

# Curso de C++: Introducción a C++

Introducción a los lenguajes de programación

# ¡BIENVENIDOS!

## José Domingo Muñoz Rodríguez

[www.josedomingo.org](http://www.josedomingo.org)

 josedom24

 @pledin\_jd



# Sistemas Informáticos



- El procesamiento de los datos lo hace un ordenador.
- El algoritmo se describe mediante un programa.
- **Programa:** Conjunto ordenado de instrucciones que se dan al ordenador indicándole las operaciones o tareas que ha de realizar para resolver un problema.
- Para escribir programas utilizamos **lenguajes de programación**.



# Lenguajes de programación





# ¿Qué lenguaje entiende...

...el ordenador?

...el programador?

```
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
11001010 11001010 11110101 00101011
11001010 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
```

Lenguaje máquina



Lenguaje natural

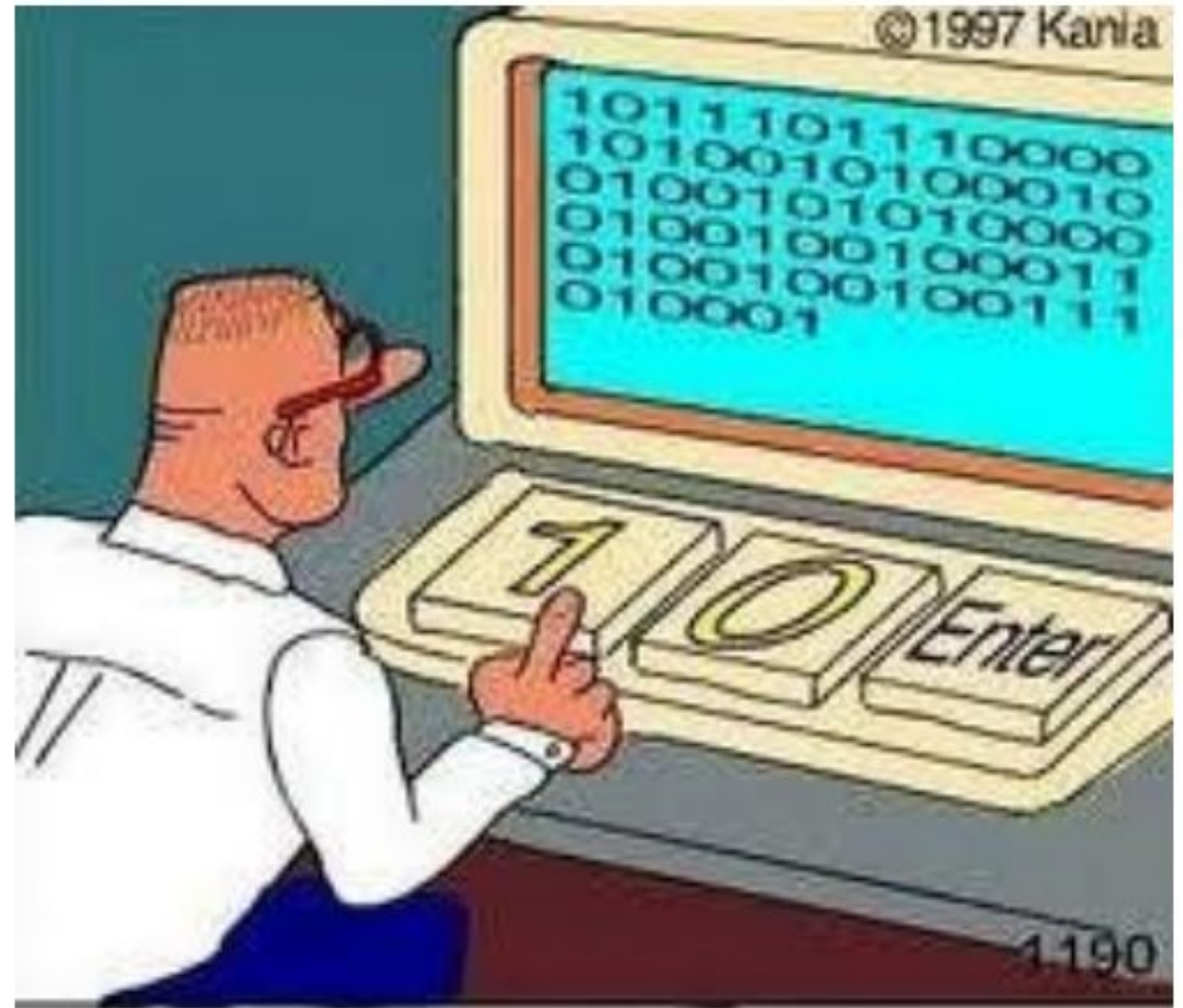




# Lenguaje máquina

El único que entiende directamente la CPU del ordenador

- Depende del modelo de ordenador
- Repertorio de instrucciones reducido (operaciones muy elementales)
- Muy difícil programar en él (en binario, con cadenas de ceros y unos)



LOS VERDADEROS PROGRAMADORES  
PROGRAMAN EN BINARIO



# Lenguaje ensamblador

Equivalente al lenguaje máquina, cada línea de código se traduce en una instrucción para la máquina.

