

**LABORATORIO: SIMULACION Y OPTIMIZACION DE UN PROGRAMA EN UN
PROCESADOR ESCALAR SEGMENTADO**

**VICTOR ALFONSO CALVACHE MOTOA
INGENIERIA INFORMATICA
HOMOLOGANTE**

**DEIVIS EDUARD RAMIREZ MARTINEZ
INGENIERO DE SISTEMAS**

**ESTRUCTURA DE COMPUTADORES
UNIR
CALI-COLOMBIA**

08 DE SEPTIEMBRE DEL 2024

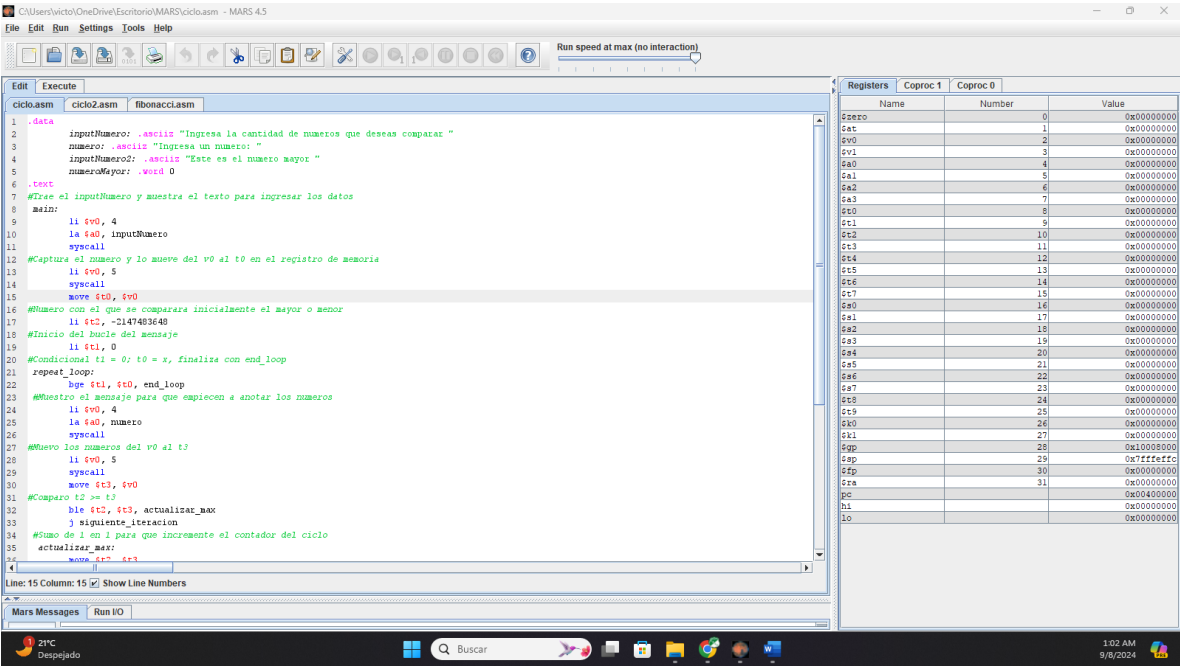
INTRODUCCION

En el siguiente trabajo se va a desarrollar ejercicios para entender como funciona el lenguaje de bajo nivel, el cual desempeña un papel muy importante al momento de trabajar con hardware y software a la vez, y entender el tema de los registros que se guardan en memoria al momento de ejecutar una porción de código. Es importante tener presente que estos ejercicios realizados tuvieron acompañamiento del profesor y de diferentes temáticas leídas y vistas en diferentes plataformas como Youtube y ChatGPT para poder entender como funciona MARS 4.5 con MIPS.

