

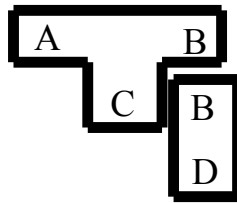
# Anexo II: Plan de la Asignatura para el Alumnado





# PLAN DE LA ASIGNATURA

## COMPILADORES I



Desarrollado en la Universidad de Zaragoza  
entre septiembre de 2006 y febrero de 2007

## Índice de contenidos

|  |           |
|--|-----------|
| <b>1.- Datos de la asignatura.....</b>                 | <b>3</b>  |
| <b>2.- Compiladores I .....</b>                        | <b>3</b>  |
| <b>3.- Objetivos generales.....</b>                    | <b>4</b>  |
| <b>4.- Objetivos específicos.....</b>                  | <b>5</b>  |
| <b>5.- Conocimientos previos.....</b>                  | <b>5</b>  |
| <b>6.- Metodología y calendario inicial .....</b>      | <b>5</b>  |
| <b>7.- Evaluación .....</b>                            | <b>8</b>  |
| <b>8.- Documentación y bibliografía .....</b>          | <b>9</b>  |
| <b>9.- Material informático necesario .....</b>        | <b>10</b> |
| <b>10.- Contenidos temáticos .....</b>                 | <b>10</b> |
| <b>11.- Objetivos específicos de los módulos .....</b> | <b>11</b> |
| Módulo 0: Presentación de la asignatura Compiladores I |           |
| Módulo 1: Introducción a los compiladores              |           |
| Módulo 2: Análisis léxico                              |           |
| Módulo 3: Fundamentos para el análisis sintáctico      |           |
| Módulo 4: Análisis sintáctico descendente              |           |
| Módulo 5: Análisis sintáctico ascendente               |           |
| Módulo 6: Introducción al análisis semántico           |           |
| Módulo 7: Repaso general del curso                     |           |
| <b>12.- Glosario.....</b>                              | <b>12</b> |

## PLAN DE LA ASIGNATURA

Este documento recoge los aspectos generales de la asignatura de *Compiladores I*; como son los objetivos generales y los específicos de cada uno de los ocho módulos que lo forman, la metodología docente, el sistema de evaluación, el temario, aspectos relacionados con las fechas de su impartición y de los entregables así como toda aquella información de interés para aquel estudiante que se quiera matricular en la asignatura y los ya matriculados a efectos de poder seguirlo mejor y disponer de un ejemplo de procedimiento y contenidos.

Es muy importante leer detenidamente este documento antes de dar inicio a las actividades de la parte no presencial del taller.

### 1.- Datos de la asignatura

|                      |  |
|----------------------|--|
| <b>Nombre:</b>       | <i>Compiladores I</i> (12044)  |
| <b>Departamento:</b> | Informática e Ingeniería de Sistemas   |
| <b>Tipo:</b>         | Troncal de segundo ciclo   |
| <b>Profesores:</b>   | Joaquín Ezpeleta, Rafael Tolosana y Raquel Trillo<br><a href="mailto:jguerrer@unizar.es">jguerrer@unizar.es</a> , <a href="mailto:gonlopez@unizar.es">gonlopez@unizar.es</a> |
| <b>Dedicación:</b>   | 3,8 créditos ECTS, 100 horas   |

### 2.- Compiladores I

*Compiladores I* es una asignatura troncal de segundo ciclo que se cursa dentro del plan de estudios de Ingeniería Informática.

Es importante que el alumnado entienda la relevancia de la asignatura y todas las motivaciones que justifican su inclusión como una materia de estudio obligatorio en los planes de estudio de Ingeniería en Informática en España y en otras muchas titulaciones a nivel mundial. Por ello, en las primeras lecciones se incidirá en esta idea.

