

## *Tutorial de construcción de un compilador con ANTLR*

### Contenido

Objetivo .....	2
Estructura del Tutorial.....	2
Descripción del Lenguaje del Tutorial .....	2
División en Capítulos .....	2
Estructura de cada Capítulo del Tutorial.....	3
Cómo hacer un capítulo .....	4

Raúl Izquierdo Castanedo  
[raul@uniovi.es](mailto:raul@uniovi.es)

Área de Lenguajes y Sistemas Informáticos  
Departamento de Informática

Escuela de Ingeniería Informática de Oviedo  
Universidad de Oviedo



Para todos los  
componentes incluidos  
con este tutorial

## Objetivo

El objetivo de este tutorial es mostrar el proceso básico de construcción de un compilador de un lenguaje de programación sencillo.

No es objetivo de este tutorial sustituir a las clases de teoría y práctica de la asignatura, sino ser un complemento de las mismas. Por tanto, no se explicarán aquí los metalenguajes ni las herramientas utilizadas. Se aconseja realizar cada capítulo del tutorial después de la clase teórica correspondiente a cada módulo del compilador.

## Estructura del Tutorial

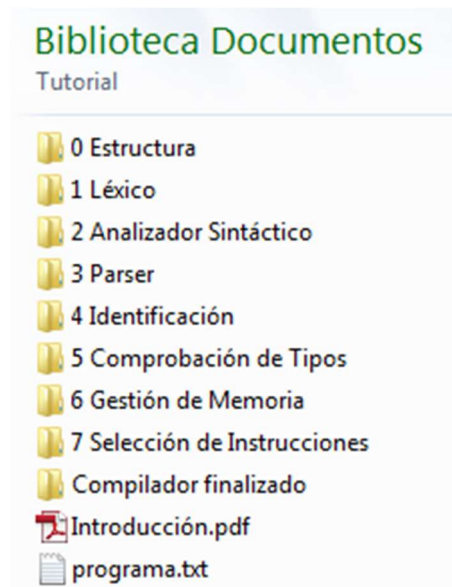
### Descripción del Lenguaje del Tutorial

En la carpeta principal del tutorial, aparte de este, hay dos ficheros más. Uno es el documento *Descripción del Lenguaje.pdf*, donde se describe el lenguaje para el cual se va a construir un compilador en este tutorial. El otro fichero, *programa.txt*, es un programa de ejemplo en dicho lenguaje.

Se aconseja que se miren dichos ficheros *antes de empezar* con los capítulos del tutorial.

### División en Capítulos

En tutorial está repartido en carpetas que deben ser visitadas en el orden indicado. Dichas carpetas son:



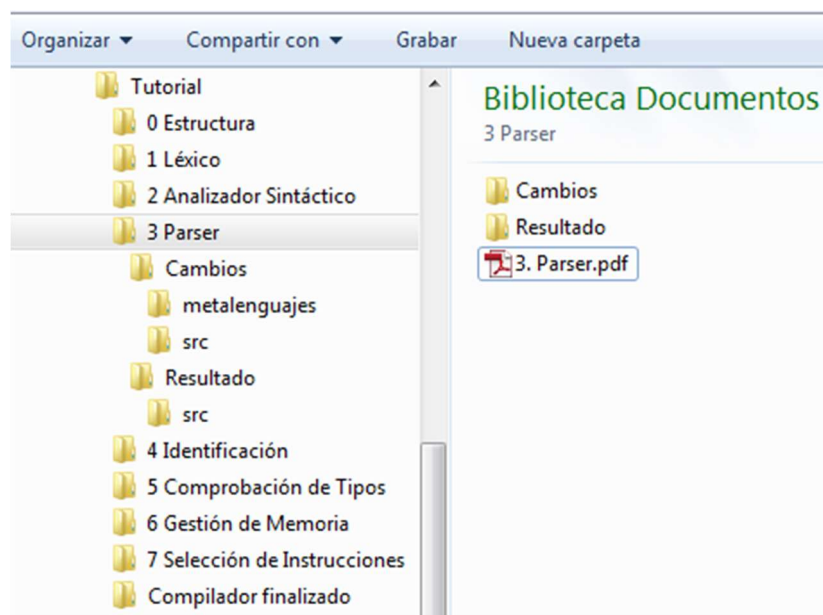
- La carpeta "0 Estructura" contiene el esqueleto de un traductor genérico el cual incluye un método *main*, unas clases auxiliares y unos paquetes (*packages*) ya creados donde ir añadiendo los fuentes que se implementen en el resto de los capítulos. Es el punto de partida del tutorial.
- A continuación, aparece el tutorial propiamente dicho dividido en 7 capítulos (de "1 Léxico" a "7 Selección de Instrucciones"), cada uno de los cuales se corresponde con una de las fases del proceso de construcción. Cada capítulo añade al resultado del

anterior la parte del traductor que le corresponde. La estructura de cada uno de estos capítulos se describe en el apartado siguiente.

- Por último, aparece la carpeta "*Compilador finalizado*" donde se encuentra el código final del traductor listo para ser compilado y ejecutado (por ejemplo, sobre "*programa.txt*").

## Estructura de cada Capítulo del Tutorial

Todas las carpetas del tutorial (numeradas de 1 a 7) tienen la misma estructura. Por ejemplo, en el capítulo 3, se tiene el siguiente contenido:



- Un documento PDF (en este caso "*3. Parser.pdf*") que indica qué se hará en dicho capítulo.
- Una carpeta "*Cambios*" con únicamente aquellos ficheros que se hayan modificado o añadido en esta fase. Es una forma de poder comprobar rápidamente lo que se ha hecho en este capítulo. Esta carpeta tendrá a su vez dos subcarpetas:
  - "*metalenguajes*" con la especificación de esta fase en el metalenguaje correspondiente.
  - "*src*" con los ficheros *Java* modificados o añadidos.
- Una carpeta "*Resultado*" que contiene todo el código del traductor con la fase actual ya implementada (compilable y ejecutable). El contenido de dicha carpeta se ha obtenido en cada capítulo copiando la carpeta "*Resultado*" del capítulo anterior y añadiendo los ficheros de la carpeta "*Cambios*" del capítulo actual<sup>1</sup>.

<sup>1</sup> Es decir, el código que se halla en la carpeta *Resultado* de un capítulo es aquel que se obtiene al añadir al código base del capítulo "*0 Estructura*" el contenido de las carpetas *Cambios* de dicho capítulo y de *todos los anteriores* a él.

## Cómo hacer un capítulo

La forma recomendada de seguir este tutorial es entrar en los capítulos por orden y, en cada uno de ellos, seguir los siguientes pasos:

1. Abrir en el documento PDF del capítulo y leer *solamente* el apartado *Objetivo*. En dicho apartado se indica qué debe hacer el nuevo módulo del compilador sobre el que trata el capítulo.
2. Intentar hacer el diseño y la implementación de dicha fase partiendo de los fuentes de la carpeta *Resultado* de la fase *anterior*. De esta manera se puede implementar cada capítulo partiendo de una base de código estable aunque no se hayan hecho los capítulos anteriores.
3. Leer el resto del PDF del capítulo (diseño de la solución y su implementación) y comparar con la solución propuesta por el alumno. Los ficheros con la solución de esta fase se encuentran en la carpeta "*Cambios*".