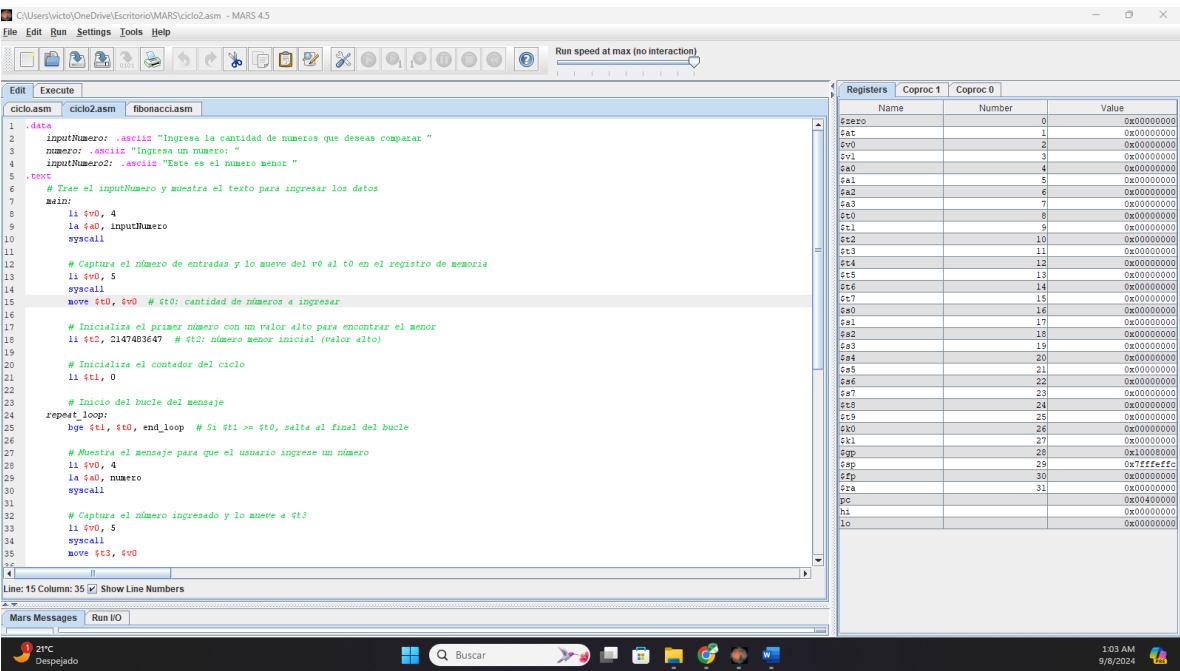
ACTIVIDAD

1. Antes de compilar

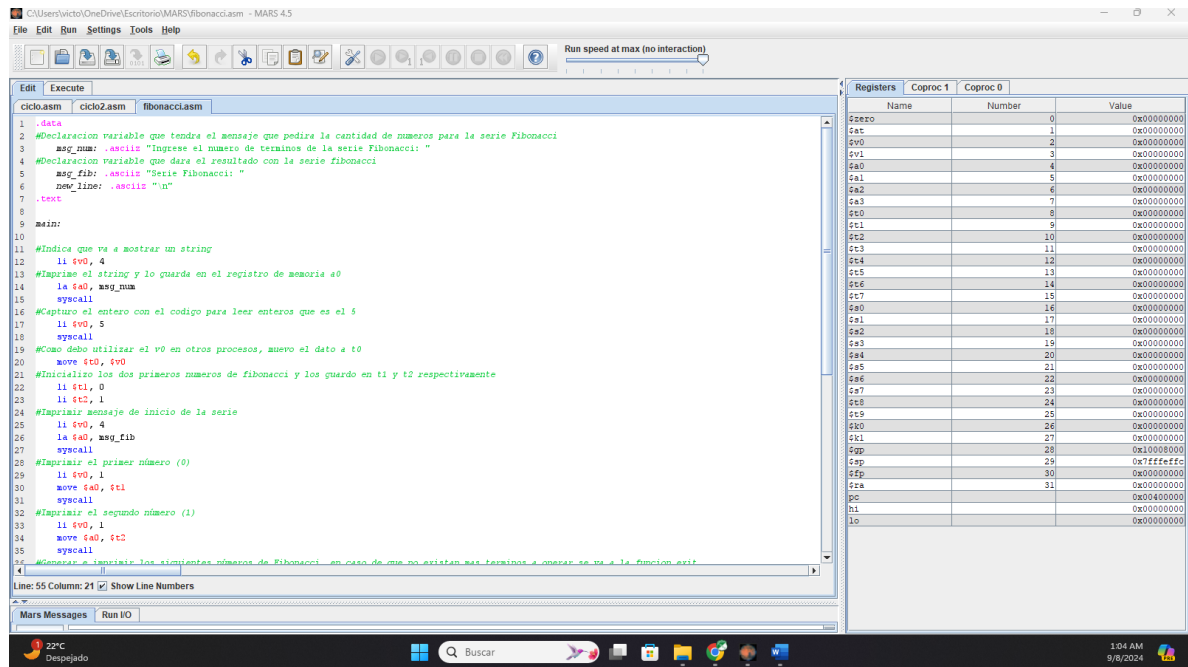
Ejercicio #1



Ejercicio #2

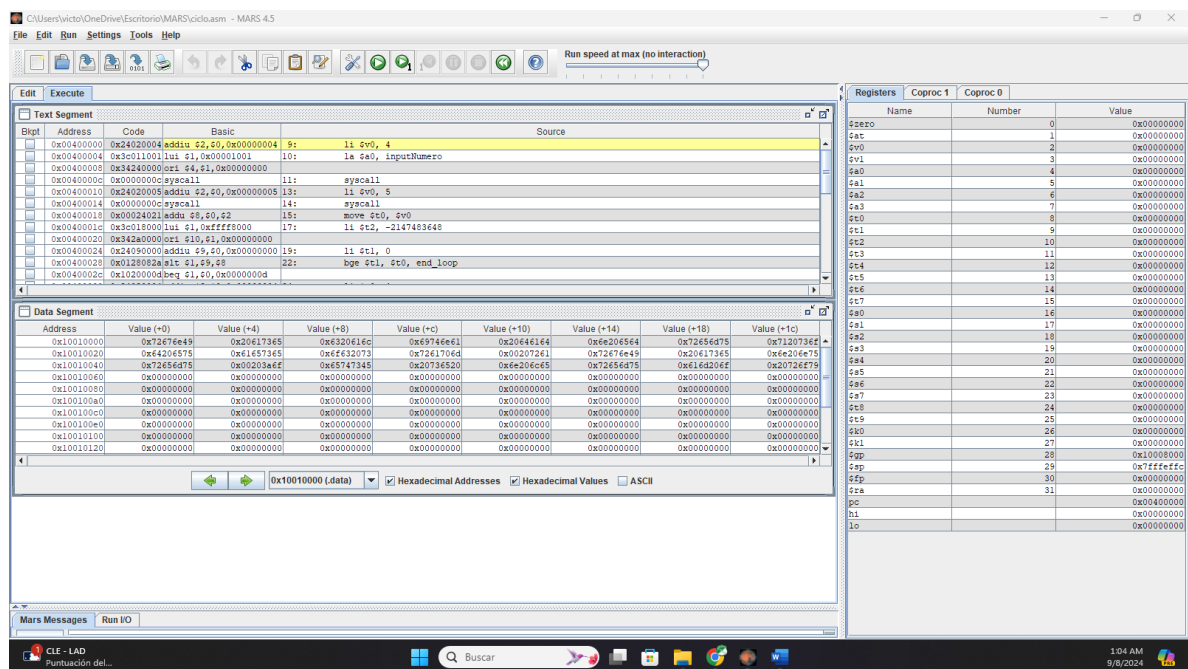


Ejercicio #3



2. Después de compilar

Ejercicio #1



Ejercicio #2

C:\Users\vicfo\OneDrive\Escritorio\MARS\ciclo2.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Inst	Address	Code	Basic	Source
0x04000000	0x24020004	addiu \$2,\$0,0x00000004	8:	li \$v0, 4
0x04000004	0x3c011001	lui \$1,0x00001001	9:	la \$a0, inputNumero
0x04000008	0x34240000	ori \$4,\$1,0x00000000		
0x0400000c	0x0000000c	syscall	10:	syscall
0x04000010	0x24020005	addiu \$2,\$0,0x00000005	13:	li \$v0, 5
0x04000014	0x0000000c	syscall	14:	syscall
0x04000018	0x00024021	addu \$5,\$0,\$2	15:	move \$t0, \$v0 # \$t0: cantidad de números a ingresar
0x0400001c	0x3c017fff	lui \$1,0x00007fff	18:	li \$t2, 2147483647 # \$t2: número menor inicial (valor alto)
0x04000020	0x342aefff	ori \$10,\$1,0x0000ffff		
0x04000024	0x24090000	addiu \$9,\$0,0x00000000	21:	li \$t1, 0
