

# Tema 1: Introducción

## Procesamiento de Lenguajes

Dept. de Lenguajes y Sistemas Informáticos  
Universidad de Alicante



## Definición de compilador

- Un compilador es un programa que traduce de un *lenguaje fuente* (normalmente un lenguaje de programación de alto nivel) a un *lenguaje objeto* (normalmente código máquina)
- **IMPORTANTE:** un compilador es un *traductor*
- Un IDE (Integrated Development Environment) no es un compilador, un IDE suele tener un buen editor, y llama al compilador y enlazador, y suele integrar un depurador.

# Tipos de procesadores de lenguaje

	LENGUAJE FUENTE	LENGUAJE OBJETO
traductor/conversor	alto	alto
compilador	alto	bajo
ensamblador	medio/bajo	bajo
preprocesador	alto	alto
intérprete	alto	medio (*)

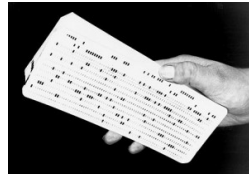
(\*) la mayoría de los intérpretes actuales precompilan el programa fuente y generan un código intermedio

## ¿Qué vamos a estudiar?

- Las técnicas y algoritmos que vamos a estudiar se aplican al diseño de compiladores, pero se pueden aplicar en otras tareas que requieran el procesamiento y/o traducción de lenguajes informáticos:
  - ▶ Traductores automáticos de lenguajes humanos (ejemplo: interNOSTRUM, Apertium, ...)
  - ▶ Visores de PDF (evince, acroread, ...)
  - ▶ Navegadores web (Firefox, Chrome, ...)
  - ▶ Ficheros de configuración (.ini ...)
  - ▶ ...

# Historia de los compiladores

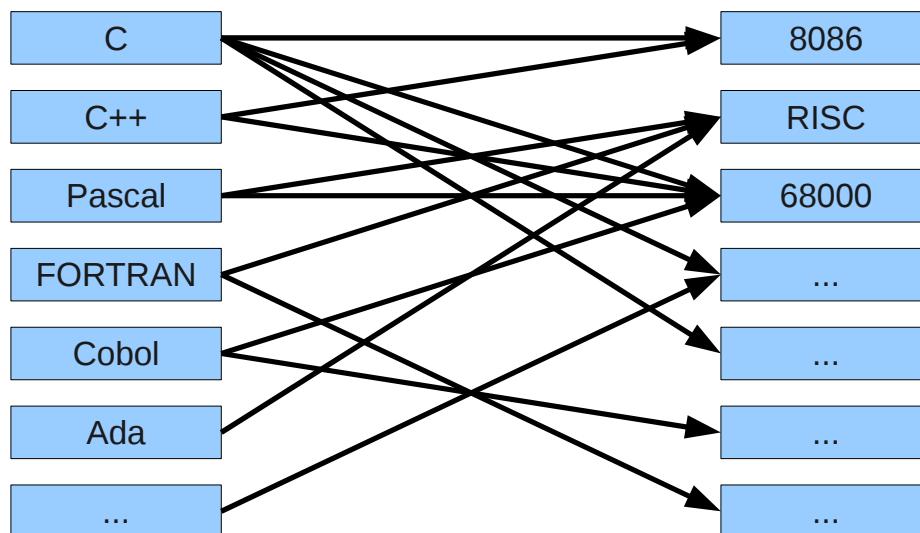
- Los primeros compiladores eran humanos:



<http://eltamiz.com/elcedazo/2009/02/17/historia-de-un-viejo-informatico-el-metodo-de-trabajo-en-proceso-de-datos-en-la-decada-de-los-setenta/>

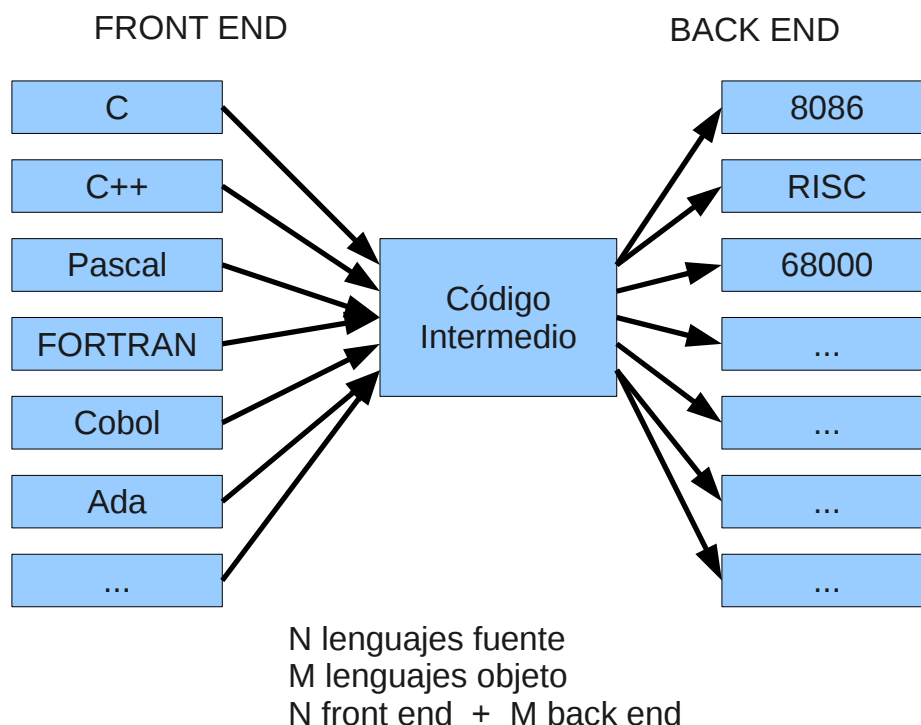
- Los primeros compiladores no humanos eran compiladores cruzados (*cross-compilers*): en una máquina se generaba código para ser ejecutado en otra máquina.
- Cuando se diseñaba una nueva arquitectura, lo primero que se hacía era diseñar un *cross-compiler* en una arquitectura conocida, para poder compilar el compilador nativo en la nueva arquitectura.

