

## UT1 5.- DESARROLLO DE SOFTWARE – EL PROCESO DE TRADUCCIÓN/COMPILACIÓN

RESULTADOS DE APRENDIZAJE ASOCIADOS
1.- Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.
CRITERIOS DE EVALUACIÓN
a) Se ha reconocido la relación de los programas con los componentes del sistema informático: memoria, procesador, periféricos, entre otros.
b) Se han identificado las fases de desarrollo de una aplicación informática.
c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable.
d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.
e) Se han clasificado los lenguajes de programación.
f) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación.

## UT1\_5.- DESARROLLO DE SOFTWARE – EL PROCESO DE TRADUCCIÓN/COMPILACIÓN

### Índice de contenido

1.- El proceso de traducción/compilación .....	3
2.- Máquinas virtuales .....	7
3.- Referencias bibliográficas .....	8

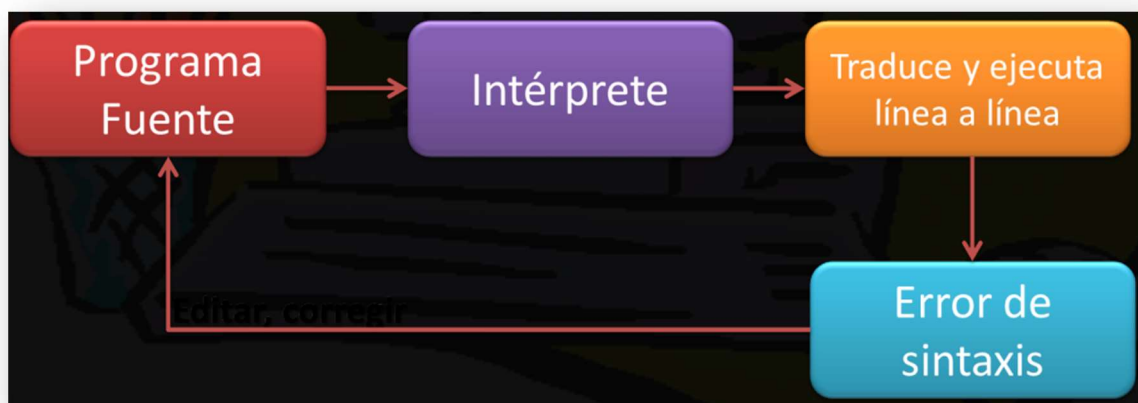
## 1.- El proceso de traducción/compilación



Los **traductores** son programas cuya finalidad es traducir lenguajes de alto nivel (en los que se programa) a lenguajes de bajo nivel como ensamblador o código máquina.

Existen dos grandes grupos de tipos de **traductores**: los compiladores y los intérpretes.

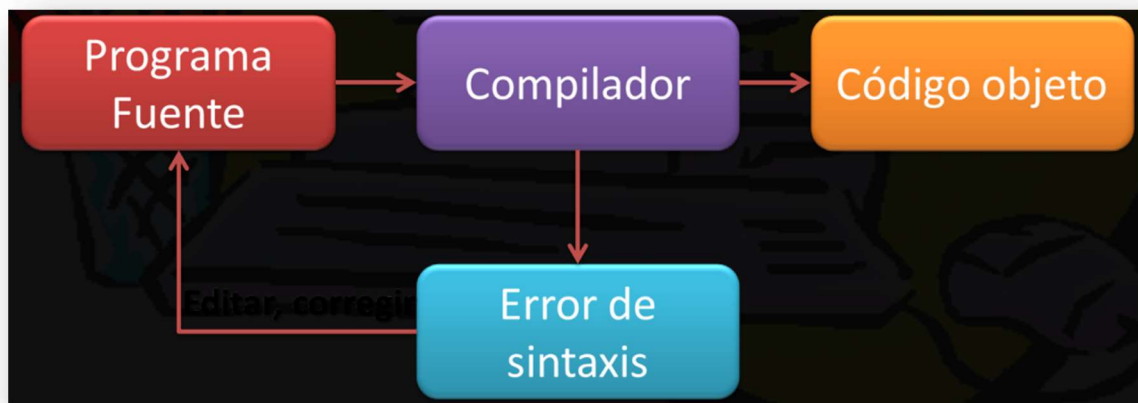
- Un **intérprete** se encarga de traducir y ejecutar el código fuente línea por línea, siguiendo este proceso:
  1. Comienza traduciendo y ejecutando la primera línea del código.
  2. Luego, pasa a la siguiente línea, traduciéndola y ejecutándola antes de avanzar.
  3. Este proceso se repite secuencialmente para todas las líneas del código.



