



**Universidad Nacional Autónoma de
México**



Facultad de Ingeniería

Semestre 2021-1

Estructura de Datos y Algoritmos

Actividad 1: Repaso

Nombre del alumno: Perera Martínez David

Fecha: 24 de febrero de 2021

A lo largo del semestre pasado realizamos un total de trece prácticas, en las cuáles pusimos a prueba nuestros conocimientos adquiridos a lo largo de las clases que nos daban en ese semestre.

La primera que hicimos, en mi caso fue hacer una infografía que nos serviría como una pequeña introducción a todos los alumnos recién llegados de la preparatoria.

Para ir conociendo un poco sobre cómo se fueron creando varios lenguajes de programación y cuál fue su proceso.

También vimos sus pros y contras, e investigamos cuáles eran las mejores opciones en la actualidad.

Cada equipo le toco un lenguaje, en mi caso nos tocó JAVA



DESVENTAJAS

- Menor rendimiento
- Sólo ejecutable en una máquina virtual JVM
- No para principiantes
- Sintaxis complicada.
- Lenta evolución a comparación de los demás lenguajes.





OBJETIVO PRINCIPAL

Java se creó como una herramienta de programación para ser usada en un proyecto de set-top-box en una pequeña operación denominada the Green Project en Sun Microsystems en 1991.

Fundamentos de programación | Grupo 10

ELABORADO POR:

CONTRERAS GODÍNEZ ERICK

CRUZ CANO OMAR

PERERA MARTÍNEZ DAVID

QUERO BAUTISTA YAXCA ALEXA

ROSALES PIÑA ALEJANDRA

Referencias:

Graussard, T. [2012]. Presentación. JAVA 7: Los fundamentos del lenguaje Java. (Primera edición, pp. 9-11). Ediciones Eri.

Graussard, T. [2012]. Características de Java. JAVA 7: Los fundamentos del lenguaje Java. (Primera edición, pp. 12-17). Ediciones Eri.

(2019, 2 de julio). ¿Cuáles son las ventajas de Java y sus usos?. Netec. <https://www.netec.com/post/cuales-son-las-ventajas-de-java-y-sus-usos/>

Luego hicimos la **práctica dos** fue algo un poco breve, fue algo acerca de Linux/Gna en la cual nos registramos en el servidor de la escuela e hicimos unos comandos un poco básicos que la verdad nunca más volvimos a ocupar esto. Siento que solo fue para cumplir con la práctica 2.

Después sigue la **práctica 3** que era sobre solución de problemas y algoritmos, en la cual elaboramos algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software.

Teníamos que identificar el conjunto de entrada y el conjunto de salida y teníamos que elaborar un algoritmo para solucionar un problema un “poco sencillo”.

Un ejemplo de los ejercicios que hicimos en equipos fue:

Problema:

Solicitar 5 números e imprimir la suma del primero y el quinto, la resta del segundo con el cuarto y la multiplicación de los cinco.

Proceso:

Inicio

Imprimir "Ingrese número 1"
Leer N1

Imprimir "Ingrese número 2"
Leer N2

Imprimir "Ingrese número 3"
Leer N3

Imprimir "Ingrese número 4"
Leer N4

Imprimir "Ingrese número 5"
Leer N5

$S = N1 + N5$

$R = N2 - N4$

$M = N1 * N2 * N3 * N4 * N5$

Imprimir "La suma del primero y el quinto es": S

Imprimir "La resta del segundo y el cuarto es": R

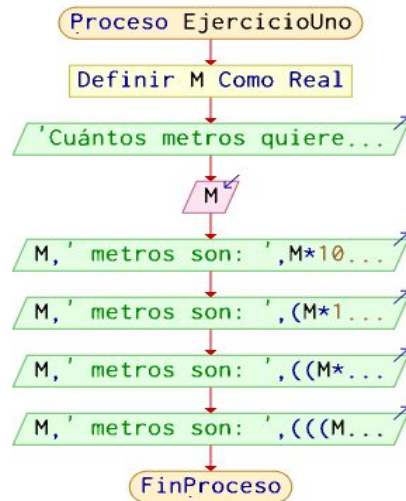
Imprimir "La multiplicación de los cinco número es": M

Fin

Después hicimos la **Práctica cuatro y la Práctica cinco** juntas en la cual elaboramos diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprendan un proceso. Así mismo elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

Un ejemplo de lo que hicimos en equipo fue:

1. Dada una longitud en metros obtener sus equivalentes en cm, pies, pulgadas y yardas. Considere que 1 metro=100 cm, 1 pulgada=2.54 cm, 1 pie=12 pulgadas, 1 yarda=3 pies.



