



PROCESADORES DE LENGUAJES

Tema 1: Introducción

ÍNDICE

- Definición de compilador
- Tipos de compiladores
- Historia de los compiladores
- Estructura de un compilador
- Especificación de compiladores
- Aplicaciones de los compiladores

DEFINICIÓN DE COMPILADOR

- Traductor: Programa que toma como entrada un texto escrito (llamado fuente) y da como salida otro texto (llamado objeto).
- Compilador: Traductor cuyo fuente es un lenguaje de alto nivel y cuyo objeto es un lenguaje de bajo nivel.
- Interprete: Compilador que ejecuta al mismo tiempo que traduce.

DEFINICIÓN DE COMPILADOR

- Ventajas de los compiladores frente a los interpretes
 - Solo se compilan una vez
 - Ejecución más rápida
 - Optimización de código e Información del error más detallada.

DEFINICIÓN DE COMPILADOR

- Ventajas de los Interpretes frente a los Compiladores
 - Menor coste espacial
 - Mayor interactividad en desarrollo
 - Añadir código “en caliente”

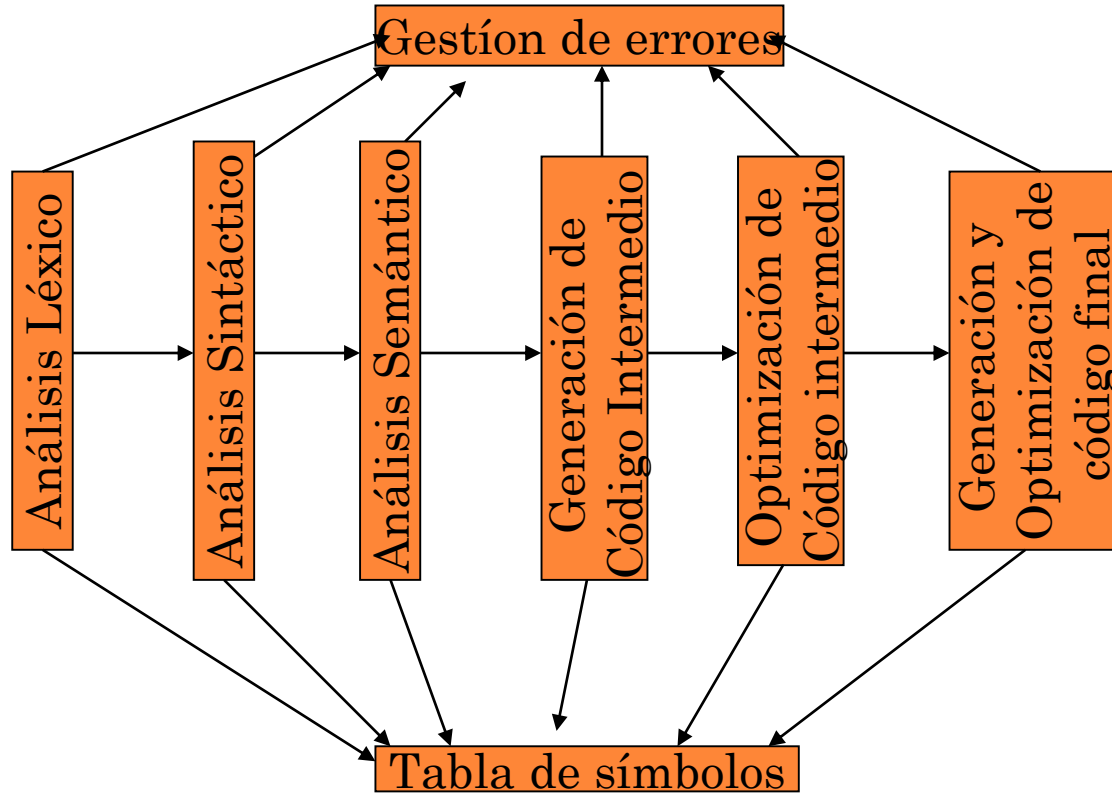
TIPOS DE COMPILADORES

- Ensamblador
- Compilador cruzado
- Compilador con montador
- Autocompilador
- Metacompilador
- Decompilador

HISTORIA DE LOS COMPILADORES

- ¿Qué fue primero el compilador o el programa?
- FORTRAN: Primer lenguaje de alto nivel
- ALGOL: Lenguaje estructurado en módulos
- Separación Front-End y Back-End para compilar lenguajes en distintas máquinas
- Creación del lenguaje intermedio UNCOL
- Definición de las distintas fases de creación de compiladores

ESTRUCTURA DE UN COMPILADOR



ESTRUCTURA DE UN COMPILADOR

- **Análisis sintáctico**
 - Comprueba los valores unitarios del programa (tokens):
 - Palabras reservadas(do, while, if, ...)
 - Palabras no específicas(identificadores, constantes)
 - Formados por tipo y lexema

Int contador

Identificador = tipo

“contador” = lexema

ESTRUCTURA DE UN COMPILADOR

○ Análisis Sintáctico

- Recibe los tokens del analizador léxico y comprueba que están ordenados conforme a la gramática

ESTRUCTURA DE UN COMPILADOR

- Análisis Semántico
 - Comprueba la validez del programa
 - Comprobación de tipos en operadores

ESTRUCTURA DE UN COMPILADOR

- Generación de código intermedio
 - Codifica el lenguaje fuente en un lenguaje intermedio entre el lenguaje objeto
 - Permite la separación entre front-end y back-end
 - Utilización de código de 3 direcciones ($A=B+C$)
 - Instrucciones condicionales y saltos