0x04000028	0x0128082a	slti \$1,\$5,\$8	25:	bge \$t1, \$t0, end_loop # Si \$t1 >= \$t0, salta al final del bucle
0x0400002c	0x1020000b	beq \$1,\$0,0x0000000d		

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x72476e49	0x20617365	0x6320616c	0x69746665	0x20646164	0x66e20654	0x72656475	0x7120736f
0x10010020	0x42467575	0x41457365	0x66632073	0x72617064	0x00207261	0x72476e49	0x20417365	0x6e206475
0x10010040	0x72456d75	0x00203a6f	0x65747345	0x20736520	0x6e206c65	0x72456d75	0x656d206f	0x20726665
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run IO

CLE - LAD Puntuación del...

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$a0	16	0x00000000
\$a1	17	0x00000000
\$a2	18	0x00000000
\$a3	19	0x00000000
\$a4	20	0x00000000
\$a5	21	0x00000000
\$a6	22	0x00000000
\$a7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$d0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$fp	29	0xffffffff
\$ra	30	0x00000000
\$ra	31	0x00000000
\$pc		0x00400000
\$t1		0x00000000
\$t0		0x00000000

Ejercicio #3

C:\Users\vicfo\OneDrive\Escritorio\MARS\fibonacci.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Inst	Address	Code	Basic	Source
