletar del texto, que esto incluso también se maneja en la terminal de Windows y que tampoco conocía yo pero conforme pasó el tiempo me acostumbre a usar la tecla de tabulador para aprovechar el autocompletar texto de los distintos entornos que manejamos. Ahora bien, como ya había pasado en el caso del servidor de Linux en la Facultad, también aprendí a moverme por la terminal de Windows y algunos comandos funcionales como es el clear que sirve para limpiar la pantalla o list que te muestra una lista de todo lo que se encuentra en el directorio en el que estás, se me había olvidado comentar que al inicio igual vimos sobre lo que era un repositorio y directorios, las distintas plataformas que trabajan en almacenamiento y los tipos de servidores que existen, otra cosa importante de esta parte fue el registro o copia de seguridad, no recuerdo ahora bien como se llamaba pero era la capacidad de tener como un registro de todas las modificaciones que le has hecho a un elemento dentro del repositorio o al mismo. El primer ejercicio que hicimos en código fue un Hola Mundo por lo que creo que el primer comando que aprendí de forma sencilla fue el printf para imprimir en pantalla, obviamente con esto viene vinculado este uso de bibliotecas de Lenguaje C, entonces también aprendí el esquema básico del código con el `#include<stdio.h>` que igual no recuerdo explícitamente como es que se escribía, creo que era screen to display in and out o algo por el estilo, este es muy importante ya que es necesario para usar printf y scanf que como en el nombre dice pues son acciones de entrada y salida. Muy importante también durante las clases aprendí que el dispositivo de entrada estándar es el teclado y de lectura estándar o salida por decirlo así es la pantalla del computador. En el Hola Mundo también pusimos en práctica comandos distintivos del entorno de Lenguaje C que instalamos, estos son los básicos como gcc para poder revisar si se instaló el programa o gcc archivo.c -o archivo.exe que se utilizaba para compilar los archivos de texto .c que realizábamos en el NotePad++ para así convertirlos en ejecutables y poder ponerlos en marcha para realizar lo que quisiéramos. Luego revisamos más el tema de Impresión en Pantalla, no solo revisamos el uso de printf, también se vio como imprimir de manera correcta en pantalla con el uso de otros comandos como `\n` que servía para poder hacer un espacio entre renglones o `\t` que funcionaba como una tabulación para dejar un espacio entre palabras del código, si estamos hablando de impresión de pantalla también aprendí del uso de caracteres ASCII para que cuando se imprima el texto con diferentes acentos use el comando `%c` para reemplazar por el código del carácter que necesitaba e incluso servía mucho para colocar distintos símbolos que también se encuentran en ASCII, por supuesto de igual manera era nuevo el concepto de este último y pues entendí que eran los códigos de los diferentes caracteres que existen en la entrada estándar de datos que es el teclado. Una vez que se vio todo lo relacionado a impresión en pantalla pasamos a lectura de datos en el teclado, en este apartado aprendí lo que es scanf, como solicitar diferentes tipos de datos y aquí también aprendí los formatos de datos, los que más llegué a usar por ejemplo

fueron %d, %i que representaban enteros y en algunas ocasiones ocupé %f que si no mal recuerdo procedía de float que se utilizaba más para decimales y que cada formato tenía como un rango de números o cantidad de memoria que podía utilizar, algunas veces comprendí esto al realizar los código ya que en algunos trabajos busqué ocupar números con decimales aplicando el formato %d me marcaba un error o si al escanear un número más grande del rango que establecía el formato entonces igual marcaba error. Otra cosa que aprendí en el curso anterior fue por ejemplo el uso de int para definir enteros o solicitar variables con uso de char para variables de tipo carácter. Lo que son los operadores matemáticos dentro de la operación, varios ya conocidos con excepción del módulo que hasta entonces no sabía lo que era y como pedir dos números para realizar dicha operación, esto acompañado de algún texto que se podía imprimir en pantalla una vez que se obtuviera en el resultado e incluso a poner el número total en un resultado. Supe como programar una calculadora quizá muy básica pero que podía realizar operaciones con números enteros. Me parece que luego empezamos con las estructuras de control y las estructuras de repetición, las estructuras de control permitían justamente como su nombre indica el poder controlar el proceso de acción con distintas condiciones dadas, tal es el caso del if – else en el que se establecía dentro del código una condición, tenías dos opciones de manera simple sí o no, en dado caso que se cumpliera con la condición entonces se realizaba una acción explícita y en el caso de lo contrario pues se podía estipular una contraposición que era diferente a la condición, estas estructuras pues son útiles y vitales en todo proceso ya que van encaminando las acciones de todo el proceso. Las estructuras de repetición, por ejemplo, for, while o do while sirven para poder realizar un proceso durante i veces mientras una condición se cumpla, es decir que mientras condicionemos algo el programa va a seguir repitiendo esa condición el tiempo que esta sea verdadera, uno de los trabajos que más recuerdo es el menú, creo fueron dos tipos de menú, uno en el que se encontraba área y perímetro de una figura en específico y una vez que te daba los datos entonces volvía al menú de inicio, el otro programa tipo menú que se realizó era uno con opción de salida y se repetía constantemente. Otro aspecto que fuera de los códigos y comandos fue que constantemente tuve que hacer un seguimiento a mi proyecto diseñado para el semestre anterior, cosas como lo eran el tiempo de planeación, los costos que podría tener este proyecto a un largo plazo, los gastos que implicaría la realización del mismo y las distintas perspectivas que tenía sobre mi proyecto dentro de todo el rubro de la informática, esto creo que me ayudó mucho a considerar no solo la parte de conocimientos académicos sino un poco las perspectiva del mercado laboral y ver que no solo tienes que hacer que tu trabajo funcione y se sustente sino que debes pensar si realmente es rentable, más o menos como está cotizado tu esfuerzo en el interés del consumidor y me parece que son conceptos muy importantes para nosotros los estudiantes de nuevo ingreso que nos preparamos para algún día salir al campo laboral y saber cómo funciona este, entonces entre más rápido conozcamos me parece mejor. En la última parte del curso creo hacia los temas finales vimos lo que son arreglos que son estructuras de datos o más o menos entendí eso, es como apartar un lugar de memoria para algún tipo de dato, como reservar lugares, estos eran unidimensionales o bidimensionales dependiendo del tipo de arreglo y esto depende del número de columnas y filas con las que

