

# Proyecto Docente de Compiladores I

Raquel Trillo Lado

Número de plaza	288
Resolución Univ. Zaragoza	6 de febrero de 2007
B.O.A.	14 de febrero de 2007
Centro	Escuela Universitaria de Estudios Empresariales de Zaragoza
Área de conocimiento	Lenguajes y Sistemas Informáticos
Departamento	Informática e Ingeniería de Sistemas
Perfil	Informática aplicada a la gestión de la empresa
Fecha de presentación	Zaragoza, 2 de mayo de 2007







# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Contextualización: ¿dónde estamos y hacia dónde vamos?</b>	<b>5</b>
2.1. El Espacio Europeo de Educación Superior . . . . .	5
2.1.1. Caminando hacia el EEES . . . . .	7
2.1.2. Sistema de Créditos Europeos (ECTS) . . . . .	7
2.1.3. Las titulaciones en el Modelo Europeo . . . . .	9
2.2. El Marco Legislativo Español . . . . .	10
2.2.1. El sistema universitario anterior: La Ley de Reforma de Universidades (LRU) . . . . .	11
2.2.2. La Ley Orgánica de Universidades (LOU) . . . . .	13
2.2.3. La Reforma de la LOU . . . . .	17
2.2.4. La adaptación al EEES en España . . . . .	19
2.3. El sistema universitario de Aragón . . . . .	24
2.3.1. La Universidad de Zaragoza . . . . .	25
2.3.2. El Centro Politécnico Superior (CPS) . . . . .	30
2.3.3. El Departamento de Informática e Ingeniería de Sistemas (DIIS) . . . . .	32
2.3.4. La adaptación al EEES en la Universidad de Zaragoza . . . . .	35
2.4. Los estudios de Informática . . . . .	37
2.4.1. Las titulaciones de Ingeniería en Informática en España . . . . .	38
2.4.2. Las titulaciones de Ingeniería en Informática en la Universidad de Zaragoza . . . . .	38
2.4.3. La materia de Procesadores del Lenguaje en Ingeniería Informática en el CPS . . . . .	41
<b>3. Contexto en la enseñanza de Compiladores</b>	<b>47</b>
3.1. Recomendaciones Curriculares ACM/IEEE . . . . .	47
3.1.1. Áreas de conocimiento . . . . .	50
3.1.2. Modelos curriculares . . . . .	52
3.1.3. Materias introductorias . . . . .	54
3.1.4. Materias intermedias . . . . .	56
3.1.5. Materias avanzadas . . . . .	61

3.2.	Espacio Europeo de Educación Superior: competencias de un/a Ingeniero/a en Informática . . . . .	61
3.2.1.	Las competencias de un/a ingeniero/a en Informática . . . . .	61
3.2.2.	Las competencias de un/a ingeniero/a en Informática en España . . . . .	63
3.2.3.	Compiladores y el Libro Blanco del título de Grado de Informática . . . . .	67
3.3.	Compiladores en Universidades de Estados Unidos . . . . .	67
3.3.1.	Universidad de Columbia . . . . .	68
3.3.2.	Universidad de Stanford . . . . .	69
3.3.3.	Conclusiones . . . . .	70
3.4.	Compiladores en las Universidades Españolas . . . . .	70
3.4.1.	Universidad Politécnica de Madrid (UPM) . . . . .	70
3.4.2.	Universidad Politécnica de Barcelona (UPC) . . . . .	73
3.4.3.	Universidad de A Coruña (UDC) . . . . .	75
3.4.4.	Conclusiones y Experiencias . . . . .	76
<b>4.</b>	<b>Metodología:¿cómo queremos desarrollar nuestra labor docente en el contexto actual?</b>	<b>79</b>
4.1.	Consideraciones generales . . . . .	79
4.2.	Metodología docente . . . . .	80
4.2.1.	Metodología didáctica . . . . .	81
4.3.	La evaluación . . . . .	88
4.3.1.	La evaluación del alumnado . . . . .	88
4.3.2.	La evaluación del profesorado y método aplicado . . . . .	89
<b>5.</b>	<b>Programa docente de Compiladores I</b>	<b>91</b>
5.1.	El alumnado . . . . .	91
5.2.	Justificación . . . . .	92
5.3.	Objetivos . . . . .	93
5.3.1.	Objetivos generales . . . . .	93
5.3.2.	Objetivos específicos . . . . .	94
5.4.	Programa detallado . . . . .	94
5.4.1.	Tema 0: Presentación de la asignatura . . . . .	95
5.4.2.	Tema 1: Introducción . . . . .	96
5.4.3.	Tema 2: Análisis léxico . . . . .	97
5.4.4.	Tema 3: Fundamentos de análisis sintáctico . . . . .	101
5.4.5.	Tema 4: Análisis sintáctico descendente . . . . .	103
5.4.6.	Tema 5: Análisis sintáctico ascendente . . . . .	108
5.4.7.	Tema 6: Análisis semántico . . . . .	113
5.4.8.	Tema 7: Repaso . . . . .	118
5.5.	Estimación de la carga de trabajo del alumnado . . . . .	118
5.5.1.	Tema 0: Presentación de la asignatura . . . . .	118
5.5.2.	Tema 1: Introducción . . . . .	118
5.5.3.	Tema 2: Análisis léxico . . . . .	119
5.5.4.	Tema 3: Fundamentos de análisis sintáctico . . . . .	120
5.5.5.	Tema 4: Análisis sintáctico descendente . . . . .	121

5.5.6. Tema 5: Análisis sintáctico ascendente . . . . .	122
5.5.7. Tema 6: Análisis semántico . . . . .	124
5.5.8. Tema 7: Repaso . . . . .	126
5.5.9. Conclusiones . . . . .	126
5.6. Problemas detectados en el alumnado . . . . .	127
5.6.1. Relacionados con la movilidad del alumnado . . . . .	127
5.6.2. Relacionados con preconcepciones erróneas de conceptos . . . .	127
5.7. Bibliografía Recomendada . . . . .	128
<b>Referencias</b>	<b>129</b>
<b>Anexo I: Ficha de la Asignatura ECTS</b>	<b>131</b>
<b>Anexo II: Plan de la Asignatura para el Alumnado</b>	<b>139</b>
<b>Anexo III: Guía de Estudio del Tema 2 para el Alumnado</b>	<b>155</b>
<b>Anexo IV: Temario para la Segunda Prueba</b>	<b>169</b>





# Lista de Figuras

1.1. Estructura del Proyecto Docente . . . . .	2
2.1. Situación respecto a la introducción de los ECTS en la Educación Superior . . . . .	9
2.2. Materias troncales en Ingeniería en Informática . . . . .	39
2.3. Materias troncales en Ingeniería Técnica en Informática de Sistemas . . . . .	40
2.4. Materias troncales en Ingeniería Técnica en Informática de Gestión . . . . .	41
2.5. Plan de estudios de Ingeniería en Informática en la Universidad de Zaragoza . . . . .	42
2.6. Optativas ofertadas en el Plan de Estudios de Ingeniería en Informática en la Universidad de Zaragoza . . . . .	43
2.7. Plan de Estudios de Ingeniería Técnica en Informática de Sistemas en la Universidad de Zaragoza . . . . .	44
2.8. Optativas ofertadas en el Plan de Estudios de Ingeniería Técnica en Informática de Sistemas en la Universidad de Zaragoza . . . . .	44
2.9. Plan de Estudios de Ingeniería Técnica en Informática de Gestión en la Universidad de Zaragoza . . . . .	44
2.10. Optativas ofertadas en el Plan de Estudios de Ingeniería Técnica en Informática de Gestión en la Universidad de Zaragoza . . . . .	45
2.11. Asignaturas relacionadas con Compiladores I en el Plan de Estudios de Ingeniería en Informática de la Universidad de Zaragoza . . . . .	45
3.1. Materias introductorias, intermedias y avanzadas en el IEEE/ACM Computing Curricula 2001 . . . . .	54
3.2. Materias en el IEEE/ACM Computing Curricula 2001 que contienen unidades de Lenguajes de Programación: tabla 1 de 2 . . . . .	57
3.3. Materias en el IEEE/ACM Computing Curricula 2001 que contienen unidades de Lenguajes de Programación: tabla 2 de 2 . . . . .	58
3.4. Horas dedicadas a unidades de Lenguajes de Programación en materias introductorias para cada aproximación (IEEE/ACM Computing Curricula 2001): Imperative first (I), Objects first (O), Functional first (F), Breadth first (B), Algorithms first (A), y Hardware first (H) . . . . .	58

3.5.	Materias intermedias con alguna unidad del área de Lenguajes de Programación, con distintas aproximaciones (IEEE/ACM Computing Curricula 2001): Topic-based (T), Compressed (C), System-based (S), Web-based (W) . . . . .	60
3.6.	Horas dedicadas a unidades de Lenguajes de Programación en materias intermedias en cada aproximación (IEEE/ACM Computing Curricula 2001): Topic-based (T), Compressed (C), System-based (S), Web-based (W). Fuente: Curricula Computing 2001 . . . . .	60
3.7.	Materias avanzadas en el IEEE/ACM Computing Curricula 2001, clasificadas por áreas . . . . .	62

# Lista de Tablas

3.1. Temas que podrían afectar a la enseñanza de Compiladores (máximos y mínimos): perfiles CE (Computer Engineering), CS (Computer Science), IS (Information Systems), IT (Information Technology), y SE (Software Engineering) . . . . .	50
3.2. Áreas de conocimiento en el IEEE/ACM Computing Curricula 2001 . . . . .	51
3.3. Unidades y temas del área de Lenguajes de Programación en el IEEE/ACM Computing Curricula 2001, tabla 1 de 2 . . . . .	52
3.4. Unidades y temas del área de Lenguajes de Programación en el IEEE/ACM Computing Curricula 2001, tabla 2 de 2 . . . . .	53
3.5. Competencias transversales genéricas en el Libro Blanco del Título de Grado en Informática . . . . .	65
3.6. Competencias transversales específicas en el Libro Blanco del Título de Grado en Informática . . . . .	66
5.1. Temas de la asignatura de Compiladores I . . . . .	95
5.2. Carga de trabajo del estudiante . . . . .	126



# Capítulo 1

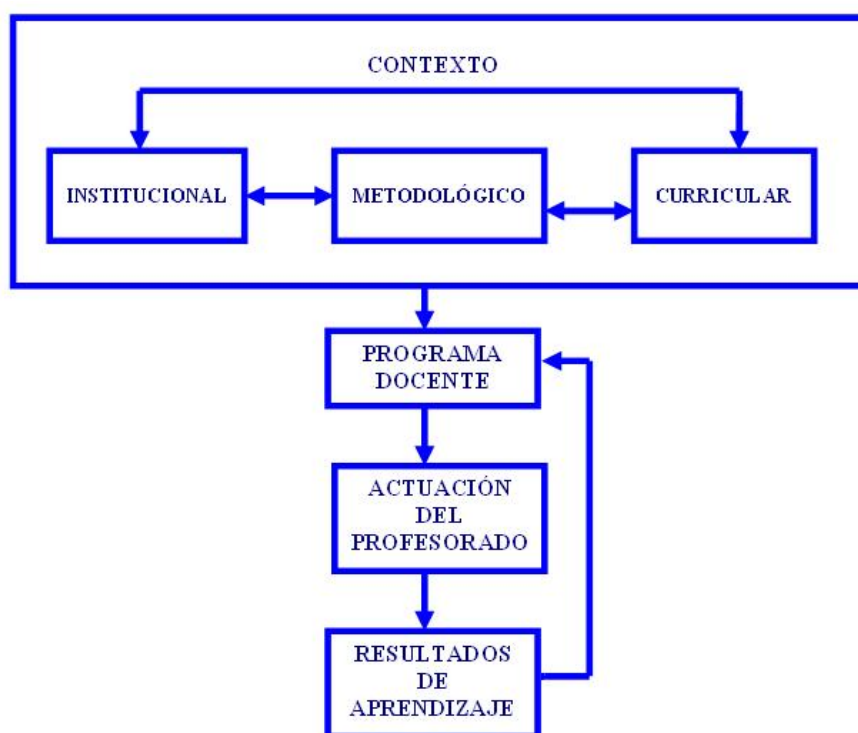
## Introducción

Este documento presenta el proyecto docente que ha sido elaborado para participar en el concurso público de la plaza de Profesor/a Colaborador número 288, publicada en el B.O.A. (Boletín Oficial de Aragón) de 14 de febrero, según resolución de 6 de febrero de 2007 de la Universidad de Zaragoza. Esta plaza, en el Departamento de Informática e Ingeniería de Sistemas (DIIS), pertenece al área de Lenguajes y Sistemas Informáticos (LSI) de la Universidad de Zaragoza y su labor se realizará en la Escuela Universitaria de Estudios Empresariales de Zaragoza. La plaza tiene el perfil *Informática aplicada a la gestión de la empresa*. Sin embargo, este trabajo está centrado en *Compiladores I*, por proximidad a la labor docente que he realizado en los últimos años y porque en las bases de la citada convocatoria se establece que “*El proyecto docente incluirá el programa de una de las asignaturas troncales u obligatorias del área de conocimiento a que corresponda la plaza*”.

El objetivo de este proyecto no es sólo dotar de contenido a la asignatura de *Compiladores I* sino también establecer las bases para el desarrollo de ésta, estructurando un programa bien coordinado entre sí y con las asignaturas afines, así como una metodología y planificación para su desarrollo. Para llevar a cabo esta labor se considera la normativa legal vigente, establecida a nivel nacional por las directrices de los planes de estudios fijadas por el Ministerio de Educación y Ciencia, y a nivel local por el plan de estudios del centro en la que se va a impartir la materia. Una vez establecido el marco legal, se considera la incorporación de todas aquellas experiencias y planteamientos ajenos y propios que permitan realizar una programación docente enmarcada en las necesidades profesionales y sociales actuales. Además, se pretende que este no sea un documento rígido y fijo sino que vaya evolucionando a lo largo del tiempo con el objetivo de mejorarlo y actualizarlo; por ello, se agradece de antemano cualquier tipo de aportación constructiva y se plantean mecanismos de evaluación para detectar sus debilidades y poder corregirlas.

La figura 1.1 proporciona una visión global del escenario donde se ubica la actividad de aprendizaje-enseñanza a realizar, y se utiliza como hilo argumental de este trabajo, condicionando la estructura del mismo. El contexto en el que se desarrolla el

trabajo está desglosado en tres grandes bloques, los cuales se interrelacionan y condicionan entre sí: 1) el contexto institucional, 2) el contexto pedagógico, y 3) el contexto curricular. El *contexto institucional* se caracteriza por el denominado *mundo universitario* (historia, normas, cultura, organización, etc.). El *contexto pedagógico* considera el proceso de enseñanza-aprendizaje o aprendizaje-enseñanza en sí mismo. Por último, el *contexto curricular* hace referencia a las competencias, capacidades, habilidades, destrezas, aptitudes y valores que se deben considerar particularizados en la asignatura de *Compiladores I* de los futuros egresados/as en Ingeniería en Informática de la Universidad de Zaragoza.



**Figura 1.1:** Estructura del Proyecto Docente

Después de analizar el contexto, se diseña el programa docente considerando la situación en la que nos encontramos. Dicho programa regirá la actuación del profesorado: organización, planificación, actividades, etc. Esa actuación provoca unos determinados resultados de aprendizaje, tanto en el alumnado como en el propio profesorado, que deben emplearse, al igual que los cambios que se produzcan en el contexto, como retroalimentación continua para la revisión constante del programa docente.

Por ello, este documento se estructura en 5 capítulos, incluyendo este. En el capítu-

lo 2, se hace una breve descripción del contexto institucional actual centrándonos en: 1) el Espacio Europeo de Educación Superior, 2) el marco legislativo español, 3) el sistema universitario en Aragón, y 4) los estudios de Ingeniería en Informática. En el capítulo 3, se consideran las directrices y recomendaciones curriculares relacionadas con el aprendizaje-enseñanza de las materias de Compiladores, así como los modelos seguidos en otras universidades. En el capítulo 4, se presentan una serie de consideraciones metodológicas generales para la impartición de la asignatura. Por último, en el capítulo 5 se describe detalladamente el programa docente de la asignatura de *Compiladores I*.





## Capítulo 2

# Contextualización: ¿dónde estamos y hacia dónde vamos?

*“La educación superior es esencial para el desarrollo científico, técnico y cultural de cualquier país. Por ello debemos plantearnos cuál es el modelo de Universidad capaz de contribuir a la transformación de la presente sociedad.”*

Francisco Michavila

En este capítulo se aborda el estudio y análisis del contexto institucional, lo que popularmente se denomina *mundo universitario*: funciones y objetivos, reglamentación, estructura, y funcionamiento. Se pretende dar una visión de la Universidad actual, particularizada en la Universidad Española y concretamente en la Universidad de Zaragoza. Después de un breve análisis de la situación actual, se hace una pequeña reflexión acerca del punto donde estamos.

### 2.1. El Espacio Europeo de Educación Superior

La Unión Europea (UE), anteriormente Comunidad Económica Europea (CEE), inicialmente persiguió objetivos económicos. Sin embargo, hoy en día, además de estos, trata de buscar la convergencia en diferentes ámbitos, entre ellos, la educación. Por ello, en los últimos años se ha impulsado un movimiento encaminado a lograr la convergencia de las titulaciones universitarias en Europa, el denominado *Espacio Europeo de Educación Superior (EEES)*. Este proyecto pretende estar implantado en la mayoría de países europeos antes del año 2010, y busca armonizar los programas de las distintas universidades, con los objetivos de:

1. *Conseguir una mayor compatibilidad y comparabilidad de los sistemas de Educación Superior.*
2. Facilitar el acceso de estudiantes y profesorado de unas universidades a otras,

*favoreciendo la movilidad.* También se pretende facilitar la movilidad de los profesionales.

3. *Incrementar el intercambio de conocimientos* entre investigadores/as de los diferentes países.
4. *Incrementar la competitividad* del Sistema Europeo de Educación Superior y aumentar su grado de atracción mundial.

A modo de resumen, es interesante destacar los grandes principios que guían este proceso:

1. La adopción de un sistema fácilmente comprensible y comparable de las titulaciones, mediante la implantación, entre otros, de un *Suplemento al Título*.
2. La adopción de un sistema universitario basado fundamentalmente en dos ciclos principales. El título concedido al acabar el primer ciclo capacitará para el ejercicio profesional y se denominará *Grado*. El segundo título se encaminará a la obtención de un *Máster* orientado a formación específica y/o *Doctorado*. Esto permitirá al alumnado obtener un primer título y continuar (posiblemente de forma paralela al ejercicio de una actividad profesional) su *formación continua a lo largo de su vida*, ya sea dentro del ámbito universitario o fuera de él.
3. La implantación de unidad de medida común, el *European Credit Transfer System (ECTS)*, un sistema de créditos europeos que difiere del que muchos países, entre ellos España, están empleando actualmente. El crédito ECTS pretende reflejar el esfuerzo real requerido por los estudiantes para conseguir una serie de objetivos relacionados con las materias.
4. *Favorecer la movilidad de estudiantes, profesores e investigadores* mediante la supresión de barreras administrativas y manteniendo los derechos del país origen.
5. *Promover la cooperación europea en el aseguramiento de la calidad de la enseñanza.* En este punto se hace referencia a los *procesos de acreditación*.
6. *Alentar a la dimensión europea en la Educación Superior* ofreciendo módulos, cursos y currículos *europeos*, especialmente en titulaciones compartidas por instituciones de países diferentes.

A continuación, en la Sección 2.1.1 se expone brevemente la evolución histórica que ha dado lugar al camino hacia el EEES, en la Sección 2.1.2 se trata en mayor profundidad el crédito ETCS y, para finalizar, en la Sección 2.1.3 se considera la nueva estructura de las titulaciones en el EEES.

### 2.1.1. Caminando hacia el EEES

En 1988, en el *IX centenario de la Universidad de Bolonia* (la más antigua de Europa), representantes de Francia, Alemania, Italia y Reino Unido redactan la *Carta Magna de las Universidades Europeas*, en la que se proponen una serie de objetivos y líneas de actuación que pretenden iniciar una reforma progresiva del Sistema Europeo de Educación Superior. No obstante, en ese momento no se toma todavía ninguna medida para llevar esto a cabo. Este acuerdo lo firman los máximos responsables de la educación de 29 estados europeos.

En mayo de 1998, los ministros encargados de la Educación Superior de Alemania, Francia, Italia y el Reino Unido, en la Universidad de Sorbona (París), suscribieron la *Declaración de la Sorbona*. En este documento se analiza la falta de homogeneidad y la inexistencia de un patrón de medida común de la carga de las materias y se insta al desarrollo de un Espacio Europeo de la Educación Superior (EEES), estableciendo la necesidad de crear un sistema de créditos Europeos (ETCS).

Un año más tarde, de nuevo en la Universidad de Bolonia, los ministros encargados de la Educación Superior de 29 países europeos celebraron una conferencia que sentó las bases para conseguir el EEES en 2010. La *Declaración de Bolonia* marca los objetivos de adoptar un sistema fácilmente legible y comparable de titulaciones basado en dos ciclos principales, establecer un sistema internacional de créditos, promover la movilidad de estudiantes, profesores e investigadores, promover la cooperación europea para garantizar la calidad de la Educación Superior y, en definitiva, promover una dimensión europea de la Educación Superior.

Los ministros, esta vez 32, se volvieron a reunir en *Praga* en mayo de 2001. En ese momento, recogen las recomendaciones de la Convención de Instituciones Europeas de Enseñanza Superior celebrada en Salamanca en marzo de ese mismo año. Posteriormente se vuelven a reunir en *Berlín* (Alemania) en septiembre de 2003, y en *Bergen* (Noruega) en mayo de 2005. La próxima conferencia tendrá lugar en *Londres* en el 2007.

La Comisión Europea se ha implicado en el proceso de convergencia de la Educación Superior Europea y ha publicado varios documentos apoyando la iniciativa, entre los que destaca la comunicación *“El papel de las universidades en la Europa del conocimiento”*, de mayo del 2003. Además, las organizaciones de universidades europeas han acogido muy positivamente la iniciativa ministerial. Así, la Asociación de la Universidad Europea (EUA) se implicó en el proceso y ha desarrollado varios estudios (*Trends in Learning Structures*) sobre la Educación Superior Europea y ha manifestado su apoyo y sugerencias a la iniciativa.

### 2.1.2. Sistema de Créditos Europeos (ECTS)

Como se comentó en la introducción de esta sección, uno de los objetivos fundamentales de la declaración de Bolonia es la implantación de un sistema común de créditos, el sistema ECTS. El nuevo sistema de créditos introduce un cambio sustancial en la unidad de medida habitual de las materias. Actualmente, en España y otros sistemas educativos europeos, existe un sistema que gira en torno a la figura

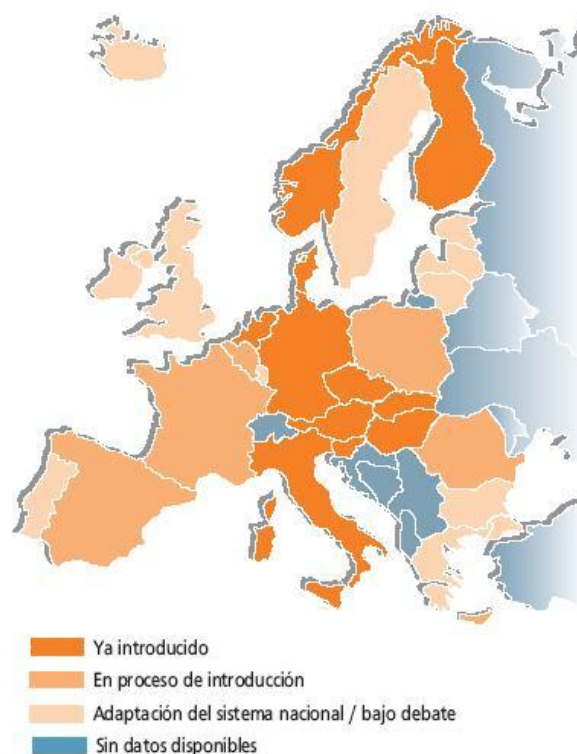
del profesorado y establece que un crédito equivale a 10 horas de clase presencial con un docente. El sistema ECTS desplaza el centro de atención del docente al esfuerzo del discente, es decir, establece un método de medida que se adapta al proceso de aprendizaje-enseñanza en lugar de al actual proceso enseñanza-aprendizaje.

Además, el sistema ECTS establece que el volumen total de trabajo de un estudiante a tiempo completo durante un curso académico debe ser de 60 créditos. Esto supone que los cursos de las titulaciones universitarias deben constar de 60 créditos ECTS. De este modo, si se considera una actividad académica aproximada de 40-45 semanas/año y una carga de trabajo en torno a las 40 horas/semana, se obtiene que un crédito ECTS supone entre 25 y 30 horas de trabajo del estudiante (de acuerdo con lo anterior, se estima su trabajo entre 1600 y 1800 horas a tiempo completo a lo largo del año). Es conveniente destacar que en esa carga de trabajo se incluye cualquier actividad que el/la estudiante deba realizar para superar las materias, es decir, se incluye como parte de esa carga, no sólo el tiempo de asistencia a la aula sino también el tiempo dedicado a la lectura, estudio, realización de trabajos y actividades complementarias, participación en seminarios y conferencias, tiempo de preparación y realización de pruebas, asistencia a tutorías, etc.

Por tanto, a la hora de diseñar una materia es necesario modificar la forma de plantearla. Además, el profesorado en este nuevo sistema deberá dedicar menos tiempo a la exposición de conocimientos, y tendrá en su lugar un trabajo más intenso en lo que respecta a la organización de programas y dirección del trabajo personal del alumnado. Esto implica una reducción en las horas lectivas o clases presenciales, aproximándonos a una enseñanza más personalizada, con un amplio uso de las tutorías y más orientado a fomentar y a incentivar el aprendizaje y la creatividad. Por otra parte, el alumnado pasa de un modelo en el que reciben clases magistrales un 80 % del tiempo y desarrollan prácticas el 20 % restante, además del estudio personal y la asistencia a tutorías *voluntariamente*, a otro modelo donde la clase magistral sólo supone el 30 % y el aprendizaje autónomo el restante 70 % y donde las tutorías pasan a ser parte del propio proceso de aprendizaje-enseñanza. En este nuevo proceso, las metodologías de aprendizaje activas (aprendizaje basado en problemas, trabajo en grupo, aprendizaje por proyectos, etc.) deberían ser lo habitual.