ESTRUCTURA DE UN COMPILADOR

- Optimización de código intermedio
 - Permite realizar mejoras en el código intermedio
 - Eliminación de saltos consecutivos
 - Factorizaciones
 - Eliminar código inútil
 - Optimización de bucles

ESTRUCTURA DE UN COMPILADOR

- Generación de código objeto
 - Generar una secuencia de instrucciones en código ensamblador o máquina a partir del código intermedio
 - $A=B+C$
 - LOAD B
 - ADD C
 - STORE A

ESTRUCTURA DE UN COMPILADOR

- Tabla de símbolos
 - Tabla donde se registran los identificadores, constantes, funciones y otros objetos especificados en el programa fuente
 - El compilador desarrolla funciones comunes de acceso a ella

ESTRUCTURA DE UN COMPILADOR

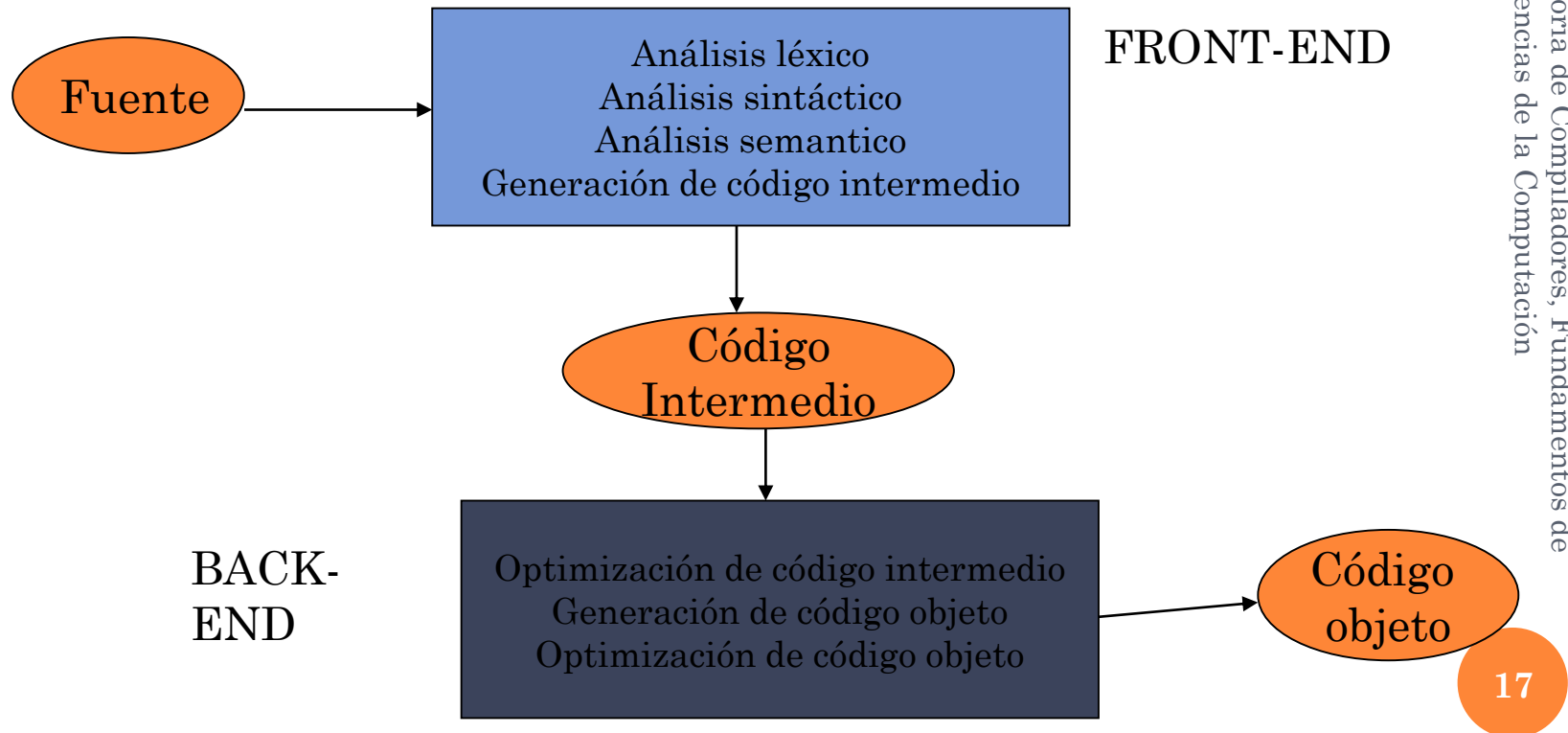
○ Manejo de errores

- Mensajes de error que el compilador emite cuando encuentra una inconsistencia en cualquiera de las fases de compilación
- Normalmente en una compilación no se muestran todos los errores producidos:
 - Ocultación de otros errores
 - Avalancha de errores
- Se puede
 - Pararse al encontrar cualquier error
 - Intentar recuperar todos los errores de una pasada

ESTRUCTURA DE UN COMPILADOR

05/08/2019

Teoría de Compiladores, Fundamentos de
Ciencias de la Computación



ESPECIFICACIÓN DE UN COMPILADOR

- Especificación léxica:
 - Tokens mediante expresiones regulares
- Especificación sintáctica:
 - Gramáticas independientes del contexto
- Especificación semántica:
 - Lenguaje natural

APLICACIONES DE LAS TÉCNICAS DE COMPILACIÓN

- Interfaces textuales
- Manejo de ficheros de texto estructurado
- Procesadores de texto
- Diseño e interpretación de lenguajes
- Gestión de bases de datos
- Procesamiento del lenguaje natural
- Traducción de formatos de ficheros
- Cálculo simbólico
- Reconocimiento sintáctico de formas