Hicimos la **práctica 6** que era sobre los editores, compilación y ejecución la verdad no recuerdo mucho sobre esta y no encuentro el archivo que ocupamos para hacerlo en equipo, pero algo un poco complejo.

Realizamos la **práctica 8** en la cual hicimos elaboración programas en lenguaje C que incluyan las estructuras de selección if, if else, switch y ternaria (o condicional) para la resolución de problemas básicos.

Un ejemplo sobre lo que hicimos fue:

Ejercicio 3.

Solicitar tres números y preguntar al usuario en qué orden se desea acomodar los números de manera ascendente o descendente.

ASCENDENTE/DESCENDENTE

- 1-Ordenar manera ascendente.
- 2-Ordenar de manera descendente.
- 3-Salir.

Introduzca opción (1-2-3): 2

Ponga sus tres numeros.

50

10

35

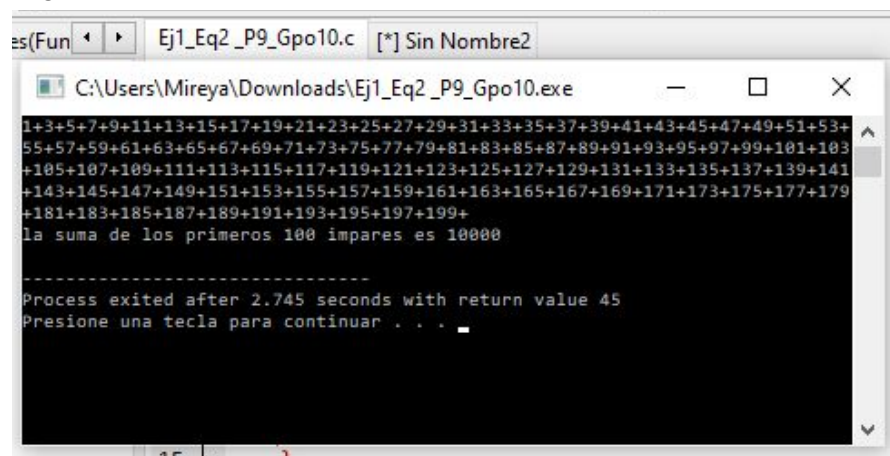
Su orden es 50, 35, 10.

Luego realizamos **la práctica 9** en la cual Elaboramos programas en lenguaje C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva define.

Un ejemplo de lo que realizamos fue:

Ejercicio 1

Diseñe un programa que sume los 100 primeros números impares.



```
es(Fun  Ej1_Eq2_P9_Gpo10.c  [*] Sin Nombre2
C:\Users\Mireya\Downloads\Ej1_Eq2_P9_Gpo10.exe
1+3+5+7+9+11+13+15+17+19+21+23+25+27+29+31+33+35+37+39+41+43+45+47+49+51+53+
55+57+59+61+63+65+67+69+71+73+75+77+79+81+83+85+87+89+91+93+95+97+99+101+103
+105+107+109+111+113+115+117+119+121+123+125+127+129+131+133+135+137+139+141
+143+145+147+149+151+153+155+157+159+161+163+165+167+169+171+173+175+177+179
+181+183+185+187+189+191+193+195+197+199+
la suma de los primeros 100 impares es 10000

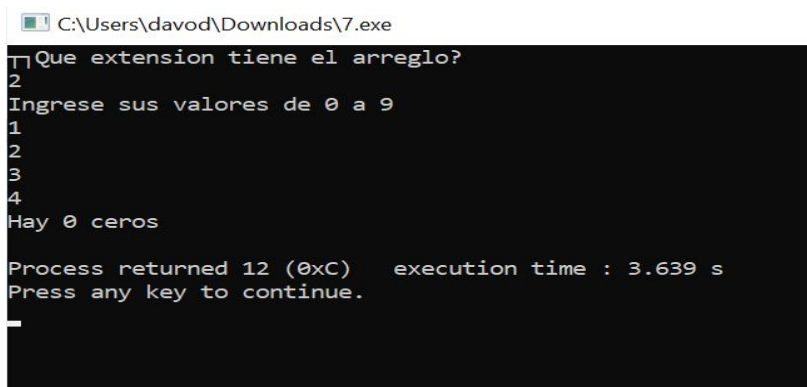
-----
Process exited after 2.745 seconds with return value 45
Presione una tecla para continuar . . .
```

En la **práctica 11** el objetivo de esta fue Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

Un ejemplo de lo que hicimos fue:

Problema 7.