contaba el arreglo, se usaban los símbolos [] para poner una cifra a y una b de la cantidad de espacios a reservar para datos, creo que igual se podía hacer con datos de tipo carácter así como nombres o datos por el estilo, los arreglos sirven para hacer un listado de distintos datos, muy comúnmente para elaborar inventarios que pues requieren del escaneo de datos y la agrupación de estos. Después de los arreglos me parece que vimos depuradores, de hecho, durante la realización de mi proyecto del curso el programa que usé para crear mi videojuego (que fue el proyecto que elegí) contaba con un depurador integrado que me ayudó a reconocer más esa herramienta para cuando la vimos en el manual de prácticas y realizamos la práctica correspondiente, en sí un depurador sirve para notar hasta los pequeños detalles de un código, se puede ir creando puntos de inflexión (no recuerdo bien si se llamaban así) que eran una especie de pautas donde le indicabas al depurador un rango de que líneas a que líneas revisar del código, en dado caso de existir un error el depurador te lo indica marcándose en rojo y te dice exactamente en cuál línea de código se encuentra el error, de cierto modo creo que también el curso de Fundamentos de Programación me enseñó por así decirlo a “depurar” por mí mismo, tuve muchos errores en mis código y en la terminal de Windows se muestra una especie de corrección de donde está dicho error así que recuerdo que para la mitad del semestre yo mismo identificaba que era lo que podía considerarse el posible error y lo corregía incluso antes de revisar donde estaba enmarcada esa alerta en el Windows. Por cierto, se me olvidó comentarlo, pero una de los conocimientos más importantes que obtuve y que no se me olvida es el uso del ampersand para siempre solicitar datos en el escaneo desde el teclado, muchísimas veces me llegué a equivocar en esa parte por lo que ya nunca se me olvida.

El último tema que recuerdo de la asignatura era el de funciones me parece, en el manual de prácticas se establecía que dentro del proceso de la codificación era mucho mejor el dividir distintas actividades de un proceso para así hacerlo por partes y este resultara más fácil y para eso servían las funciones, para establecer lapsos o ramas de un proceso que llevaban a pequeños resultados que al final te daban el resultado final, un ejemplo muy común pues es la función main o función principal que es donde se realizan todas las demás funciones. Igual al final aprendí sobre la IEEE y los diferentes lineamientos que buscan que el mundo tecnológico tenga una paridad o sincronía que alimente el desarrollo y evolución en términos de tecnología para que se mejore la comunidad en sí. Muchos, muchos detalles como los símbolos ocupados para programar en Lenguaje C, cosas básicas como los corchetes, llaves, etc., pero que realmente son muy importantes a la hora de trabajar este lenguaje de programación. Y quiero cerrar con lo que considero más importante de mi aprendizaje en ese curso que fue yo creo la capacidad de resolver problemas, porque en muchas tareas y trabajos tuve que buscar solución de distintos problemas que nos proponían, en cada caso creo que fui mejorando y desarrollando una intuición natural a pensar en la forma en que podía resolver tal situación o representar a manera de código y con lo que había aprendido hasta el momento un ejecutable capaz de no solo realizar la actividad que me dejaban, sino a reproducirla a mi manera con mis lineamientos y como yo la estaba imaginando en mi cabeza, el cálculo matemático y no solo eso sino como representarlo con operadores del lenguaje de programación de manera clara.