- Le asocia mnemónicos a las operaciones que entiende la CPU
- Repertorio de instrucciones reducido (operaciones muy elementales)
- Programas difíciles de entender

004012A7	90	NOP	
004012A8	53	PUSH EBX	
004012A9	56	PUSH ESI	
004012AA	57	PUSH EDI	
004012AB	8BF2	MOV ESI,EDX	
004012AD	8BD8	MOV EBX,EAX	
004012AF	85F6	TEST ESI,ESI	
004012B1	8BF8	MOV EDI,EBX	
004012B3	74 35	JE SHORT RTRACE.004012EA	
004012B5	6A 04	PUSH 4	
004012B7	68 00100000	PUSH 1000	
004012BC	68 00100000	PUSH 1000	
004012C1	53	PUSH EBX	
004012C2	E8 15870000	CALL <JMP.&KERNEL32.VirtualAlloc>	Protect = PAGE_READWRITE AllocationType = MEM_COMMIT Size = 1000 (4096.) Address VirtualAlloc
004012C7	85C0	TEST EAX,EAX	
004012C9	75 0F	JNZ SHORT RTRACE.004012DA	
004012CB	8BD3	MOV EDX,EBX	
004012CD	8BC7	MOV EAX,EDI	
004012CF	2BD7	SUB EDX,EDI	
004012D1	E8 1E000000	CALL RTRACE.004012F4	
004012D6	33C0	XOR EAX,EAX	
004012D8	EB 15	JMP SHORT RTRACE.004012EF	
004012DA	81C3 00100000	ADD EBX,1000	
004012E0	81EE 00100000	SUB ESI,1000	
004012E6	85F6	TEST ESI,ESI	
004012E8	75 CB	JNZ SHORT RTRACE.004012B5	
004012EA	B8 01000000	MOV EAX,1	
004012EF	5F	POP EDI	
004012F0	5E	POP ESI	
004012F1	5B	POP EBX	
004012F2	C3	RETN	
004012F3	90	NOP	



# Lenguaje de alto nivel

Permiten que el programador exprese el procesamiento de datos de forma simbólica, sin tener en cuenta los detalles específicos de la máquina.

