

# Utilidades para el desarrollo y pruebas de programas. Compiladores. Interpretes. Depuradores

---

- [Utilidades para el desarrollo y pruebas de programas. Compiladores. Interpretes. Depuradores](#)
- [Introducción](#)
  - [Compiladores](#)
    - [Ventajas](#)
    - [Tipos de compiladores](#)
    - [Etapas del proceso de traducción](#)
    - [Elementos importantes en las etapas](#)
    - [Módulos externos al compilador](#)
  - [3. Intérpretes](#)
    - [Estructura de un intérprete](#)
    - [Tipos de intérpretes](#)
  - [4. Depuradores](#)
    - [Herramientas de depuración](#)
    - [Tipos de depuradores](#)
  - [5. Entornos de desarrollo](#)
    - [Componentes de un IDE](#)
  - [6. Bibliografía](#)

## Introducción

---

- Lenguaje ordenadores formado por nistrucciones codificadas en código binario
- Lenguajes de programación formado por símbolos y reglas que permiten escribir programas
- Creados para facilitar la tarea de programación
- Compuestos por un léxico, sintaxis y semántica
- Necesidad de traducir de un programa escrito en un lenguaje fuente a un lenguaje máquina.

Tipos de traductores:

- Compiladores
- Intérpretes
- Según lenguajes de programación uno u otro
- C++ compilado
- Java compilado e interpretado
- SQL, interpretado

## Compiladores

- Convierte un programa de un lenguaje a otro, informando de posibles errores y generando los resultados a partir de datos de entrada.
- A partir de un lenguaje fuente se crea un programa objeto ejecutable

## Ventajas

- Solo una compilación y el código objeto se puede ejecutar muchas veces sin necesitar de recompilar
- Si se encuentra un error no se genera un programa objeto
- Programas objeto codificados, por lo que son "confidenciales" al no mostrar el código
- Mayor eficiencia que los lenguajes interpretados
- Consumen mayor cantidad de memoria
- Los archivos compilados solo serán ejecutables en el tipo de máquina para el que han sido compilados

## Tipos de compiladores

- De una o varias pasadas
- Compiladores incrementales
- Compiladores cruzados

## Etapas del proceso de traducción

El proceso de traducción consta de dos etapas principales: análisis y síntesis.

La **etapa de análisis** incluye un análisis léxico, un análisis sintáctico mediante parsers, y un análisis semántico.

Por su parte, la **etapa de síntesis** comprende la generación de un código intermedio, la optimización del mismo y la generación de código.

## Elementos importantes en las etapas

- Tabla de símbolos: Estructura de datos que guarda información generada durante el proceso de compilación
- Manejador de errores: decide acciones a realizar al encontrar un error en el proceso de compilación
- Gestión de errores
- Errores léxicos, semánticos o sintácticos
- Errores de compilación
- Errores de ejecución

## Módulos externos al compilador

- Preprocesador: por ejemplo incluir código de librerías
- Ensamblador: traduce a código máquina
- Enlazador: crea ejecutables a partir de código objeto y librerías

## 3. Interpretes

### Estructura de un intérprete

- Traductor a representación interna
- Tabla de símbolos
- Evaluador de representación interna
- Tratamiento de errores

### Tipos de intérpretes

- Iterativos o recursivos

- Puros, avanzados, incrementales

## 4. Depuradores

Los **depuradores** son herramientas esenciales para localizar errores durante la ejecución de un programa. Estos permiten detener la ejecución temporalmente y observar la situación de las variables, entre otras cosas. De esta forma, los errores que no se detectaron durante la compilación pueden ser identificados y corregidos.

### Herramientas de depuración

Herramientas que ofrecen al programador:

- Puntos de parada o breakpoints
- Ejecución paso a paso
- Ventana de seguimiento de variables
- Modificar valores de variables o añadir instrucciones nuevas

### Tipos de depuradores

- Dentro del entorno de programación
- Independientes del compilador

## 5. Entornos de desarrollo

- Combinación de herramientas para automatizar y simplificar desarrollo de software.
- Etapas: escribir código fuente. Compilar. Enlazar código y librerías, y depurar
- Antiguamente con herramientas diferentes para cada proceso. Un IDE combina las herramientas conjuntamente
- Editor comprende sintaxis y marca partes del código y reconoce errores
- El editor permite situarse en los diferentes puntos del programa al depurarlo

### Componentes de un IDE

- Componentes: Editor de textos, compilador, intérprete, depurador
- Componentes adicionales:
  - Ejecución automática de pruebas
  - Control de versiones
  - Generador de documentación
  - Optimización de código

## 6. Bibliografía