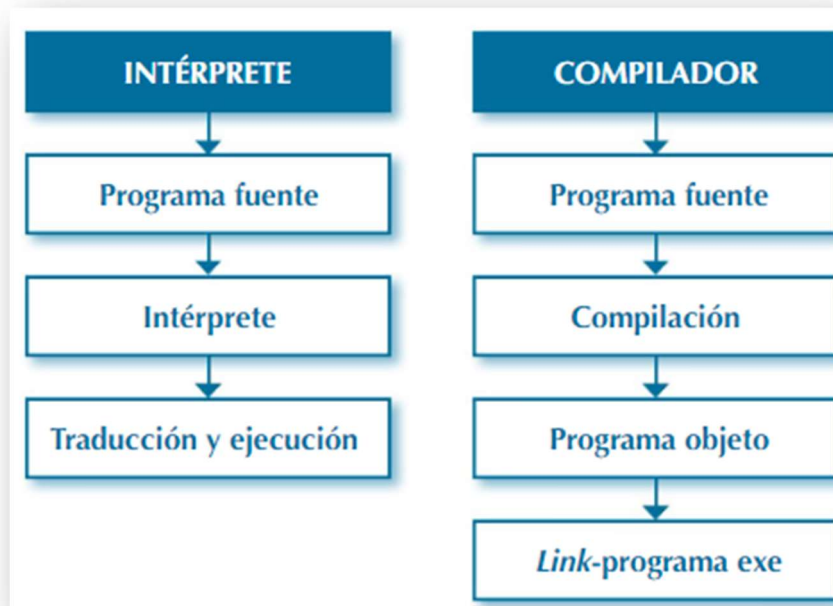


Es importante destacar que el intérprete debe estar en ejecución en la memoria para poder llevar a cabo la ejecución del programa. De manera similar, el propio código fuente también debe estar presente en la memoria para que el intérprete pueda acceder a él y realizar la traducción y ejecución línea por línea.

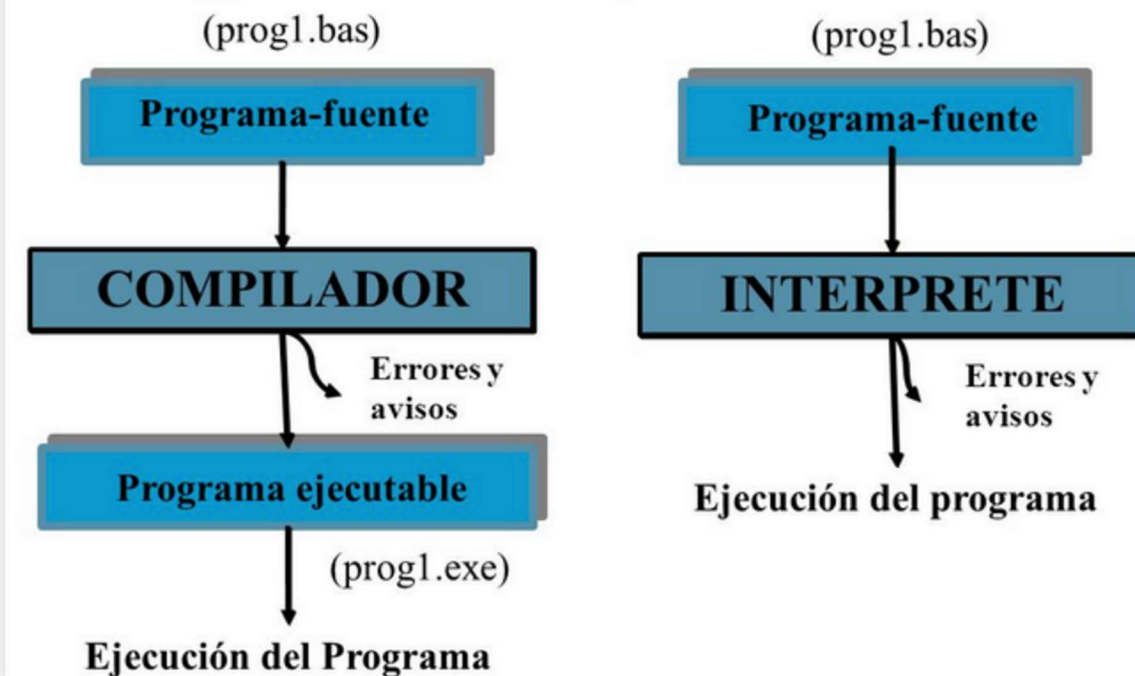
- Un **compilador** realiza la traducción del código fuente a código máquina. Es importante destacar que el compilador está presente exclusivamente en la máquina de desarrollo. El código resultante de esta traducción está altamente vinculado al hardware y software específicos de dicha máquina.