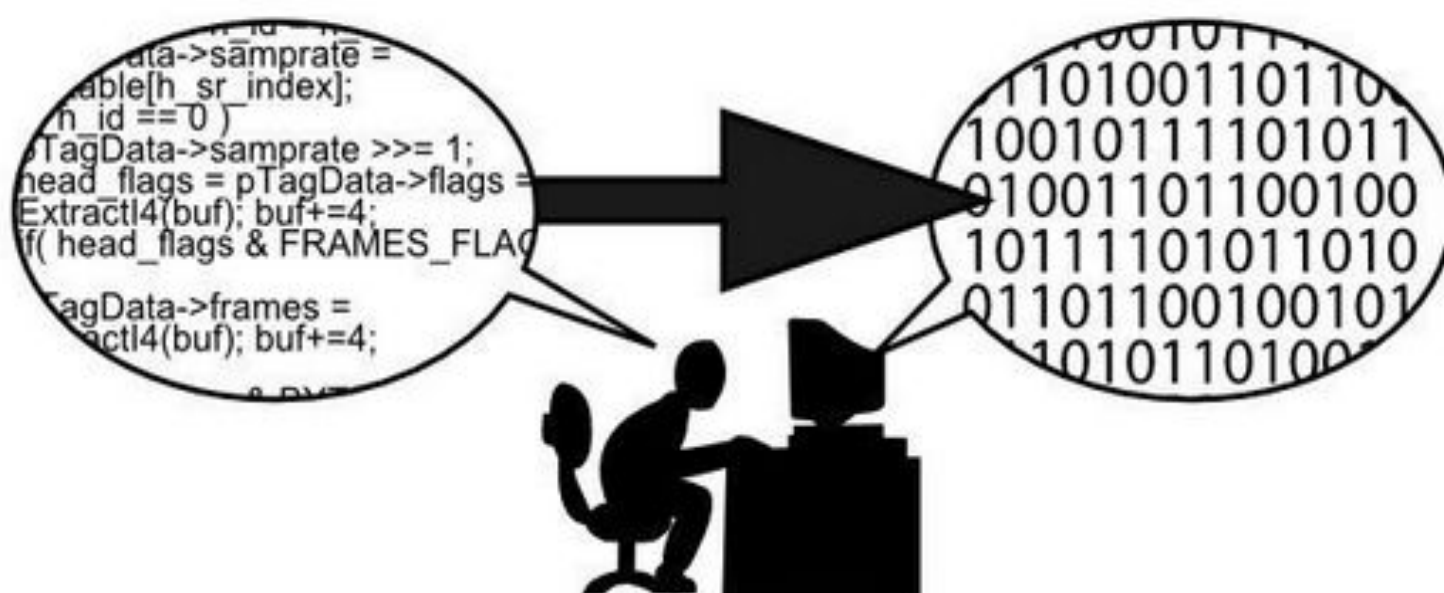
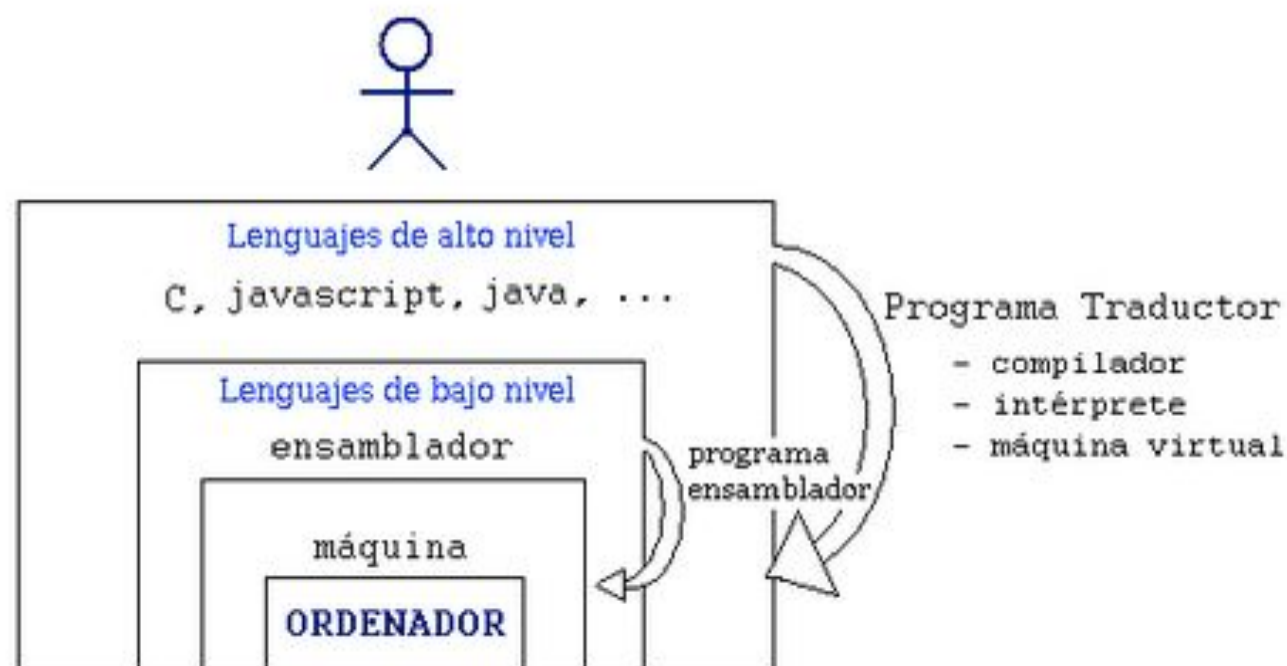
- Independientes del modelo de ordenador
- Proporcionan un mayor nivel de abstracción

```
readln(b);  
writeln('ingresa un numero c');  
readln(c);  
  
if (a>b )and (a>c) then  
begin  
writeln('el mayor es a');  
end;  
begin;  
if (b>c) and (b>a) then  
writeln('el mayor es b');  
end;  
begin;  
if (c>a) and (c>b)then  
writeln('el mayor es c')  
end;  
  
readkey;
```