Resaltar y no perder de vista el hecho de que los principios, técnicas y herramientas que se aprenden en los cursos de diseño de compiladores son aplicables a multitud de problemas. Así, probablemente ningún estudiante tendrá que utilizar en su vida profesional los conocimientos y las habilidades desarrolladas en estos cursos para diseñar un compilador propiamente dicho (entendido como algo que transforma código en un lenguaje de alto nivel a lenguaje ensamblador). Sin embargo, sí se enfrentará a una variedad de situaciones similares donde los conocimientos y habilidades adquiridas son aplicables: traductores de un lenguaje a otro (por ejemplo, de Latex a HTML, de un formato de fichero binario a gráficos, de preguntas en lenguaje natural en un dominio específico a SQL, intérpretes, procesamiento de XML, etc.). Estas generalizaciones se harán patentes en la clase, dada la dificultad que experimentan los estudiantes a la hora de transferir sus conocimientos a otros ámbitos de aplicación similares.

En la asignatura se aborda la construcción de las primeras fases de un compilador, continuándose posteriormente en la asignatura de *Compiladores II*. Un proyecto de

construcción de un compilador es relevante para un currículum básico incluso si el propósito no es el propio estudio de los compiladores. Así, la construcción de un compilador no tiene por que ser el fin, sino un medio a través del cual se consiguen una serie de objetivos de gran importancia para la formación profesional de un ingeniero: se trabaja en un proyecto completo, y se aprende a desarrollar documentación y a depurar y realizar pruebas de funcionamiento. Además el desarrollo del proyecto final implica varias iteraciones, siendo necesario en ocasiones mejorar los resultados de iteraciones previas antes de continuar con las siguientes (se introduce así el concepto de ciclo de vida del software de forma natural).

La construcción de un compilador conlleva interesantes desafíos algorítmicos: se precisa la utilización de distintos tipos de estructuras de datos (árboles, grafos, autómatas, tablas, etc.) y algoritmos (análisis LR, recorridos de árboles, ordenación, reescritura de términos, coloreado de grafos, etc.). Además los estudiantes tendrán que utilizar distintas herramientas de desarrollo al mismo tiempo (Automake, Doxygen, Flex, Bison, depuradores, yacc, lex, sistemas de control de versiones, etc.).

El estudio de los compiladores tiene además una función muy importante desde el punto de vista de aprender cómo funcionan y entender cómo se construyen los lenguajes (entender como funcionan las herramientas que utilizamos habitualmente en ingeniería informática). Por ejemplo, es muy típico que un estudiante novato utilizando un compilador se vea abrumado cuando obtiene algún tipo de error; esta situación cambia radicalmente una vez que el estudiante ha participado en un curso de compiladores. En definitiva, estudiar compiladores ayuda a los desarrolladores a hacer mejor su trabajo y muchos de los conocimientos y habilidades involucrados en el aprendizaje de los compiladores son esenciales para entender los conceptos y el funcionamiento interno de los lenguajes de programación.

Se consideran a continuación un conjunto de objetivos tanto de carácter general o transversal como de carácter específico, que se pretenderá que alcance el alumnado a lo largo del curso. Estos objetivos son que regirán el modo de actuar del profesorado. Además de los objetivos propuestos para Compiladores I como disciplina universitaria de la Ingeniería en Informática, se consideran competencias generales comunes a todas las disciplinas, que se consideran fundamentales para un titulado universitario.

### **3.- Objetivos generales**

La asignatura pretende:

- Iniciar al alumnado en procedimientos básicos de investigación: consulta de diversas fuentes, lectura de material que muestre diferentes opiniones, clasificación de las fuentes, etc.
- Fomentar el análisis crítico y evaluación de diferentes alternativas.
- Integrar el trabajo académico en el contexto social y cultural, procurando establecer elementos de relación con el mundo exterior.
- Fomentar el espíritu de mejora continua.
- Fomentar el trabajo en grupo.