En determinados grupos de profesorado, la pregunta que surge es: ¿cuántas horas de clase presencial se corresponden a un crédito ECTS?. Dicho de otra manera, la pregunta es cuántas horas de clase presencial se corresponden con 25-30 horas de trabajo del alumnado. Algunos buscan la respuesta en el Espacio de Educación Superior de Estados Unidos, donde se considera que por término medio una hora de clase presencial debe ocasionarle al alumnado de 2 a 3 horas de trabajo personal. Si se aplicara la proporción de que 1 hora de clase presencial genera 2 horas de trabajo para el estudiante, entonces se concluiría que a 1 crédito ECTS le corresponden 10 horas de clase y 20 de trabajo personal, y 1 crédito ECTS supondría 30 horas totales de trabajo del estudiante.

Actualmente, el grado de implantación en la Unión Europea (UE) del nuevo sistema de créditos es alto debido a la cercanía del curso 2010-2011, como se puede observar en la figura 2.1.



**Figura 2.1:** Situación respecto a la introducción de los ECTS en la Educación Superior

Para finalizar, señalar que el término ECTS no debe confundirse con el término *ECTS grades*, ya que este último representa el rendimiento obtenido en una determinada materia (la calificación), no la carga de trabajo para el estudiante. La introducción del *ECTS grades* permite la comparación de las calificaciones obtenidas en diferentes universidades europeas. Además, este nuevo sistema de calificación se corresponde con el utilizado en Estados Unidos.

### 2.1.3. Las titulaciones en el Modelo Europeo

Como se dejaba ver al inicio de esta sección, otro aspecto clave del Espacio Europeo de Educación Superior afecta a la estructura de las titulaciones, puesto que en la actualidad es muy heterogénea y dificulta la comparabilidad. Para tratar de homogeneizar la situación, se establece un sistema basado en *niveles o ciclos*. En la *Declaración de Bergen* se destaca que: “La Educación Superior Europea se estructura alrededor de tres ciclos, donde cada nivel tiene la función de preparar al estudiante para el mercado laboral, para construir futuras competencias y para formarlo como ciudadano activo.”.

Se han establecido dos ciclos principales: el *Grado* (primer ciclo) y el *Máster* (segundo ciclo), pero no establece una norma general en cuanto a la duración de los ciclos se refiere (las estructuras más generalizadas son de tipo 3+2, 4+1 y 4+2). El *Grado* se obtendrá al acabar el primer ciclo y capacitará para el ejercicio profesional, mientras que el *Máster* se orientará o bien a formación específica en una determinada área para mejorar alguna capacidad profesional o bien a iniciar una carrera investigadora con el objetivo de obtener un *Doctorado* (3º ciclo); en general, el acceso al Doctorado es directo tras completar los dos ciclos anteriores. Este sistema pretende permitir al egresado ingresar en el mercado laboral una vez realice el Grado y fomentar la *formación continua a lo largo de toda la vida*, compatibilizándola con el ejercicio profesional.

Además, en la *Declaración de Bolonia* se establece: “La adopción de un sistema de grados<sup>1</sup> comparables y legibles, a través de la implementación del Suplemento al Diploma, para promover la empleabilidad de los ciudadanos europeos y la competitividad internacional del sistema de Educación Superior europeo”. Es decir, junto con el título de un determinado nivel se deberá adjuntar un *suplemento*. En este se describirá el nivel de la titulación, las materias cursadas en créditos ECTS y sus calificaciones, el contexto, y las competencias alcanzadas. El suplemento al título debe ser bilingüe, para potencial la movilidad internacional. Por esa misma razón, el programa de estudio también debe ser bilingüe, accesible vía web, y con una determinada estructura.

Por otro lado se incide en la necesidad del aseguramiento de la calidad. Por ello, prácticamente todos los países de la Unión Europea disponen de un comité miembro de la *European Network for Quality Assurance in Higher Education (ENQA)* para evaluar periódicamente las titulaciones, aunque todavía no es obligatorio hacerlas.

## 2.2. El Marco Legislativo Español

El cambio en la sociedad en las 3 últimas décadas y el de la Universidad en el marco de la *Ley de Reforma Universitaria* (LRU, 1983) supusieron un acercamiento de la Universidad a los ciudadanos y un aumento del número de estudiantes en las aulas, dando lugar a la denominada *Universidad de masas*. Además, se produjo un incremento e intensificación de la investigación en España. Con el paso del tiempo, y después de 18 años, la realidad social ya no es la misma y se aconseja una actualización de la LRU que corrija los defectos y supla las carencias detectadas. Por ello surge la *Ley Orgánica de Universidades* (LOU, 2001) y posteriormente la *Reforma de la LOU* (que ha sido aprobada recientemente). Hoy en día, además, todos los países firmantes de la Declaración de Bolonia (1998), entre los que está España, han emprendido las reformas legislativas pertinentes para adaptarse al Espacio Europeo de Educación Superior.

A continuación, se analiza el marco legislativo actual en España. En concreto, la Ley de Reforma Universitaria (LRU) en la Sección 2.2.1, la Ley Orgánica de Universidades (LOU) en la Sección 2.2.2, y la Reforma de la Ley Orgánica de Universidades en la Sección 2.2.3. La LRU, a pesar de estar derogada por el punto 1 de la dis-

---

<sup>1</sup>Entendiéndose aquí grado como título y no como primer ciclo de la formación.

posición derogatoria única de la LOU, es interesante comentarla porque rige para el desarrollo no establecido en la LOU, la LOU por ser el marco normativo que ha estado vigente en España en los últimos años, y la Reforma de la LOU porque ha sido aprobada recientemente. Además, se analiza la situación de España dentro del proceso de convergencia Europea en la Sección 2.2.4.

### 2.2.1. El sistema universitario anterior: La Ley de Reforma de Universidades (LRU)

Ya en la *Ley General de Educación* (1970) se estructuró la educación universitaria en España en tres ciclos:

1. *Primer ciclo*: se establece que el primer ciclo tiene una duración de 3 años y se imparte en las Escuelas Universitarias, ofreciendo una formación de profesionales de grado medio. El título obtenido es el de Diplomatura, Ingeniero Técnico o Arquitecto Técnico.
2. *Segundo ciclo*: se establece que el segundo ciclo tiene una duración de 5 años y se imparte en las Facultades, o una duración de 6 años si se imparte en los Centros de Enseñanza Técnica Superior. El título obtenido es el de Licenciado, Arquitecto o Ingeniero.
3. *Tercer ciclo*: se establece que el tercer ciclo es una formación complementaria que tiene una duración de 2 años, orientada a capacitar al estudiante a afrontar posteriormente el desarrollo de la Tesis Doctoral. Está destinado a formar investigadores/as y profesorado de Universidad.

Después, la *Ley de Reforma Universitaria (LRU)*, en 1983 (Ley Orgánica 11/1983 de 25 de agosto), supuso una reforma de la Universidad y de la enseñanza superior en España. La LRU dictaba en su primer artículo que “*el servicio público de la Educación Superior corresponde a la Universidad*”, y que se debía realizar a través de “*la docencia, el estudio y la investigación*”. Además, establecía que “*los principios que rigen el funcionamiento de la Universidad son la autonomía y la libertad académica*”. En este punto, cabe destacar la autonomía financiera y la capacidad de seleccionar y promocionar al profesorado.

El cambio de régimen estatuario de las universidades, la organización departamental y el régimen del profesorado fueron consecuencias significativas de la LRU. Sin embargo, todavía había que tratar la ordenación académica de la enseñanza. Ésta empieza a realizarse a partir de la constitución del *Consejo de Universidades* en 1985. A este organismo, el artículo 28.1 de la LRU le atribuye “*proponer los títulos de carácter oficial y de validez en todo el territorio estatal, así como las directrices generales de los planes de estudios que deberán cursarse para su obtención y homologación*”.

Posteriormente, el *Consejo de Universidades*, en febrero de 1987, elaboró un documento titulado “*La reforma de las Enseñanzas Universitarias*”, con los siguientes objetivos:

1. *Facilitar y actualizar la enseñanza y los conocimientos* impartidos en las universidades españolas, promoviendo el desarrollo cultural, científico, técnico, y la formación interdisciplinar. Además, se subraya la importancia de la formación de carácter instrumental.
2. *Armonizar el carácter estatal de los títulos académicos y la autonomía de las universidades*, promoviendo la flexibilidad de las enseñanzas, de forma que se garantice que para cada título exista un núcleo común a todo el Estado, y se particularice y complete según los criterios de cada universidad, en función de sus características específicas. Además, se debe facilitar al estudiante que diseñe su propio currículo en función de sus intereses.
3. *Promover la relación entre la Universidad y la sociedad*, teniendo en cuenta las necesidades sociales. Para ello, se propone diversificar el catálogo de títulos oficiales y la ordenación cíclica de la enseñanza. La *diversificación* facilita la creación de especializaciones sujetas al criterio de cada universidad, mientras que el *carácter cíclico* pretende facilitar el acceso simultáneo de los estudiantes al mercado laboral y a su formación universitaria.

Como consecuencia de este documento, se publicó el *Real Decreto 1497/1987*, donde se establecen las *Directrices Generales Comunes* para los *Planes de Estudio* de carácter oficial en el ámbito estatal. Se estableció que los *Planes de Estudio* debían ser elaborados por la universidades teniendo en cuenta las *Directrices Generales Comunes* del Real Decreto y las *Directrices Generales Propias* de cada Título, establecidas también a nivel estatal. Algunas de las características más relevantes de las directrices, presentadas en los documentos anteriores, en relación con los objetivos perseguidos son:

1. *Racionalidad*. La carga lectiva de las titulaciones con frecuencia era excesiva y además se pretendía dar una mayor relevancia a las enseñanzas prácticas. Debido a esto, se establece el crédito como unidad de medida de las enseñanzas (corresponde a 10 horas de enseñanza por parte del docente), y una carga lectiva de entre 60 y 90 créditos al año. Además, establecida la carga semanal de 20 a 30 horas, se determina que, como máximo, se corresponderán 15 horas con enseñanzas teóricas.
2. *Homogeneidad y flexibilidad*. Se pretendía flexibilizar las enseñanzas, garantizando que para cada título existiera un núcleo común de carácter estatal. Debido a esto, se distinguen tres bloques de materias entre los que deben distribuirse los contenidos de los Planes de Estudio:
  - *Materias troncales*. Deben ser incluidas en todos los planes de estudio conducentes a una misma titulación. Sus contenidos son establecidos por el *Consejo de Universidades* en las directrices propias de cada titulación. La carga lectiva en créditos para el conjunto de las materias troncales debe ser como mínimo el 30 % de los créditos totales del plan de estudio en primer ciclo, y el 25 % en segundo ciclo.



- *Materias no troncales.* Son materias seleccionadas por la universidad que establece el Plan de Estudios. Se dividen en dos tipos:
    - *Materias obligatorias.* Son de estudio obligado para todo el alumnado y representan el carácter distintivo que cada universidad pretende en su Plan de estudios.
    - *Materias optativas.* Son de estudio opcional, debiéndose completar un número mínimo de estas. El/la estudiante puede seleccionar las que más se adecúen a sus intereses, orientando de este modo su formación.
  - *Materias de libre configuración.* Ofrecen a la/el estudiante la posibilidad de completar sus estudios universitarios con materias que no están directamente relacionadas con el Título al que opta, o realizar otras actividades como asistencia a conferencias, pertenencia a asociaciones, prácticas en empresa, etc. La carga lectiva en créditos para el conjunto de materias de libre elección debe ser el 10 % de los créditos totales del plan de estudio.
3. *Ciclicidad.* Se pretendía facilitar el acceso simultáneo de los/las estudiantes al mercado laboral y a su formación universitaria. Debido a esto, los estudios se organizan en ciclos, al igual que se proponía en la Ley General de Educación. El primero ciclo tiene una duración de 3 cursos y comprende enseñanzas básicas y de formación general orientadas a la preparación para el ejercicio de actividades profesionales. Por su parte, el segundo ciclo tiene una duración de 2 cursos o, excepcionalmente, de 3 y está dedicado a la profundización y especialización en las correspondientes enseñanzas orientadas a la preparación para el ejercicio de actividades profesionales. La superación de los estudios de primer ciclo da derecho, si así lo establecen las directrices propias de cada titulación, a la obtención de una titulación oficial de Diplomatura, Arquitectura Técnica o Ingeniería Técnica. De forma similar, la superación del segundo ciclo da derecho a la obtención de una Licenciatura, Arquitectura o Ingeniería.

Tras la publicación de los decretos en los que se establecían las directrices propias de cada titulación, el Consejo de Universidades, debido a la confusión reinante en la elaboración de los planes de estudio, elaboró una serie de recomendaciones adicionales. La mayoría de estas recomendaciones están recogidas en el documento “*Recomendaciones del Consejo de Universidades para el diseño y la homologación de planes de estudio*”, de 25 de junio de 1992, y en la “*Modificación parcial del Real Decreto 1497/1987*”, de 28 de junio de 1998.

### 2.2.2. La Ley Orgánica de Universidades (LOU)

En primer lugar, es conveniente señalar que aunque la LOU (Ley Orgánica 6/2001 de Universidades, de 21 de diciembre de 2001) deroga a la LRU, acepta muchos de sus principios, de modo que muchos aspectos de la LRU permanecen no sólo aceptados sino reforzados en la propia LOU. Algunos aspectos que se mantienen son: 1) la estructura departamental de la universidades ortogonal a la organización en centros,

2) la consolidación de la función investigadora del profesorado universitario, 3) la consolidación de los principios de autonomía universitaria y libertad de cátedra, 4) el establecimiento de algunas figuras de profesorado (como las de Catedrático, Titular de Universidad y Asociado), 5) la utilización del crédito como unidad de medida, 6) la ciclicidad de los estudios universitarios, 7) la organización semestral, etc. Otros aspectos se han modificado parcialmente, como es el caso de la elección democrática de los componentes de los órganos rectores (unipersonales y pluripersonales) de las diferentes estructuras (departamentos, centros y la propia universidad). La LOU asigna al profesorado funcionario doctor más peso en la votación y representación de los nuevos claustros<sup>2</sup>.

La LOU surge en una atmósfera favorable para la renovación del sistema universitario, pero desafortunadamente no fue una reforma consensuada por los diferentes actores de la vida universitaria y hubo numerosas protestas de los distintos colectivos. Los objetivos más destacados son:

1. Adaptar la enseñanza superior a los cambios sociales, económicos y tecnológicos de los últimos años en consonancia con la evolución de la Unión Europea. Así, en la exposición de motivos señala: *“es necesaria una nueva ordenación de la actividad universitaria. Ésta, de forma coherente y global, debe sistematizar y actualizar múltiples aspectos académicos, de docencia, de investigación y de gestión, que permitan a las Universidades abordar, en el marco de la sociedad de la información y el conocimiento, los retos derivados de la innovación en las formas de generación y transmisión del conocimiento.”*, y más adelante, *“responder a los retos derivados tanto de la enseñanza superior no presencial a través de las nuevas tecnologías de la información y la comunicación como de la formación a lo largo de la vida”*.
2. Mejorar la calidad docente, investigadora y de gestión de la enseñanza superior, fomentando la cultura de la evaluación y la acreditación. Así en la exposición de motivos señala: *“...integrarse competitivamente junto a los mejores centros de enseñanza superior en el nuevo espacio universitario de educación superior”*.
3. Fomentar la movilidad de estudiantes, profesorado e investigadores. Así, en la exposición de motivos señala: *“fomentar la movilidad de estudiantes y profesores; profundizar en la creación y transmisión del conocimiento como eje de la actividad académica”*.

Para alcanzar estos objetivos, la LOU articula diferentes niveles de competencias para el Gobierno, las Comunidades Autónomas, y las Universidades. Además, modifica la composición y algunos nombres de diferentes Organismos de Gobierno de la Universidad y crea nuevos mecanismos para el acceso del alumnado y profesorado. Algunas de los cambios más relevantes que se llevan a cabo, en relación con los objetivos perseguidos, son:

---

<sup>2</sup>La composición de los claustros la regulan las Juntas de Gobierno de cada Universidad dentro de los márgenes establecidos en la LOU.

1. La elección de nuevos claustros<sup>3</sup>, la elaboración de nuevos estatutos que regirán las universidades<sup>4</sup>, y la creación de nuevos órganos nacionales de gestión y evaluación de la calidad como la *ANECA*, y el Consejo de *Coordinación Interuniversitaria*, que sustituye al *Consejo Superior de Universidades* y amplía sus funciones. Así, este actualmente tiene un papel activo en la selección de profesorado universitario, ya que una de sus funciones es la de convocar y gestionar las pruebas de habilitación.
2. *El establecimiento de la necesidad de evaluación sistemática*, no sólo del profesorado sino también de las titulaciones que se imparten en las universidades (tanto públicas como privadas) y, por lo tanto, de las cualidades del profesorado, tanto en los aspectos relacionados con la docencia como en los relacionados con la investigación y la gestión. Con este objetivo se crea, y en cumplimiento del artículo 32 de la LOU, por acuerdo del Consejo de Ministros, previo informe del Consejo de Ordenación Interuniversitaria, la *Agencia Nacional de Evaluación de la Calidad y Acreditación (ANECA)* en 2002. En numerosas autonomías también se crean *Agencias de Evaluación de la Calidad autonómicas* (por ejemplo, la *ACPUA* en Aragón). Además, también se establece el *II Plan de Calidad de las Universidades (2001-2006)*, para dar continuidad al *I Plan Nacional de Evaluación de la Calidad de las Universidades (1996-2000)*.
3. El establecimiento de la *necesidad de desvinculación* de la universidad donde se ha obtenido la titulación para poder ser contratado en determinadas figuras de profesorado laboral. Esta necesidad se ha eliminado en la Reforma de la Ley Orgánica de Universidades.

Debido a que esta plaza se enmarca dentro de la normativa de la LOU para la selección de profesorado, y por ser uno de los puntos en los que se produce un mayor cambio con respecto a la LRU, a continuación se describe brevemente algunos aspectos relativos a las figuras de profesorado en la LOU.

### Las nuevas figuras de profesorado y su selección

En primer lugar hay que señalar que la LOU establece dos tipos profesorado:

1. *Laboral*. Este tipo de personal docente investigador aparece por primera vez reflejado en la LOU. Cada universidad especificará en sus estatutos el sistema de selección, garantizando que se cumplen los principios de igualdad, publicidad y competencia.
2. *Funcionario*. Este tipo de personal es el que tradicionalmente ha tenido la Universidad para realizar tareas de docencia e investigación. Hasta la entrada en vigor de la LOU, todo el profesorado universitario era funcionario.

---

<sup>3</sup>Julio de 2002 fue la fecha límite para la constitución de los nuevos claustros.

<sup>4</sup>Los nuevos claustros se encargaron de elaborar los nuevos estatutos y tenían para ello un plazo de 9 meses.

Además, dentro del profesorado universitario laboral se establecen cinco categorías: *Ayudante*, *Ayudante Doctor*, *Contratado Doctor*, *Asociado*, *Colaborador*, y *Funcionario*. A continuación, se resumen brevemente las características de cada una de las categorías.

1. *Ayudante*. Se entiende que esta figura constituye la entrada al sistema de enseñanza universitaria como profesor/a, por lo que no es necesario recibir ningún tipo de acreditación positiva de la ANECA. (u órgano autonómico correspondiente) ni tampoco estar en posesión del título de Doctor: el único requisito es haber superado el período investigador correspondiente a tercer ciclo. Este tipo de contratos tienen una duración limitada de 4 años<sup>5</sup>.
2. *Ayudante Doctor*. Se entiende que esta figura trata de consolidar al profesorado ayudante en la labor docente y reconoce su capacidad investigadora. Para poder acceder a este tipo de plaza es necesario recibir una evaluación positiva de la ANECA (u órgano autonómico correspondiente) y estar en posesión del título de Doctor. Actualmente también es necesario, para tener este tipo de contrato, haber pasado al menos dos años realizando trabajo docente o investigador en una entidad diferente a aquella en la que se ingresa<sup>6</sup>. Este tipo de contratos tienen una duración limitada de 4 años<sup>7</sup>.
3. *Contratado Doctor*. Se entiende que esta figura consolida al profesorado en su labor docente e investigadora. Para poder acceder a este tipo de plaza es necesario recibir una evaluación positiva de la ANECA (u órgano autonómico correspondiente) y estar en posesión del título de Doctor con 3 años de experiencia postdoctoral. Este tipo de contratos no tienen una duración limitada.
4. *Asociado*. Se entiende que esta figura trata de acercar a la Universidad a profesionales de la empresa para que aporten su punto de vista y experiencia. Por ello, se trata de contratos con dedicación a tiempo parcial y nunca a tiempo completo, a diferencia de lo que establecía la LRU. Para poder acceder a este tipo de plazas, no es necesario recibir evaluación positiva de la ANECA (u órgano autonómico correspondiente) ni estar en posesión del título de Doctor, pero sí experiencia profesional acreditada. Este tipo de contratos no tienen una duración limitada.
5. *Colaborador*. Se entiende que esta figura trata acercar a la Universidad a personas con experiencia en la empresa para que aporten sus experiencias. Para poder optar a este tipo de plaza es necesario recibir una evaluación positiva de la ANECA (u órgano autonómico correspondiente). Este tipo de contratos no tienen una duración limitada<sup>8</sup>.

<sup>5</sup>En la Ley Orgánica de Reforma Universitaria, este límite es de 5 años.

<sup>6</sup>Este requisito se ha eliminado en la Ley Orgánica de Reforma Universitaria.

<sup>7</sup>En la Ley Orgánica de Reforma Universitaria, este límite es el mínimo entre 5 años y 8 años menos los años que se haya ejercido como Ayudante.

<sup>8</sup>Esta figura desaparece en la Ley Orgánica de Reforma Universitaria, pero el personal con este tipo de contrato mantendrá sus derechos.

6. *Visitante.*

7. En lo que respecta a *profesorado funcionario*, las figuras generales que establece la LOU son las mismas figuras de *Titular de Universidad* y *Catedrático de Universidad* ya existentes en ese momento, y también considera las de *Catedrático de Escuela Universitaria* y *Titular de Escuela Universitaria*<sup>9</sup>. Por tanto, desaparecen con carácter general las figuras de profesorado de Escuela Universitaria, aunque con carácter excepcional en algunas titulaciones que tradicionalmente fueron de grado medio (como, por ejemplo, Enfermería) se mantienen.

Para el acceso del profesorado a los cuerpos de funcionarios docentes, la LOU introduce un sistema de *Habilitación Nacional* en el *Real Decreto 774/2002, de 26 de julio de 2002*. Corresponde al *Consejo de Coordinación Interuniversitaria* la convocatoria de las pruebas de habilitación a nivel estatal, así como el sorteo de los tribunales que evaluarán a las personas que se presenten, siguiendo la composición diseñada por la LOU. En cada convocatoria, el número de habilitaciones para un área de conocimiento estará en función de las vacantes existentes en las diferentes universidades del Estado en ese momento. Actualmente, el número de habilitaciones es el número de vacantes multiplicado por un factor de 1.5 y redondeado al mayor entero, de modo que se garantiza que el número de personas habilitadas es superior al de vacantes, para dar a las universidades cierta posibilidad de elección entre las personas habilitadas para cubrir sus vacantes. Una vez celebradas las pruebas de habilitación correspondientes, las universidades que tienen vacantes pueden convocar un concurso-oposición para cubrirlas. Cada universidad establece en sus estatutos el sistema que rige el concurso-oposición, pero para poder presentarse será requisito mínimo estar habilitado/a en el área de la plaza a concurso. Además, en la LOU se establece que una universidad sólo podrá dejar sin cubrir una vacante durante un año.

### 2.2.3. La Reforma de la LOU

Desde que se promulgó la LOU se vinieron detectando una serie de deficiencias, que aconsejaron su revisión. Además, en la LOU existían numerosos puntos en los que la *Conferencia de Rectores de las Universidades Españolas (CRUE)* no se mostraba de acuerdo, como queda de manifiesto en el “*Documento de la asamblea general de la CRUE sobre cambios en la LOU y orientaciones de las políticas universitarias*”, de 24 de junio de 2004. Por ello, se elabora el *Anteproyecto de Ley Orgánica de 30 de junio de 2006*, que posteriormente se convierte en el *Proyecto de Ley Orgánica de 1 de septiembre de 2006*, y que ha sido aprobado recientemente. En esta reforma se pretenden corregir algunas deficiencias en el funcionamiento de sistema anterior, como por ejemplo el sistema de habilitación nacional, y buscar un consenso mayor. Además, debido a que otros elementos del entorno han cambiado, entre los que destacan los acuerdos en política del EEES, se precisan modificaciones. Los objetivos más destacados de esta reforma son:

---

<sup>9</sup>En la Ley Orgánica de Reforma Universitaria, estas dos últimas figuras desaparecen.

1. *Potenciar la autonomía de las universidades.* Se pretende que esta autonomía permita a las universidades responder con flexibilidad y rapidez a los cambios en las necesidades de una sociedad del conocimiento dinámica y globalizada.
2. *Responder a las demandas de la sociedad y el sistema productivo, facilitando la transferencia con el mundo empresarial.* Se pretende que las universidades persigan una mejor formación de sus graduados para que sean capaces de adaptarse a las demandas sociales y del sistema científico y tecnológico. Así, en el artículo 39 (y también en el 41) se enfatiza como objetivo esencial de la Universidad la transferencia de conocimiento a la sociedad.
3. *Potenciar el papel y la responsabilidad de todos los agentes del sistema universitario, articulando mejor la relación entre ellos.* Actualmente, las Comunidades Autónomas son las responsables de la política universitaria, de acuerdo con la Constitución y los Estatutos de Autonomía, mientras que el Estado tiene que establecer las normativas básicas para su desarrollo. Por ello, es necesario establecer mecanismos de coordinación y cooperación adecuados.
4. *Reforzar la necesidad de evaluación para garantizar la calidad.* El fomento de la calidad y su evaluación para llegar a ser más competitivos ha sido una constante en la ordenación universitaria en los últimos años. En esta nueva ley se incide de nuevo en estos aspectos.
5. *Formar globalmente y en valores.* Se considera que la práctica del deporte es un aspecto importante en la formación de los estudiantes universitarios. Además, se destaca el papel de la Universidad como transmisor esencial de valores para alcanzar una sociedad tolerante e igualitaria.

Para alcanzar estos objetivos, la Reforma de la LOU promueve diversos cambios. Algunos de los más relevantes son:

1. *La normativa en lo referente a la elección del rector.* La vigente normativa al respecto fue uno de los aspectos que mayor controversia despertó de la LOU. La nueva normativa pretende flexibilizar el sistema de elección, permitiendo que cada universidad pueda elegir la opción que considere más adecuada.
2. *La vinculación entre la investigación universitaria y el entorno productivo.* Una de las medidas importantes que considera al respecto es la creación de institutos mixtos de investigación. Además, prevé potenciar los mecanismos de intercambio de personal investigador entre el sistema universitario y el productivo.
3. *La creación de órganos que faciliten la coordinación y cooperación entre los distintos agentes del ámbito universitario.* Entre otras medidas, se determina la creación de la *Conferencia General de Política Universitaria*, y se constituye el *Consejo de Universidades* con funciones de asesoramiento, cooperación, coordinación, verificación, y aprobación en el ámbito académico. Además, se pretende incentivar la participación de los estudiantes en el sistema universitario; para

ello, se prevé la elaboración de un *Estatuto del Estudiante Universitario* y la creación del *Consejo de Estudiantes Universitarios*.

