

## Intro. a la Teoría de la Computación - COMPILADORES

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

## SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES

- **CONSIDERACIONES GENERALES**
  - Proceso de desarrollo de software
    - Especificación de requerimientos, Análisis de sistemas, Diseño de sistemas, **Programación**, Verificación, Integración, validación y mantenimiento, que pueden variar según la metodología
  - **Ingeniería de software:** respuesta a la crisis de la programación
  - Generaciones de los lenguajes de programación
    - 1ª Binario
    - 2ª Ensamblador
    - **3ª Lenguajes de alto nivel ⇒ COMPILADOR**
    - 4ª Lenguajes de muy alto nivel
    - 5ª Generación y más allá



## **Intro. a la Teoría de la Computación - COMPILADORES**

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

## **SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES**

- **CONCEPTOS RELACIONADOS**
  - Editores de texto
  - Compilación
  - Una y varias pasadas
  - Preprocesador
  - Intérprete
  - Conversor fuente-fuente
  - Ensamblador
  - Administrador de librerías
  - Depurador
  - Enlazador
  - Librerías de enlace dinámico



## Intro. a la Teoría de la Computación - COMPILADORES

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

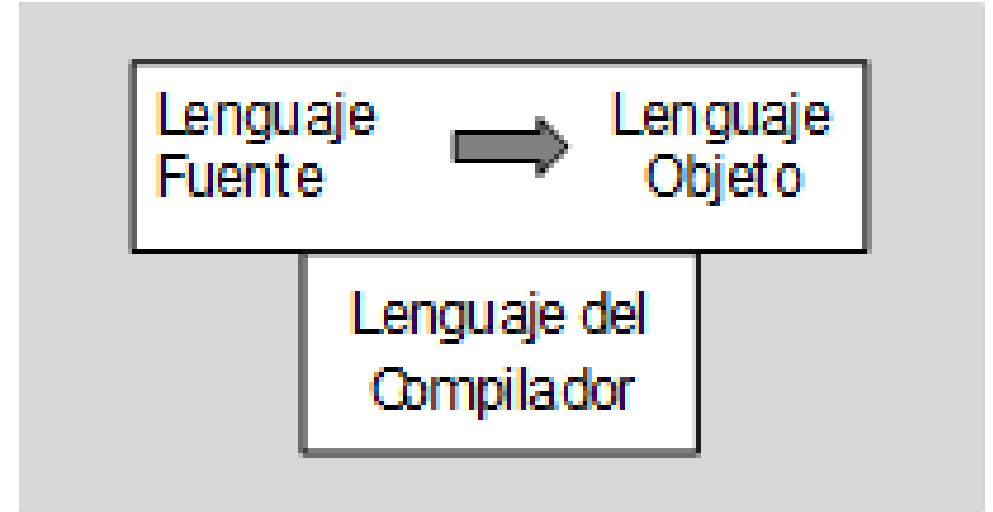
## SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES

### • TIPOS DE COMPILADORES

- Compilador cruzado
- Autocompilador
- Metacompilador
- Decompilador
- Compilador optimizador
- Compilador gráfico
- Compilador intérprete
- Ambiente integrado de desarrollo (IDE)