0x04000000	0x24020004	addiu \$2,\$0,0x00000004	12:	li \$v0, 4
0x04000004	0x3c011001	lui \$1,0x00001001	14:	la \$a0, msg_num
0x04000008	0x34240000	ori \$4,\$1,0x00000000		
0x0400000c	0x0000000c	syscall	15:	syscall
0x04000010	0x24020005	addiu \$2,\$0,0x00000005	17:	li \$v0, 5
0x04000014	0x0000000c	syscall	18:	syscall
0x04000018	0x00024021	addu \$5,\$0,\$2	20:	move \$t0, \$v0
0x0400001c	0x24090000	addiu \$9,\$0,0x00000000	22:	li \$t1, 0
0x04000020	0x240a0001	addiu \$10,\$0,0x00000001	23:	li \$t2, 1
0x04000024	0x24020004	addiu \$2,\$0,0x00000004	25:	li \$v0, 4
0x04000028	0x3c011001	lui \$1,0x00001001	26:	la \$a0, msg_fib
0x0400002c	0x34240036	ori \$4,\$1,0x00000036		

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x72476e49	0x20617365	0x6320616c	0x69746665	0x6564206f	0x72457420	0x66e20654	0x65642073
0x10010020	0x20416c20	0x6f724573	0x69420655	0x61666361	0x61666361	0x65550020	0x20456972	0x6f622946
0x10010040	0x63631e1e	0x00203a69	0x0000000a	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run IO

CLE - LAD Puntuación del...