4. *El refuerzo del papel de la Agencia Nacional de Evaluación de la Calidad y Acreditación (ANECA)*. Dado el importante papel que ésta desempeña en el binomio autonomía-rendición de cuentas, se autoriza su creación como *agencia de acuerdo* con la *Ley de Agencias Estatales* para la mejora de los servicios públicos. Con ello, se facilita la coordinación en los procesos de garantía de calidad y la definición de criterios de evaluación. Además, la ANECA también será la responsable de establecer un *modelo de acreditación* para la selección del profesorado funcionario. De este modo, *se elimina el sistema de habilitación* que resultó ser poco eficiente y muy caro. Las universidades seleccionarán a su profesorado entre los previamente acreditados.
5. *La inclusión de un título dedicado al deporte universitario*, debido a que la práctica del deporte se considera un aspecto importante en la formación de los estudiantes universitarios. Además, la ley tampoco olvida el papel de la Universidad como transmisor esencial de valores para alcanzar una sociedad tolerante e igualitaria, en la que se respeten los derechos y libertades fundamentales y de igualdad entre hombres y mujeres. Entre otras medidas, se considera el establecimiento de sistemas que permitan alcanzar la paridad en los órganos de representación y una mayor participación de la mujer en los grupos de investigación. Además, se introduce la creación de programas específicos sobre la igualdad de género, de ayuda a las víctimas del terrorismo, y el impulso de políticas activas para garantizar la igualdad de oportunidades a las personas con discapacidad.

Otro aspecto importante de la Reforma son los diversos cambios que se establecen en lo referente a las distintas figuras de contratación del Personal Docente e Investigador de las universidades. Los más importantes son la desaparición de las figuras de Catedrático de Escuela Universitaria, Titular de Escuela Universitaria, y Profesor Colaborador. También conviene señalar que el 2 de febrero de 2005 la *Comisión Académica del Ministerio de Educación* aprobó el *Procedimiento para el reconocimiento de equivalencias de categorías de profesorado de las Universidades de la Unión Europea y del Espacio Europeo Común, a efectos de la habilitación nacional*.

#### 2.2.4. La adaptación al EEES en España

Todos los países firmantes de la Declaración de Bolonia han emprendido las reformas legislativas pertinentes para adaptarse al Espacio Europeo de Educación Superior. En lo que respecta a la legislación española, ya en la LOU se dejan ver una serie de actuaciones. Así, el *Título XIII de la LOU* se dedica exclusivamente al Espacio Europeo de Educación Superior y se señala que: “*En el ámbito de sus respectivas competencias el Gobierno, las Comunidades Autónomas y las Universidades adoptarán las medidas necesarias para la plena integración del sistema español en el espacio europeo de enseñanza superior*”. En este título fundamentalmente se hace referencia a la adaptación

de los títulos y movilidad de alumnado y profesorado. De hecho, conviene destacar que en uno sus artículos se asigna al Gobierno, tras informe del Consejo de Coordinación Universitaria, la adaptación de créditos y títulos.

En febrero del 2003, el Ministerio de Educación Cultura y Deporte, elabora el *Documento Marco para la Integración del Sistema Universitario Español en el Espacio de Enseñanza Superior*. Este documento pretendía ser una propuesta del proceso seguido para armonizar las enseñanzas universitarias y asume los principios fundamentales de Bolonia y los documentos posteriores. De hecho, se asume que existirá un primer ciclo de Grado, que conducirá a la obtención del título de Licenciatura (o Ingeniería) que concederá cualificación profesional en el mercado europeo, y un segundo ciclo de Posgrado que conducirá a la obtención de los títulos de Máster. Con respecto a las titulaciones, uno de los párrafos que suscitó mayor controversia en el momento de su publicación fue el tercero del artículo 4: “... deberá definirse un catálogo de títulos de primer nivel, tomando como punto de partida el actual pero propiciando una disminución mediante fusiones o agrupaciones necesarias para racionalizar el conjunto tanto desde el punto de vista nacional como europeo”.

También se asume el *Suplemento Europeo al Título* y el *Sistema Europeo de Créditos ECTS*, que posteriormente fueron elaboradas en los siguientes Reales Decretos:

1. *Real Decreto 1044/2003, de 1 de agosto*, por el que se establece el procedimiento para la expedición por las universidades del Suplemento Europeo al Título. El Consejo de Coordinación Universitaria, a raíz de la publicación del Real Decreto 1044/2003, pone a disposición de las universidades un *Modelo de Suplemento Europeo*, de conformidad con los modelos que se incluyen como anexos en el citado Real Decreto, con el objeto de facilitar a las universidades la expedición del Suplemento Europeo.
2. *Real Decreto 1125/2003, de 5 de septiembre*, por el que se establece el Sistema Europeo de Créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y validez en todo el territorio nacional.

Además, en el *Documento Marco* también se dedica un capítulo a la acreditación y a la calidad académica. Por último, en el apartado 7 se proponía cómo realizar la implantación paulatina del nuevo sistema y la transición de las titulaciones actuales a las nuevas.

Posteriormente, la *Conferencia de Rectores de Universidades Españolas (CRUE)*, el 12 de septiembre de 2003, manifiesta su apoyo a las medidas tomadas y coincide en una de las más polémicas; así en la *Declaración sobre el Espacio Europeo de Enseñanza Superior* puede leerse: “Ha de mantenerse un solo título de Grado, aunque pueda tener duración distinta, tal como se enuncia en el Real Decreto, con objetivos amplios y formación polivalente, con el fin de potenciar la adecuación al empleo de los titulados universitarios”. Además, mantiene la intención de seguir colaborando activamente en el proceso de convergencia europea en materia de educación, y así lo vuelve a manifestar en la *Declaración sobre el Espacio Europeo de Enseñanza Superior* un mes más tarde.



En los meses siguientes culminaron las medidas para la homologación de títulos, aprobándose una serie de Reales Decretos, entre los que conviene destacar los siguientes:

1. *Real Decreto 1272/2003, de 10 de octubre*, por el que se regulan las condiciones para la declaración de equivalencia de títulos.
2. *Real Decreto 49/2004, de 19 de enero*, sobre homologación de planes de estudios y títulos de carácter oficial y validez en todo el territorio nacional.
3. *Real Decreto 285/2004, de 20 de febrero*, por el que se regulan las condiciones de homologación y convalidación de títulos y estudios extranjeros de Educación Superior. Fue modificado posteriormente por el *Real Decreto 309/2005 de 18 de marzo*. Esto también se ve reflejado en las diferentes órdenes emitidas por el Ministerio de Educación y Ciencia (*ECI*). Así, en la *Orden ECI/1712/2005 de 2 de junio*, se modifica la Orden ECI/3686/2004 de 3 de noviembre, por la que se dictaban normas para la aplicación del Real Decreto 285/2004 de 20 de febrero.

En actual camino hacia la convergencia, dos hitos importantes fueron la elaboración de los Libros Blancos de las titulaciones (2003-2004) y la Intervención de la Excm. Sra. Ministra de Educación y Ciencia en el Pleno del Consejo de Coordinación Universitaria el 28 de Julio de 2004:

- Un *Libro Blanco* es un documento oficial que contiene información sobre el estado de alguna cuestión o materia, en forma de programa, proyecto o propuesta de actuación, elaborada sobre un tema concreto para su posterior discusión o sometimiento a la opinión pública. En el marco de la convergencia del sistema de Educación Superior español al EEES, la red de universidades españolas, apoyada por la ANECA, y con el objetivo explícito de estudiar supuestos prácticos útiles en el diseño de títulos de Grado adaptados al EEES, elaboró Libros Blancos para diferentes titulaciones: Ingeniería en Informática, Enfermería, Logopedia, Ingeniería Industrial, Derecho, etc. En el caso de los estudios de Informática, que es donde se encuadra este proyecto docente, se elaboró dentro del proyecto EICE el Libro Blanco "*Título de Grado en Ingeniería Informática*".
- En el discurso de la Excm. Sra. Ministra de Educación, ésta manifiesta las líneas de actuación que el Ministerio se propone desarrollar en los siguientes meses en relación con la política universitaria y destaca tres líneas básicas: 1) el impulso y la colaboración en la definición y desarrollo del EEES, 2) la elaboración de un marco de referencia para la financiación de las universidades públicas, y 3) la modificación y desarrollo de la legislación de ese momento buscando consenso.

En este discurso de la Excm. Sra. Ministra, ya se dejaba ver lo que pocos meses después se concretó en los Reales Decretos que establecieron las bases para los títulos de Grado y Posgrado:

1. *Real Decreto 55/2005 de 21 de enero*, por el que se establece la estructura de las enseñanzas universitarias y se regulan los estudios universitarios oficiales de Grado. En este documento se establece que el Gobierno realizará un nuevo catálogo de títulos académicos de Grado, de carácter oficial y con validez en todo el territorio estatal, y deja claro que la enseñanza que conduzca a la obtención de títulos de Grado tendrá un carácter generalista, por lo que no podrá incorporar en ningún caso el reconocimiento de especialidades.

Además, se establece que cada título oficial de Grado será oficializado mediante un Real Decreto donde se detallarán todas las características y Directrices Generales para su implantación. En este punto se especificarán necesariamente las materias troncales de las cuales debe constar, el número de créditos ECTS de cada una de ellas, y las áreas de conocimiento a que debe pertenecer el profesorado que las imparta. En lo que respecta a materias obligatorias, optativas y de libre elección, se especifica que el número máximo de créditos que pueden representar en una titulación son el 30 %, 20 %, y 5 %, respectivamente. También se aclara que las diferentes titulaciones de Grado serán de 180 ó 240 créditos, con cursos de 60 créditos por año, y que las titulaciones tendrán la misma duración en todo el Estado.

También es de destacar que en el párrafo 6 del artículo 7 se aclara la relación que debe existir entre horas de clase presencial y créditos ECTS, indicando que “... las horas dedicadas a la docencia, teóricas y prácticas, de los planes de estudios no podrán exceder del 50 % del total de los créditos asignados a la materia de que se trate”. Por tanto, 1 crédito ECTS no podrá corresponder a más de 15 horas de clase presencial.

En este documento, también se establece el proceso a desarrollar para la elaboración y aprobación de un Plan de Estudios. Una vez que el Gobierno fije las directrices de un nuevo título académico, los nuevos planes de estudios deberán ser aprobados e iniciados en un plazo máximo de tres años.

2. *Real Decreto 56/2005 de 21 de enero*, por el que se regulan los estudios universitarios oficiales de Posgrado. Lo primero que se establece es que los títulos de Posgrado pueden estar orientados a la obtención de un título de Máster o a la de un título de Doctorado. Respecto a los estudios oficiales de Máster, se establece que podrán constar de 60 ó 120 créditos, según correspondan a 1 ó 2 años de duración. En cualquier caso, estos estudios se orientarán a la formación académica avanzada, especializada y/o multidisciplinar, o bien a la especialización académica o profesional o, en su caso, al inicio de la investigación. Para finalizar los estudios de Máster y obtener el título correspondiente, el/la estudiante deberá realizar una exposición pública ante un tribunal. En relación a los estudios de Doctorado, se especifica que para ser admitido a un programa de Doctorado es necesario haber obtenido el título de Máster o superado al menos 60 ECTS en Programas Oficiales de Posgrado y un mínimo de 300 entre Grado y Posgrado (260 de Grado + 60 de Máster o 180 de Grado + 120 de Máster). Para la obtención del título de Doctorado habrá que elaborar un trabajo original

y propio de investigación sobre alguna materia relativa al campo del programa de Posgrado que dará lugar a la Tesis de Doctorado.

3. *Real Decreto 1509/2005 de 16 de diciembre*, por el que se modifican el Real Decreto 55/2005 y el Real Decreto 56/2005.
4. *Real Decreto 189/2007 de 9 de febrero*, por el que se modifican el determinadas disposiciones del Real Decreto 56/2005 de 21 de enero, por el que se regulan los estudios universitarios oficiales de Posgrado.
5. El Senado aprueba una enmienda a la LOU, presentada por el grupo socialista, por la que se introduce la *figura de Investigador de Universidad*, el día 16 de febrero de 2007.

Resumiendo, según lo anterior, en España habrá dos niveles, el Grado y el Posgrado, y tres ciclos: Grado (1ª ciclo), Máster (2ª ciclo), y Doctorado (3ª ciclo). En general, las titulaciones de Grado tendrán un carácter generalista y una carga de 240 créditos ECTS. El Máster puede tener una carga de 60 ó 120 créditos ECTS, y se centrará en la formación especializada. Para poder realizar un Máster es necesario tener el título de Grado correspondiente, o haber superado al menos 180 ECTS de este incluyendo la totalidad de los contenidos formativos comunes. Por último, el Doctorado es muy similar al actual y concluirá con la defensa de la tesis.

En los últimos años el trabajo se ha centrado en el diseño de los nuevos planes de estudio y en la realización de proyectos piloto. Desde el Ministerio y la ANECA se han lanzado numerosas convocatorias para financiar esto; algunas de más conocidas son:

1. *Orden ECI/924/2005, de 21 de marzo*, del Ministerio de Educación y Ciencia, por la que se establecen las bases reguladoras y se convocan ayudas para la financiación de propuestas orientadas a diseñar la adecuación de las instituciones universitarias dentro del marco del Proceso de Bolonia e impulsar acciones para promover la construcción del Espacio Europeo de Educación Superior (EEES) y la adaptación progresiva de nuestro sistema de Educación Superior al mismo.
2. *Las diferentes convocatorias de la ANECA para acciones sobre el EEES* como la I, II y III convocatoria para el diseño de planes de estudio y títulos de Grado.

Además, también existen numerosos documentos de trabajo publicados por el Ministerio de Educación, entre los que destacamos:

1. Propuesta de *la organización de las enseñanzas universitarias en España* de 26 de septiembre de 2006.
2. Aclaraciones sobre el documento de 26 de septiembre de 2006 sobre *la organización de las enseñanzas universitarias en España*, de 6 de noviembre de 2006.
3. Propuesta de las directrices para la *elaboración de títulos universitarios de Grado y Máster*, de 21 de diciembre de 2006.

4. *Borrador Real Decreto 806/2006 de Acceso a Estudiantes Extranjeros.*
5. Informe de las subcomisiones del Consejo de Coordinación universitaria acerca de *La organización de las enseñanzas universitarias en España*, de 14 de noviembre de 2006.
6. Propuesta para el debate por las subcomisiones del Consejo de Coordinación universitaria acerca de las *Materias Básicas por Ramas* de 15 de febrero de 2007.

Para finalizar, me gustaría destacar que se está realizando un gran esfuerzo. Sin embargo, las ayudas financieras son todavía insuficientes, como se refleja en el documento *Trends in European Higher Education V*.

### 2.3. El sistema universitario de Aragón

La Universidad Española creció de forma relevante durante la segunda mitad del siglo pasado. Actualmente, el número de centros universitarios está alrededor del millar, agrupados en aproximadamente 70 universidades (a principios de los 80 había alrededor de 30), la mayoría de ellas de carácter público y presencial con más de un millón de alumnos matriculados. Al mismo tiempo que se experimentaba el crecimiento, aumentó la oferta formativa, dando lugar a la aparición de nuevos títulos como el de Ingeniería en Informática.

En Aragón, que es donde se enmarca este proyecto docente, hay que tener en cuenta también su normativa autonómica, ya que sus competencias en materia de universidades le fueron transferidas a través del Real Decreto 96/1996 de 26 de enero. El marco legislativo que rige actualmente a nivel autonómico es la *Ley de Ordenación del Sistema Universitario de Aragón (LOSUA)*, de julio de 2005, que tiene como objeto ordenar y coordinar el sistema universitario de Aragón y detallar los principios y objetivos de este.

Actualmente, en Aragón existen dos universidades: la *Universidad de Zaragoza*, de carácter público, y la recientemente creada *Universidad de San Jorge*, de carácter privado. La Universidad de Zaragoza es la más antigua y grande de Aragón, y ofrece un abanico muy amplio de titulaciones en las diferentes ramas: literaria, técnica, sanitaria, y social, no estando especializada en un campo concreto de conocimiento. Por otro lado, la Universidad de San Jorge oferta las siguientes titulaciones: Periodismo, Publicidad, Relaciones Públicas e Ingeniería en Informática.

Nos centraremos en la Universidad de Zaragoza (en la Sección 2.3.1), y en concreto en los organismos que más influyen en el desarrollo de la docencia de *compiladores*: el centro donde se imparte (*Centro Politécnico Superior*, en la Sección 2.3.2), y el departamento que tiene asignada su docencia (*Departamento de Informática e Ingeniería de Sistemas*, en la Sección 2.3.3). Además, para finalizar se hará un recorrido sobre las diferentes actuaciones que están llevando a cabo la Universidad, el Departamento, el Área, y el centro en el proceso de convergencia hacia el Espacio Europeo de Educación Superior (en la Sección 2.3.4).

### 2.3.1. La Universidad de Zaragoza

La Universidad de Zaragoza es una de las más antiguas de las Universidades Españolas. En esta sección nos centramos en la Universidad de Zaragoza, en su normativa, su estructura y sus recientes actuaciones.

#### Normativa de la Universidad de Zaragoza

La normativa legal a tener en cuenta en el contexto de la Universidad de Zaragoza, además de la LOU, sus decretos asociados y la normativa autonómica, es principalmente la siguiente:

1. *El Decreto 84/2003 de 29 de abril, del Gobierno de Aragón*, por el que se regula el régimen jurídico y retributivo del Personal Docente e Investigador contratado de la Universidad de Zaragoza (BOA de 19 de mayo).
2. *El Decreto 1/2004, de 13 de enero, del Gobierno de Aragón*, por el que se aprueban los Estatutos de la Universidad de Zaragoza (BOA de 19 de enero y BOA de 27 de febrero, donde se realizan ligeras correcciones de errores, y BOE de 24 de marzo de 2004).
3. *El Convenio Colectivo del personal laboral de la Universidad de Zaragoza*, (BOA de 19 de julio de 2000).
4. *El I Convenio Colectivo de la Universidad de Zaragoza para el Personal Docente e Investigador contratado Laboral* (BOA de 30 de junio de 2006).

Es importante destacar aquí que en el artículo 3 de los *Estatutos de la Universidad de Zaragoza* se establecen los fines al servicio de la sociedad y en el ejercicio de su autonomía, y en el artículo 4 las acciones a desarrollar en un marco de compromiso con la calidad en sus diferentes actuaciones en la docencia, la investigación y la promoción de valores y deporte. Así, en la línea marcada por los estatutos la Universidad de Zaragoza, elaboró su *Plan Estratégico (2002-2005)*<sup>10</sup>, y programas como el *Programa de Actuación 2006*, los cuales afectan a los diferentes ámbitos: docencia (centros), investigación (departamentos) y gestión (servicios).

En relación a la calidad, se puede destacar la creación de la *Unidad de Calidad y Racionalización*<sup>11</sup>, encargada del asesoramiento de los órganos directivos de la Universidad de Zaragoza para la toma de decisiones, mediante la elaboración de informes y estudios, así como del apoyo técnico para lograr un sistema de gestión basado en la mejora continua, con el objetivo de hacer efectiva la política de calidad de la Universidad de Zaragoza. Concretando en la evaluación de la docencia, hay que señalar que el *Comité de Empresa del Personal Docente e Investigador Laboral* de la Universidad de Zaragoza acordó aprobar una propuesta de normativa de la evaluación docente el 22 de septiembre de 2006.

<sup>10</sup>[http://www.unizar.es/plan\\_estrategico/](http://www.unizar.es/plan_estrategico/)

<sup>11</sup>[http://www.unizar.es/unidad\\_calidad/](http://www.unizar.es/unidad_calidad/)

En relación a la promoción de valores, hay que destacar la creación del *Servicio de Gestión Social* en mayo de 2004 y la puesta en marcha de un sitio web para fomentar la igualdad:

- El *Servicio de Gestión Social* integra todas las actividades sociales que se realizan desde el *Vicerrectorado de Proyección Social y Cultural y Relaciones Institucionales*. Sus principales objetivos son: informar, fomentar, gestionar y difundir iniciativas desarrolladas en la Universidad de Zaragoza destinadas a fines solidarios, de justicia social, de voluntariado, de cooperación al desarrollo, etc., además de colaborar con diferentes organizaciones que persigan los mismos fines.
- El sitio web <http://www.unizar.es/igualdad> surge a petición de los integrantes de la *Iniciativa para la Igualdad de oportunidades entre hombres y mujeres en el ámbito de la comunidad universitaria*, apoyada por el *Vicerrectorado de Ordenación Académica*. Sus objetivos prioritarios son: informar y promover en toda la comunidad la reflexión acerca de los temas de igualdad, en los ámbitos de docencia, investigación y relaciones laborales, así como fomentar la tolerancia y la convivencia entre grupos y personas de la Universidad de Zaragoza en particular y de la sociedad en general.

En relación a la promoción del deporte, hay que destacar la creación del *Servicio de Actividades Deportivas (S.A.D)*, que tiene como objetivo principal facilitar la actividad físico-deportiva a todos los miembros de la comunidad universitaria, para contribuir a su formación personal y mejorar su calidad de vida. También existe una *Comisión de Actividades Deportivas de la Universidad*, *Comisiones de Actividades Deportivas de Centros*, y diferentes secciones, asociaciones deportivas y clubes.

### Estructura de la Universidad de Zaragoza

La Universidad de Zaragoza, fundada en 1583, tiene sus dependencias distribuida en cinco poblaciones aragonesas: Jaca, Huesca, La Almunia de Doña Godina, Teruel y Zaragoza.

- En **Jaca** se encuentra la *Residencia Universitaria*, donde se realizan diferentes actividades, entre las que pueden destacarse las *Escuelas de Verano*.
- En **Huesca**, el segundo campus más grande, se encuentran:
  - La *Escuela Politécnica Superior*, donde se imparten las titulaciones oficiales de Ingeniería Técnica Industrial, Ingeniería Técnica Agrícola e Ingeniería Agrónoma.
  - La *Escuela Universitaria de Enfermería*, donde se imparte la titulación oficial de Diplomatura en Enfermería.
  - La *Facultad de Ciencias Humanas y de la Educación*, donde se imparten las titulaciones oficiales de Licenciatura en Humanidades, Diplomatura de Magisterio, y el programa de Doctorado “Realidad, Ficción y Mentira en

las Humanidades”. Además, se imparte el Máster propio en “Museos: Educación y Comunicación”.

- *Facultad de Ciencias de la Salud y del Deporte*, donde se imparten las titulaciones oficiales de Diplomatura en Nutrición Humana y Dietética, Licenciatura en Ciencias de la Actividad Física y del Deporte, Licenciatura en Medicina (sólo el primer ciclo), y Licenciatura en Odontología. Dentro del mismo edificio en el que se encuentra ubicada esta facultad, también se encuentra el Colegio Mayor Universitario “Ramón Acín”.
  - *Escuela Universitaria de Estudios Empresariales*, donde se imparten los títulos oficiales de Diplomatura en Ciencias Empresariales, Diplomatura en Relaciones Laborales, y Diplomatura en Gestión y Administración Pública. Además, se imparten los estudios propios de segundo ciclo en Dirección y Planificación del Turismo, Diplomatura de Especialización en Recursos Técnicos y Humanos para el Desarrollo de la Comunicación en la Empresa Turística, Diplomatura de Especialización en Funcionamiento y Mejora de las Instituciones Públicas, y el Posgrado en Políticas Demográficas y Económicas frente a la Despoblación.
- En **La Almunia de Doña Godina** se encuentra la *Escuela Politécnica de La Almunia de Doña Godina*, donde se imparten las titulaciones de Ingeniería Técnica en Informática de Sistemas, Ingeniería Técnica Industrial, Ingeniería Técnica Agrícola, Arquitectura Técnica, e Ingeniería Técnica de Obras Públicas.
- En **Teruel** se encuentran:
- La *Facultad de Humanidades y Ciencias Sociales*, donde se imparten las titulaciones oficiales de Licenciatura en Humanidades, Diplomatura en Magisterio, Diplomatura en Relaciones laborales, y Licenciatura en Ciencias del Trabajo. Además, se imparten los estudios propios de Diploma de Educación en el Medio Rural, Diploma de Especialización en Formación Práctica en Geografía e Historia, Diploma de Especialización en Educación Ambiental, y Diploma de Especialización en Restauración Hostelera.
  - La *Escuela Universitaria Politécnica*, donde se imparten las titulaciones oficiales de Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Telecomunicaciones. Además, se imparten el título propio de Diploma de Especialización de Administración de Sistemas de Información.
  - El *Colegio Mayor Universitario Pablo Serrano*.
  - La *Escuela Universitaria de Enfermería*, donde se imparte la titulación oficial de Diplomatura en Enfermería.
- En **Zaragoza** se encuentra la sede de la Universidad, y existe un mayor número de centros; por ello, se distribuyen en 5 campus:
- *Campus Río Ebro*, donde se encuentran;

- La *Escuela Universitaria de Ingeniería Técnica Industrial*, donde se imparten las titulaciones oficiales de Ingeniería Técnica en Diseño Industrial e Ingeniería Industrial. Además, se imparten los siguientes Diplomas de Especialización: en Análisis Tecnológicos y Sociolaboral de los Sistemas Productivos, en Director de Seguridad, en Diseño, Instalación y Mantenimiento de Sistemas Eléctricos de Control y Accionamientos Industriales, en Gestión de Sistema de Calidad, y en Técnicas Medioambientales en la Empresa.
- El *Centro Politécnico Superior*, distribuido en los edificios Ada Byron y Torres Quevedo, donde se imparten las titulaciones oficiales de Ingeniería en Informática, Ingeniería en Telecomunicaciones, Ingeniería Industrial, e Ingeniería Química. Además, también se ofertan estudios propios y programas de Doctorado: Máster de Logística, Máster en Mercados Energéticos (dos modalidades: presencial y online), Máster Europeo de Energías Renovables (dos modalidades: presencial y online), Máster en Ahorro Energético (dos modalidades: presencial y online), Máster on-line en Generación Eléctrica Ciclos Combinado, Cogeneración y Sistemas Avanzados, Máster en Servicios Web, Seguridad Informática y Aplicaciones de Comercio Electrónico, Máster Interuniversitario Europeo en Polímeros y Procesos de Transformación y Manufactura del Producto, Máster en Ingeniería del Medio Ambiente, Máster Europeo en Dirección de Proyectos, Máster en Tecnologías de la Información y Comunicaciones en Redes Móviles, Posgrado en Ingeniería de los Recursos Hídricos, Posgrado en Ingeniería de Organización Industrial, Posgrado en Ingeniería Biomédica, Posgrado en Calidad Ambiental y Desarrollo Sostenible, Posgrado en Ingeniería de Climatización, Posgrado en Nuevas Tecnologías Energéticas Sostenibles (on-line), Posgrado en Generación Combinada de Electricidad, Calor, Frío y Agua (on-line), Diploma de Especialización en Tecnologías del Hidrógeno y Pilas de Combustible, Diploma de Especialización en Sistemas de Ayuda a la Discapacidad y Rehabilitación, Diploma de Especialización en Técnicas de Ayuda a la Cirugía y Planificación Preoperatoria, Diploma de Especialización en Técnicas de Diagnóstico Avanzado usando las Tecnologías de la Información y las Comunicaciones, Diploma de Especialización en Telemedicina y Sistemas de Información en Salud. Este es el centro donde se imparte la docencia de *Compiladores I*, por lo que se darán algunos datos más acerca de él en la siguiente sección.
- La *Escuela Universitaria de Estudios Empresariales*, donde se imparten la titulación oficial de Diplomatura en Ciencias Empresariales, el Máster en Mercado Inmobiliario, el Diploma de Especialización en Mercados Inmobiliarios, y el Diploma de Especialización en Asesoramiento Integral de Empresas.
- *Escuela Universitaria de Turismo*, donde se imparte la titulación oficial de



Diplomatura en Turismo.