# Traductores

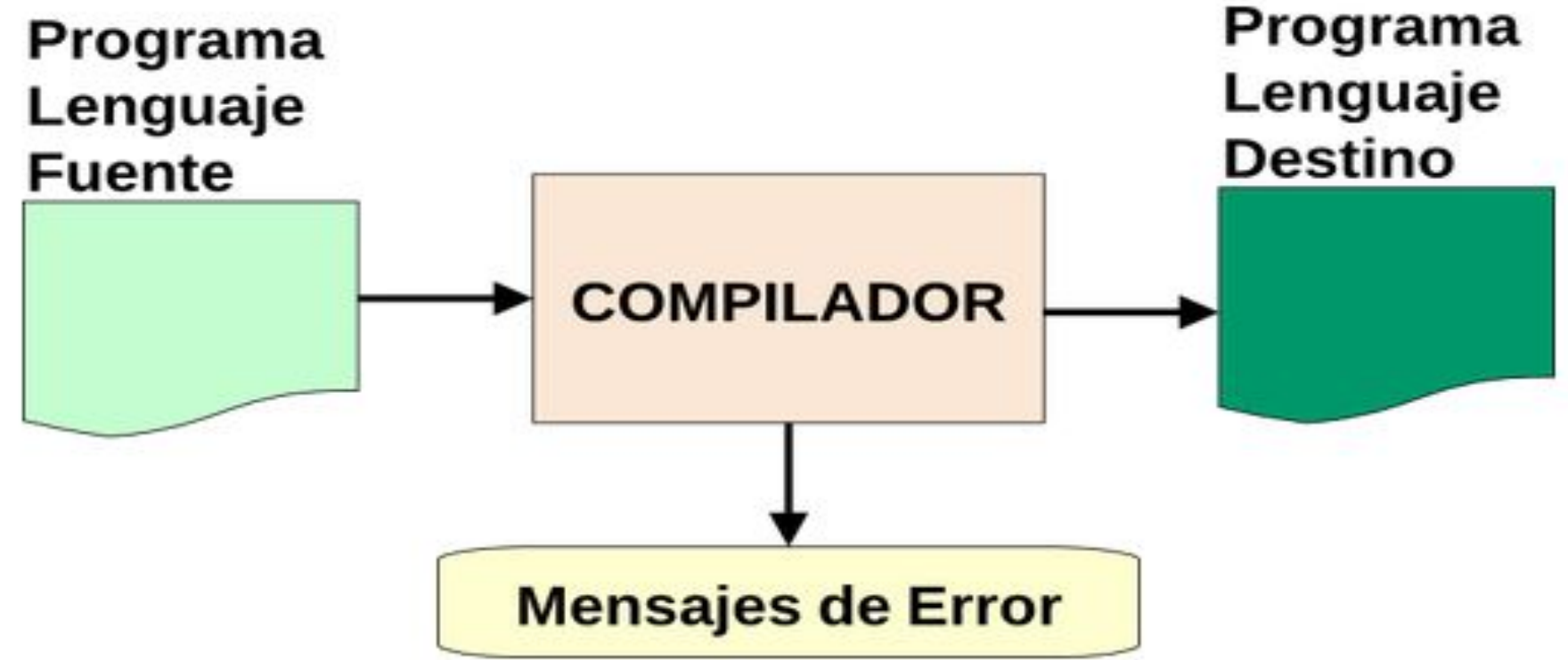
Los traductores transforman programas escritos en un lenguaje de alto nivel en programas escritos en código máquina.





