
PROCESADORES DE LENGUAJES

Laura Climent -- laura.climent@uam.es

¿QUÉ ES UN PROCESADOR DE LENGUAJES?

Un procesador para un lenguaje de programación es un sistema capaz de manipular programas escritos en dicho lenguaje con el objeto de ejecutarlo o prepararlo para su ejecución.

```
C:\FP\Unidad02>g++ -o hola.exe hola.cpp
hola.cpp: In function 'int main()':
hola.cpp:4: error: 'cout' was not declared in this scope
hola.cpp:4: error: 'endl' was not declared in this scope
C:\FP\Unidad02>_
```

ESTRUCTURA DE LA ASIGNATURA

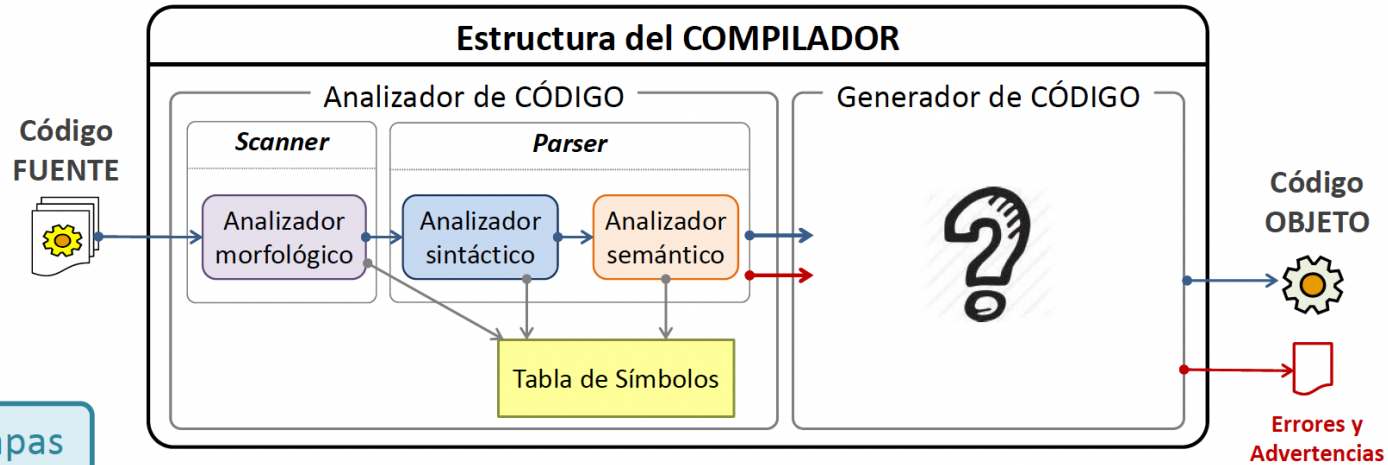
Asignatura **totalmente práctica**, en la que el alumno programará un compilador (**80% de la nota**). El examen (**20% de la nota**), consistirá en una pequeña modificación de su compilador.

No habrá preguntas teóricas. Asignatura recomendada para alumnos a los que les guste mucho programar

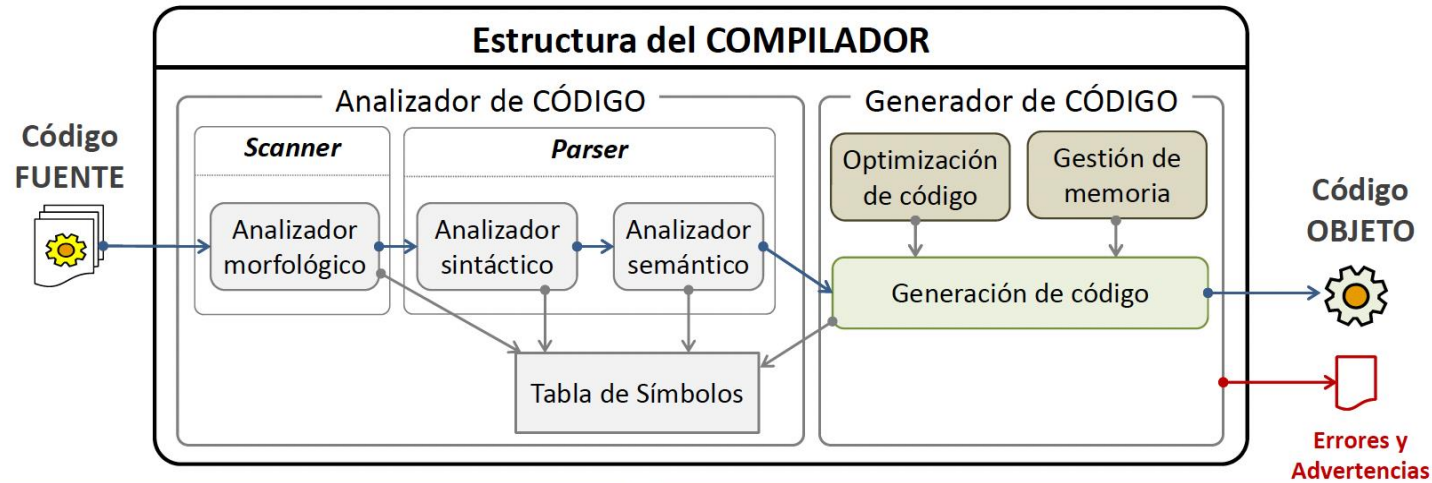
```
C:\FP\Unidad02>g++ -o hola.exe hola.cpp
hola.cpp: In function 'int main()':
hola.cpp:4: error: 'cout' was not declared in this scope
hola.cpp:4: error: 'endl' was not declared in this scope

C:\FP\Unidad02>_
```

Etapas



- 1 **Analizador morfológico:** transforma el código fuente del programa de una secuencia de caracteres, a una secuencia de unidades sintácticas (tokens).
 - 2 **Analizador sintáctico:** interpreta las unidades sintácticas identificadas (tokens) por el analizador morfológico como un programa estructurado según una gramática (ya se verá en Autómatas y Lenguajes).
 - 3 **Analizador semántico:** realiza comprobaciones semánticas como por ejemplo que las variables están declaradas antes de su uso, que los tipos de las expresiones son correctos, etc.
-
- T** **Tabla de símbolos o identificadores.** Se encarga de todos los aspectos dependientes del contexto relacionados con los *nombres* (variables, constantes, funciones, palabras reservadas, etc.) que puedan aparecer en los programas.



Generación de código

- Traduce el *código fuente* al *código objeto* utilizando toda la información proporcionada por las restantes partes del compilador.
- Antes de generar el código objeto, se puede crear el correspondiente código en **lenguajes de simbólicos** (lenguaje ensamblador NASM).
- En esta tarea se puede acceder a la tabla de símbolos si es necesario.

Gestión de memoria y optimización de código:

- Mejora la eficiencia del programa objeto en ocupación de memoria o en tiempo de ejecución.
- La optimización de código es la fase cuyo objetivo consiste en modificar el código objeto generado por el generador de código, para mejorar su rendimiento. Esta fase puede realizarse, bien en un paso independiente, posterior a la generación de código, o bien mientras éste se genera.



Un compilador puede tener distintos módulos de generación de código para generar código para distintas plataformas.