

REGISTRO E INSTRUCCIONES

CÓDIGO DE ALTO NIVEL

//ejemplo de programa de alto nivel

```
int main () {  
    int a[5]=10;  
    int b=3;  
    c=2+b  
    d[5]=c+a[5];  
    return d[5];  
}
```

LENGUAJE ENSAMBLADOR

```
OR %G0, 10, %L0  
OR %G0, 3, %L1  
ST %L4, [%L0+(5*4)]  
ADD %L2, %L1, %L1  
LD [%L3+(5*4)], %L5  
ADD %L1, %L4, %L6  
ST %L2, [%L5+(5*4)]
```

LENGUAJE MAQUINA

FORMATO 3						
Dir.Memoria	OP	RD	OP3	RS1	i	UNUSED(ZERO) / RS2
0X0000	10	10000	000010	00000	1	0000000001010
0X0004	10	10001	000010	00000	1	0000000000011
0X0008	11	10100	000100	10000	1	0000000010100
0X000C	10	10001	000000	10010	0	00000000/10001
0X0010	11	10101	000000	10011	1	0000000010100
0X0014	10	10110	000000	10001	0	00000000/10100
0X0018	11	10010	000100	10101	1	0000000010100

ANGELICA MARIA LOAIZA LOAIZA
COD. 1088310806

PROGRAMA: INGENIERIA DE SISTEMAS Y COMPUTACION

PROCEDIMIENTO DEL LENGUAJE

ENSAMBLADOR

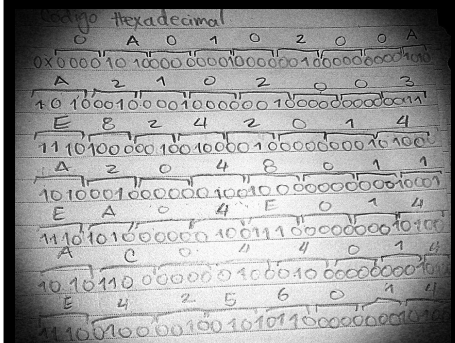
//Inicializando con OR, el %G0 siempre será de valor cero para que no genere basura, seguido por un registro inmediato de valor x(en este caso 10 y 3), y por ultimo un registro destino.

//El Store es quien me permite guardar en memoria

//Instruccion aritmetologica que permite ejecutar una suma con registro Fuente uno, registro Fuente 2 y un registro destino donde lo queremos almacenar.

// El Load es quien me permite extraer de memoria el dato indicado en el programa.

//pasamos a codigo hexadecimal de a cuatro bits separando de derecho a izquierda los resultants del format 3. Como este:



CODIGO HEXADECIMAL:

A010200A
A2102003
E8242014
A2048011
EA04E014
AC044014
E4256014