### **4.- Objetivos específicos**

La asignatura pretende:

- Identificar los elementos constitutivos de un compilador: comprender y dominar su funcionamiento.
- Aplicar conocimientos previos de gramáticas y autómatas para la especificación y construcción de traductores.
- Diferenciar y dominar las diferentes técnicas para llevar a cabo el análisis léxico, sintáctico y semántico de lenguajes.
- Manejar herramientas para la construcción de analizadores léxicos, sintácticos y semánticos.

## 5.- Conocimientos previos

Dado que esta asignatura se realiza en el segundo ciclo se presupone al alumnado los conocimientos básicos adquiridos en el primer ciclo de una carrera de ingeniería informática. Es imprescindible que el estudiante maneje con soltura el uso de varios compiladores (C, Ada, Pascal,...) y tenga nociones de lenguajes de programación, gramáticas y autómatas. También se supondrá que el alumnado tiene conocimientos de arquitectura de computadores básicos.

En general, la mayoría del alumnado matriculado ha cursado el primer ciclo de la titulación de Ingeniería en Informática en el CPS, por lo que han superado ya asignaturas de primer ciclo de teoría de autómatas y lenguajes formales y programación. Sin embargo, hoy en día existe un porcentaje elevado de alumnos/as (sobre el 15 %) que es la primera vez que se matriculan en el centro y provienen en su mayoría de las titulaciones de Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Informática de Sistemas, impartidas en la Universidad de Zaragoza, y de Ingeniería Técnica en Informática de Gestión de la Universidad de la Rioja. A este tipo de alumnado se le recomienda cursar la asignatura de "Lenguajes, autómatas y gramáticas" antes de abordar la asignatura de Compiladores I y II en caso de que no hayan abordado su temario.

## 6.- Metodología y calendario inicial

A continuación, y en función de las horas de las que dispone actualmente la asignatura de "Compiladores I" (25 h de clases teóricas, 5 horas de clases de problemas y 15 horas de prácticas) se establece el tiempo que se le va a dedicar a cada uno de los módulos. Señalar que, aunque se indica una duración en horas de cada una de las diferentes lecciones comprendidas en cada módulo se será flexible y se adaptará a las necesidades y cambios que puedan surgir durante el desarrollo del curso. Para realizar de forma adecuada esa adaptación hay que disponer de información, por lo que es necesaria una retroalimentación continua al profesorado. Por último se considera que en el caso de la asignatura de "Compiladores I" las clases en el aula se corresponden con clases de 50 minutos 2 días por semana, mientras que las clases en el laboratorio de prácticas se corresponden con clases de 2 horas y 50 minutos cada quincena.

| Tema                                  | Horas<br>teóricas | Horas<br>problemas | Horas<br>laboratorio | Total |
|---------------------------------------|-------------------|--------------------|----------------------|-------|
| 0. Presentación de la asignatura      | 1                 | 0                  | 0                    | 1     |
| 1. Introducción                       | 1                 | 0                  | 0                    | 1     |
| 2. Análisis Léxico                    | 3                 | 1                  | 6                    | 10    |
| 3. Fundamentos de Análisis Sintáctico | 3                 | 0                  | 0                    | 3     |
| 4. Análisis Sintáctico Descendente    | 4                 | 1                  | 3                    | 8     |
| 5. Análisis Sintáctico Ascendente     | 5.5               | 1.5                | 3                    | 10    |
| 6. Introducción al Análisis Semántico | 5.5               | 1.5                | 3                    | 10    |
| 7. Repaso                             | 2                 | 0                  | 0                    | 2     |
| TOTAL                                 | 25                | 5                  | 15                   | 45    |

Las sesiones prácticas de la asignatura, se estructuran de la siguiente forma:

- Práctica 1: Definición de un lenguaje de programación propio para el cual se va a realizar un compilador a lo largo del curso.
- Práctica 2: Elaboración de un analizador léxico para el lenguaje de programación propio definido en la práctica anterior.
- Práctica 3: Elaboración de un analizador sintáctico descendente predictivo para el lenguaje de programación propio definido en la práctica 1 que esté integrado con el analizador léxico de la práctica 2.
- Práctica 4: Elaboración de un analizador sintáctico ascendente para el lenguaje de programación propio definido en la práctica 1 que esté integrado con el analizador léxico de la práctica 2.
- Práctica 5: Elaboración de un traductor dirigido por la sintaxis a partir del analizador sintáctico realizado en la práctica 4.