# Compilador

- Convierte un programa escrito en alto nivel (código fuente) a un programa máquina (código ejecutable).
- Para generar el código ejecutable el código no debe tener errores de sintaxis.
- Necesitamos un compilador para cada arquitectura y sistema operativo.
- Los programas ejecutables no son compatibles entre plataformas.
- Una vez generado el programa ejecutable, no es necesario tener el código fuente.

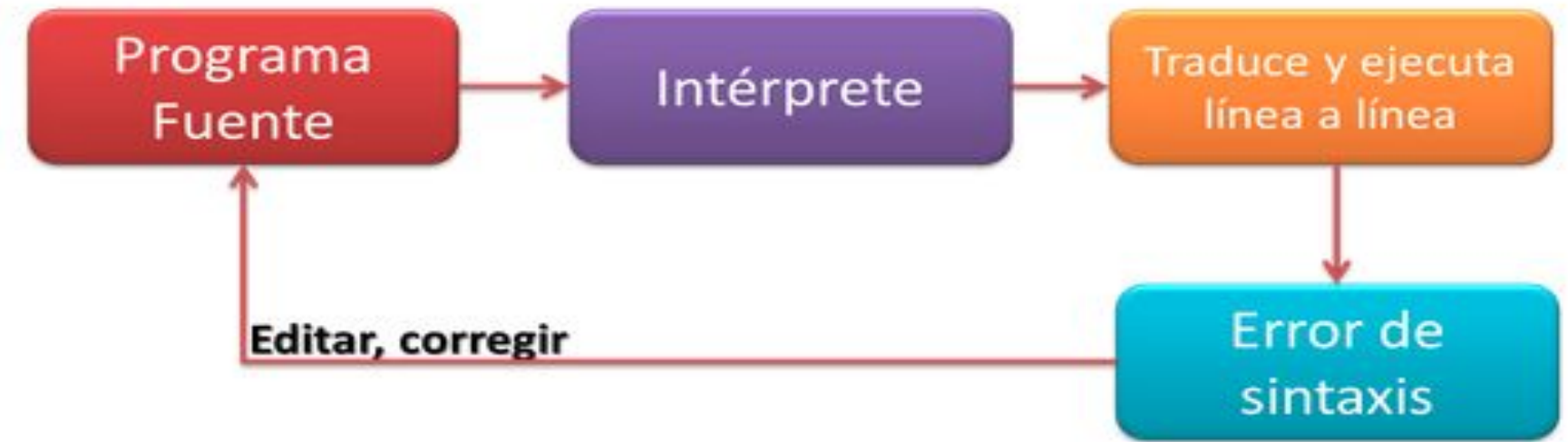


Ejemplos:  
C, Pascal, ...



# Interprete

- La traducción y ejecución de código fuente a código máquina se hace línea por línea.
- Los errores de sintaxis aparecen cuando se interpreta la instrucción con error.
- Necesitamos el código fuente para ejecutar el programa.
- Los lenguajes interpretados suelen ser más lentos en su ejecución.