- *Plaza de Basilio Paraíso*, donde se encuentra el Paraninfo de la Universidad y la Facultad de Ciencias Económicas y Empresariales, donde se imparten las titulaciones oficiales de Licenciatura en Administración y Dirección de Empresas y Licenciatura en Economía.
- *Campus Plaza San Francisco*. Se trata del campus más grande de la Universidad de Zaragoza, y en él se encuentran la mayoría de los edificios dedicados a servicios de gestión y administrativos, como:
  - *Edificio Interfacultades*, dedicado fundamentalmente a servicios de gestión y administración, aunque allí también tiene su sede el *Instituto de Idiomas* de la Universidad.
  - *Servicio de Información*, que ofrece información de carácter general sobre la Universidad de Zaragoza: acceso, oferta de estudio, planes de estudio, títulos propios, becas, convalidaciones, cursos de verano, etc.

Los edificios donde se imparten las diferentes titulaciones son:

- *Escuela Universitaria de Ciencias de la Salud*. Imparte: Diplomatura en Enfermería, Diplomatura en Fisioterapia, y Diplomatura en Terapia Ocupacional, Máster en Gerontología Social y Máster Interuniversitario en Ciencias de la Enfermería. Además, se imparten: Diploma de Especialización en Accidentes de Tráfico: Emergencias, Reanimación y Transporte Sanitario, Máster en Fisioterapia Manual O.M.T. Concepto de Kalterborn-Evjenth, Máster en Fisioterapia Manual Osteopática Estructural, Posgrado en Diagnóstico y Tratamiento Logopédico de las Patologías de la Voz, Habla y Lenguaje,
- *Escuela Universitaria de Estudios Sociales*. Imparte: Diplomatura en Relaciones Laborales y Diplomatura en Trabajo Social.
- *Facultad de Educación*. Imparte Licenciatura en Psicopedagogía y Diplomatura en Magisterio. Además, se imparten programas de Doctorado Interdepartamental: Departamento de Ciencias de la Educación y Didácticas Específicas, y Máster en Orientación Profesional. También imparte los Posgrados Profesor de Español para Alumnos Inmigrantes, Psicomotricidad y Educación, Musicoterapia, y “Pedagogía Social: Intervención educativo-social en Infancia y Adolescencia en Situaciones de Maltrato, Desprotección y Riesgo”.
- *Facultad de Medicina* (Edificio A y Edificio B), donde se imparte la Licenciatura en Medicina. Estudios propios: Asistentes Sociales Psiquiátricos, Medicina Naturista, Alimentación y Dietoterapia en el Niño y en el Adolescente, Medicina de Urgencias, Valoración Médico Legal del Daño a la Persona, Medicina en Catástrofes, Atención Sanitaria en Catástrofes.
- *Facultad de Ciencias*, que dispone de varios edificios: Químicas, Físicas, Matemáticas y Geológicas. Imparte: Licenciatura en Física, Licenciatura

ra en Geología, Licenciatura en Matemáticas, Licenciatura en Química, Licenciatura en Bioquímica (sólo segundo ciclo), Diplomatura en Estadística, y Diplomatura en Óptica y Optometría. Posgrados oficiales: Geología y Física. Máster oficial en Química Sostenible. Estudios propios: Herramientas Informáticas y Comunicación para Científicos (Posgrado), Bases de Datos e Internet (Máster), Diploma de Especialización en Bases de Datos, Posgrado en Iniciación a la Investigación en Áreas Científicas.

- *Facultad de Derecho*. Imparte: Licenciatura en Derecho y Licenciatura doble en Derecho-Administración y Dirección de Empresas
- *Facultad de Filosofía y Letras*. Imparte: Licenciatura en Filología Clásica, Licenciatura en Filología Hispánica, Licenciatura en Filología Francesa, Licenciatura en Filología Inglesa, Licenciatura en Geografía, Licenciatura en Historia, Licenciatura en Historia del Arte, Licenciatura en Filosofía, y Diplomatura en Biblioteconomía y Documentación.

Además, también se encuentra los colegios mayores: Colegio Mayor Universitario Pedro Cerbuna y Colegio Mayor Universitario Santa Isabel.

- *Facultad de Veterinaria*, donde se imparten las titulaciones oficiales de: Licenciatura en Veterinaria y Licenciatura en Ciencia y Tecnología de los Alimentos. Además, también se imparten: Diploma de Especialización en Antropología de la Alimentación: Alimentación, Salud, Tecnología y Sociedad.

### 2.3.2. El Centro Politécnico Superior (CPS)

El Centro Politécnico Superior nace de la transformación de la *Escuela Técnica Superior de Ingenieros Industriales* (ETSIIZ) fundada en 1974, 3 años después de que ésta se traslade al actual Campus Río Ebro en el Actur (BOE de 10 de agosto de 1989). En este nuevo marco, comienzan a impartirse las titulaciones de Ingeniería de Telecomunicaciones en el curso 1990/1991, Ingeniería en Informática en el curso 1992/1993, e Ingeniería Química en el curso 1994/1995. Actualmente, el CPS (como se puede ver en la sección anterior) es el único centro de esta Universidad de Zaragoza que imparte estudios técnicos superiores.

En el entorno del CPS conviven otros organismos relacionados con la tecnología y el mundo empresarial, como el *Instituto Tecnológico de Aragón (ITA)*, el *Centro Europeo de Empresas e Innovación (CEEI)*, y el *Instituto de Carboquímica del Consejo Superior de Investigaciones Científicas*. Además, la biblioteca técnica de la Universidad de Zaragoza, la *Biblioteca Hypatía de Alejandría* está ubicada al lado del centro. Esto supone que la comunidad universitaria disponga no sólo de un lugar idóneo para la consulta de fuentes de datos (libros, bases de datos técnicas, revistas, etc.) sino también de numerosas salas de estudio, como por ejemplo la sala de autoaprendizaje del Instituto de Idiomas.

Hoy en día, al igual que los demás centros, el CPS está inmerso en el proceso de adaptación al EEES. Algunas de las actuaciones que está llevando a cabo en esa línea

son las siguientes:

- *Elaboración de los programas de Posgrado, Máster y Doctorado que va a ofertar en el próximo curso académico.* En esta línea, destacamos los trabajos llevados a cabo para ofertar los siguientes títulos:
  - Programa Oficial de Posgrado en Ingeniería de Telecomunicación: Máster en Tecnologías de la Información y Comunicaciones en Redes Móviles (TI-CRM), y Doctorado en Tecnologías de la Información y Comunicaciones en Redes Móviles (TICRM).
  - Programa Oficial de Posgrado en Ingenierías Transversales: Máster en Ingeniería Biomédica, y Doctorado en Ingeniería Biomédica.
  - Programa Oficial de Posgrado en Ingeniería Informática: Máster en Ingeniería de Sistemas e Informática, y Doctorado en Ingeniería de Sistemas e Informática.
- *El programa de tutorización de nuevos alumnos.* Está es una de las acciones llevadas a cabo para la mejora de la calidad de la enseñanza y el rendimiento académico del alumnado. Este Programa de Acción Tutorial (PAT) está implantado en el CPS desde el curso 2003/2004 y cuenta con la colaboración del *Instituto de Ciencias de la Educación (ICE)* de la Universidad de Zaragoza. El programa pretende que las tradicionales tutorías académicas se amplíen a otros aspectos relativos al desarrollo integral del alumno, ofreciéndose así una atención personalizada similar a la que pueda tener el alumnado de tercer ciclo.
- *Creación del grupo InDoTec.* Este grupo, *Innovación Docente en Tecnologías de las Comunicaciones*, surge en el año 2003 con el objetivo de mejorar la docencia en las ingenierías de telecomunicaciones e informática. Desde su creación, han participan activamente en la realización de proyectos de Innovación Docente orientados a encontrar el camino más adecuado hacia la convergencia europea.
- *Fomento de Proyectos de Innovación Docente*, desde el curso 2004/2005. Entre ellos, destacan aquellos que abordan la coordinación entre diferentes materias y los que estudian la carga de trabajo del alumnado.
- *Fomento de la movilidad.* La *Oficina de Relaciones Internacionales (ORI)* facilita información sobre cómo cursar parte de los estudios en el extranjero, como por ejemplo: programas Sócrates y Erasmus y la posibilidad de obtener un doble título Europeo-Español de Ingeniería en algunas de las universidades con las que el CPS mantiene acuerdos de reconocimiento de estudios. La ORI también es un centro de referencia para los estudiantes extranjeros que vienen al CPS.
- *Fomento de las relaciones con ámbito empresarial.* La *Oficina de Relaciones con la Empresa (ORE)* se encarga de la tramitación de los aspectos relacionados con las prácticas en empresa. Además, es un punto de referencia para la búsqueda del primer empleo.

Además de la implicación en la docencia y la investigación, el CPS apoya otro tipo de actividades para favorecer el desarrollo integral del alumnado. Así, en el CPS existen unas 20 *asociaciones de estudiantes* con intereses variados. Algunas de ellas son de carácter profesional, como In-FORUM o ISC-Junior Empresa, la Asociación de Ingenieros Químicos de Aragón, la Asociación de Ingenieros en Informática de Aragón, la Asociación de Alumnos de Ingeniería de Telecomunicaciones de la Universidad de Zaragoza, la Asociación de Estudiantes de la Universidad de Zaragoza de Ingeniería Industrial, y el IEEE Student Branch Universidad de Zaragoza. Otras asociaciones son de carácter lúdico-deportivo, como *La cueva de SMAUG* (juegos de rol), el club de montaña, el Coro de Ingenieros, la Tuna de Ingenieros, y el Club de Malabares. Por último, algunas se dedican a la difusión de conocimiento y a los intercambios con el extranjero (DIR-Europa o IAESTE), a la cultura (Comisión de Cultura), y a la cooperación al desarrollo (Ingenieros sin Fronteras).

Por último, hay que destacar que el CPS es un centro innovador y dinámico que a lo largo de los años ha generado una gran cantidad de actividad docente e investigadora a su alrededor, y hoy en día se puede considerar el centro neurálgico del campus universitario Río Ebro.

### 2.3.3. El Departamento de Informática e Ingeniería de Sistemas (DIIS)

El Departamento de Informática e Ingeniería de Sistemas (DIIS) de la Universidad de Zaragoza se crea como tal en abril de 1995 y actualmente tiene su sede en el Centro Politécnico Superior, Edificio Ada Byron. Incluye las áreas de conocimiento que se detallan a continuación, las cuales hasta el momento de su creación pertenecían al *Departamento de Ingeniería Eléctrica e Informática*:

1. *Arquitectura y Tecnología de Computadores (ATC)*, con alrededor de 22 profesores/as, con la siguiente distribución aproximada: 1 Catedrático de Universidad, 5 Titulares de Universidad, 1 Profesor Colaborador, 3 Titulares de Escuela, 5 Asociados LRU, 2 Ayudantes LOU, 2 Ayudantes Doctores, y 3 Asociados a Tiempo parcial.
2. *Ciencias de la Computación e Inteligencia Artificial (CCIA)*, con alrededor de 9 profesores/as, de las que aproximadamente hay: 1 Catedrático de Universidad, 2 Titulares de Universidad, 2 Titulares de Escuela, 1 Asociado LRU, 1 Ayudante, 1 Ayudante Doctor, y 1 Asociado a Tiempo parcial.
3. *Ingeniería de Sistemas y Automática (ISA)*, con alrededor de 20 profesores/as, aproximadamente con: 2 Catedráticos de Universidad, 7 Titulares de Universidad, 1 Profesor Colaborador, 3 Titulares de Escuela, 2 Asociados LRU, 2 Ayudantes, 1 Ayudante Doctor, 1 Investigador Ramón y Cajal, y 1 Asociado a tiempo parcial.
4. *Lenguajes y Sistemas Informáticos (LSI)*, con alrededor de 71 profesores/as, de las que aproximadamente existen: 4 Catedráticos de Universidad, 16 Titulares

de Universidad, 7 Titulares de Escuela, 16 Asociados LRU, 16 Ayudantes, 2 Profesores Colaboradores, y 10 Asociados a Tiempo parcial.

La plantilla del DIIS, distribuida en las diferentes áreas, tiene aproximadamente unos 120 profesores, 58 de ellos doctores, 37 becarios, y 10 miembros del Personal de Administración y Servicios, distribuidos en 10 centros universitarios. Los en los que se imparte docencia son:

1. *Centro Politécnico Superior (CPS)*, en las titulaciones de:
  - a) Ingeniería en Informática, donde aproximadamente imparte unas 54 materias.
  - b) Ingeniería en Telecomunicaciones, donde aproximadamente imparte unas 19 materias.
  - c) Ingeniería Industrial, donde aproximadamente imparte unas 15 materias.
  - d) Ingeniería Química, donde aproximadamente imparte unas 4 materias.
2. *Escuela Politécnica Superior de Huesca (EPSH)*, en las titulaciones de:
  - a) Ingeniería Técnica Industrial, donde imparte 1 materia.
  - b) Ingeniería Técnica Agrícola, donde imparte 1 materia.
3. *Escuela Universitaria de Estudios Empresariales de Huesca (EUEEH)*, en las titulaciones de:
  - a) Diplomatura en Ciencias Empresariales, donde imparte 1 materia.
  - b) Diplomatura en Relaciones Laborales, donde imparte 1 materia.
  - c) Diplomatura en Gestión y Administración Pública, donde imparte 2 materias.
4. *Escuela Universitaria de Estudios Empresariales de Zaragoza (EUEEZ)*, en la Diplomatura en Ciencias Empresariales, donde aproximadamente imparte 3 materias.
5. *Escuela Universitaria de Estudios Sociales de Zaragoza (EUESZ)*, en las titulaciones de:
  - a) Diplomatura en Trabajo Social, donde imparte 1 materia.
  - b) Diplomatura en Relaciones Laborales, donde imparte 1 materia.
6. *Escuela Universitaria de Ingeniería Técnica Industrial de Zaragoza (EUITIZ)*, en las titulaciones de:
  - a) Ingeniería Técnica Industrial, especialidad de Electricidad, donde aproximadamente imparte 4 materias.

- b) Ingeniería Técnica Industrial, especialidad de Electrónica Industrial, donde aproximadamente imparte 5 materias.
  - c) Ingeniería Técnica Industrial, especialidad de Mecánica, donde aproximadamente imparte 3 materias.
  - d) Ingeniería Técnica Industrial, especialidad de Química Industrial, donde aproximadamente imparte 5 materias.
  - e) Ingeniería Técnica en Diseño Industrial, donde imparte 1 materia.
7. *Escuela Universitaria Politécnica de Teruel (EUPTE)*, en las titulaciones de:
- a) Ingeniería Técnica en Informática de Gestión, donde aproximadamente imparte 18 materias.
  - b) Ingeniería Técnica en Telecomunicaciones, donde aproximadamente imparte 3 materias.
8. *Facultad de Ciencias (FC)*, en las titulaciones de:
- a) Diplomatura en Estadística, donde aproximadamente imparte 9 materias.
  - b) Licenciatura en Matemáticas, donde imparte 1 materia.
9. *Facultad de Ciencias Humanas y de la Educación (FCHE)*, en las titulaciones de:
- a) Licenciatura en humanidades, donde aproximadamente imparte 2 materias.
  - b) Maestro, especialidad Educación Física, donde imparte 1 materia.
  - c) Maestro, especialidad Educación Infantil, donde imparte 1 materia.
  - d) Maestro, especialidad Educación Primaria, donde imparte 1 materia.
10. *Facultad de Humanidades y Ciencias Sociales (FHCS)* en la Licenciatura en Humanidades, donde imparte 1 materia.

Además, es responsable del *Programa de Doctorado en Ingeniería de Sistemas e Informática*, que obtuvo la *Mención de Calidad* por el Ministerio de Educación y Ciencia en la resolución de 28 de mayo de 2003 (referencia MCD2003-00466), y a partir de ahí la ha ido manteniendo año tras año. Actualmente, está trabajando en la elaboración del *Programa Oficial de Posgrado en Ingeniería de Sistemas e Informática*, aprobado en la Junta de Centro de 10 de octubre de 2006.

Dentro del Departamento podemos distinguir principalmente 7 grupos de investigación:

- *Grupo de Arquitectura de Computadores de la Universidad de Zaragoza (gaZ)*. Sus áreas de investigación son principalmente: las memorias cache, la prebúsqueda en memorias caché, la medida de prestaciones de los computadores, la simulación de microprocesadores eficientes, el multithreading, la gestión de registros de los microprocesadores, y las arquitecturas de los multiprocesadores.

- *Grupo de Informática Gráfica Avanzada, Ambientes Inteligentes y Audio Digital*. Sus principales áreas de investigación son: modelado bidimensional y tridimensional, modelado y simulación de la mecánica de fluidos, modelado y simulación de la actividad geofísica, herramientas de información geográfica, el modelado sísmico, y la simulación y validación de proyectos de iluminación.
- *Grupo de Ingeniería de Sistemas de Eventos Discretos (GISED)*. Sus principales áreas de investigación son: el análisis, diseño y modelado de sistemas de eventos discretos, continuos e híbridos mediante Redes de Petri, evaluación de prestaciones de sistemas, y el modelado, simulación y optimización de sistemas logísticos.
- *Grupo de Robótica, Percepción y Tiempo Real (RoPeRT)*. Sus principales áreas de investigación son: la robótica, la visión artificial y los sistemas de tiempo real.
- *Grupo de Sistemas de Información Avanzados (IAAA)*. Su principal área de investigación son los Sistemas de Información Geográficos (GIS) aunque también realizan trabajos en áreas como la telemedicina y sistemas de información web.
- *Grupo de Sistemas de Información Distribuidos (SID)*. Sus principales áreas de investigación son: las bases de datos distribuidas, la computación móvil, y la web semántica.
- *Grupo Noesis*. Su principal línea de investigación es la topología digital y la dinámica y evolución de los sistemas de información.

También existen otros grupos de investigación realizando trabajos en otros campos como: *grid computing*, sistemas de información en tiempo real, generación de *workflows*, etc.

#### 2.3.4. La adaptación al EEES en la Universidad de Zaragoza

La Universidad de Zaragoza, al igual que la mayoría de las universidades españolas, se encuentra inmersa en el proceso de adaptación al EEES. Así, a lo largo de los últimos años se han realizado numerosas acciones, entre las que destacamos las siguientes:

- *Los programas de asesorías para jóvenes*. La Universidad dispone en el Campus de San Francisco y en el Campus Río Ebro de un grupo de profesionales que están a disposición del alumnado para ofrecerles asesoría jurídica, psicológica, sexológica y orientación en los estudios.
- *El fomento de cursos de formación de profesorado*. Aquí conviene destacar la labor del Instituto de Ciencias de la Educación (ICE) y en especial el Diploma de Formación Pedagógica de Profesorado Universitario, que vienen ofertando desde hace varios años.

- *El Documento Marco del Proyecto Tutor* (tutoría académica de titulación), dentro del Plan integral en Convergencia Europea para los centros de la Universidad de Zaragoza, aprobado el curso 2005/2006.
- El plan estratégico “*Planificación Estratégica de la Convergencia Europea en la Universidad de Zaragoza 2006-2008*”, aprobado en Junta de Gobierno el 10 de mayo de 2006. En este documento se detallan las acciones para llevar a cabo: la implantación de Grados, programas oficiales de Posgrado con DGP, y la de programas oficiales de Posgrado, la elaboración de un Mapa de Titulaciones y un Mapa de Centros, fomentar e implantar nuevas metodologías docentes activas y colaborativas, los cambios en los aspectos administrativos implicados, y acciones dirigidas a los estudiantes y la formación a lo largo de la vida. Para todo esto, además de una voluntad política decidida por parte del Consejo de Dirección y del Consejo de Gobierno de la Universidad, se necesita aunar voluntades, motivaciones y esfuerzos por parte del resto de nuestra comunidad, requisitos imprescindibles para conseguir los objetivos señalados.
- *El fomento del aprendizaje de lenguas extranjeras*, a través del Instituto de Idiomas y con otro tipo de programas como la impartición de diversas asignaturas optativas en inglés. También se ofrecen cursos de español para extranjeros.
- *El fomento de la movilidad* de los estudiantes a través de diferentes oficinas, como por ejemplo “*The Office of the Erasmus Institucional Coordinator*” que coordina todas las iniciativas de cooperación y colaboración en el área de la educación y entrenamiento, y centraliza todas las iniciativas que conciernen al establecimiento de relaciones entre la Universidad de Zaragoza y otras instituciones de enseñanza o/e investigación. También podemos destacar que existen diferentes sitios web orientados a estudiantes extranjeros, donde se les ofrece la información necesaria para poder incorporarse a la Universidad de Zaragoza.

Además, la Universidad cuenta con docentes de excelencia y es necesario aprovechar sus buenas prácticas para que sirvan de ejemplo y transmisión a la comunidad. Así, también existen numerosos equipos docentes que vienen desarrollando interesantes e importantes actividades de innovación docente de cara a la convergencia europea. En este punto, es de justicia señalar los siguientes trabajos que se han llevado a cabo:

- La celebración de la *I Jornada sobre convergencia en el EEES* de la Universidad de Zaragoza, el 19 de noviembre de 2002.
- Las *experiencias piloto* llevadas a cabo durante el curso 2005/2006 al amparo de las ayudas obtenidas del Ministerio de Educación y Ciencia (Orden ECI/924/2005 de 21 de marzo), del Gobierno de Aragón, y de la propia Universidad de Zaragoza. El objetivo es dar un impulso decidido de cara a adaptar nuestras titulaciones al EEES. Se llevaron a cabo 3 tipos de acciones:
  - Experiencias piloto desarrolladas en un curso académico completo.



- Experiencias piloto restringidas a una o varias materias.
  - Proyectos de innovación relacionados con la puesta en práctica de metodologías activas.
- 
- La celebración de las *I Jornadas de Innovación Docente* en noviembre de 2006, donde se presentaron numerosos trabajos llevados a cabo por los distintos colectivos de la Universidad de Zaragoza.
  - La publicación, por parte del Dr. José Luis Bernal Agudo (Profesor Titular de Ciencias de la Educación y director del Departamento de Ciencias de la Educación), del documento “*Pautas para el diseño de una asignatura desde la perspectiva de los ECTS*”, así como de un modelo de ficha de asignatura en ECTS y un documento aclaratorio sobre cómo medir los créditos ECTS en la Universidad de Zaragoza. También conviene señalar en este punto que actualmente la Universidad de Zaragoza emplea los créditos ECTS para los programas de intercambio (ERASMUS), pero se limita sólo a multiplicar los créditos docentes por un factor constante común. Para ajustarse a la carga real de las “*horas de trabajo*”, se han realizado algunos estudios como “*Evaluación de la carga docente de la titulación de Ingeniería de Telecomunicación: asignación de créditos ECTS y adecuación a los perfiles y competencias profesionales*”, elaborado y llevado a cabo por el grupo InDoTec (Innovación Docente en Tecnologías de las Comunicaciones).

Para finalizar, cabe señalar que las acciones necesarias para abordar los retos del *Espacio Europeo de Investigación (EEI)* serán objeto de un plan estratégico específico.

## 2.4. Los estudios de Informática

La informática ha ido constituyéndose, a lo largo de las últimas décadas, como una rama de importancia creciente e independiente, como así lo evidencia el estudio realizado en el marco del proyecto *EICE (Estudios Universitarios de Informática y Convergencia Europea)*. Hace algunos años, el contexto profesional de las Tecnologías de la Información y Comunicación estaba marcado por la gran carencia de personal cualificado en el mercado laboral. Actualmente, sigue existiendo una gran demanda y paralelamente existe un interés creciente por empezar a consolidar la identidad profesional informática. Esto se ve reflejado, por ejemplo, en la aparición de colegios profesionales de informática, siendo el último en España *El Colegio Profesional de Ingeniería en Informática de Galicia* (15 de diciembre de 2006).

En esta sección, se hace un breve repaso de las directrices generales propias de los estudios de informática en España y de los planes de estudios relacionados con estos en la Universidad de Zaragoza.

### 2.4.1. Las titulaciones de Ingeniería en Informática en España

La informática en España nace como disciplina académica en 1969, con la creación del *Instituto de Informática*, bajo la dependencia del Ministerio de Educación y Ciencia, al considerarse que para el correcto ejercicio profesional era preciso la obtención previa de formación técnica y profesional.

A través del *Decreto 327/1976* de 26 de febrero se establece la necesidad de que las enseñanzas en informática se desarrollen a través de la educación universitaria y de la formación profesional. Posteriormente, el *Real Decreto 1459/1990*, *1460/1990* y *1461/1990* de 26 de octubre, estableció el título universitario oficial de Ingeniería en Informática, Ingeniería Técnica en Informática de Gestión, e Ingeniería Técnica en Informática de Sistemas, respectivamente, y se aprobaron las directrices generales propias de los planes de estudios conducentes a su obtención, las cuales están brevemente resumidas en las Figuras 2.2, 2.3, y 2.4. Años después, la Licenciatura en Informática fue homologada a la de la Ingeniería en Informática por el *Real decreto 1954/1994*, de 30 de septiembre.

Como se puede observar en la Figura 2.2, la temática relacionada con los *procesadores del lenguaje*, que es la tratada en este proyecto docente, se incluye en el segundo ciclo de Ingeniería en Informática. Esta temática tiene una fuerte relación con la materia de *Teoría de Autómatas y Lenguajes Formales*, incluida en el primer ciclo de Ingeniería en Informática.

Hoy en día, en el marco del proceso de convergencia a Europa y tras la publicación en marzo de 2004 del *Libro Blanco Título de Grado en Ingeniería en Informática*, se están planteando una nueva estructura de contenidos a la que habrá que prestar atención.