Tanto las clases de teoría en las que se desarrolla la parte teórica de los módulos como las sesiones prácticas son presenciales. La asistencia a las clases teóricas es opcional, aunque aconsejable, ya que es en ellas donde se presentan los contenidos de aplicación para desarrollar las sesiones prácticas. La asistencia a las prácticas es obligatoria y es condición necesaria haber asistido a todas con provecho para aprobar la asignatura.

La realización de las prácticas está planificada de forma que cada sesión se lleva a cabo una vez que el alumno dispone de los conocimientos necesarios correspondientes a la teoría del módulo que se corresponde con cada sesión práctica.

Antes de cada sesión práctica el alumno debe obligatoriamente preparar un trabajo previo correspondiente a las cuestiones que se plantean en el guión de cada práctica para realizar previamente. Después de realizar la práctica el alumno debe preparar



una memoria de la misma que se entregará y presentará al final del curso. El trabajo previo y posterior de cada sesión práctica es de carácter no presencial.

Las actividades no presenciales consisten, básicamente, en la preparación y posterior análisis del trabajo de las sesiones prácticas; además la resolución de ejercicios, casos o problemas planteados en las clases teóricas. Para estas actividades se dejará plena libertad horaria con la restricción de encontrar el tiempo necesario indicado en las guías de estudio para poder realizar los trabajos dentro de los plazos fijados.

Para el desarrollo de la asignatura se contará con el anillo Digital Docente (ADD). Esta Intranet o campus virtual será el principal medio para proporcionar los materiales de estudio necesarios como son las transparencias utilizadas en clase, los enunciados y manuales de prácticas y parte de los programas utilizados en las sesiones prácticas (Flex y Bison).

### **Sugerencias sobre pautas de actuación**

Al inicio del curso se publicarán en el ADD los documentos necesarios para todo el curso. Para poder cumplir con las actividades de forma conveniente es interesante seguir la siguiente pauta:

1. Preparación de un módulo
  - a. Descargar el material disponible en el ADD (Guía, materiales, etc.,)
  - b. Organizar y clasificar los documentos después de si, opcionalmente, se decide imprimirlos. En cualquier caso es interesante abrir carpetas diferentes para clasificar el material, a criterio de cada cual.
  - c. Analizar la dedicación requerida utilizando las indicaciones de las guías de estudio
  - d. Programar los días y las horas que se van a dedicar a lo largo de las semanas de duración del módulo.
2. Desarrollo del módulo
  - a. Realizar las actividades indicadas en las guías de estudio
  - b. Escribir las dudas, dificultades, etc., que se vayan planteando
  - c. Anotar el tiempo efectivo que se dedica a cada actividad
  - d. Preguntar las dudas a los profesores utilizando las horas de tutorías.
  - e. Al final de cada módulo, utilizar los objetivos formativos de cada uno de ellos para autoevaluarse.
3. Preparación de sesión práctica
  - a. Leer el guión de prácticas correspondiente que está disponible en el ADD.
  - b. Si la práctica lo requiere, consultar manual de usuario de la/s herramientas a usar en esa práctica.
  - c. Realizar las cuestiones planteadas en el guión previo
  - d. Realizar la sesión práctica con aprovechamiento
  - e. Después de la práctica escribir una memoria de la misma

Es importante señalar en la agenda personal de cada cual las fechas y horas de dedicación a cada actividad así como las fechas límite de las actividades.