### • COMPILADORES E INTÉRPRETES

### • NOTACIÓN T

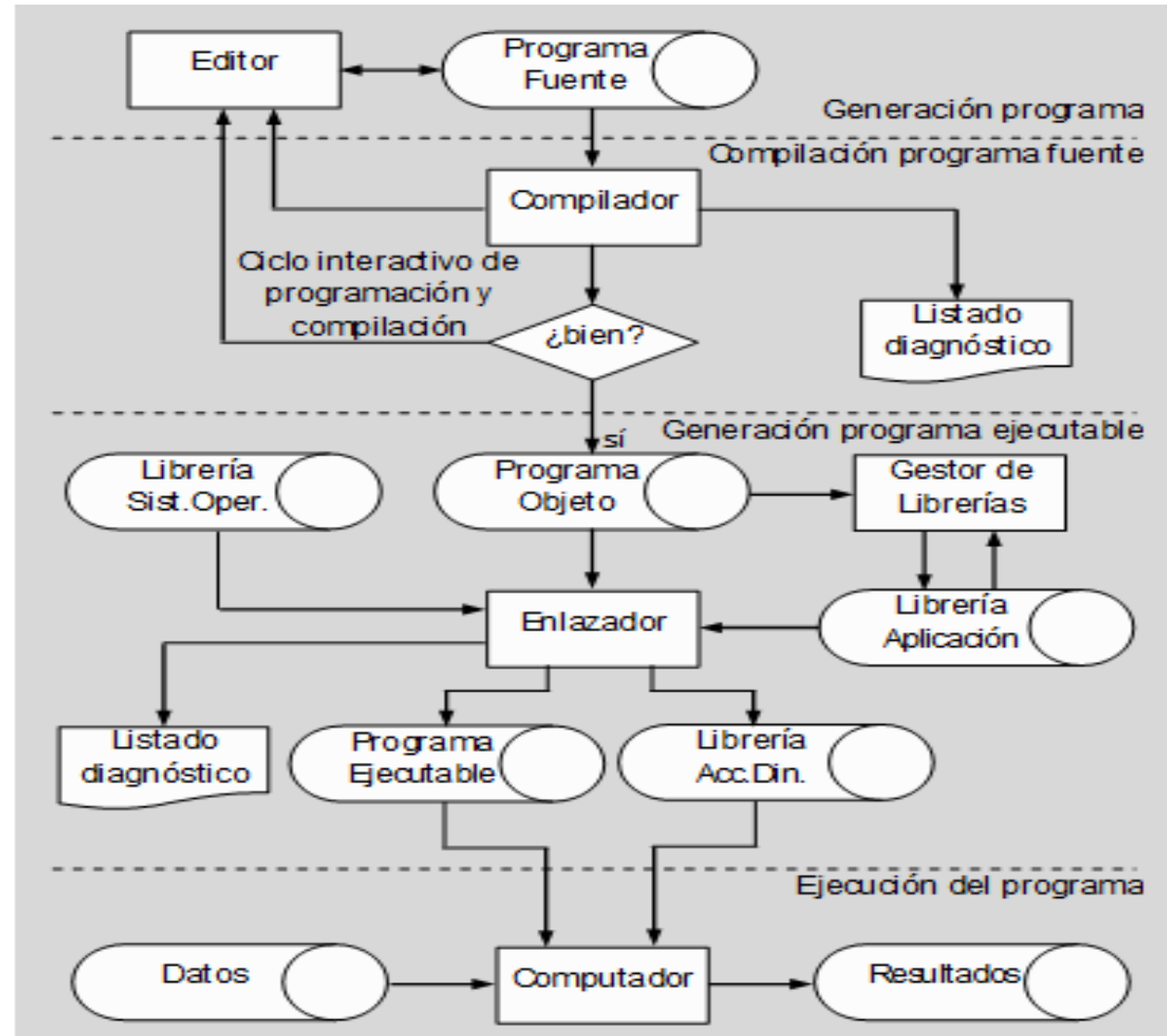


## Intro. a la Teoría de la Computación - COMPILADORES

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

### • CONTEXTO DE UN COMPILADOR

## SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES



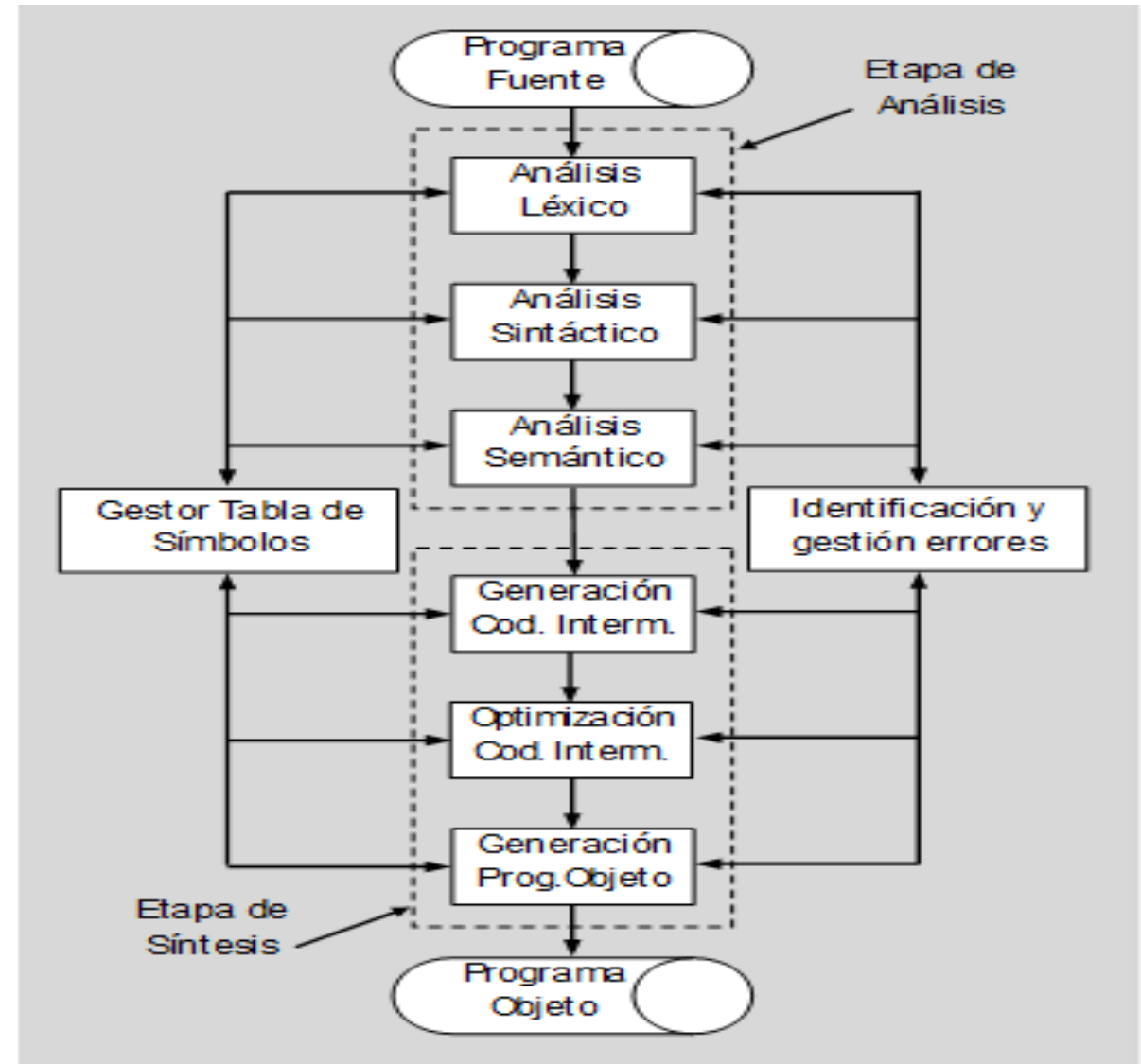
## Intro. a la Teoría de la Computación - COMPILADORES

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

### • ESTRUCTURA DE UN COMPILADOR

- Etapa de Análisis
- Etapa de Síntesis
- Errores
- Información de soporte:  
Tabla de Símbolos

## SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES



## Intro. a la Teoría de la Computación - COMPILADORES

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

## SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES

### • ESTRUCTURA DE UN COMPILADOR

#### • Etapa de Análisis

GR / AF

GIC / AP  
ASD/ASA

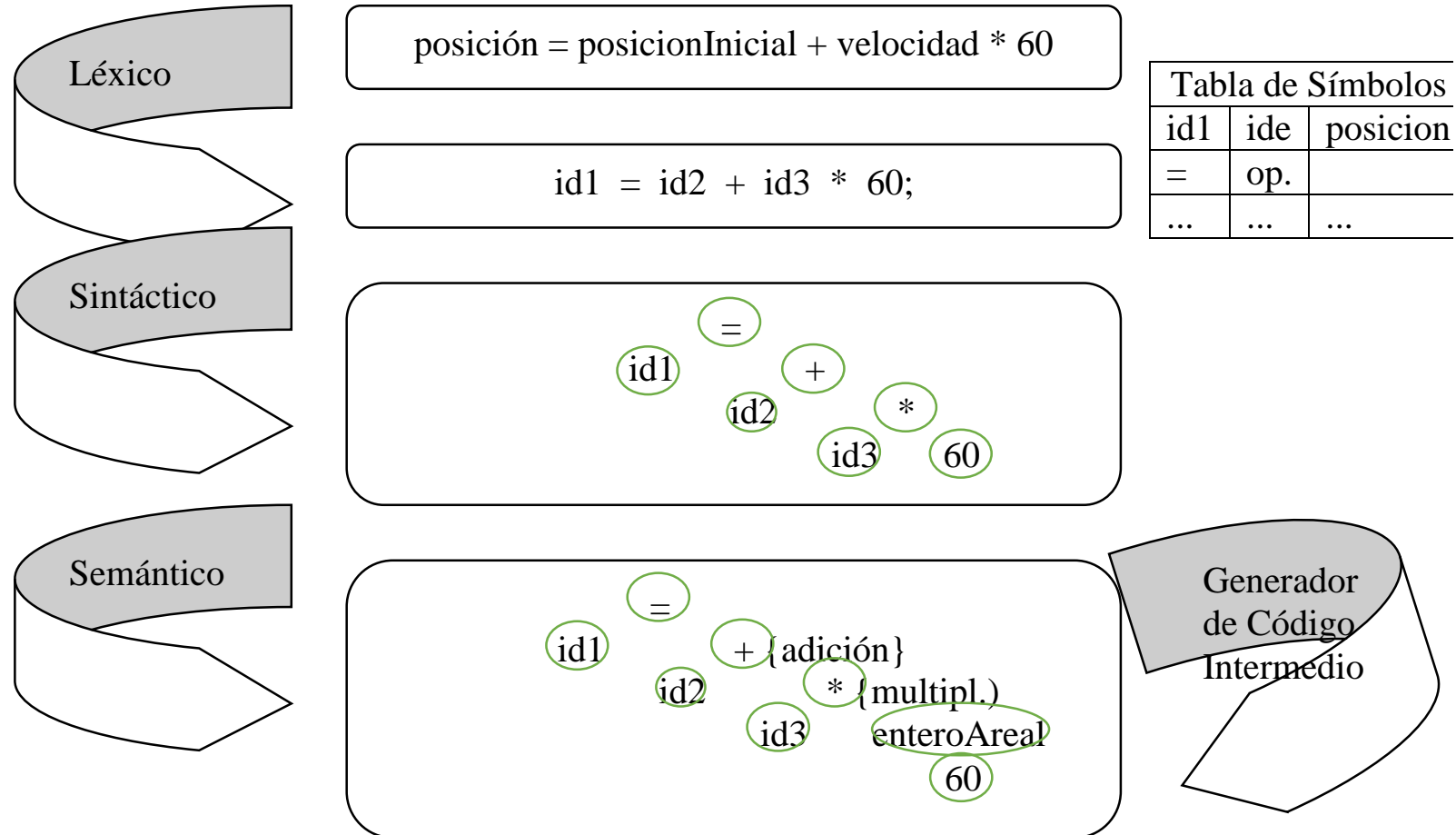
- Léxico: separa el programa fuente en componentes léxicos.
  - Patrón, lexema, componente léxico o token.
  - Identificadores, operadores, constantes, puntuación, etc.
- Sintáctico: estudio de la estructura del programa fuente.
  - Verifica que se ajuste a la gramática del lenguaje fuente.
- Semántico: asigna significado y verifica todo lo demás.
  - Tipos, unicidad, flujo de control, coherencia, etc.
- Etapa de Síntesis
  - Generación de código intermedio (tres direcciones, UNCOL, etc.)
  - Optimización (tiempo, memoria, especialización a un hardware)
  - Generación de código objeto (específico para un hardware)
- Administrador de la tabla de símbolos y gestor de errores