Si se producen cambios en el hardware o el software, se hace necesario llevar a cabo una nueva compilación del código para adaptarlo a las nuevas condiciones.



## Compilador e Interpretador





### Resumen:

Es fundamental comprender de manera clara las diferencias entre los siguientes conceptos:

- **Código fuente:** Se trata del código escrito en un lenguaje de programación específico.
- **Código objeto:** Este representa el resultado de la compilación del código fuente. Puede tomar la forma de código máquina directamente ejecutable o *bytecode*, si se trata de un lenguaje interpretado que luego se ejecuta en una Máquina Virtual (VM).
- **Código ejecutable:** Este es el resultado final tras compilar y enlazar el código con las librerías necesarias. Es un programa completamente funcional que puede ser ejecutado directamente en una máquina concreta sin necesidad de procesamiento adicional.

## ACTIVIDAD 5.1

Elabora una lista que incluya al menos 3 lenguajes de programación interpretados, junto con al menos otros 3 que sean compilados, generando código objeto.

## ACTIVIDAD 5.2

Cita las ventajas y desventajas:

- Lenguajes compilados
- Lenguajes interpretados

## 2.- Máquinas virtuales

Se refiere a un entorno de ejecución virtualizado en el que se puede ejecutar código de programa.

- En el desarrollo de un programa informático, a menudo se utiliza un lenguaje de programación de alto nivel para escribir el código fuente del programa. Este código fuente se traduce a un lenguaje de nivel más bajo o a un código intermedio que no es específico de una arquitectura de hardware particular.

**Este código intermedio puede ser ejecutado por una máquina virtual que simula un entorno de hardware específico.**

- La generación de código intermedio permite que el mismo programa se ejecute en diferentes plataformas o arquitecturas de hardware sin necesidad de reescribir el código fuente para cada una de ellas. Esto se logra mediante la utilización de una máquina virtual que interpreta o compila el código intermedio en código ejecutable para la plataforma de destino.

Ejemplos de máquinas virtuales:

- Máquina Virtual Java (JVM) para programas escritos en Java
- Common Language Runtime (CLR) de Microsoft para programas escritos en C#.

En resumen, una máquina virtual en este contexto se refiere a un entorno de ejecución que permite la ejecución de código de programa en un formato intermedio que es **independiente de la plataforma de hardware subyacente**.



**Esto facilita la portabilidad y la ejecución en diferentes sistemas informáticos.**



Investiguemos un poco...

¿Qué es el Bytecode?

### 3.- Referencias bibliográficas

- ❖ Moreno Pérez, J.C. *Entornos de desarrollo*. Editorial Síntesis.
- ❖ Ramos Martín, A. & Ramos Martín, M.J. *Entornos de desarrollo*. Grupo editorial Garceta.