### **Consejos sobre estrategias de trabajo**

Las guías de estudio contienen las instrucciones y la estimación del tiempo que se debe dedicar a cada actividad, Es muy importante que se realice una dedicación sistemática al estudio de esta asignatura. Estrictamente hablando, la única fecha límite es el día de la presentación de los trabajos de la asignatura. Sin embargo es muy aconsejable que el alumno realice el trabajo correspondiente semana a semana. Esta asignatura tiene una carga práctica muy importante, si se lleva al día durante el curso se tendrá una carga constante de trabajo pudiendo finalizar la asignatura antes del periodo habitual de exámenes.

## 7.- Evaluación

En esta sección, se considera tanto la evaluación del alumnado como la del profesorado y el método aplicado.

### La evaluación del alumnado

Aunque es posible aplicar muchas y variadas técnicas para la evaluación del alumnado, es francamente difícil asegurar que una técnica concreta se ajuste a los objetivos de imparcialidad, objetividad y justicia que serán deseables. La valoración tratará de hacerse sobre el grado de conocimientos adquiridos y sobre la formación global en el campo concreto de la asignatura, se primará la capacidad de razonamiento más que la memorística y el enciclopedismo.

Para evaluar según estos criterios, el diseño de pruebas y exámenes será muy cuidadoso y se propondrán a lo largo del curso una serie de ejercicios sobre los contenidos del temario que mejora el proceso de evaluación. Lamentablemente, su uso generalizado sólo será viable en grupos poco numerosos, pues el profesorado deberá analizarlos y devolverlos comentados y calificados.

En concreto, en la asignatura de Compiladores I se propone calificar al alumnado a través del siguiente sistema:

- El 75% de la calificación final (entre 0 y 10) se corresponde a la evaluación de un examen final.
- El 25% de la calificación final (entre 0 y 10) se corresponde a la evaluación de las prácticas. En caso de que el alumnado haya hecho las entregas y respectivas defensas en las clases de prácticas, se realizará una evaluación continua en la que se irá proporcionando retroalimentación al alumnado. En otro caso la evaluación de las prácticas será análoga a la que se realiza en el examen final.
- Además, se podrá optar a un punto extra que servirá para inclinar la balanza en casos dudosos (alumnos/as entre 4 y 5, 6 y 7, etc.). En este punto se considerará la participación en clase, la realización de mapas conceptuales de cada tema, la realización de fichas de ejercicios, y el interés mostrado.

### La evaluación del profesorado y método aplicado

El verdadero profesorado también ha de aprender y transformarse al enseñar. Por ello, es sumamente importante la evaluación del profesorado y del método aplicado, de modo que esta evaluación contribuya a mejorar la calidad de su trabajo, haciendo que la reflexión propia de sus funciones esté orientada a la acción y al cambio

cuando sea necesario. En este caso, es necesario considerar el resultado del proceso de (aprendizaje/enseñanza o enseñanza/aprendizaje) y la opinión del alumnado.

El análisis de los resultados de las pruebas planteadas al alumnado permite sacar conclusiones interesantes de cara a replantear el programa del siguiente curso. Básicamente, cuando se corrige una prueba se están percibiendo los errores de concepto más comunes. Estos errores son, sobre todo cuando se imparte una asignatura por primera vez, los que suelen sorprendernos y ante los que habrá que reaccionar. Fundamentalmente, es la forma de realizar la explicación, el tiempo dedicado a trabajar conceptos, o los métodos y medios utilizados, lo que se puede tratar de replantear con el objeto de reducir las deficiencias de comprensión detectadas.

La universidad de Zaragoza, se encarga de realizar distintos trabajos relativos a la calidad de la docencia y, especialmente, las encuestas que recogen la opinión de los estudiantes sobre la calidad de esta. Estas encuestas, entre otros resultados, proporcionan al profesorado una visión de la opinión de su alumnado sobre la docencia que han recibido, reflejando distintos aspectos. Se considera que hay que hacer un estudio autocrítico, pero relativo, de las opiniones por parte del alumnado. Los principales problemas que presentan este tipo de cuestionarios son dos: 1) el resultado suele tardar mucho tiempo en llegar al docente debido a causas administrativas, y 2) las preguntas son demasiado generales para poder aplicarse a todas las asignaturas. Por estas razones, se elaboran también encuestas propias que se le proporcionarán al alumnado al finalizar la asignatura enfocadas a la asignatura en cuestión.