### 2.4.2. Las titulaciones de Ingeniería en Informática en la Universidad de Zaragoza

Tras la creación del Centro Politécnico Superior, la Junta de Gobierno de la Universidad de Zaragoza acordó aprobar el plan de estudios de las enseñanzas de Ingeniería en Informática y solicitó la homologación de su primer ciclo al Consejo de Universidades (23 de mayo de 1990). Posteriormente, debido a la publicación del RD 1459/1990, hubo que remitir un nuevo plan de estudios que fue aprobado por el Consejo de Universidades el 26 de mayo de 1994 (BOE 1 de febrero de 1995).

Es conveniente hacer notar que para la creación del plan de estudios de Ingeniería en Informática se tuvieron en cuenta planes de otras universidades donde la titulación ya estaba más asentada, como la Universidad Politécnica de Madrid (UPM), la Universidad Politécnica de Catalunya (UPC), la Universidad Politécnica de Valencia (UPV), y la Universidad del País Vasco (UPV/EHU). Igualmente, se consideraron las recomendaciones de ACM, así como los planes de estudios de otras universidades europeas y estadounidenses. Finalmente, también se tuvieron en cuenta las tendencias del mercado laboral y las necesidades de formación empresariales en este sector, debido a que los planes de estudios tienen que ser sensibles a las demandas. El plan de estudios pretende que la titulación capacite para desempeñar múltiples actividades

Relación de materias troncales (por orden alfabético)	Créditos (I)			Áreas de conocimiento
	Técnicos	Prácticos	Total	
<i>Estadística.</i> Estadística descriptiva. Probabilidades. Métodos estadísticos aplicados.	-	-	6	«Ciencia de la Computación e Inteligencia Artificial», «Estadística e Investigación Operativa» y «Matemática Aplicada».
<i>Estructura de datos y de la información.</i> Tipos abstractos de datos. Estructura de datos y algoritmo de manipulación. Estructura de información: Ficheros, bases de datos.	-	-	12	«Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Estructura y tecnología de computadores.</i> Unidades funcionales: Memoria, procesador, periferia, lenguajes máquina y ensamblador, esquema de funcionamiento. Electrónica. Sistemas digitales. Periféricos.	-	-	15	«Arquitectura y Tecnología de Computadores», «Electrónica», «Ingeniería de Sistemas y Automática» y «Tecnología Electrónica».
<i>Fundamentos físicos de la Informática.</i> Electromagnetismo. Estado sólido. Circuitos.	-	-	6	«Electrónica», «Electromagnetismo», «Física Aplicada», «Física de la Materia Condensada», «Ingeniería Eléctrica» y «Tecnología Electrónica».
<i>Fundamentos matemáticos de la Informática.</i> Álgebra. Análisis matemático. Matemática discreta. Métodos numéricos.	-	-	18	«Álgebra», «Análisis Matemático», «Ciencia de la Computación e Inteligencia Artificial» y «Matemática Aplicada».
<i>Metodología y tecnología de la programación.</i> Diseño de algoritmos. Análisis de algoritmos. Lenguajes de programación. Diseño de programas: Descomposición modular y documentación. Técnicas de verificación y pruebas de programas.	-	-	15	«Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Sistemas operativos.</i> Organización, estructura y servicio de los sistemas operativos. Gestión y administración de memoria y de procesos. Gestión de entrada/salida. Sistemas de ficheros.	-	-	6	«Arquitectura y Tecnología de Computadores», «Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Teoría de autómatas y lenguajes formales.</i> Máquinas secuenciales y autómatas finitos. Máquinas de Turing. Funciones recursivas. Gramáticas y lenguajes formales. Redes neuronales.	-	-	9	«Álgebra», «Ciencia de la Computación e Inteligencia Artificial», «Ingeniería de Sistemas y Automática», «Lenguajes y Sistemas Informáticos» y «Matemática Aplicada».
SEGUNDO CICLO				
<i>Arquitectura e Ingeniería de computadores.</i> Arquitecturas paralelas. Arquitecturas orientadas a aplicaciones y lenguajes.	-	-	9	«Arquitectura y Tecnología de Computadores», «Electrónica», «Ingeniería de Sistemas y Automática» y «Tecnología Electrónica».
<i>Ingeniería del software.</i> Análisis y definición de requisitos. Diseño, propiedades y mantenimiento del software. Gestión de configuraciones. Planificación y gestión de proyectos informáticos. Análisis de aplicaciones.	-	-	18	«Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Inteligencia artificial e Ingeniería del conocimiento.</i> Heurística. Sistemas basados en el conocimiento. Aprendizaje. Percepción.	-	-	9	«Ciencia de la Computación e Inteligencia Artificial», «Ingeniería de Sistemas y Automática» y «Lenguajes y Sistemas Informáticos».
<i>Procesadores de lenguaje.</i> Compiladores. Traductores e intérpretes. Fases de compilación. Optimización de código. Macroprocesadores.	-	-	9	«Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Redes.</i> Arquitectura de redes. Comunicaciones.	-	-	9	«Arquitectura y Tecnología de Computadores», «Ciencia de la Computación e Inteligencia Artificial», «Ingeniería de Sistemas y Automática», «Ingeniería Telemática», «Lenguajes y Sistemas Informáticos».
<i>Sistemas informáticos.</i> Metodología de análisis. Configuración, diseño, gestión y evaluación de sistemas informáticos. Entornos de sistemas informáticos. Tecnologías avanzadas de sistemas de información, bases de datos y sistemas operativos. Proyectos de sistemas informáticos.	-	15	-	«Arquitectura y Tecnología de Computadores», «Ciencia de la Computación e Inteligencia Artificial», «Estadística e Investigación Operativa», «Ingeniería de Sistemas y Automática», «Ingeniería Telemática», «Lenguajes y Sistemas Informáticos» y «Organización de Empresas».

Figura 2.2: Materias troncales en Ingeniería en Informática

en la industria de la informática y de los ordenadores: diseño, análisis, y evaluación de sistemas informáticos en general, contemplando los distintos aspectos de gestión, organización, y dirección de proyectos informáticos, mantenimiento de equipos e infraestructuras, inteligencia artificial, paralelismo y gestión masiva de la información, y manejo de bases de datos y comunicaciones en cualquier sector industrial y de servicios. Además, el plan de estudios está orientado a que los egresados puedan desarrollar sus actividades tanto en la Administración y Organismos Públicos como en empresas privadas.

Relación de materias troncales (por orden alfabético)	Créditos (I)			Áreas de conocimiento
	Teóricos	Prácticos	Total	
<i>Estadística</i> . Estadística descriptiva. Probabilidades. Métodos estadísticos aplicados.	-	-	6	«Ciencia de la Computación e Inteligencia Artificial», «Estadística e Investigación Operativa» y «Matemática Aplicada».
<i>Estructura de datos y de la información</i> . Tipos abstractos de datos. Estructura de datos y algoritmo de manipulación. Estructura de información: Ficheros, bases de datos.	-	-	12	«Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Estructura y tecnología de computadores</i> . Unidades funcionales: Memoria, procesador, periferia, lenguajes máquina y ensamblador, esquema de funcionamiento. Electrónica. Sistemas digitales. Periféricos.	-	-	15	«Arquitectura y Tecnología de Computadores», «Computadores», «Electrónica», «Ingeniería de Sistemas y Automática» y «Tecnología Electrónica».
<i>Fundamentos físicos de la Informática</i> . Electromagnetismo. Estado sólido. Circuitos.	-	-	6	«Electromagnetismo», «Electrónica», «Física Aplicada», «Física de la Materia Condensada», «Ingeniería Eléctrica» y «Tecnología Electrónica».
<i>Fundamentos matemáticos de la Informática</i> . Álgebra. Análisis matemático. Matemática discreta. Métodos numéricos.	-	-	18	«Álgebra», «Análisis Matemático», «Ciencia de la Computación e Inteligencia Artificial» y «Matemática Aplicada».
<i>Metodología y tecnología de la programación</i> . Diseño de algoritmos. Análisis de algoritmos. Lenguajes de programación. Diseño de programas: Descomposición modular y documentación. Técnicas de verificación y pruebas de programas.	-	-	12	«Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Redes</i> . Arquitectura de redes. Comunicaciones	-	-	6	«Arquitectura y Tecnología de Computadores», «Ciencia de la Computación e Inteligencia Artificial», «Ingeniería Telemática», «Lenguajes y Sistemas Informáticos» e «Ingeniería de Sistemas y Automática».
<i>Sistemas operativos</i> . Organización, estructura y servicio de los sistemas operativos. Gestión y administración de memoria y de procesos. Gestión de entrada/salida. Sistemas.	-	-	6	«Arquitectura y Tecnología de Computadores», «Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Teoría de autómatas y lenguajes formales</i> . Máquinas secuenciales y autómatas finitos. Máquinas de Turing. Funciones recursivas. Gramáticas y lenguajes formales. Redes neuronales.	-	-	9	«Álgebra», «Ciencia de la Computación e Inteligencia Artificial», «Ingeniería de Sistemas y Automática», «Lenguajes y Sistemas Informáticos» y «Matemática Aplicada».

Figura 2.3: Materias troncales en Ingeniería Técnica en Informática de Sistemas

El plan de estudios de Ingeniería en Informática consta de 10 cuatrimestres estructurados en dos ciclos de cinco cuatrimestres cada uno, donde se enmarcan las diferentes asignaturas (ver Figuras 2.5 y 2.6).

El alumnado debe cursar 351.5 créditos. En el primer ciclo (2.5 primeros años), la mayoría de las asignaturas son troncales y obligatorias para tratar de que el alumnado adquiriera un perfil generalista, mientras que en el segundo ciclo se ofrece un mayor volumen de optatividad para que el propio alumnado cree su propio perfil. A partir del segundo curso, el alumnado selecciona, en función de sus intereses, una serie de asignaturas optativas y de libre elección. En la Figura 2.6, se puede ver la oferta para las asignaturas optativas en el CPS. El alumnado puede seleccionar las asignaturas de libre elección de entre las asignaturas optativas, pero además tienen la posibilidad de convalidarlas con la realización de prácticas en empresa, asistencia a cursos, seminarios o reuniones de carácter científico, asignaturas realizadas en otras titulaciones de la Universidad de Zaragoza o de otra universidad, etc.

En el momento en que se implantó este plan de estudios, se era consciente de que la implantación de los planes de estudios estructurados por créditos LRU y cuatrimestres había desencadenado un aumento del fracaso escolar. Para tratar de evitar esto, se colocaron las asignaturas en los cuatrimestres de modo que facilitaran al alumnado el seguimiento de grupos de optativas, en un intento de mostrar posibles itinerarios de especialización, y también se intentó ofrecer el mismo número de optativas en el cuatrimestre de otoño que en el de primavera. Por otro lado, se tuvo en cuenta

Relación de materias troncales (por orden alfabético)	Créditos (1)			Áreas de conocimiento
	Teóricos	Prácticos	Total	
<i>Estadística</i> . Estadística descriptiva. Probabilidades. Métodos estadísticos aplicados.	-	-	9	«Ciencia de la Computación e Inteligencia Artificial», «Estadística e Investigación Operativa» y «Matemática Aplicada».
<i>Estructura de datos y de la información</i> . Tipos abstractos de datos. Estructura de datos y algoritmo de manipulación. Estructura de información: Ficheros, bases de datos.	-	-	12	«Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Estructura y tecnología de computadores</i> . Unidades funcionales: Memoria, procesador, periferia, lenguajes máquina y ensamblador, esquema de funcionamiento. Electrónica. Sistemas digitales. Periféricos.	-	-	9	«Arquitectura y Tecnología de Computadores», «Electrónica», «Ingeniería de Sistema y Automática» y «Tecnología Electrónica».
<i>Fundamentos matemáticos de la Informática</i> . Álgebra. Análisis matemático. Matemática discreta. Métodos numéricos.	-	-	18	«Álgebra», «Análisis Matemático», «Ciencia de la Computación e Inteligencia Artificial» y «Matemática Aplicada».
<i>Ingeniería del software de gestión</i> . Diseño, propiedades y mantenimiento del software de gestión. Planificación y gestión de proyectos informáticos. Análisis de aplicaciones de gestión.	-	-	12	«Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Metodología y tecnología de la programación</i> . Diseño de algoritmos. Análisis de algoritmos. Lenguajes de programación. Diseño de programas: Descomposición modular y documentación. Técnicas de verificación y pruebas de programas.	-	-	15	«Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Sistemas operativos</i> . organización, estructura y servicio de los sistemas operativos. Gestión y administración de memoria y de procesos. Gestión de entrada/salida. Sistemas de ficheros.	-	-	6	«Arquitectura y Tecnología de Computadores», «Ciencia de la Computación e Inteligencia Artificial» y «Lenguajes y Sistemas Informáticos».
<i>Técnicas de organización y gestión empresarial</i> . El sistema económico y la Empresa. Técnicas de administración y técnicas contables.	-	-	12	«Economía Financiera y Contabilidad» y «Organización de Empresas».

**Figura 2.4:** Materias troncales en Ingeniería Técnica en Informática de Gestión

el proceso actual de revisión de los contenidos troncales de las titulaciones del área de informática y las tendencias del mercado laboral y las necesidades de formación empresariales.

El mismo curso en que empezó a impartirse la titulación de Ingeniería en Informática en el CPS (1992/1993) comenzó también a impartirse la titulación de Ingeniería Técnica en Informática de Sistemas en la Escuela Politécnica de la Almunia de Doña Godina. Su plan de estudios (publicado en el BOE de 8 de Junio de 1994) consta de 3 cursos que forman un único ciclo (primer ciclo) donde se enmarcan las diferentes asignaturas (ver Figura 2.7 y 2.8).

Años después, en el curso 1999/2000, se comienza a impartir la titulación de Ingeniería Técnica de Informática de Gestión en la Escuela Politécnica de Teruel. Este plan de estudios, publicado en el BOE de 7 de enero de 2000, consta de 6 cuatrimestres pertenecientes todos ellos a un único ciclo (primer ciclo) donde se enmarcan las diferentes asignaturas (ver Figuras 2.9 y 2.10).

### 2.4.3. La materia de Procesadores del Lenguaje en Ingeniería Informática en el CPS

La materia troncal de *Procesadores del lenguaje*, tiene como *descriptores*: *Compiladores. Traductores e intérpretes. Fases de compilación. Optimización de código. Macroprocesadores*. Además, tiene asignados un total mínimo de 9 créditos LRU que cada universidad puede desglosar en las asignaturas que considere oportunas para

Ciclo	Curso	Cuatrimes	Troncal	Obligatoria	Optativa	Libre
Primero	Primer	1º	Cálculo (7.5)	Sistemas Lógicos (6)		
			Matemática Discreta (7.5)			
			Álgebra (6)			
			Introducción a la programación (7.5)			
		2º	Estadística (7.5)	Ecuaciones Diferenciales (6)		
			Fundamentos Físicos de la Informática (6)			
			Arquitectura de computadores (7.5)			
			Metodología de la programación (7.5)			
	Segundo	3º	Organización de computadores (7.5)	Cálculo numérico (6)		
			Sistemas operativos I (6)	Fundamentos de electrónica (4.5)		
			Estructuras de datos y algoritmos (7.5)			
		4º	Ficheros y bases de datos (7.5)	Tecnología electrónica (7.5)	6 créditos	6 créditos
Segundo	Tercero	5º	Lenguajes, gramáticas y autómatas (4.5)	Sistemas operativos II (6)		
			Modelos abstractos de cálculo (4)	Economía (6)	6 créditos	9 créditos
				Laboratorio de computadores (6)		
				Laboratorio de programación (6)		
		6º	Diseño de arquitecturas (6)	Lenguajes de programación (7.5)	12 créditos	
			Conceptos básicos de redes (4.5)			
	Cuarto	7º	Ingeniería del software I (4.5)			
			Fundamentos de arquitecturas paralelas (6)	Inglés Técnico (6)	7.5 créditos	
			Sistemas de transporte de datos (4.5)			
			Ingeniería del software II (7.5)			
			Compiladores I (4.5)			
			Inteligencia artificial e ingeniería del conocimiento I (4.5)			
		8º	Proyectos (6)	Administración de empresas (6)	6 créditos en técnicas y 3 créditos en no técnicas	
			Inteligencia artificial e ingeniería del conocimiento II (4.5)			
	Quinto	9º	Compiladores II (4.5)			
		10º		Organización de la producción y gestión de la calidad (6)	12 créditos	9 créditos
			Sistemas informáticos (15)	Proyecto fin de carrera (5)	12 créditos	9 créditos

Figura 2.5: Plan de estudios de Ingeniería en Informática en la Universidad de Zaragoza

cubrir los contenidos abarcados por la misma. En el caso de la titulación de Ingeniería en Informática de la Universidad de Zaragoza, se desglosa en *Compiladores I* y *Compiladores II*, cada una de ellas con 4.5 créditos LRU (3 teóricos y 1.5 prácticos).

En ambas asignaturas se trata de adquirir el conocimiento sobre los métodos y técnicas para construir compiladores y traductores e intérpretes. La asignatura de *Compiladores I* está centrada en el *front-end*, es decir en los fundamentos y técnicas para realizar de las fases del análisis léxico, sintáctico y semántico, mientras que la de *Compiladores II* se centra en el *back-end*, es decir en las fases de generación de código y optimización.

La asignatura en la que nos centramos, *Compiladores I*, además de estar relacionada evidentemente con *Compiladores II*, está fuertemente relacionada con *Introducción a la Programación* y con *Lenguajes, Gramáticas y Autómatas* (ver figura 2.11). Así, es necesario haber cursado y superado al menos dichas asignaturas para poder cursar *Compiladores I* (dependencia fuerte). Las flechas punteadas indican una dependencia media, es decir, son requisitos recomendados pero no obligatorias para poder cursar la asignatura, mientras que las flechas discontinuas indican una dependencia débil, es decir, es recomendable tener conocimientos de *Arquitecturas de Computadores* para poder seguir con mayor facilidad la asignatura de *Compiladores I*. En *Compiladores II*, sin embargo, se exige la superación de la asignatura de *Arquitecturas de Computadores*, puesto que se trabaja a más bajo nivel. En el capítulo 5, se detallan en

Cuatrimestre	Optativas Primer Ciclo	Optativas Segundo Ciclo
Otoño	Diseño de experimentos y regresión (6)	Estructura interna de los sistemas operativos (6)
	Teoría de señales y sistemas (6)	Diseño de bases de datos relacionales (6)
	Laboratorio de electrónica (6)	Materiales en las tecnologías de la información y comunicación (6)
		Modelos estocásticos en ingeniería (6)
		Conceptos avanzados de sistemas operativos (6)
		Subsistemas e/s y periféricos (6)
		Bases de datos avanzadas (6)
		Modelado visual y animación (6)
		Programación paralela (6)
		Redes de comunicaciones de banda ancha (6)
		Ingeniería de los sistemas basados en el conocimiento (6)
		Control y programación de robots (6)
		Fundamentos de microelectrónica (6)
		Sistemas empujados (6)
		Accesos digitales (6)
		Simulación de sistemas dinámicos (6)
		Esquemas algorítmicos (6)
		Electrónica digital para comunicaciones (6)
		Modelado geométrico (6)
		Administración de sistemas informáticos (6)
		Creatividad e innovación (no técnica) (3)
		Historia de la tecnología (no técnica) (3)
		Ciencia, tecnología y sociedad (no técnica) (3)
Primavera	Bases físicas de la ingeniería (6)	Procesado digital de la señal (3)
	Expresión gráfica (6)	Programación concurrente (6)
	Lógica (6)	Sistemas de tiempo real (6)
	Transmisión de datos (6)	Sistemas informáticos para el rim (6)
		Visión por computador (6)
		Interacción hombre-máquina (6)
		Informática gráfica (6)
		Modelado y evaluación de prestaciones en sistemas informáticos (6)
		Transmisión de imágenes: técnicas y sistemas (6)
		Servicios de alto nivel en redes informáticas (6)
		Diseño y evaluación de redes (3)
		Paralelismo en procesadores (6)
		Sistemas de eventos discretos (6)
		Informática industrial (6)
		Sistemas de información (6)
		Técnicas avanzadas de programación (6)
		Criptografía y seguridad en comunicaciones (6)
		Creatividad e innovación (no técnica) (3)
		Comunicación oral y escrita en español (no técnica) (3)
		Ética y legislación para ingenieros (no técnica) (3)
		Introducción al ejercicio profesional de la ingeniería (no técnica) (3)
		Historia de la tecnología (no técnica) (3)
		Psicosociología industrial (no técnica) (3)
		Ingeniería y desarrollo tecnológico (no técnica) (3)

**Figura 2.6:** Optativas ofertadas en el Plan de Estudios de Ingeniería en Informática en la Universidad de Zaragoza

profundidad algunos problemas detectados relacionados con las dependencias de las asignaturas, sobre todo con el alumnado que accede directamente al segundo ciclo procedente de otras titulaciones.

Por último, señalar que es necesaria una fuerte colaboración en la planificación y coordinación de los programas de las asignaturas de *Lenguajes*, *Gramáticas y Autómatas*, *Compiladores I* y *Compiladores II* para que se imparta una docencia adecuada.



Ciclo	Curso	Troncal	Obligatoria	Optativa	Libre
Primero	Primer	Algebra (9)	Ampliación de la física (15)	48 créditos	24 créditos
		Análisis matemático (12)	Prácticas de programación I (3)		
		Fundamentos Físicos de la Informática (6)			
		Programación (12)			
	Segundo	Sistemas Lógicos y estructura de compiladores (6)			
		Estructuras de datos y la información (12)	Sistemas de comunicación (4)		
		Electrónica digital (6)	Prácticas de programación II (4)		
		Teoría de Automatas y lenguajes formales (9)			
		Métodos estadísticos de la ingeniería (6)			
		Sistemas Operativos I (6)			
		Arquitectura de ordenadores (9)			
	Tercero	Redes (6)	Proyectos (6)		
			Proyecto Final de Carrera (9)		

**Figura 2.7:** Plan de Estudios de Ingeniería Técnica en Informática de Sistemas en la Universidad de Zaragoza

Optativas
Idioma instrumental técnico I
Idioma instrumental técnico II
Regulación automática
Periféricos de ordenador
Electrónica analógica
Cálculo numérico
Sistemas operativos II
Concurrencia y tiempo real
Ingeniería de software
Arquitectura de ordenadores II
Diseño de sistemas basados en microprocesador
Inteligencia artificial
Administración de empresas y organización de la producción
Compiladores
Informática gráfica
Robótica

**Figura 2.8:** Optativas ofertadas en el Plan de Estudios de Ingeniería Técnica en Informática de Sistemas en la Universidad de Zaragoza

Ciclo	Curso	Cuatrimes	Troncal	Obligatoria	Optativa	Libre
Primero	Primer	1*	Fundamentos de sistemas digitales (6)			6 créditos
			Algebra (6)			
			Análisis matemático (6)			
		2*	Programación I (7.5)			
			Economía de la empresa (6)			
			Estadística (6)	Tecnología de computadores (6)		
	Segundo	3*	Estructuras de datos (7.5)		6 créditos	9 créditos
			Matemática discreta (6)			
			Métodos numéricos (6)			
		4*	Arquitectura de computadores I (6)	Gestión de la producción (9/2)		
			Algoritmia (7.5)	Conceptos básicos de redes de computadores (6)		
				Programación II (7.5)		
Tercero	Tercero	5*	Ingeniería de software I (6)	Gestión de la producción (9/2)	12 créditos	6.5 créditos
			Sistemas operativos I (7.5)	Redes de computadores (6)		
			Administración de empresas (9/2)	Ingeniería de software II (6)		
		6*	Investigación operativa (6)	Sistemas operativos II (6)		
			Bases de datos I (6)	Configuración, evaluación y explotación de sistemas informáticos (6)		
				Bases de datos II (6)		
			Administración de empresas (9/2)	Proyectos informáticos (6)		
				Trabajo fin de carrera (6)		

**Figura 2.9:** Plan de Estudios de Ingeniería Técnica en Informática de Gestión en la Universidad de Zaragoza



Cuatrimestre	Optativas
Primavera	Planificación y control de la empresa
	Inglés I
	Comercio electrónico
	Gestión de la calidad
	Estrategias y sistemas de información
	Sistemas distribuidos
	Interfaces de usuario
	Arquitectura de computadores II

**Figura 2.10:** Optativas ofertadas en el Plan de Estudios de Ingeniería Técnica en Informática de Gestión en la Universidad de Zaragoza

Cuatrimestre	Troncal	Obligatoria
1º	Cálculo (7.5)	Sistemas Lógicos (6)
	Matemática Discreta (7.5)	
	Álgebra (6)	
	Introducción a la programación (7.5)	
2º	Estadística (7.5)	Ecuaciones Diferenciales (6)
	Fundamentos Físicos de la Informática (6)	
	Arquitectura de computadores (7.5)	
	Metodología de la programación (7.5)	
3º	Organización de computadores (7.5)	Cálculo numérico (6)
	Sistemas operativos I (6)	
	Estructuras de datos y algoritmos (7.5)	
4º	Ficheros y bases de datos (7.5)	Tecnología electrónica (7.5)
	Lenguajes, gramáticas y autómatas (4.5)	
5º	Modelos abstractos de cálculo (4)	Sistemas operativos II (6)
6º	Diseño de arquitecturas (6)	Economía (6)
	Conceptos básicos de redes (4.5)	
	Ingeniería del software I (4.5)	
7º	Fundamentos de arquitecturas paralelas (6)	Laboratorio de computadores (6)
	Sistemas de transporte de datos (4.5)	
	Ingeniería del software II (7.5)	
	Compiladores I (4.5)	
8º	Inteligencia artificial e ingeniería del conocimiento I (4.5)	Laboratorio de programación (6)
	Proyectos (6)	
	Inteligencia artificial e ingeniería del conocimiento II (4.5)	
9º	Compiladores II (4.5)	Lenguajes de programación (7.5)
10º		Inglés Técnico (6)
	Sistemas informáticos (15)	
		Administración de empresas (6)
		Organización de la producción y gestión de la calidad (6)
		Proyecto fin de carrera (5)

Dependencia Fuerte  
 Dependencia Media  
 Dependencia Débil

**Figura 2.11:** Asignaturas relacionadas con Compiladores I en el Plan de Estudios de Ingeniería en Informática de la Universidad de Zaragoza



## Capítulo 3

# Contexto en la enseñanza de Compiladores

A la hora de definir el contenido de una materia es necesario conocer, además de los marcos normativos o directrices, las recomendaciones de carácter internacional que periódicamente son revisadas en función del escenario cambiante donde se enmarca la titulación de Ingeniería en Informática. Así, el objetivo de este capítulo es revisar dichos marcos y extraer conclusiones respecto a los objetivos genéricos y específicos que se deben perseguir en la impartición de la asignatura de *Compiladores I* en el plan de estudios de Ingeniería en Informática de la Universidad de Zaragoza. En primer lugar, analizaremos las recomendaciones curriculares de ACM/IEEE. Después, se considera la influencia del Espacio Europeo de Educación Superior. Posteriormente, se analizan algunos modelos que siguen otras universidades de Estados Unidos y España para la enseñanza de esta materia. Finalmente, se muestran nuestras conclusiones y algunas experiencias.