Se requiere determinar cuántos ceros se encuentran en un arreglo de “n” renglones y “n” columnas, las cuales almacenan valores comprendidos entre 0 y 9.



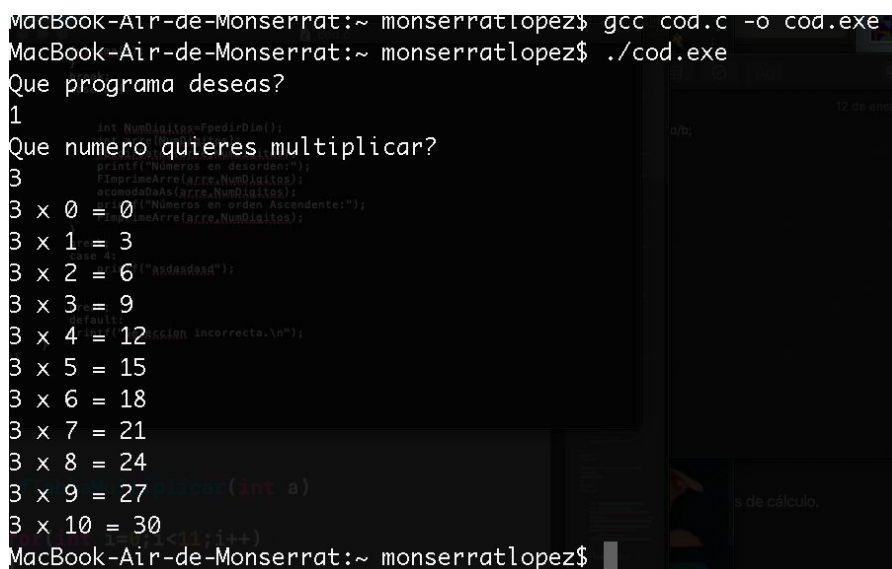
```
C:\Users\davod\Downloads\7.exe
Que extension tiene el arreglo?
2
Ingrese sus valores de 0 a 9
1
2
3
4
Hay 0 ceros

Process returned 12 (0xC)   execution time : 3.639 s
Press any key to continue.
```

En la **práctica 12** Elaboramos programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

Ejercicio 1

Crear una función escribirTablaMultiplicar, que reciba como parámetro un número entero, y escriba la tabla de multiplicar de ese número (por ejemplo, para el 3 deberá llegar desde 3x0=0 hasta 3x10=30).



```
MacBook-Air-de-Monserrat:~ monserratllopez$ gcc cod.c -o cod.exe
MacBook-Air-de-Monserrat:~ monserratllopez$ ./cod.exe
Que programa deseas?
1
Que numero quieres multiplicar?
3
3 x 0 = 0
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
MacBook-Air-de-Monserrat:~ monserratllopez$
```

Y por último en **la práctica 13** el objetivo fue elaborar programas en lenguaje C que requieran el uso de archivos de texto plano en la resolución de problemas, entendiendo a los archivos como un elemento de almacenamiento secundario.

Ejercicio 1.

Diseñe un programa, que sume dos variables de tipo entero, por medio de apuntadores. Deberá imprimir la dirección de memoria de cada una de las variables y el contenido de cada variable en un archivo txt. El cual se generará cada vez que se ejecute el programa.

```
MacBook-Air-de-Monserrat:~ monserratlópez$ gcc cod.c -o cod.exe
MacBook-Air-de-Monserrat:~ monserratlópez$ ./cod.exe
Que programa quieres?
1) Programa 1 2) Programa 2 3) Programa 3 4) Programa 4
1
Ingrese dos numeros:
20
30
El resultado ya esta en tu archivo
Desea repetir el programa? 1) SI 2) NO
2
MacBook-Air-de-Monserrat:~ monserratlópez$
```

```
suma.txt
El contenido de a es:20
El contenido de b es:30

La suma del contenido de los apuntadores es: 50

Direccion de a es 0x7ffeea868abc
Direccion de b es 0x7ffeea868ab8
```