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$a0	16	0x00000000
\$a1	17	0x00000000
\$a2	18	0x00000000
\$a3	19	0x00000000
\$a4	20	0x00000000
\$a5	21	0x00000000
\$a6	22	0x00000000
\$a7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$d0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$fp	29	0xffffffff
\$ra	30	0x00000000
\$ra	31	0x00000000
\$pc		0x00400000
\$t1		0x00000000
\$t0		0x00000000

3. Después de ejecutar

Ejercicio #1

The screenshot shows the MARS 4.5 IDE with the following components:

- Text Segment:** Assembly code for a program that compares two numbers. The code includes instructions for loading, comparing, and branching.
- Data Segment:** Memory addresses and values for the program's data.
- Registers:** A table showing the state of registers, including Name, Number, and Value.
- Mars Messages:** A window showing the program's output, including the prompt "Ingresar la cantidad de numeros que desea comparar 3" and the result "Este es el numero mayor 6".

Ejercicio #2

The screenshot shows the MARS 4.5 IDE with the following components:

- Text Segment:** Assembly code for a program that compares two numbers. The code includes instructions for loading, comparing, and branching.
- Data Segment:** Memory addresses and values for the program's data.
- Registers:** A table showing the state of registers, including Name, Number, and Value.
- Mars Messages:** A window showing the program's output, including the prompt "Ingresar la cantidad de numeros que desea comparar 3" and the result "Este es el numero menor 1".

Ejercicio #3

C:\Users\victor\OneDrive\Escritorio\MARS\fibonacci.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Offset	Address	Code	Basic	Source
	0x00400000	0x24020004	addiu \$2,\$0,0x00000004	li \$v0, 4
	0x00400004	0x3c011001	lui \$1,0x00001001	la \$a0, msg_num
	0x00400008	0x34240000	ori \$4,\$1,0x00000000	
	0x0040000c	0x0000000c	syscall	15: syscall
	0x00400010	0x24020005	addiu \$2,\$0,0x00000005	li \$v0, 5
	0x00400014	0x0000000c	syscall	18: syscall
	0x00400018	0x00024021	addu \$5,\$0,\$2	20: move \$t0, \$v0
	0x0040001c	0x24090000	addiu \$9,\$0,0x00000000	li \$t1, 0
	0x00400020	0x240a0001	addiu \$10,\$0,0x00000001	li \$t2, 1
	0x00400024	0x24020004	addiu \$2,\$0,0x00000004	li \$v0, 4
	0x00400028	0x3c011001	lui \$1,0x00001001	la \$a0, msg_fib
	0x0040002c	0x34240036	ori \$4,\$1,0x00000036	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x72676e49	0x26457365	0x6e206e48	0x72656d79	0x6564206f	0x72657420	0x6f66696d	0x65642073
0x10010020	0x20416c20	0x6f726573	0x69462065	0x71666f62	0x3a696363	0x65530020	0x20464972	0x6f626946
0x10010040	0x6363616e	0x00203a69	0x0000000a	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000004
\$v1	3	0x00000000
\$a0	4	0x10010048
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000002
\$t2	10	0x00000003
\$t3	11	0x00000003
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$t8	16	0x00000000
\$t9	17	0x00000000
\$s0	18	0x00000000
\$s1	19	0x00000000
\$s2	20	0x00000000
\$s3	21	0x00000000
\$s4	22	0x00000000
\$s5	23	0x00000000
\$s6	24	0x00000000
\$s7	25	0x00000000
\$s8	26	0x00000000
\$s9	27	0x00000000
\$k0	28	0x10008000
\$k1	29	0xffffffff
\$fp	30	0x00000000
\$ra	31	0x00000000
\$pc		0x00400094
\$hi		0x00000000
\$lo		0x00000000

Mars Messages Run IO

Ingrese un numero de terminos de la serie Fibonacci: 5
Este es el numero menor 1
-- program is finished running --

Clear

Ingrese el numero de terminos de la serie Fibonacci: 5
Serie Fibonacci: 01123
-- program is finished running --

22°C Despejado

Buscar

1:07 AM 9/8/2024

CONCLUSIONES

Gracias al trabajo desarrollado en MARS 4.5, se ha podido entender como funcionan las aplicaciones con lenguaje de bajo nivel, la importancia de reutilizar código y de aplicar las diferentes técnicas para organizar de una manera mas rápida y segura el flujo del proceso en los diferentes ejercicios desarrollados.