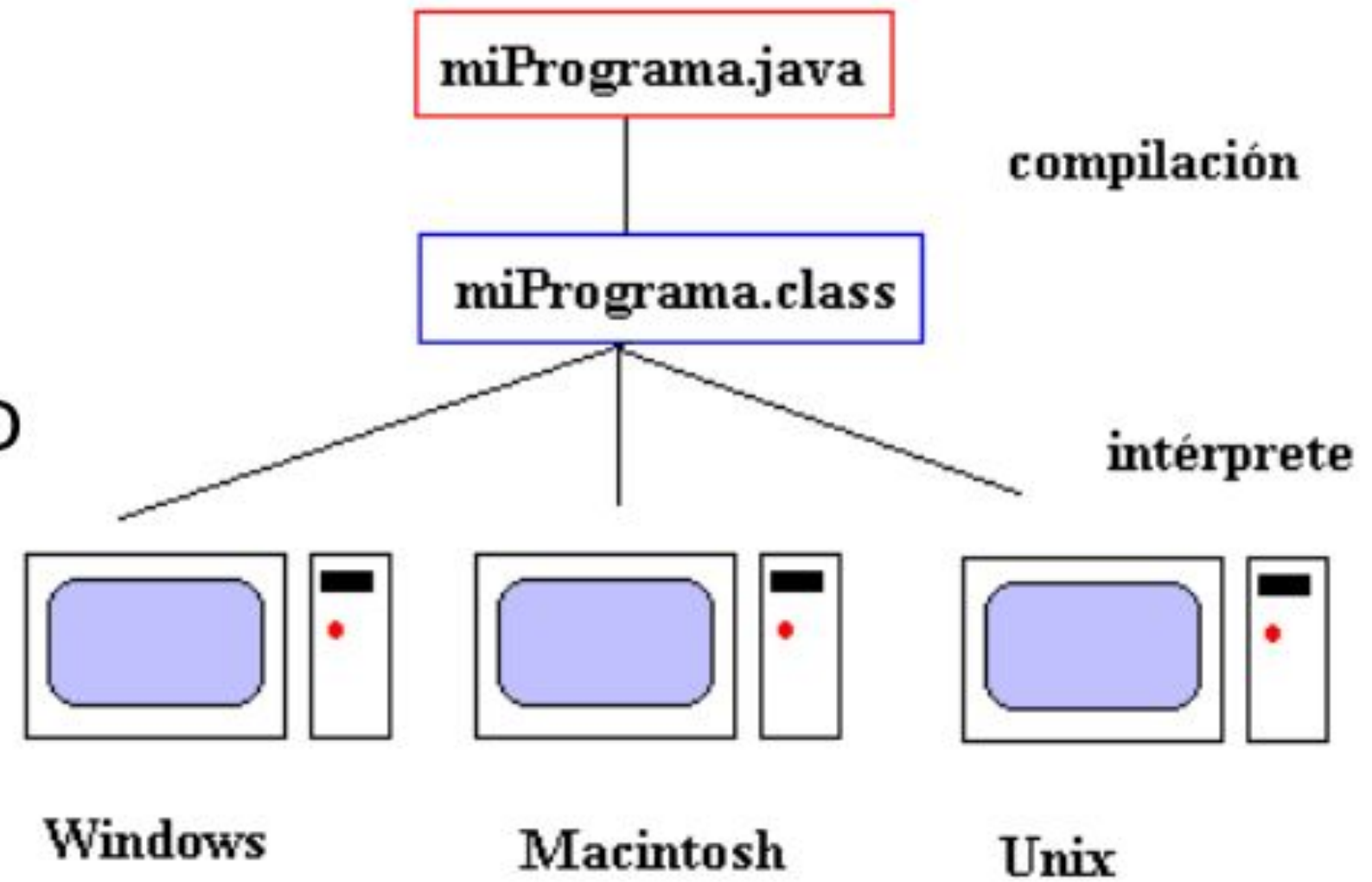


Ejemplos: python, php, ...



# Máquina virtual

- La traducción se hace en dos pasos.
- Primero se compila el código fuente a un código intermedio (bytecode).
- Segundo, este bytecode se interpreta y ejecuta por una "máquina virtual".
- El bytecode es multiplataforma.
- Necesito una MV para cada plataforma.
- No necesito el código fuente.



Ejemplos: java, c#, ...