### 3.1. Recomendaciones Curriculares ACM/IEEE

Una fuente de información que sin duda tiene una gran importancia en la elaboración de los programas de estudio de Informática son las recomendaciones y evaluaciones que el grupo de trabajo formado por la *Association for Computer Machinery (ACM)* y la *Computer Society* del *Institute of Electrical and Electronics Engineers (IEEE)* realizan periódicamente. En esta sección se describen los resultados obtenidos por este grupo en su última publicación: el *Computing Curricula* de 2005, publicado en 2006.

El primer currículo de Informática publicado por *ACM* data de 1968. Más tarde, se produjeron varias revisiones que fueron criticadas por considerarse obsoletas y por no dar un enfoque demasiado profundo. Posteriormente en, 1983, la *Computer Society* del *IEEE* publicó su propio plan de estudios con defectos similares a los que

tenían los planes de ACM. Debido a esto, en 1988 se formó el *Joint Curriculum Task Force*, que integraba a expertos de ambas sociedades, con el objetivo de elaborar un programa que estableciera la base común de los planes de estudio en Informática, donde se incluyeran tanto propuestas de temario como aspectos más metodológicos. En una de las primeras publicaciones de este grupo, definen la disciplina como “*el estudio sistemático de procesos algorítmicos que describen y transforman información: su teoría, análisis, diseño, eficiencia, implementación y aplicación*”. Además, dividen la materia en 9 áreas temáticas, cada una de las cuales tiene suficiente base teórica, nivel de abstracción y nivel para el diseño y la implementación, que debe cubrir cualquier plan de estudios de Licenciatura o Ingeniería en Informática. Estos trabajos cristalizaron en la propuesta conjunta *ACM/IEEE-CS Joint Curriculum Task Force Report* [11], en 1991, en la que se consideraron las mismas 9 áreas:

- *Algorithms and Data Structures*
- *Architecture Methods*
- *Artificial Intelligence and Robotics*
- *Database and Information Retrieval*
- *Human-Computer Communication*
- *Numerical and Symbolic Computation*
- *Operating Systems*
- *Programming Languages*
- *Software Methodology and Engineering*

Estas áreas se detallan en *unidades de conocimiento* con el objetivo de que cada institución pudiera elaborar su propio programa de estudios en función de sus objetivos pedagógicos. En relación a los compiladores, se define la materia avanzada *Programming Language Translation* (a veces llamado *Compiler Design* o *Compiler Construction*) [9], y también la materia *Language Translation Systems (PL9)*.

En 1998, IEEE y ACM establecieron las bases para comenzar la revisión de las principales directrices del Computing Curricula de 1991 que culminó en el desarrollo de una versión actualizada de éste, *Computing Curricula 2001*, que constaba de 4 volúmenes más especializados en áreas:

- *Computer Science* publicado en 2002.
- *Computer Engineering* publicado en 2004.
- *Information Systems* publicado en 2002.
- *Software Engineering* publicado en 2004.

En estos volúmenes se incluyen los objetivos de aprendizaje para las diferentes áreas, modelos de currículo y descripciones de cursos para lograr los objetivos de aprendizaje.

Recientemente, en el año 2006, se publicó la última de las actualizaciones, el *Computing Curricula 2005: The overview report*. En este volumen se señala que actualmente existen muchos tipos de programas de Grado relacionados con la Informática y que la variedad de los nombres usada es todavía más amplia. En este caso la publicación se centra en 5 de ellas, que considera las más prometedoras:

- *Computer Engineering (CE)*. Involucra el estudio de hardware, software, comunicaciones y la interacción entre ellas con el objetivo de diseñar y construir ordenadores (*computadores*) y sistemas basados en ordenadores.
- *Computer Science (CS)*. Involucra un amplio abanico de estudios que van desde los fundamentos teóricos y algorítmicos hasta el desarrollo de sistemas inteligentes, bioinformática, visión artificial, robótica, etc. Sus trabajos se pueden dividir en tres categorías:
  - Diseño e implementación de software.
  - Nuevas formas de uso de los ordenadores, abarcando áreas de redes, bases de datos, e interfaces hombre máquina en el marco de la *World Wide Web*.
  - Desarrollo de mecanismos y formas eficientes para resolver problemas de computación; por ejemplo determinar la mejor forma de almacenar la información en bases de datos, de enviar datos a través de la red, etc.
- *Information Systems (IS)*. Se centra en integrar soluciones tecnológicas y procesos de negocio para que las compañías logren de forma eficiente y eficaz sus objetivos.
- *Information Technology (IT)*. Con este término se refieren al estudio de las necesidades de tecnologías de computación en áreas de negocio, gobierno, salud, escuelas y cualquier tipo de organización.
- *Software Engineering (SE)*. Involucra el desarrollo y mantenimiento de sistemas software que se comporten de forma fiable y eficientemente y que satisfagan todos sus requisitos.

En este estudio se incluye además una comparativa de la ponderación de los diferentes temas que se deben tratar en la formación de graduados de esos 5 perfiles. En la tabla 3.1 se muestra un breve resumen de los temas que podrían afectar a la enseñanza de Compiladores. En la tabla se indican valores mínimos y máximos (en el rango de 0 a 5) para cada una de las áreas de conocimiento mencionadas anteriormente. Las descripciones correspondientes a las áreas de conocimiento incluidas en la misma son:

Área de Conocimiento	CE	CS	IS	IT	SE
Teoría de Lenguajes de Programación	1-2	3-5	0-1	0-1	2-4
Análisis de Requisitos Técnicos	2-5	2-4	2-4	3-5	3-5
Diseño y Principios de Sistemas Operativos	2-5	3-5	1-1	1-2	3-4

**Tabla 3.1:** Temas que podrían afectar a la enseñanza de Compiladores (máximos y mínimos): perfiles CE (Computer Engineering), CS (Computer Science), IS (Information Systems), IT (Information Technology), y SE (Software Engineering)

- *Teoría de Lenguajes de Programación:* principios y diseño de lenguajes de programación incluyendo gramáticas (sintaxis), semántica, sistemas de tipos, y varios modelos de lenguajes (por ejemplo lenguajes declarativos, funcionales, procedurales, y orientados a objetos).
- *Análisis de Requisitos Técnicos:* el proceso a través del cual en un proyecto de desarrollo computacional se determina el hardware y el software en que se basa para alcanzar los objetivos de los individuos u organizaciones usuarios/as.
- *Diseño y Principios de Sistemas Operativos:* trata los principios subyacentes y de diseño para el software de sistema que maneja todos los recursos hardware (incluyendo los mecanismos de procesadores, memoria, almacenes externos y entrada/salida), y proporciona la interfaz entre las aplicaciones software y la máquina.

Además, se destaca que en todos y cada uno de los perfiles se deberían tener en cuenta una serie de aspectos. Entre ellos, nos gustaría resaltar, por su relación con la materia de Compiladores, el del entendimiento de los conceptos de procesos esenciales en lo que se refiere a la ejecución de un programa y las operaciones del sistema, y en lo que se refiere a la relación entre la calidad del producto y el despliegue de los procesos humanos apropiados durante el desarrollo del producto.

Hay que destacar que en el *Computing Curricula 2005* no se incluyen los objetivos de aprendizaje ni los currículos de los cursos; en esto se está trabajando actualmente. Debido a esto, y porque el *Computer Science* (conocido como *Computing Curricula 2001*) es el más relacionado con la materia que tratamos en este proyecto docente, nos centramos en el volumen de *Computer Science* de 2001. Concretamente, se describen algunos aspectos de este, como las áreas de conocimiento y la organización en cursos de los contenidos de dichas áreas según diferentes enfoques, centrándonos en el área que tiene más relación con Compiladores.

### 3.1.1. Áreas de conocimiento

Se establecen 14 áreas de conocimiento. Se extiende así la propuesta de 1991, que constaba de 5 áreas. Las nuevas áreas incluidas son: *Redes*, *Informática Gráfica*, *Aspectos Sociales y Profesionales*, y *Ciencias de la Computación*. Además, se separan en

dos áreas *Fundamentos de la Programación y Algorítmica*, que antes estaban unidas. Las áreas resultante y sus acrónimos se muestran en la Tabla 3.2

Discrete Structures (DS)	Human-Computer Interactions (HC)
Programming Fundamentals (PF)	Graphics & Visual Computing (GV)
Algorithms & Complexity (AL)	Intelligent Systems (IS)
Architecture & Organization (AR)	Information Management (IM)
Operating Systems (OS)	Social & Professional Issues (SP)
Net-Centric Computing (NC)	Software Engineering (SE)
Programming Languages (PL)	Computational Science (CN)

**Tabla 3.2:** Áreas de conocimiento en el IEEE/ACM Computing Curricula 2001

Cada área de conocimiento se compone de una serie de unidades temáticas (*knowledge units*), que se describen mediante un conjunto de temas (*topics*). De este modo, los temas constituyen el nivel más bajo en el que se estructura el currículo. En la siguiente tabla, mostramos las unidades temáticas (junto con sus temas) que tienen relación con el área de conocimiento de Lenguajes de Programación (*Programming Languages, PL*). La tabla incluye tanto las unidades nucleares (*core*) como las electivas (*elective*). Los números entre paréntesis, representados para las unidades *core* (subrayadas en la tabla), representan el número mínimo de horas requerido para cubrir el material en un formato de clase tradicional, pero siempre es apropiado incluir más.

En lo que sigue, nos centraremos en el área de lenguajes de programación (*Programming Languages, PL*), dado que incluye las 2 unidades más relacionadas con el presente proyecto docente: *PL3. Introduction to language translation* y *PL8. Language translation systems*. En las Tablas 3.3, y 3.4 mostramos las distintas unidades y temas de dicha área.

Por su relación con el presente proyecto docente, indicamos los objetivos de aprendizaje de las siguientes unidades:

- *PL3. Introduction to language translation (core)*. Learning objectives: 1) compare and contrast compiled and interpreted execution models, outlining the relative merits of each; 2) describe the phases of program translation from source code to executable code and files produced by these phases; and 3) explain the differences between machine-dependent and machine-independent translation and where these differences are evident in the translation process.
- *PL8. Language translation systems (elective)*. Learning objectives: 1) describe the steps and algorithms used by language translators; 2) recognize the underlying formal models such as finite state automata, push-down automata and their connection to language definition through regular expressions and grammars; 3) discuss the effectiveness of optimization; and 4) explain the impact of a separate

PL. Programming Languages	
Units	Topics
PL1. Overview of programming languages (2)	-History of programming languages -Brief survey of programming paradigms -The effects of scale in programming methodology
PL2. Virtual machines (1)	-The concept of a virtual machine -Hierarchy of virtual machines -Intermediate languages -Security issues arising from running code on an alien machine
PL3. Introduction to language translation (2)	-Comparison of interpreters and compilers -Language translation phases (lexical analysis, parsing, code generation, optimization) -Machine-dependent and machine-independent aspects of translation
PL4. Declaration and types (3)	-The concept of type as a set of values together with a set of operations -Declaration models (bindings, visibility, scope and lifetime) -Overview of type checking-Garbage collection
PL5. Abstraction mechanisms (3)	-Procedures, functions, and iterators as abstraction mechanisms -Parametrization mechanisms (reference vs. value) -Activation records and storage management -Type parameters and parameterized types -Modules in programming languages
PL6. Object-oriented programming (10)	-Object-oriented design -Encapsulation and information-hiding -Separation of behavior and implementation -Classes and subclasses -Inheritance (overriding, dynamic dispatch) -Polymorphism (subtype polymorphism vs. inheritance) -Class hierarchies -Collection classes and iteration protocols -Internal representations of objects and method tables

**Tabla 3.3:** Unidades y temas del área de Lenguajes de Programación en el IEEE/ACM Computing Curricula 2001, tabla 1 de 2

*compilation facility and the existence of program libraries on the compilation process.*

### 3.1.2. Modelos curriculares

Para obtener un modelo curricular determinado, es necesario estructurar el cuerpo de conocimiento considerado en un conjunto de cursos de acuerdo con una determinada ordenación temporal. El *IEEE/ACM Computing Curricula* de 2001 lo que hace es establecer un modelo marco y proporcionar una serie de posibles estrategias a seguir para obtener un modelo curricular concreto.

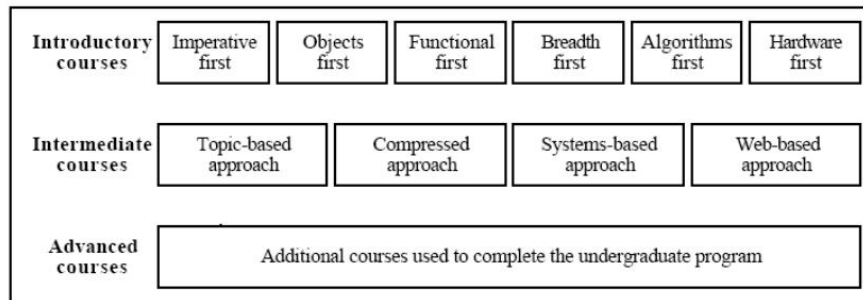
Las materias se clasifican en 3 niveles (ver Figura 3.1): hay *materias introductorias, intermedias y avanzadas*. Para las materias introductorias se plantean seis posibles estrategias: *imperative first, objects first, functional first, breadth first, algorithms first*, y *hardware first*. De forma similar, se plantean 4 posibles estrategias para las materias intermedias: *topic-based approach, compressed approach, systems-based approach*, y *web-based approach*. Estos dos primeros niveles de materias introductorias e intermedias están destinados a cubrir los contenidos fundamentales del cuerpo de cono-



PL. Programming Languages (continuación)	
Units	Topics
PL7. Functional programming	<ul style="list-style-type: none"> <li>-Overview and motivation of functional languages</li> <li>-Recursion over lists, natural numbers, trees, and other recursively-defined data</li> <li>-Pragmatics (debugging by divide and conquer; persistency of data structures)</li> <li>-Amortized efficiency for functional data structures</li> <li>-Closures and uses of functions as data (infinite sets, streams)</li> </ul>
PL8. Language translation systems	<ul style="list-style-type: none"> <li>-Application of regular expressions in lexical scanners</li> <li>-Parsing (concrete and abstract syntax, abstract syntax trees)</li> <li>-Application of context-free grammars in table-driven and recursive-descent parsing</li> <li>-Symbol table management</li> <li>-Code generation by tree walking</li> <li>-Architecture-specific operation: instruction selection and register allocation</li> <li>-Optimization techniques</li> <li>-The use of tools in support of the translation process and the advantages thereof</li> <li>-Program libraries and separate compilation</li> <li>-Building syntax-directed tools</li> </ul>
PL9. Type systems	<ul style="list-style-type: none"> <li>-Data type as set of values with set of operations</li> <li>-Data types</li> <li>-Type-checking models</li> <li>-Semantic models of user-defined types</li> <li>-Parametric polymorphism</li> <li>-Subtype polymorphism</li> <li>-Type-checking algorithms</li> </ul>
PL10. Programming language semantics	<ul style="list-style-type: none"> <li>-Informal semantics</li> <li>-Overview of formal semantics</li> <li>-Denotational semantics</li> <li>-Axiomatic semantics</li> <li>-Operational semantics</li> </ul>
PL11. Programming language design	<ul style="list-style-type: none"> <li>-General principles of language design</li> <li>-Design goals</li> <li>-Typing regimes</li> <li>-Data structure models</li> <li>-Control structure models</li> <li>-Abstraction mechanisms</li> </ul>

**Tabla 3.4:** Unidades y temas del área de Lenguajes de Programación en el IEEE/ACM Computing Curricula 2001, tabla 2 de 2

cimiento, existiendo la posibilidad de incorporar unidades opcionales gradualmente. Estos niveles vendrían a equivaler a un primer ciclo clásico (3 años de la titulación de Ingeniería en Informática). El objetivo de las materias avanzadas es fundamentalmente el de incluir temas opcionales que contribuyan a completar la formación; no obstante, también podrían incluirse aquí contenidos fundamentales que por requerir formación previa no puedan darse en los primeros años de la titulación. A continuación se analizan los 3 niveles de materias en más detalle.



**Figura 3.1:** Materias introductorias, intermedias y avanzadas en el IEEE/ACM Computing Curricula 2001

### 3.1.3. Materias introductorias

Para las materias introductorias, el *IEEE/ACM Computing Curricula 2001* considera 6 posibles enfoques:

- *Imperative first.* Es el enfoque tradicional por el cual se comienza enseñando programación imperativa. La motivación de esta aproximación es que los lenguajes imperativos continúan siendo utilizados ampliamente hoy en día. Los defensores de este modelo también afirman que facilita a los estudiantes comprender la potencia de los lenguajes orientados a objetos. Como desventaja, los conceptos de programación orientada a objetos, que tienen una creciente importancia, suelen retrasarse hasta el segundo curso. Dentro de este modelo, se proponen las siguientes materias:
  - CS101. Programming Fundamentals
  - CS102. The Object-Oriented Paradigm
  - CS103. Data Structures and Algorithms
  - CS111. Introduction to Programming
  - CS112. Data Abstraction
- *Objects first.* También se centra en programación, pero en este caso se enfatizan los principios de la programación orientada a objetos. Las materias propuestas son:
  - CS101. Introduction to Object-Oriented Programming
  - CS102. Objects and Data Abstraction
  - CS103. Algorithms and Data Structures
  - CS111. Object-Oriented Programming

- CS112. Object-Oriented Design and Methodology
- *Functional first.* Propone utilizar un lenguaje funcional sencillo (como *Scheme*) en el primer curso. Algunas ventajas de este modelo son: 1) al utilizarse un lenguaje no utilizado ampliamente fuera del entorno universitario, reduce los efectos de la diversidad de conocimientos previos que puedan tener los estudiantes; 2) la minimalidad de la sintaxis de los lenguajes funcionales permite centrarse en los aspectos fundamentales; y 3) algunas ideas importantes de programación (como la recursión o las estructuras de datos enlazadas) ocurren de forma natural en este dominio. Como principales desventajas, podemos citar que puede reducirse la motivación de los alumnos (al tratarse de lenguajes que no se utilizan ampliamente) y que se les exige pensar en un nivel de abstracción mucho más elevado. Las materias de ejemplo que se proponen en este modelo son:
  - CS111. Introduction to Functional Programming
  - CS112. Objects and Algorithms
- *Breadth-first.* Este modelo rompe con la aproximación tradicional orientada a la programación. El objetivo es dar a los estudiantes una visión general de todo el alcance de la informática, permitiéndoles decidir si quieren estudiar el campo en profundidad o no. Si bien la idea es interesante, en la práctica se ha demostrado que es complicada de implementar. Se proponen 2 posibles implementaciones de este modelo, con las siguientes materias, una de ellas con la materia *CS100. Preview of Computer Science*, y la otra con las materias *CS101. Introduction to Computer Science*, *CS102. Algorithms and Programming Techniques*, y *CS103. Principles of Object-Oriented Design*.
- *Algorithm first.* Con esta aproximación, los conceptos básicos se introducen utilizando pseudo-código. Esto abstrae a los estudiantes de los detalles sintácticos de los lenguajes, permitiéndoles centrarse en los aspectos algorítmicos. La principal desventaja de esta aproximación es que puede ser desmotivadora para los estudiantes, al no poder probar sus algoritmos en un ordenador. En esta aproximación, se proponen las materias:
  - CS111. Introduction to Algorithms and Applications
  - CS112. Programming Methodology
- *Hardware first.* Este modelo propone enseñar los conceptos básicos empezando a nivel de máquina y sucesivamente construyendo conceptos más abstractos. Se proponen las materias:
  - CS111. Introduction to the Computer
  - CS112. Object-Oriented Programming Techniques

Se destacan las materias introductorias propuestas que incluyen alguna temática de lenguajes de programación (*PL Programming Languages*) en las Tablas 3.2 y 3.3. En la Tabla 3.4, se recogen un resumen referente a la cobertura de las distintas unidades del área de lenguajes de programación en materias de cada una de las aproximaciones.

Como conclusión, se observa que las unidades más relacionados con asignaturas de compiladores (*PL3. Introduction to language translation* y *PL8. Language translation systems*) tienen un tratamiento reducido en las materias introductorias.

### 3.1.4. Materias intermedias

Las recomendaciones del *IEEE/ACM Computing Curricula 2001* consideran 4 grupos de materias, uno para cada una de las 4 aproximaciones consideradas. El objetivo es organizar las materias por áreas generales en lugar de centrarse en tareas concretas. Se sigue así la filosofía del *IEEE/ACM Computing Curricula 1991*, donde se proponía las materias intermedias *arquitectura, desarrollo de software, algorítmica, y teoría* (una por área). Las aproximaciones consideradas en el *IEEE/ACM Computing Curricula 2001* para las materias intermedias son:

- *Basada en conceptos clásicos.* Es la aproximación tradicional, donde se divide el material en unidades de acuerdo con las divisiones tradicionales del campo (arquitectura, sistemas operativos, algoritmos, etc.). Como implementación de ejemplo de este modelo, se proponen las materias:

- CS210. Algorithm Design and Analysis
- CS220. Computer Architecture
- CS225. Operating Systems
- CS230. Net-centric Computing
- CS260. Artificial Intelligence
- CS270. Databases
- CS280. Social and Professional Issues
- CS290. Software Development
- CS490. Capstone Project

La principal ventaja de este modelo es que ha sido implementado y probado durante mucho años. Como desventaja, tiene el peligro de que podría llevar a un diseño curricular demasiado compartimentalizado.

- *Comprimida.* La idea es agrupar temas individuales en materias temáticas que abarquen varias áreas. De este modo, se pretende resolver el inconveniente de la aproximación tradicional y crear un currículo más integrado. Como implementación de ejemplo, se proponen las materias:

- CS210. Algorithm Design and Analysis

<b>Materia introductoria</b>	<b>Unidad</b>	<b>Horas</b>
<i>Imperative first</i>		
CS101L. Programming Fundamentals	PL3 Introduction to language translation	1
	PL4 Declarations and types	3
CS102. The Object-Oriented Paradigm	PL1 Overview of programming languages	1
	PL2 Virtual machines	1
	PL3 Introduction to language translation	1
CS103. Data Structures and Algorithms	PL1 Overview of programming languages	1
CS111. Introduction to Programming	PL1 Overview of programming languages	1
	PL4 Declarations and types	1
	PL5 Abstraction mechanisms	2
	PL6 Object-oriented programming	3
CS112. Data Abstraction	PL1 Overview of programming languages	1
	PL2 Virtual machines	1
	PL3 Introduction to language translation	2
	PL4 Declarations and types	2
	PL5 Abstraction mechanisms	1
	PL6 Object-oriented programming	7
<i>Object first</i>		
CS101. Introduction to Object-Oriented Programming	PL1 Overview of programming languages	2
	PL4 Declarations and types	2
	PL6 Object-oriented programming	7
	PL8 Language translation systems	1
CS111. Object-Oriented Programming	PL4 Declarations and types	2
	PL5 Abstraction mechanisms	1
	PL6 Object-oriented programming	7
CS112. Oriented Design and Methodology	PL1 Overview of programming languages	2
	PL2 Virtual machines	1
	PL4 Declarations and types	1
	PL5 Abstraction mechanisms	2
	PL6 Object-oriented programming	4
<i>Functional first</i>		
CS111. Introduction to Functional Programming	PL1 Overview of programming languages	1
	PL4 Declarations and types	1
	PL5 Abstraction mechanisms	1
	PL7 Functional programming	4
CS112. Objects and Algorithms	PL1 Overview of programming languages	1
	PL2 Virtual machines	1
	PL4 Declarations and types	2
	PL5 Abstraction mechanisms	2
	PL6 Object-oriented programming	8
<i>Breadth first</i>		
CS100. Preview of Computer Science	PL6 Object-oriented programming	4
CS101. Introduction to Computer Science	PL1 Overview of programming languages	1
	PL3 Introduction to language translation	1
	PL4 Declarations and types	1
CS102. Algorithms and Programming Techniques	PL1 Overview of programming languages	2
	PL2 Virtual machines	1
	PL4 Declarations and types	2
	PL5 Abstraction mechanisms	2
	PL6 Object-oriented programming	3
CS103. Principles of Object-Oriented Design	PL3 Introduction to language translation	1
	PL5 Abstraction mechanisms	1
	PL6 Object-oriented programming	5

**Figura 3.2:** Materias en el IEEE/ACM Computing Curricula 2001 que contienen unidades de Lenguajes de Programación: tabla 1 de 2

<i>Algorithms first</i>		
CS111. Introduction to Algorithms and Applications	PL1 Overview of programming languages	1
	PL5 Abstraction mechanisms	2
	PL6 Object-oriented programming	4
CS112. Programming Methodology	PL1 Overview of programming languages	1
	PL2 Virtual machines	1
	PL3 Introduction to language translation	2
	PL4 Declarations and types	3
	PL5 Abstraction mechanisms	1
	PL6 Object-oriented programming	4
<i>Hardware first</i>		
CS111. Introduction to the Computer	PL1 Overview of programming languages	1
	PL4 Declarations and types	1
	PL5 Abstraction mechanisms	2
CS112. Object-Oriented Programming Techniques	PL1 Overview of programming languages	1
	PL2 Virtual machines	1
	PL4 Declarations and types	2
	PL5 Abstraction mechanisms	1
	PL6 Object-oriented programming	8
<i>Other first year courses</i>		
CS120. Introduction to Computer Organization	PL3 Introduction to language translation	1

**Figura 3.3:** Materias en el IEEE/ACM Computing Curricula 2001 que contienen unidades de Lenguajes de Programación: tabla 2 de 2

	I	O	F	B	A	H	Other
PL1. Overview of programming languages	4	4	2	3	2	2	0
PL2. Virtual machines	2	1	1	1	1	1	0
PL3. Introduction to language translation	4	0	0	2	2	0	1
PL4. Declaration and types	6	3	3	3	3	3	0
PL5. Abstraction mechanisms	3	3	3	3	3	3	0
PL6. Object-oriented programming	10	18	8	8	8	8	0
PL7. Functional programming	0	0	4	0	0	0	0
PL8. Language translation systems	0	1	0	0	0	0	0
PL9. Type systems	0	0	0	0	0	0	0
PL10. Programming language semantics	0	0	0	0	0	0	0
PL11. Programming language design	0	0	0	0	0	0	0
TOTAL Programming Languages	29	30	21	20	19	17	1

**Figura 3.4:** Horas dedicadas a unidades de Lenguajes de Programación en materias introductorias para cada aproximación (IEEE/ACM Computing Curricula 2001): Imperative first (I), Objects first (O), Functional first (F), Breadth first (B), Algorithms first (A), y Hardware first (H)

- CS220. Computer Architecture
- CS226. Operating Systems and Networking
- CS262. Information and Knowledge Management

- CS292. Software Development and Professional Practice
- *Basada en sistemas.* La idea es desarrollar un curricula vertebrado en el desarrollo de sistemas como tema unificador. Incluye material más profesional y técnico que en los otros modelos, manteniendo una cobertura razonable de los temas teóricos. Una implementación mínima constaría de las siguientes materias, además de las introductorias:
  - CS120. Introduction to Computer Organization
  - CS210. Algorithm Design and Analysis
  - CS220. Computer Architecture
  - CS226. Operating Systems and Networking
  - CS240. Programming Language Translation
  - CS255. Computer Graphics
  - CS260. Artificial Intelligence
  - CS271. Information Management
  - CS291. Software Development and Systems Programming
  - CS490. Capstone Project

Se destaca, por su relación con el presente proyecto docente, la materia *CS240. Programming Language Translation*.