El resultado de las pruebas finales o este tipo de encuestas no es útil para replantear el curso actual; para ello, se consideran fundamentales los Cuestionarios de Incidencias Críticas (CUIC), realizados en cada clase, y las pruebas consideradas para la evaluación continua, para poder actuar de la forma que se considere conveniente ante la detección de desviaciones sobre los resultado esperados. Además, el profesorado tratará de grabar sus clases para luego reflexionar sobre ellas o analizarlas con otros miembros del equipo docente.

## **8.- Documentación y bibliografía**

- "Compiladores: principios, técnicas y herramientas". A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- "Construcción e intérpretes: teoría y práctica". M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006
- "Crafting a compiler with C". C.N. Fischer, R.J. LeBlanc Jr. The Benjamin/Cummings Publishing Co. 1991.
- "Construcción de compiladores. Principios y práctica". K. C. Loudon. Thomson, 2004.
- "Diseño de compiladores". A. Garrido, J.M. Iñesta, F. Moreno, J.A. Pérez. Publicaciones Universidad de Alicante, 2002.

- "Lex and Yacc". J.Levine, T. Mason, D. Brown. O'Reilly and Associates, 1992.
- "Modern compiler implementation in C". A.W. Appel Cambridge University Press, 1997.

En el ADD estarán oportunamente disponibles los materiales necesarios para realizar las diversas actividades. El material básico para el seguimiento de curso son las Transparencias y el material de prácticas.

## 9.- Material informático necesario

Como se desprende del presente plan es imprescindible disponer de un cierto material informático para poder seguir el curso. En concreto es aconsejable disponer o tener acceso a:

### Hardware:

- Ordenador con conexión a Internet
- Impresora

### Software:

- Flex
- Bison

El alumno podrá trabajar en su ordenador personal con los programas proporcionados en el anillo digital. En cualquier caso, la sala donde se realizan las sesiones prácticas está a disposición del alumnado para que puedan trabajar en ella cuando lo requieran y no esté ocupada por otros grupos.

## 10.- Contenidos temáticos

La asignatura se organiza en 8 módulos:

Módulo 0: Presentación de la asignatura Compiladores I

Módulo 1: Introducción a los compiladores

- Cuestiones generales sobre compiladores.
- Elementos constitutivos de un compilador.

Módulo 2: Análisis léxico

- Fundamentos teóricos
- Cuestiones sobre implementación de un analizador léxico
- FLEX: un generador de analizadores léxicos

Módulo 3: Fundamentos para el análisis sintáctico

- Cuestiones generales sobre gramáticas y su uso en el análisis sintáctico
- Tipos de analizadores sintácticos: descendentes y ascendentes.

Módulo 4: Análisis sintáctico descendente

- Fundamentos y construcción de analizadores sintácticos descendentes
- Cuestiones sobre implementación de un analizador sintáctico descendente

#### Módulo 5: Análisis sintáctico ascendente

- Fundamentos y construcción de analizadores sintácticos ascendentes
- Cuestiones sobre implementación de un sintácticos ascendente
- BISON: un generador de analizadores sintácticos ascendentes

#### Módulo 6: Introducción al análisis semántico

- Manipulación de atributos semánticos

#### Módulo 7: Repaso general del curso

### 11.- Objetivos específicos de los módulos

Los objetivos que a continuación se detallan deben leerse, cada uno de ellos, precediendo su enunciado con la frase:

“Al finalizar el módulo el asistente debe ser capaz de...”