## Diseño de compiladores

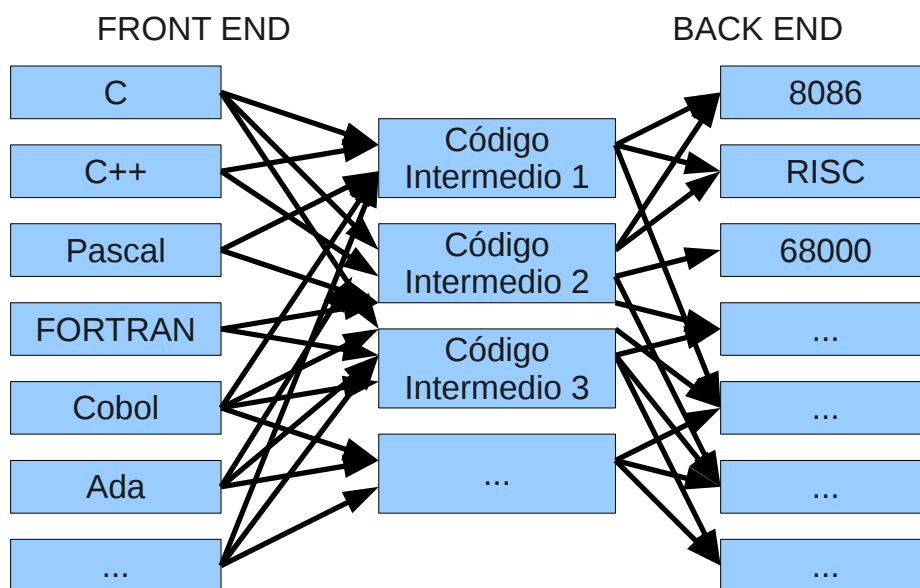


N lenguajes fuente  
M lenguajes objeto  
NxM compiladores

# Código intermedio



## Código intermedio (mundo real)

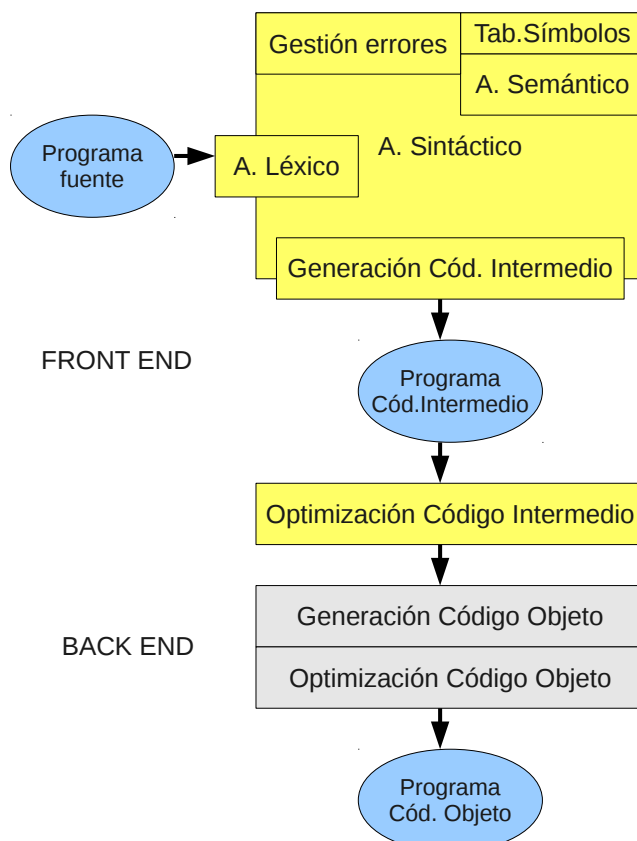


# Fases de un compilador

- Internamente, un compilador se organiza en varias fases o partes que interaccionan entre sí:

- 1 Análisis léxico
- 2 Análisis sintáctico
- 3 Análisis semántico
- 4 Gestión de errores
- 5 Tabla de símbolos
- 6 Generación de código intermedio
- 7 Optimización de código intermedio
- 8 Generación de código objeto
- 9 Optimización de código objeto

## Estructura estándar de un compilador



# ¿Cómo se especifica un traductor/compilador?

- Especificación léxica: expresiones regulares, AFD, diagramas de transiciones
- Especificación sintáctica: gramáticas independientes del contexto
- Especificación semántica: sistema de tipos, restricciones semánticas
- Especificación de la traducción: gramáticas de atributos, DDS, ETDS
- ...

## Ejemplo de traducción

$2+35*4$	<code>suma(2,prod(35,4))</code>
$2-3/(4+5*6)$	<code>resta(2,div(3,suma(4,prod(5,6))))</code>
$2-3+4$	<code>suma(resta(2,3),4)</code>
$2-3+4-5$	<code>resta(suma(resta(2,3),4),5)</code>

Hay dos formas de resolver este problema:

- 1 *A mano (difícil)*
- 2 Utilizando técnicas para el diseño de compiladores: análisis léxico, análisis sintáctico, **traducción dirigida por la sintaxis (chupao)**