

Arquitectura de Máquina Virtual

- **Código de instrucción:** 6 dígitos decimales (2 código de Operación, 4 dirección de memoria)
- Registro **A (acumulador)**
- Contador de programa **PC (memPointer)**
- Apuntador a memoria de la pila **SP (stackPointer)**.
- Datos: **enteros** (6 dígitos decimales) y **cadenas** (1 carácter ocupa 3 dígitos decimales).

Instrucción Ensamblador	Instrucción Máquina	Descripción	Ejemplo Máquina	Ejemplo Ensamblador
ASP	08	Asigna un valor entero positivo al acumulador.	080005	ASP 5
ASN	09	Asigna un valor entero negativo al acumulador.	090007	ASN 7
RD	10	Lee entero del teclado y guarda en memoria.	100020	RD Y
WD	11	Escribe en pantalla entero leído desde la memoria.	110020	WD Y
NL	12	Salta a línea siguiente en la pantalla.	120000	NL
RS	13	Lee cadena y guarda a partir de dirección de memoria. (máximo 40 caracteres)	130100	RS S1
WS	14	Escribe en pantalla cadena a partir de dirección de memoria.	140100	WS S1
LDA	20	Carga en A valor desde la memoria.	200035	LDA Z
STA	21	Guarda en memoria el valor actual de A.	210007	STA W
ADD	30	Suma A con variable, resultado queda en A.	300035	ADD Z
SUB	31	Resta A con variable ,resultado queda en A.	310007	SUB Y
DIV	32	Divide A con variable, resultado queda en A.	320020	DIV Y
MUL	33	Multiplica A con variable, resultado queda en A.	330005	MUL X
MOD	34	Residuo de la división, de A modulo variable, resultado queda en A.	340007	MOD X
EXP	35	Exponente de $A ^ \text{variable}$, resultado en A.	350005	EXP E
J	40	Salto incondicional, a dirección de memoria (una etiqueta).	400110	J J1
JLT	41	Salta si A menor a cero a dirección de memoria.	410050	JLT J2
JEQ	42	Salta si igual $A==0$ a dirección de memoria.	420060	JEQ J3
JGT	43	Salta si A mayor a cero a dirección de memoria.	430120	JGT J4
COMP	44	Compara: Resta A menos variable, valor queda en A.	440025	COMP Z
JSUB	50	Salto a subrutina (identificada con etiqueta), coloca en Pila la dirección de retorno.	500090	JSUB S1
RSUB	51	Regresa de subrutina, coloca en PC el valor de regreso desde la Pila.	510000	RSUB
PUSH	52	Coloca en Pila valor desde A.	520000	PUSH
POP	53	Saca de la Pila valor, coloca en A.	530000	POP
END	64	Fin de programa	640000	END

Notas:

Las variables serán sustituidas por direcciones de memoria. Las operaciones de salto y llamado a subrutinas requieren la definición de etiquetas que serán sustituidas por direcciones de memoria.

Ejemplo de programa en ensamblador

```
#LEE NOMBRE Y EDAD Y DICE SI ES JOVEN O VIEJO SI EDAD >=60,  
#S1 y S2, SON LOS TEXTOS A MOSTRAR.  
ASP 60  
STA M  
RS S1  
RS S2  
RS N  
RD E  
LDA M  
COMP E  
JGT J1  
WS N  
WS S1  
END  
J1  
WS N  
WS S2  
END
```

Descripción:

El símbolo # al inicio de una línea, indica comentario y se ignoran los caracteres siguientes. Las líneas deben iniciar con una instrucción válida o etiqueta de salto, las instrucciones pueden tener un operando o sólo el código de instrucción. Las variables deben iniciar con letras y pueden ser de varios caracteres.

Ejemplo de archivo en lenguaje máquina (resultado de la traducción)

```
@ 138  
80060  
210016  
130017  
130057  
130097  
100137  
200016  
440137  
430012  
140097  
140017  
640000  
140097  
140057  
640000
```

Descripción:

La primera línea indica la cantidad aproximada de memoria que requiere el programa, se calcula al ser traducido por el ensamblador. El total es la cantidad de líneas, más las variables enteras, más las variables cadena (multiplicando por 40 cada variable cadena).

El resto de las líneas son las instrucciones en lenguaje máquina de la máquina virtual (en notación decimal). Las operaciones con argumentos requieren direcciones de memoria.

Al cargar al simulador se omite la primer línea, de manera que las instrucciones hacen referencia a partir de la posición 0 de memoria, que es donde se carga la primer instrucción del programa. La memoria máxima de la computadora es de 10,000 localidades, los programas pueden cargarse en cualquier posición, por ello, al ejecutar se emplea un método de direccionamiento relativo al sumar la posición de inicio del programa a la dirección de memoria que indique la instrucción a ejecutar (desplazamiento en base a la dirección de carga del programa).