- *Basada en la web.* En esta aproximación, se utiliza Internet y la World Wide Web como fundamento común para el curricula conjunto. Como ejemplo, se proponen las materias:
  - CS130. Introduction to the World-Wide Web
  - CS210. Algorithm Design and Analysis
  - CS221. Architecture and Operating Systems
  - CS222. Architectures for Networking and Communication
  - CS230. Net-centric Computing
  - CS250. Human-Computer Interaction
  - CS255. Computer Graphics
  - CS261. AI and Information
  - CS292. Software Development and Professional Practice
- *Híbridas.* Los anteriores modelos son sólo algunas posibilidades representativas. Sin embargo, se pueden considerar otras, donde se combinen elementos de varias de ellas. En la siguiente figura se muestran algunos ejemplos de materias que, combinadas con alguna de las secuencias de materias introductorias propuestas, cubren totalmente el conjunto de temas core.

En la Figura 3.5, se muestran las materias intermedias que tienen como contenido alguna unidad temática del área de lenguajes de programación. Mostramos en la Figura 3.6 el número de horas totales que cada aproximación dedica a las unidades temáticas de lenguajes de programación. Como se observa en la tabla, en todas las aproximaciones se considera la introducción a la traducción entre lenguajes (*PL3. Introduction to language translation*), y los sistemas de traducción (*PL8. Language translation systems*) son de gran importancia en la aproximación basada en sistemas.

Materia intermedia (aproximación)	Unidad	Horas
CS210. Algorithm Design and Analysis (C,S,T,W)	PL3 Introduction to language translation	2
CS226. Operating Systems and Networking (C,S)	PL6 Object-oriented programming	2
CS230. Net-centric Computing (T,W)	PL6 Object-oriented programming	2
CS240. Programming Language Translation (S)	PL1 Overview of programming languages	2
	PL2 Virtual machines	1
	PL3 Introduction to language translation	2
	PL8 Language translation systems	15
	PL9 Type systems	4
	Elective topics	16
CS250. Human-Computer Interaction (W)	PL6 Object-oriented programming	2
CS291. Software Development and Systems Programming (S)	PL6 Object-oriented programming	4

**Figura 3.5:** Materias intermedias con alguna unidad del área de Lenguajes de Programación, con distintas aproximaciones (IEEE/ACM Computing Curricula 2001): Topic-based (T), Compressed (C), System-based (S), Web-based (W)

	T	C	S	W
PL1. Overview of programming languages	0	0	2	0
PL2. Virtual machines	0	0	2	0
PL3. Introduction to language translation	2	2	2	2
PL4. Declaration and types	0	0	0	0
PL5. Abstraction mechanisms	0	0	0	0
PL6. Object-oriented programming	2	2	6	4
PL7. Functional programming	0	0	0	0
PL8. Language translation systems	0	0	15	0
PL9. Type systems	0	0	4	0
PL10. Programming language semantics	0	0	0	0
PL11. Programming language design	0	0	0	0
TOTAL Programming Languages	4	4	31	6

**Figura 3.6:** Horas dedicadas a unidades de Lenguajes de Programación en materias intermedias en cada aproximación (IEEE/ACM Computing Curricula 2001): Topic-based (T), Compressed (C), System-based (S), Web-based (W). Fuente: Curricula Computing 2001



### 3.1.5. Materias avanzadas

En la figura 3.7, se muestran las distintas materias avanzadas propuestas, clasificados por área. No se procede a analizar en detalle las distintas materias avanzadas, debido a que no se dispone de la descripción completa de las mismas. Sin embargo, es conveniente destacar la presencia de la materia *CS340 Compiler Construction*, dentro del área de lenguajes de programación, por su relación con el proyecto docente presentado. Se entiende que esta materia ampliará y profundizará en los conocimientos adquiridos con la materia intermedia *CS240. Programming Language Translation*.

Otras materias avanzadas también incluirán unidades del área de lenguajes de programación, obviamente las clasificadas en dicha área (*CS341. Programming Language Design*, *CS342. Programming Language Semantics*, *CS343. Programming Paradigms*, *CS344. Functional Programming*, *CS345. Logic Programming*, y *CS346. Scripting Languages*), pero probablemente también materias no asociadas al área (por ejemplo, como parte de *CS338. Programming for the World-Wide Web* o *CS397 Programming Environments*). Se puede observar cómo los nombres de algunas de las materias propuestas dentro del área de lenguajes de programación coinciden incluso con los nombres de unidades consideradas dentro del área (por ejemplo, *PL7. Functional programming*, *PL10. Programming language semantics*, y *PL11. Programming language design*).

## 3.2. Espacio Europeo de Educación Superior: competencias de un/a Ingeniero/a en Informática

En el contexto del *Espacio Europeo de Educación Superior*, la modificación del contexto tecnológico-productivo ha hecho emerger nuevos retos que, actualmente, apuntan hacia la construcción de *sistemas de enseñanza basados en competencias*.

Tradicionalmente, el sistema educativo se ha caracterizado por proporcionar un conocimiento fundamentalmente teórico, mientras que el sistema productivo se ha caracterizado por el desarrollo de capacidades y habilidades prácticas. La noción de *competencia profesional* pretende mejorar la relación del sistema educativo con el productivo, con el objetivo de promover una adecuada formación de los profesionales.

Las competencias profesionales se caracterizan por un conjunto de *conocimientos, procedimientos, actitudes, valores y rasgos*, que se complementan entre sí, de manera que el individuo debe *saber, saber hacer, saber estar, y saber ser*, para actuar de forma eficiente y eficaz en las diferentes situaciones en el ámbito profesional que se le presenten.

### 3.2.1. Las competencias de un/a ingeniero/a en Informática

En este marco, en el año 2000 surgió el *Proyecto Tuning*, que tiene como objetivo compartir y aprovechar la experiencia de las universidades para elaborar orientaciones que permitan diseñar y poner en práctica planes de estudios propios para las distintas

<b>Discrete Structures (DS)</b> CS301. Combinatorics CS302. Probability and Statistics CS303. Coding and Information Theory <b>Computational Science (CN)</b> CS304. Computational Science CS305. Numerical Analysis CS306. Operations Research CS307. Simulation and Modeling CS308. Scientific Computing CS309. Computational Biology <b>Algorithms and Complexity (AL)</b> CS310. Advanced Algorithmic Analysis CS311. Automata and Language Theory CS312. Cryptography CS313. Geometric Algorithms CS314. Parallel Algorithms <b>Architecture and Organization (AR)</b> CS320. Advanced Computer Architecture CS321. Parallel Architectures CS322. System on a Chip CS323. VLSI Development CS324. Device Development <b>Operating Systems (OS)</b> CS325. Advanced Operating Systems CS326. Concurrent and Distributed Systems CS327. Dependable Computing CS328. Fault Tolerance CS329. Real-Time Systems <b>Net-Centric Computing (NC)</b> CS330. Advanced Computer Networks CS331. Distributed Systems CS332. Wireless and Mobile Computing CS333. Cluster Computing CS334. Data Compression CS335. Network Management CS336. Network Security CS337. Enterprise Networking CS338. Programming for the World-Wide Web <b>Programming Languages (PL)</b> CS340. Compiler Construction CS341. Programming Language Design CS342. Programming Language Semantics CS343. Programming Paradigms CS344. Functional Programming CS345. Logic Programming CS346. Scripting Languages	<b>Human-Computer Interaction (HC)</b> CS350. Human-Centered Design and Evaluation CS351. Graphical User Interfaces CS352. Multimedia Systems Development CS353. Interactive Systems Development CS354. Computer-Supported Cooperative Work <b>Graphics and Visual Computing (GV)</b> CS355. Advanced Computer Graphics CS356. Computer Animation CS357. Visualization CS358. Virtual Reality CS359. Genetic Algorithms <b>Intelligent Systems (IS)</b> CS360. Intelligent Systems CS361. Automated Reasoning CS362. Knowledge-Based Systems CS363. Machine Learning CS364. Planning Systems CS365. Natural Language Processing CS366. Agents CS367. Robotics CS368. Symbolic Computation CS369. Genetic Algorithms <b>Information Management (IM)</b> CS370. Advanced Database Systems CS371. Database Design CS372. Transaction Processing CS373. Distributed and Object Databases CS374. Data Mining CS375. Data Warehousing CS376. Multimedia Information Systems CS377. Digital Libraries <b>Social and Professional Issues (SP)</b> CS380. Professional Practice CS381. Social Context of Computing CS382. Computers and Ethics CS383. Computing Economics CS384. Computer Law CS385. Intellectual Property CS386. Privacy and Civil Liberties <b>Software Engineering (SE)</b> CS390. Advanced Software Development CS391. Software Engineering CS392. Software Design CS393. Software Engineering and Formal Specification CS394. Empirical Software Engineering CS395. Software Process Improvement CS396. Component-Based Computing CS397. Programming Environments CS398. Safety-Critical Systems
--	---

**Figura 3.7:** Materias avanzadas en el IEEE/ACM Computing Curricula 2001, clasificadas por áreas

titulaciones, facilitando de este modo la convergencia. Hasta el año 2003, este proyecto desarrolló tres líneas:

- Identificación de las competencias que deseablemente deberían desarrollarse en la formación de todas las titulaciones, es decir las denominadas *competencias*

*genéricas o transversales.*

- Identificación de las *competencias específicas* en las titulaciones estudiadas.
- Definición del *crédito europeo* como unidad de medida común que permita comparar los currículos de las diferentes universidades.

La segunda fase de este proyecto, que comenzó en el 2003, aborda también una cuarta línea, referida a la enseñanza y al aprendizaje: *evaluación y rendimiento, y calidad.*

Al mismo tiempo, *Career Space*, un consorcio formado por grandes compañías de *Tecnologías de la Información y las Comunicaciones* (BT, Cisco Systems, IBM Europe, Intel, Microsoft Europe, Nokia, Nortel Networks, Philips Semiconductors, Siemens AG, Telefónica S.A. y Thales) y por la *Asociación Tecnológica Europea de Industrias de la Electrónica, la Información y las Comunicaciones* (EICTA), elaboró y publicó en el año 2001 el informe “*Directrices para el desarrollo curricular. Nuevos currículos de TIC para el siglo XXI: el diseño de la educación del mañana*”. Este trabajo recibió el apoyo de diversas organizaciones, como el Comité Europeo de la Normalización/Sistema de normalización de la Sociedad de la Información (CEN/ISSS), las sociedades nacionales de ingenieros electrónicos de Europa (Eurel) y la organización nacional de formación en TIC del Reino Unido (e-skills NTO), entre otros. Dicho informe concluye que:

- “*Los graduados en TIC necesitan una sólida base de capacidades técnicas, tanto en el campo de la ingeniería como de la informática, con especial atención a una perspectiva sistémica amplia. Precisan aprender a trabajar en equipo y tener alguna experiencia real en este sentido en proyectos donde se realicen distintas actividades en paralelo. Precisan también conocimientos básicos de economía, mercados y empresas.*”.
- “*Además, es necesario que los graduados en TIC adquieran unas buenas capacidades personales, como capacidad para la resolución de problemas, conciencia de la necesidad de la formación permanente, agudeza para comprender plenamente las necesidades de los clientes y de sus compañeros de proyecto, y conciencia de las diferencias culturales cuando actúen en un contexto mundial.*”.

Además, en el informe también se sugiere que las universidades definan en primer lugar el grupo de perfiles para los que se desea formar a sus estudiantes, y que para ello tengan en cuenta las empresas del sector y otras partes interesadas, recogiendo comentarios sobre los resultados previstos.

### **3.2.2. Las competencias de un/a ingeniero/a en Informática en España**

Los informes y estudios de *Career Space* provocaron la aparición en España de tres estudios de las profesiones relacionadas con la informática:

- *Propuestas de Acciones para la Formación de profesionales en Electrónica, Informática y Telecomunicaciones (PAPET, 2003)*, promovido por ANIEL, la Fundación de Tecnologías de la Información, y El Colegio Oficial y la Asociación Española de Ingenieros de Telecomunicación.
- *Perfil de la Profesión de Ingeniero Informática y Definición del Currículo Académico (COPITI, 2003)*, promovido por la Conferencia de la Profesión de Ingeniero e Ingeniero Técnico en Informática.
- *Profesiones y Perfiles en Informática (ALI, 2004)*, promovido por la Asociación de Doctores, Licenciados e Ingenieros en Informática.

Posteriormente, en el “*Libro Blanco de las titulaciones en Informática*”, publicado en el 2004, se realiza un análisis y comparación de todos estos trabajos en el capítulo “9. Competencias y perfiles profesionales del título académico de Grado”. En esta publicación, también se realiza un estudio de las *competencias transversales*, considerando la opinión de empresarios, docentes universitarios, y titulados en disciplinas relacionadas con la Informática; se concluye que: “*Las competencias transversales capacidad para resolver problemas, trabajo en equipo, capacidad de análisis y de síntesis, capacidad de organización y planificación, y capacidad de gestión de la información (captación y análisis de la información) son considerados por los tres colectivos (empresarios, docentes universitarios y titulados) las más importantes*”.

Debido a esto, y considerando que existen 3 perfiles profesionales claros y diferenciados del Ingeniero/a en Informática (*Desarrollo de Software, Sistemas, y Gestión y Explotación de las Tecnologías de la Información*) establecen cuáles deberían ser las competencias transversales genéricas para cada uno de ellos, valorándolas entre 0 y 4 y considerando su importancia en el perfil. (ver las Tablas 3.5 y 3.6).

Considerando estas tablas y el análisis comparativo del que se ha hablado anteriormente en el capítulo 9 del Libro Blanco, se propone una relación de las competencias específicas en relación con los tres perfiles considerados. Destacamos a continuación aquellas que, a mi juicio, pueden incidir en la enseñanza de Compiladores<sup>1</sup>:

- Para el perfil de *Desarrollo de Software*:
  - Dominar todas las etapas de la vida de un proyecto (análisis de concepción, análisis técnico, programación, pruebas, documentación y formación de usuarios).
  - Interpretar las especificaciones funcionales encaminadas al desarrollo de las aplicaciones informáticas.
  - Definir la estructura modular y de datos para llevar a cabo las aplicaciones informáticas que cumplan con las especificaciones funcionales y restricciones del lenguaje de programación.
  - Realizar pruebas que verifiquen la validez funcional, la integridad de los datos y el rendimiento de las aplicaciones informáticas.

---

<sup>1</sup>Datos extraídos del Libro Blanco.

Competencias Transversales Genéricas	Comunes a los perfiles: Desarrollo de Software, Sistemas, y Gestión y Explot. TI
Instrumentales	
1. Capacidad de análisis y síntesis	4
2. Capacidad de organización y planificación	4
3. Comunicación oral y escrita en la lengua nativa	3
4. Conocimiento de una lengua extranjera	3
5. Conocimientos de informática relativos al ámbito de estudio	3
6. Capacidad de gestión de la información	3
7. Resolución de problemas	3
8. Toma de decisiones	3
Personales	
9. Trabajo en equipo	4
10. Trabajo en un equipo de carácter interdisciplinar	3
11. Trabajo en un contexto internacional	2
12. Habilidades en las relaciones interpersonales	3
13. Reconocimiento a la diversidad y la multiculturalidad sistémicas	2
14. Razonamiento crítico	3
15. Compromiso ético	3
16. Aprendizaje autónomo	3
17. Adaptación a nuevas situaciones	3
18. Creatividad	3
19. Liderazgo	3
20. Conocimiento de otras culturas y costumbres	2
21. Iniciativa y espíritu emprendedor	3
22. Motivación por la calidad	4
23. Sensibilidad hacia temas medioambientales	2

**Tabla 3.5:** Competencias transversales genéricas en el Libro Blanco del Título de Grado en Informática

- Elaborar y mantener documentación descriptiva de la génesis, producción y operatividad de las aplicaciones informáticas.

- Para el perfil de *Sistemas*:

Competencias Transversales Específicas	Desarrollo Software	Sistemas	Gestión y Expl. TI
Análisis estadístico	3	3	4
Arquitectura de Computadores	2	4	2
Arquitectura de redes	3	4	3
Bases de datos	4	3	4
Capacidad para entender y evaluar especificaciones internas y externas	4	3	3
Cifrado y protección de datos	2	3	2
Conocimiento de productos tecnológicos y tendencias de la tecnología, asociados al segmento del mercado	4	4	4
Conocimientos creativos y artísticos	3	2	2
Dirección, planificación y gestión de proyectos	4	4	4
Diseño y arquitectura de sistemas de información	4	1	4
Documentación técnica	3	3	3
Evaluación de requisitos hardware	2	4	3
Gestión del cambio y del conocimiento	3	2	3
Ingeniería de software	4	1	3
Integración de sistemas	2	4	2
Interfaz con el usuario final	3	3	4
Matemáticas	2	3	3
Metodologías de configuración		3	
Métodos y Herramientas para el diseño y desarrollo de sistemas basados en computadores	4	3	3
Planificación, estrategia y organización empresarial	3	2	4
Programación	4	3	3
Robótica y automatización de procesos		3	2
Tecnología hardware		4	2
Visión comercial y empresarial	4	3	4

**Tabla 3.6:** Competencias transversales específicas en el Libro Blanco del Título de Grado en Informática

- Estudio de Métodos, Técnicas y Herramientas de Análisis y Diseño (en este caso, restringidas al ámbito de los compiladores y traductores así como a las especificaciones de diferentes lenguajes).

- Para el perfil de *Gestión y Explotación de las TI*:
  - Asegurar el seguimiento de los proyectos y su realización.
  - Definir con mayor precisión la necesidad técnica del cliente. Elabora la parte técnica de la propuesta.

### 3.2.3. Compiladores y el Libro Blanco del título de Grado de Informática

En el Libro Blanco de la titulación de Grado de los estudios relacionados con la Informática, existe una propuesta de *Contenidos Formativos Comunes (CFC)* para el Título de Grado. En dicha propuesta no aparece el estudio de técnicas de construcción de compiladores ni la palabra *compiladores* ni *intérpretes*. Sin embargo, en el anexo 9 de dicho Libro Blanco se recoge el documento “*Perfil de la profesión de Ingeniero en Informática y definición del Currículo Académico*”, presentado por la *Conferencia de la Profesión de Ingeniero e Ingeniero Técnico en Informática (COPITI)*. Hay que destacar que en el informe de la COPITI se considera que los contenidos troncales del Título de Grado deberían suponer al menos el 65-70 % de los créditos de la titulación, y que entre ellos se incluye el descriptor “*Procesadores de Lenguaje - Teoría de autómatas, gramáticas y lenguajes formales, computabilidad, compiladores, traductores e intérpretes, teoría de los lenguajes de programación*”.

Además, es interesante resaltar que en otro de los anexos del Libro Blanco, donde se proporciona información de programas de educación superior adaptados al EEEES en otros países europeos, se muestran algunos programas relacionados con el estudio de la Informática que consideran el estudio de la construcción de Compiladores: el “*Master of Science in Software Systems Engineering Institución*” de la Rheinisch-Westfälische Technische Hochschule Aachen (RWTH Aachen) en Alemania, y el “*Programa de Bachelor en Informática*” (equivalente al Grado en España) de la Universidad de Twente en Holanda.

## 3.3. Compiladores en Universidades de Estados Unidos

Para complementar el capítulo, se ha considerado interesante analizar el currículo de los principales cursos que tratan temas relacionados con la construcción de compiladores de dos universidades estadounidenses. Se ha optado por la Universidad de Stanford (California) y la Universidad de Columbia (Nueva York). La elección de estos cursos en estas universidades está motivada por las siguientes razones:

- En la mayoría de los rankings que existen sobre las universidades de Estados Unidos (EEUU), la Universidad de Standford se encuentra en un lugar preferente entre las universidades que imparten estudios relacionados con la Informática.

- En la Universidad de Stanford se encuentra uno de los grupos de investigación que más ha trabajado en temas relacionados con la construcción de compiladores, el dirigido por Jeffrey D. Ullman, co-autor de los principales libros de referencia sobre las técnicas y métodos para la construcción de compiladores en las universidades de todo el mundo, los famosos “libros del dragón”.
- El grupo dirigido por J.D. Ullman colabora activamente en temas relacionados con la construcción de compiladores con el grupo de investigación dirigido por Alfred V. Aho de la Universidad de Columbia; además A.V. Aho es co-autor de los denominados “libros del dragón”.

### 3.3.1. Universidad de Columbia

En la Universidad de Columbia, A.V. Aho imparte un curso introductorio, *COMS W4115 Programming Languages and Translators*, de un semestre (de enero a junio). Este curso está dirigido a estudiantes de Grado en informática, y se centra en los elementos semánticos y sintácticos de los lenguajes de programación y las técnicas y algoritmos usados por los compiladores para traducirlos en lenguaje máquina u otros lenguajes. El temario reducido del curso es el siguiente:

- 1.- Introducción a la traducción de lenguajes de programación.
- 2.- Temas relacionados con el diseño de lenguajes.
- 3.- Herramientas de procesamiento de lenguaje.
- 4.- Lenguajes de Scripting.
- 5.- Análisis Léxico.
- 6.- Análisis Sintáctico.
- 7.- Diseño de Gramáticas.
- 8.- El generador de analizadores léxicos y analizadores sintácticos descendentes ANTLR.
- 9.- El generador de analizadores léxicos Lex y el generador de analizadores sintácticos Yacc.
- 10.- Análisis sintáctico predictivo.
- 11.- Análisis sintáctico LR.
- 12.- Resolución de Conflictos en Análisis sintáctico.
- 13.- Traducción dirigida por la sintaxis.
- 14.- Análisis semántico.



- 15.- Organización del almacenamiento.
- 16.- Representaciones intermedias.
- 17.- Generación de código intermedio.
- 18.- Generación de código para el control de flujo.
- 19.- Generación de código.

Conviene señalar que la mayor parte del curso se dedica al análisis léxico y sintáctico, tratándose de forma más resumida el análisis semántico y generación de código.

### 3.3.2. Universidad de Stanford

En la Universidad de Stanford, por su parte, J.D. Ullman imparte el curso avanzado *CS243: Advanced Compiling Techniques*, de un semestre (de enero a junio), dirigido al alumnado que pretenda especializarse en técnicas de compilación. El temario reducido del curso es el siguiente:

- 1.- Introducción a la compilación.
- 2.- Flujos de datos (3 Lecciones).
- 3.- Eliminación de redundancias parciales.
- 4.- Propagación de constantes y variables.
- 5.- Asignación simple estática.
- 6.- Control de dependencias entre bloques.
- 7.- Emplazamiento en los registros.
- 8.- La recolección de *basura*.
- 9.- Paralelización y dependencias de datos.
- 10.- *Prefetch* de datos y *Software Pipelining*.
- 11.- Planificación de instrucciones.
- 12.- Análisis de punteros e IPO.
- 13.- Construcción de un compilador de producciones.

Además, J.D. Ullman también imparte un curso de verano de similar contenido: *Code Optimizations - A Quantitative Approach*. Este curso se centra en el estado del arte de las técnicas de optimización de código usadas en compiladores para procesadores modernos, con una base cuantitativa para el diseño y evaluación de estas técnicas.

Los estudiantes se introducen en un amplio rango de técnicas de optimización y transformaciones reestructuradas que mejoran el rendimiento del código generado por el compilador.

Conviene señalar que en los dos cursos impartidos por J.D. Ullman la mayor parte del contenido se dedica a la optimización y generación de código, obviándose los módulos de análisis léxico y sintáctico.

### 3.3.3. Conclusiones

Si se comparan estos cursos con la oferta del plan de estudios de Ingeniería en informática en la Universidad de Zaragoza, se observa que la asignatura de *Compiladores I* se puede considerar similar a la impartida por A.V. Aho *COMS W4115 Programming Languages and Translators*, y debería tomarse ésta como referencia. De forma similar, la asignatura de *Compiladores II* se contextualiza en el mismo marco de trabajo que la materia impartida por J.D. Ullman en los cursos *CS243: Advanced Compiling Techniques* y *Code Optimizations - A Quantitative Approach*.

## 3.4. Compiladores en las Universidades Españolas

En este apartado se describe el resultado del estudio realizado sobre la impartición de asignaturas de Compiladores en los planes de estudio de Ingeniería en Informática de algunas universidades públicas españolas. En concreto, se han seleccionado las siguientes: *Universidad Politécnica de Madrid*, *Universidad Politécnica de Catalunya*, y *Universidad de A Coruña*. El motivo de esta selección es que las dos primeras tienen las Facultades de Informática más antiguas de España (la creación de las facultades de Informática de Barcelona y Madrid se anunció en el Real Decreto 593/1976, de 4 de marzo), y la tercera es muy cercana a la autora de este trabajo y también tiene gran experiencia en el campo de la enseñanza de esta titulación (desde 1986).

### 3.4.1. Universidad Politécnica de Madrid (UPM)

En la Facultad de Informática de la Universidad Politécnica de Madrid, el actual plan de estudios de Ingeniería en Informática se imparte desde el curso 1996/1997. Su duración es de 5 años, siguiendo un modelo 2+3. En el plan de estudios existe una asignatura troncal anual de 9 créditos de cuarto curso, *Compiladores*. No se incluye en el plan de estudios ninguna otra asignatura en la que se haga referencia a la materia *Procesadores del lenguaje* del BOE de la titulación de Ingeniería en Informática. A continuación, se presenta esta asignatura en el curso 2006/2007, a partir de la información disponible en el sitio web mantenido por el profesorado de ésta [6]. El temario de la asignatura engloba:

- 1. Introducción.
  - 1.1 Introducción a la compilación.

