



Actividad 2.8: Simulador de Instrucciones

Simula instrucciones básicas tipo ensamblador



Objetivo:

Diseñar e implementar en Java un pequeño simulador de instrucciones básicas.

Usa `LOAD`, `ADD`, `JMP`, `PRINT`, `HALT` y crea tu propio programa paso a paso.

¿Qué debe hacer tu simulador?

Tu programa debe ser capaz de:

- Leer una **lista de instrucciones en texto**
- Ejecutarlas en orden (controlado por un contador)
- Usar un **acumulador** como registro
- Terminar correctamente con `HALT`

Instrucciones a implementar

Instrucción	Efecto esperado
LOAD x	Guarda x en el acumulador
ADD x	Suma x al acumulador
JMP x	Salta a la instrucción número x
PRINT	Muestra el contenido del acumulador
HALT	Finaliza la ejecución



Fragmento inicial (pista)

```
int acc = 0;
int pc = 0;

String[] program = {
    "LOAD 5",
    "ADD 3",
    "PRINT",
    "HALT"
};
```

¿Cómo harías que el programa lea cada instrucción y actúe?

⚙ Pista: estructura general de ejecución

```
while (pc < program.length) {  
    String[] parts = program[pc].split(" ");  
    String inst = parts[0];  
    // ¿Cómo interpretas y ejecutas según el valor de inst?  
  
    pc++; // ¡Solo si no hay salto!  
}
```

¿Qué pasa si la instrucción es `JMP` ?

Reto adicional

Haz una **tabla de ejecución** de tu programa como esta:

Paso	PC	Instrucción	ACC antes	ACC después
1	0	LOAD 5	0	5
2	1	ADD 3	5	8
...	

Te ayudará a depurar y razonar tu simulador.

! ¿Qué puede fallar?

Prueba esto y analiza el resultado:

```
String[] program = {  
    "LOAD 2",  
    "ADD 10",  
    "PRINT",  
    "JMP 10",    // <- línea inválida  
    "HALT"  
};
```

¿Cómo podrías evitar errores como este?

Ampliación opcional

¿Te animas a extender el lenguaje?

Instrucción extra	Acción
SUB x	Resta x del acumulador
IFZERO x	Salta a x si el acumulador es cero
STORE y	Guarda el valor en mem[y]
LOADM y	Carga el valor de mem[y] en el acumulador



Pista para añadir memoria

```
int[] mem = new int[10]; // Simulación de memoria

// Ejemplo:
case "STORE":
    mem[arg] = acc;
    break;

case "LOADM":
    acc = mem[arg];
    break;
```

¡Piensa cómo leer o guardar datos más allá del acumulador!



Qué debes entregar

- ✓ Tu código Java del simulador
- ✓ Un programa propio de ejemplo
- ✓ Tabla de ejecución del programa
- ✓ (Opcional) Nuevas instrucciones o mejoras

¡Acepta el reto!

- ¿Puedes simular un pequeño procesador?
- ¿Cómo evitas errores como bucles infinitos?
- ¿Qué estructura de control es más adecuada?

 ¡Tu simulador, tu lógica, tus reglas!