## Intro. a la Teoría de la Computación - COMPILADORES

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

## SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES

### • ESTRUCTURA DE UN COMPILADOR – Traducción de una sentencia

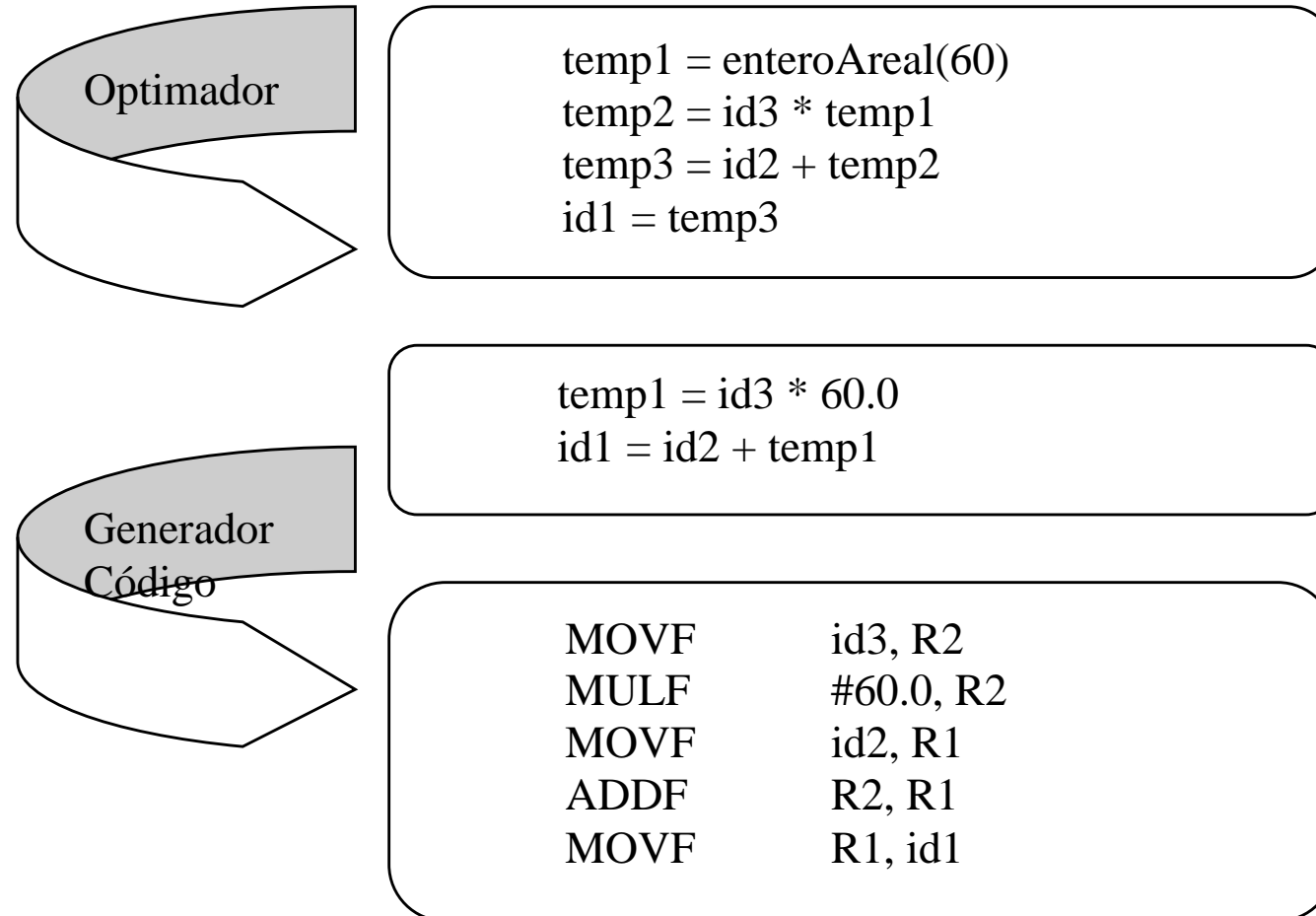


## Intro. a la Teoría de la Computación - COMPILADORES

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

## SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES

### • ESTRUCTURA DE UN COMPILADOR – Traducción de una sentencia





## Intro. a la Teoría de la Computación - COMPILADORES

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

## SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES

- **ERRORES DE PROGRAMACIÓN**
  - **Verificación**: producto construido correctamente, sin errores, y
  - **Validación**: producto correcto según especificaciones requeridas
  - Para verificar, el compilador tiene que informar clara y oportunamente los errores detectados. IDE y depuradores ayudan.
  - Clases de errores:
    - **Léxicos**: símbolos incorrectos en lugares incorrectos.
    - **Sintácticos**: estructura del programa no respeta las reglas.
    - **Semánticos**: errores de tipos, argumentos de funciones, etc.
    - **Por falla del compilador**: excede los límites del compilador.
    - **En tiempo de ejecución**: lógica, periféricos, imprevisiones, etc.
  - Verificación **estática** y verificación **dinámica**.
  - Permiten detectar errores pero nunca aseguran la ausencia de ellos.
  - Gestión y recuperación de errores.

## Intro. a la Teoría de la Computación - COMPILADORES

- Consideraciones generales
- Conceptos relacionados
- Tipos de compiladores
- Compiladores e intérpretes
- Notación T
- Contexto de un compilador
- Estructura de un compilador
- Errores de programación
- Evolución y tendencias

## SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES

### • **EVOLUCIÓN Y TENDENCIAS**

- Los lenguajes evolucionan por necesidades del mercado y de los usuarios, por avances tecnológicos y en las ciencias informáticas.
- En general, se intenta:
  - Mejorar la eficiencia y aumentar la productividad de los programadores.
  - Aumentar la calidad de los productos de software.
  - Facilitar el mantenimiento y la actualización de sistemas.
- Cada vez más, se diseñan lenguajes de propósito específico.
- Se avanza hacia los lenguajes dependientes del contexto, aún en fase experimental y no comerciales.
- Todas las técnicas y herramientas diseñadas para la construcción de compiladores pueden usarse con provecho en el desarrollo de software convencional (en general, en sistemas de estado finito).