#### ***Módulo 1: Introducción a los compiladores***

- Definir y diferenciar los términos: compilador, intérprete y compilador-intérprete.
- Identificar y enumerar los elementos constitutivos de un compilador: conocerlos, diferenciarlos y comprender su función.
- Identificar y enumerar los elementos constitutivos de un intérprete: conocerlos, diferenciarlos y comprender su función.
- Comparar diferentes cualidades y características de los diferentes compiladores e intérpretes existentes.

#### ***Módulo 2: El análisis léxico***

- Identificar las distintas técnicas y métodos que intervienen en la construcción de un analizador léxico.
- Utilizar el vocabulario relacionado con el análisis léxico de la forma apropiada.
- Utilizar las diferentes técnicas y métodos para realizar análisis léxico de la forma apropiada.
- Determinar los elementos fundamentales de un lenguaje de programación.
- Manejar herramientas para la construcción de analizadores léxicos.

#### ***Módulo 3: Fundamentos para el análisis sintáctico***

- Identificar las distintas técnicas y métodos que intervienen en la construcción de un analizador sintáctico.
- Utilizar el vocabulario relacionado con el análisis sintáctico de la forma apropiada.
- Utilizar las diferentes técnicas y métodos para realizar análisis léxico de la forma apropiada.
- Distinguir entre el modo de funcionamiento y las características de analizadores sintácticos descendentes y ascendentes.

**Módulo 4: El análisis sintáctico descendente**

- Identificar las distintas técnicas y métodos que intervienen en la construcción de un analizador sintáctico descendente.
- Utilizar el vocabulario relacionado con el análisis sintáctico descendente de la forma apropiada.
- Construir analizadores sintácticos descendentes recursivos y por tablas.
- Manejar herramientas para la construcción de analizadores sintácticos descendentes.

**Módulo 5: El análisis sintáctico ascendente**

- Identificar las distintas técnicas y métodos que intervienen en la construcción de un analizador sintáctico ascendente.
- Utilizar el vocabulario relacionado con el análisis sintáctico ascendente de la forma apropiada.
- Construir analizadores sintácticos ascendentes SLR, LR canónicos y LALR.
- Manejar herramientas para la construcción de analizadores sintácticos ascendentes.

**Módulo 6: El análisis semántico**

- Identificar las distintas técnicas y métodos que intervienen en la construcción de un analizador semántico.
- Utilizar el vocabulario relacionado con el análisis semántico de la forma apropiada.
- Diferenciar entre los diferentes tipos de analizadores semánticos analizando sus ventajas e inconvenientes.
- Identificar los distintos tipos de atributos de gramáticas atribuídas.
- Manejar herramientas para la construcción de analizadores semánticos.

**12.- Glosario***Bison*

Bison es una herramienta de generación automática de analizadores sintácticos ascendentes LALR. A partir de una serie de especificaciones Bison construye analizadores sintácticos ascendentes en lenguaje C. Se trata de una herramienta de libre distribución basada en la herramienta propietaria Yacc. Existen versiones para plataformas Windows, Linux y Unix.

*Compilador*

Programa que lee un programa (código fuente) escrito en un lenguaje y lo traduce a un programa EQUIVALENTE en otro lenguaje (código objeto).

*Compilador intérprete*

Programa(s) que traduce(n) un programa fuente a un lenguaje intermedio que posteriormente se interpreta. Ejemplos: Java, familia .Net

*Intérprete*

Programa que a medida que va leyendo un programa (código fuente) escrito en un lenguaje, lo va interpretando (ejecutando).

*Flex*

Flex es una herramienta de generación automática de analizadores léxicos. A partir de una serie de especificaciones Flex construye analizadores léxicos en lenguaje C. Se trata de una herramienta de libre distribución basada en la herramienta propietaria Lex. Existen versiones para plataformas Windows, Linux y Unix.

## Nota:

Este documento puede sufrir modificaciones a lo largo del desarrollo de la asignatura. En ese caso se informará a los alumnos de los cambios introducidos y se colocará la última versión en el Anillo Digital Docente (<http://add.unizar.es/>).