- 1.2 Esquema de un compilador.
- 1.3 Componentes de lenguajes y gramáticas.
- 1.4 Máquinas reconocedoras y traductor.
- 2. Análisis léxico.
  - 2.1 Especificación y reconocimiento de componentes léxicos.
  - 2.2 Descripción y diseño del analizador léxico.
- 3. Tabla de símbolos.
  - 3.1 Estructura.
  - 3.2 Descripción y diseño.
- 4. Análisis sintáctico.
  - 4.1 Análisis ascendente y descendente.
  - 4.2 Análisis con y sin retroceso.
- 5. Análisis sintáctico ascendente.
  - 5.1 Precedencia de operador.
  - 5.2 Analizadores LR.
- 6. Análisis sintáctico descendente.
  - 6.1 Descendente recursivo predictivo.
  - 6.2 Analizadores LL.
- 7. Análisis semántico y generación de código intermedio.
  - 7.1 Traducción dirigida por la sintaxis.
  - 7.2 Comprobación de tipos.
  - 7.3 Lenguajes intermedios.
  - 7.4 Sentencias y expresiones.
- 8. Entorno de ejecución.
  - 8.1 Organización de la memoria en tiempo de ejecución.
  - 8.2 Estrategias de asignación de memoria.
  - 8.3 Acceso a variables locales, no locales y globales.
  - 8.4 Paso de parámetros.

- 9. Generación de Código.
  - 9.1 Códigos de máquina final.
  - 9.2 Esquemas de generación.
- 10. Optimización de código.
  - 10.1 Optimizaciones independientes de la máquina.
  - 10.2 Optimizaciones dependientes de la máquina.
- 11. Tratamiento de errores.
  - 11.1 Tipos de errores.
  - 11.2 Detección. Recuperación.
- 12. Tratamiento incremental e intérpretes.
  - 12.1 Compilación incremental.
  - 12.2 Intérpretes.

Con respecto a la evaluación de la asignatura, el 65 % de la nota corresponde a un examen teórico (que se puede realizar en dos parciales: uno en febrero, y otro en junio) y el 35 % restante corresponde a la nota de prácticas. Las prácticas consisten en la realización de un proyecto en grupos 3 personas como máximo. El proyecto consiste en el diseño e implementación de un compilador para un subconjunto del lenguaje Ada. Para la implementación del compilador se puede utilizar cualquier lenguaje de programación o herramienta. Se aconsejan:

- Bison o Yacc para el lenguaje C y generadores de analizadores sintácticos ascendentes LALR.
- COCO/R para los lenguajes C/C++ y generadores de analizadores léxicos y sintácticos descendentes recursivos.
- Flex o Lex para el lenguaje C y generador de analizadores léxicos.
- SDGLL1 exe Sistema Detector de Gramáticas LL(1).
- TS 2006 para los lenguaje C/C++ como tipo abstracto de datos de la Tabla de Símbolos sencilla.
- TS para el lenguaje C como tipo abstracto de datos de la Tabla de Símbolos.
- TS-OO para el lenguaje C++ como tipo abstracto de datos Tabla de Símbolos.

En la realización de la práctica a cada grupo se le asigna un método de Análisis Sintáctico diferente:

- Análisis LR.
- Descenso recursivo.
- Descendente con tablas.
- Precedencia de operador.

Y, de forma similar, también se le asigna a cada grupo un *Lenguaje Intermedio* diferente:

- Cuartetos.
- Tercetos.
- Tercetos indirectos.
- Notación Polaca Inversa (RPN).

Existen entregas parciales de los diferentes módulos que constituyen un compilador:

- 14 de diciembre de 2006: Analizador Léxico y Tabla de Símbolos y 12 casos de prueba.
- 28 de marzo de 2007: analizadores sintáctico y semántico.
- 6 de junio de 2007: entrega del compilador completo (generación de código intermedio, generación de código objeto, y los registros de activación).

### 3.4.2. Universidad Politécnica de Barcelona (UPC)

El actual plan de estudios de Ingeniería en Informática de la Universidad Politécnica de Catalunya fue aprobado en el año 2003. Su duración es de 5 años, siguiendo un modelo 2+3, y consta de 375 créditos. El plan de estudios incluye la asignatura *Compilación*, de 9 créditos, aunque anteriormente esta asignatura estaba distribuida en dos diferentes: *Compilación I (CL I)* y *Compilación II (CL II)*. No se incluye en el plan de estudios ninguna otra asignatura en la que se haga referencia a la materia *Procesadores del lenguaje* del BOE de la titulación de Ingeniería en Informática. A continuación, se presenta esta asignatura en el curso 2006/2007 a partir de la información disponible en el sitio web mantenido por el profesorado de esta [5]. El temario de la asignatura engloba:

- 1. Introducción.
  - 1.1 Fases de un compilador-Pequeño inciso sobre sintaxis, semántica y corrección.
  - 1.2 Organización de un curso de compiladores.

- 2. Análisis léxico.
  - 2.1 Un algoritmo de particionamiento y clasificación de la entrada.
  - 2.2 Descripción de componentes léxicos con PCCTS.
  - 2.3 Construcción de un generador de analizadores léxicos.
- 3. Análisis sintáctico.
  - 3.1 Algunas notaciones preliminares.
  - 3.2 Un algoritmo genérico de parsing.
  - 3.3 Parsing descendente y ascendente.
    - 3.3.1 Parsing descendente.
    - 3.3.2 Parsing ascendente.
  - 3.4 Generación de parsers descendentes recursivos.
  - 3.5 Programas PCCTS.
  - 3.6 Construcción de un generador de parsers.
- 4. Análisis semántico.
  - 4.1 Aspectos que influyen en el análisis semántico.
    - 4.1.1 Ámbito de declaración y visibilidad.
    - 4.1.2 Paso de parámetros.
    - 4.1.3 Tipo de datos estructurados.
  - 4.2 Análisis semántico en Compiladores.
    - 4.2.1 Cuestiones de implementación.
- Intérpretes y semántica del lenguaje.
  - 5.1 Ejemplo para un subconjunto de CL.
- 6. Generación de código intermedio.
  - 6.1 Traducción a Q-código.
    - 6.1.1 Variaciones respecto al modelo original.
    - 6.1.2 Arquitectura.
    - 6.1.3 Las Q-instrucciones.
    - 6.1.4 Consideraciones generales en la generación de Q-código.
    - 6.1.5 Llamadas a subrutinas y bloques de activación.
    - 6.1.6 Generación para tipos de datos estructurados.
    - 6.1.7 Variables locales y paso de parámetros.
    - 6.1.8 Paso de procedimientos formales como parámetro.

- 6.2 Generación de R-código.
  - 6.2.1 Consideraciones en la generación de R-código.
  - 6.2.2 Subrutinas.
  - 6.2.2 Generación para tipos de datos estructurados.

Con respecto a la evaluación de la asignatura, el 60 % de la nota corresponde a un examen teórico final y el 40 % restante a la nota de prácticas. Se realizan dos prácticas sobre las que se realiza un examen en el laboratorio. El primer examen de prácticas es a mediados de curso y consiste en realizar una modificación de la *Práctica 1* (la nota de dicho examen se denota  $P1$ ), mientras que el segundo examen se realiza al finalizar el curso y consiste en realizar una modificación de la *Práctica 2* (la nota de este examen se denota  $P2$ ). La nota global de prácticas ( $P$ ) se calculará mediante la fórmula  $P = \max(P2, (P1 + P2)/2)$ .

Las prácticas son individuales y se realizan en *PCCTS*, un método de *parsing* moderno y sencillo (*PCCTS* reemplaza a *Flex+Bison*). La primera práctica se centra en el análisis léxico y sintáctico, mientras que la segunda práctica se centra en la generación de código (concretamente, *R-código*). Por tanto, en conjunto se está realizando un compilador.

### 3.4.3. Universidad de A Coruña (UDC)

El actual plan de estudios de Ingeniería en Informática de la Universidad de A Coruña fue aprobado en el año 1995. Su duración es de 5 años, siguiendo un modelo 3+2, y consta de 366 créditos. El plan de estudios incluye la asignatura anual obligatoria de *Compiladores*, de 9 créditos. No se incluye en el plan de estudios ninguna otra asignatura en la que se haga referencia a la materia *Procesadores del lenguaje* del BOE de la titulación de Ingeniería en Informática. A continuación se presenta esta asignatura en el curso 2006/2007 a partir de la información disponible en el sitio web mantenido por el profesorado de ésta [5]. El temario de la asignatura engloba:

- 1. Introducción a los lenguajes y gramáticas.
  - 1.1 Teoría de autómatas finitos deterministas.
  - 1.2 Gramáticas independientes de contexto.
- 2. Análisis léxico.
  - 2.1 Introducción a Lex.
- 3. Análisis sintáctico.
  - 3.1 Análisis sintáctico por precedencia de operador.
  - 3.2 Análisis sintáctico LL (1).
  - 3.3 Análisis sintáctico SLR.

- 3.4 Análisis sintáctico LR.
- 3.5 Análisis sintáctico LALR.
- 3.6 Introducción a Yacc.
- 4. Acciones semánticas.
- 5. Generación de código intermedio.
- 6. Optimización de código.
  - 6.1 Propagación de constantes.
  - 6.2 Eliminación de estructuras de selección y bucles.
  - 6.3 Optimización de bucles.

Con respecto a la evaluación de la asignatura, hasta el curso pasado las prácticas eran de carácter obligatorio pero no influían en la calificación final (excepto en casos extremos). En una última revisión de la asignatura se ha optado por modificar los criterios de evaluación, considerando que:

- La nota de teoría supondrá el 75 % de la nota final y vendrá determinada por la media de 2 exámenes parciales, uno a finales de marzo y otro en junio.
- Las prácticas supondrán un 25 % de la nota final. Se realizan 3 prácticas. Las 2 primeras supondrán un 5 % de nota de la asignatura, y la última (*práctica especial*) supondrá un 20 %.

Las prácticas se realizan en parejas. La primera práctica consiste en la realización de un pequeño analizador léxico utilizando *Lex*. La segunda práctica consiste en la implementación de un analizador sintáctico SLR para una gramática sencilla en lenguaje C. Las dos primeras prácticas se entregan a final de marzo. Por último, la práctica especial es propuesta por los alumnos, y se entrega al final de curso.

#### 3.4.4. Conclusiones y Experiencias

En la mayoría de las universidades españolas, como en los ejemplos seleccionados en esta sección, los cursos de compiladores abarcan 9 créditos. En muchas ocasiones, la división de un curso de compiladores en *Compiladores I* y *Compiladores II* obedece más a razones administrativas que a consideraciones pedagógicas [9].

En general, el temario abarcado en los diferentes cursos que se imparten en la Universidad española es bastante similar. Las partes en las que existe mayor heterogeneidad son:

- Las herramientas de generación de analizadores léxicos, sintácticos y semánticos utilizadas. En la Universidad Politécnica de Catalunya se opta por *PCCTS*, en la Universidad de A Coruña por *Lex* y *Yacc*, y en la Universidad Politécnica de Madrid se ofrecen varias opciones (entre otras, *Flex* y *Bison*).



- Las elección del lenguaje al que se realiza la generación de código: *R-Código*, *Máquina virtual Java*, *P-Código*, etc.

Además de una carga teórica centrada en los contenidos mencionados, los cursos de compiladores consideran clases de laboratorio o prácticas que suponen aproximadamente un 30 % de las clases. En las prácticas, se propone al alumnado un proyecto en el que hay que construir un compilador para un subconjunto de un lenguaje. En este momento, surge la pregunta: ¿se elige un lenguaje existente o uno inventado?. Las ventajas de que sea inventado es que puede especificarse adecuándolo a los objetivos del curso y los estudiantes no tienen preconcepciones acerca de cómo hay que interpretar la sintaxis del lenguaje o cómo funciona. La desventaja es que están menos familiarizados con dicho lenguaje y, por tanto, a la hora de elaborar bancos de pruebas suele generarse mayor confusión. Además hay que decidir cuál va a ser el lenguaje destino al que se realiza la compilación/traducción: un lenguaje ensamblador real [2–4, 8] o una máquina virtual, como se sugiere en [10]. En los cursos de *Compiladores I* se suele realizar la traducción a un lenguaje de alto nivel (Ada, Pascal, C, etc.), mientras que en *Compiladores II* se opta por un lenguaje ensamblador o máquina virtual, donde, como se ha comentado, hay múltiples opciones. Relacionado con las prácticas de compiladores, nos parece interesante referenciar dos experiencias:

- En [10] se propone una aproximación iterativa para construir un compilador de forma incremental. Su motivación es que muchos estudiantes se quedan atrás y no llegan a profundizar en las etapas posteriores del diseño de los compiladores. Por ello, proponen una estructura mediante la cual el estudiante construye un compilador completo desde el primer día (pasando por todas las etapas de su diseño) y posteriormente se va refinando para añadir nuevas funcionalidades e introducir conceptos más avanzados. De este modo, se aseguran de que el estudiante entiende desde el principio las ideas básicas de cada etapa. Si bien parece un trabajo muy interesante, en nuestro caso exigiría la unificación de las asignaturas *Compiladores I* y *Compiladores II*. También advierten de la dificultad de realizar una planificación precisa, estimando bien la carga de trabajo (en este sentido, no les salió todo como hubiera sido deseable).
- En [12] se propone diseñar un compilador para un controlador de robot sencillo. Según los autores, esto resulta altamente motivador para los estudiantes, siendo uno de los elementos más atrayentes el que a la finalización del curso realizan una competición de robots donde los estudiantes escriben programas del controlador y utilizan sus compiladores para convertirlos a código máquina y ejecutarlos en los robots.

Para finalizar, considero interesante citar el artículo [9], donde se plantean aplicar el modelo de *aprendizaje como una actividad de investigación* para enseñar un curso de dos semestres en diseño de compiladores (de la Universidad de Alicante), basándose en las ideas de la *teoría constructivista del aprendizaje*: 1) aprender es construir conocimiento (no basta recuperar información, además hay que construir relaciones

entre conceptos); 2) si la información está aislada, se olvida fácilmente y pierde sentido; y 3) todo conocimiento depende del conocimiento previo, los nuevos conceptos se aprenden en relación a los conceptos aprendidos previamente. En sus planteamientos, el aprendizaje se centra en el alumno, que trabaja en grupos en un proyecto, y el profesor actúa de guía.

## Capítulo 4

# Metodología: ¿cómo queremos desarrollar nuestra labor docente en el contexto actual?

*“Los humanos no poseemos un saber innato ni podemos adquirirlo como el que se apropia de un bien mostrenco, por el simple procedimiento de fotocopiar unos apuntes, mirar la pantalla de un ordenador, escuchar una clase o leer un libro.”*

A. Llano

*“Siempre que enseñes, enseña también a dudar de lo que enseñas.”*  
Ortega y Gasset

Un proyecto docente debe ser un trabajo que refleje de forma razonable las ideas de quien lo realiza sobre el *qué*, *cuánto*, *cuándo*, *dónde* y *cómo* desarrollar la enseñanza de la materia. Los tres primeros aspectos (qué, cuánto y cuándo) quedan reflejados en la programación de la materia, que se va a introducir en el siguiente capítulo, mientras que *el dónde* y *cómo* se va a tratar en este capítulo. Con respecto *al dónde*, se tendrán en cuenta los diferentes recursos con los que cuenta el centro en el que se imparte la asignatura, mientras que en *el cómo* se hace referencia a la metodología docente que se va a seguir para alcanzar los objetivos que se proponen, considerando el tipo de alumnado, los recursos disponibles y la normativa vigente.

### 4.1. Consideraciones generales

Tradicionalmente, se le ha prestado más atención e importancia al *qué enseñar* y *cuánto enseñar* (en qué profundidad) que al *cómo enseñar*, lo que llevó a planificar una enseñanza centrada en el contenido y olvidando a los sujetos que debían aprender.

Esta situación fue analizada y criticada, por primera vez, por las diferentes co-

rientes pedagógicas de principios del siglo pasado, promoviéndose la programación de la *enseñanza centrada en objetivos*. Los objetivos plantean lo que el alumnado debe saber-hacer como consecuencia del proceso de aprendizaje, y además proporcionan un medio de evaluación del éxito de la docencia, debido a que se podrá observar el grado de cumplimiento de los mismos.

Es conveniente destacar que la enseñanza centrada en objetivos plantea por primera vez que una enseñanza de calidad debe perseguir no sólo que el alumnado adquiera ciertos *conocimientos teóricos* (como sucedía en la enseñanza tradicional), sino que también destaca la importancia de la adquisición de destrezas o *competencias* (conocimientos más prácticos), así como de *actitudes, valores y comportamientos*. Debido a esto, se comienza a prestar atención a la necesidad de realizar prácticas para alcanzar los objetivos relacionados con las destrezas o competencias y los aspectos psicoafectivos del alumnado.

Por otro lado, hoy en día nos encontramos en el proceso de convergencia al Espacio Europeo de Educación Superior, lo cual supone un gran cambio que afecta también a la metodología docente. En este contexto, habrá que diseñar un modelo centrado en el/la estudiante y el *modelo de aprendizaje-enseñanza-evaluación por competencias*, entendiendo las competencias como “*un saber hacer complejo resultado de la integración, movilización y adecuación de capacidades, habilidades, conocimientos, actitudes y valores en el desempeño de la ocupación que se trate, que además sean utilizados eficazmente y puedan dar respuesta a los problemas imprevistos*”. El trabajo del docente al diseñar una asignatura consiste en identificar las competencias genéricas y específicas (normalmente ya establecidas en el plan de estudios de la titulación, ver Sección 3.2) y desarrollarlas coherentemente en objetivos, contenidos y actividades, con una secuenciación apropiada.

Desde mi punto de vista, el proceso de convergencia europeo, como sucede con la mayoría de los cambios, va a exigir un enorme esfuerzo y un proceso de adaptación que no puede realizarse de un día para otro en el momento en que entre en vigor una ley. Debido a esto, considero conveniente plantear este proyecto teniendo en cuenta ese camino a la convergencia, y también considerando el marco actual en el que nos encontramos.

## 4.2. Metodología docente

El *modelo docente tradicional* de la Universidad se ha basado principalmente en la transmisión y recepción de conocimientos ya elaborados, *primando la enseñanza en lugar del aprendizaje*. Una consecuencia directa del uso indiscriminado de este método docente es la adopción frecuente, por parte del alumnado, de *actitudes pasivas* frente al aprendizaje, y de hábitos memorísticos y de reproducción mecánica de los modos de razonamiento presentados por el profesorado. Además, de este modo no se garantizan los principios de calidad de aprendizaje a los cuales se debe aspirar puesto que no se alcanza un aprendizaje significativo.

Frente a este modelo de aprendizaje, se pretende lograr un modelo dinámico en el que se incluya la motivación y orientación del alumno en su aprendizaje, haciendo que

el protagonismo del proceso recaiga en el alumnado y no en el profesorado. Enseñar ha de ser “*dirigir con técnicas apropiadas el proceso de aprendizaje de los alumnos*”; es decir, “*encaminarlos hacia los hábitos de aprendizaje significativo que los acompañarán a lo largo de la vida*”. Por consiguiente, el profesorado debe motivar, orientar y facilitar al alumnado el trabajo activo de buscar conocimientos. Ha de ser capaz de transmitir desde el primer momento la utilidad de lo que enseña, renunciar, cuando sea preciso, al rigor de las demostraciones en favor de un mejor seguimiento, usar un lenguaje apartado de formalismos pero preciso, etc. Utilizando una cita de Einstein “*se trataría de hacer las cosas tan simples como sea posible, pero sin simplificarlas*”.

La selección de los métodos no ha de tener en cuenta únicamente las características de la materia que se vaya a impartir, sino también las peculiaridades del entorno donde se desarrolla la actividad docente y a quién va dirigida. También se debe tener en cuenta que no se puede hablar de métodos docentes totalmente buenos o malos, ya que todos ellos tienen su parte positiva y su parte negativa. Las opciones que se dan a continuación están delimitadas por las condiciones del entorno y no se consideran inamovibles. Además, consideran metodologías activas.

#### 4.2.1. Metodología didáctica

Se distingue entre clases en el aula (teóricas y de problemas) y clases en el laboratorio (prácticas), aunque la idea fundamental es plantear constantemente situaciones problemáticas, ejercicios y otro tipo de actividades que mantengan a los estudiantes activos.

##### Las clases teóricas en el aula

En general, el método que se va a emplear en este tipo de clases va a ser el conocido como *lección magistral participativa*. Según diversos estudios pedagógicos, existen varios elementos fundamentales a tener presentes en una lección de tipo magistral de este tipo:

- *Delimitar, seleccionar y ordenar los contenidos y estructurar la clase:* ¿cuál es el número de conceptos nuevos a introducir?, ¿en qué profundidad se incide en cada uno de ellos?, ¿en qué orden se introducen?.
- *En la exposición, se debe combinar el rigor con la intuición.* El rigor expositivo debe fundamentarse en la calidad de los contenidos y apoyarse en un lenguaje claro que evite puntos oscuros y ambigüedades. Sin embargo, no hay que olvidar que estas explicaciones formales, en muchas ocasiones, no facilitan en absoluto la comprensión de los conocimientos que se intentan transmitir. En estas ocasiones, un enfoque intuitivo, aplicado con oportunidad, contribuye a facilitar la transmisión de conocimientos. Este enfoque puede preceder a la formalización rigurosa del contenido, consiguiendo que el alumnado capte mejor el sentido de esa formalización. Otras veces, conviene proceder a la inversa, para

cerrar un tema y para confirmar o descartar conclusiones teóricamente plausibles. En cualquier caso, la clase debe estar convenientemente complementada con la realización de ejercicios por parte del alumno.

- *La primera vez que se vaya a exponer un tema conviene desarrollarlo con antelación*, por ejemplo, ante los miembros de la propia área de conocimientos y, si fuese necesario, remodelar la exposición con las sugerencias que aporten.
- *Entregar por adelantado apuntes al alumnado*, sobre todo en aquellos temas en los que abundan los gráficos o esquemas, para apoyar la exposición oral. Con esto, se consigue que el alumnado no tengan que utilizar el tiempo de la clase para escribir lo que el profesor expone y pueda aprovechar las explicaciones que se le dan. Además, es conveniente adaptar los apuntes para permitir al alumnado anotar explicaciones y comentarios. En caso de que no se entregue el material, la tendencia observada es la de copiar las transparencias con los esquemas y gráficos y lo que se dice el profesorado, lo que supone un esfuerzo de seguimiento mayor.

En las clases en el aula también se va a utilizar el método activo conocido como *Cuestionario de Incidencias Críticas (CUIC)*. Este método tiene como objetivo recoger la opinión del alumnado de los aspectos positivos y negativos que se han dado a lo largo de la clase. De este modo, el profesorado va a tener una continua retroalimentación de las sensaciones que experimentan, y se podrá adaptar a ellas. El método consiste en pedir al alumnado, en el último minuto de clase de cada día, que escriba en una hoja primero las incidencias positivas (*IC+*) y luego las negativas (*IC-*). Algunos elementos fundamentales a tener presentes en este método son:

- Las incidencias deben escribirse muy rápido y en un tiempo muy corto. Se aconseja que el alumnado escriba sólo dos o tres frases o simplemente ponga palabras clave sin estructurarlas en una frase.
- Es fundamental que al principio de la siguiente sesión en el aula el profesorado dedique 2 ó 3 minutos a comentar el CUIC del día anterior, incidiendo en las estrategias que se van a llevar a cabo para solucionar los problemas detectados.

Otras estrategias que se propone utilizar en el aula para llevar a cabo la lección magistral participativa e implantar los principios metodológicos expuestos al comienzo de la Sección 4.2, evitando las desventajas de la lección magistral tradicional, son:

- Llegar al aula 5 minutos antes del comienzo para acondicionarla para el desarrollo de la clase: borrar la pizarra, encender el proyector de transparencias o vídeo, encender las luces, etc.
- Cuando llegue la hora de inicio, cerrar las puertas y saludar al alumnado.
- *Iniciar siempre la clase con un repaso* de los principales conceptos o temas explicados en las clases anteriores, a modo de resumen.

- Después del repaso inicial, *indicar de forma general lo que se pretende ver* en la clase de ese día *y el sentido que tiene* en el marco general del tema o asignatura, destacando la utilidad para que quede bien justificado. Una estrategia que se usará a menudo es plantear una situación problemática a la que dan respuesta los conocimientos que se van a explicar a continuación. Es decir, primero se pretende que el alumnado entienda el problema y la necesidad de que exista una solución al mismo. Además, si se dispone del tiempo necesario, se le pedirá al alumnado que piense estrategias de solución y se analizarán estas antes de proporcionar la solución ya elaborada; de este modo, se podrá sacar partido a las ventajas del *aprendizaje por descubrimiento*.
- *Utilizar el retroproyector y fotocopias* para transmitir información como definiciones, algoritmos, gráficos etc., para permitir al profesorado una mayor calidad expositiva.
- *Explicar muy detalladamente los nuevos conceptos, enmarcándolos* en la medida de lo posible en el problema para el que constituyen una solución. No importa que sea sencilla, se planteará de forma razonada la construcción de la solución, así como el estudio de las posibles soluciones alternativas que puedan aparecer en su gestación. Sólo así podrá evitarse la sensación de aparición mágica que la simple presentación de una solución puede producir.
- *Hacer paralelismos* de lo explicado con cosas que ya conocen, por ejemplo comparando las fases de un compilador con el análisis morfo-sintáctico utilizado en lingüística.
- *Estimular al máximo la participación* y atención del alumnado, creando el clima adecuado para conseguir que hablen, que den ideas, que discutan soluciones, o al menos que manifiesten si comprenden o no lo explicado. Algunas *estrategias para conseguir la participación* que se van a aplicar son:
  - Realizar *preguntas directas* (por ejemplo, pidiendo ejemplos o soluciones o preguntando por conceptos explicados con anterioridad).
  - Pedir a personas concretas *que resuelvan en la pizarra* ejercicios planteados (habiéndoselos revisado primero para que salgan a la pizarra con la seguridad de que lo tienen bien).
  - *Dar una solución incompleta o con alguna incorrección* y preguntar si la solución está bien para todos los casos, poniendo a la clase a buscar el fallo o la incompletitud de lo explicado.
  - Es importante también tener en cuenta que es *imprescindible que nunca se ridiculice* lo dicho por un estudiante.

Dado que la curva de atención decae tras los primeros veinte minutos de exposición, se considerarán las siguientes *estrategias para renovar la atención*:

- Dar un *descanso* de un par de minutos en la exposición *planteando un pequeño problema*.
- Plantear un *grupo de cuchicheo*, es decir, dar la respuesta primero a un/a compañero/a y luego hacer una puesta en común.
- 5 minutos antes de finalizar la clase, *resumir lo que se ha visto* ese día y recordar/indicar al alumnado uno o más libros, artículos o direcciones de Internet donde puede ampliar la información. De este modo, se evitará que los apuntes (a veces muy deficientes) sean el único material de estudio del alumnado.
- Al finalizar un tema o bloque temático, pedir al alumnado que elabore un *mapa conceptual* de media carilla, que deberán entregar al comienzo de la siguiente sesión en el aula. Con la elaboración de los mapas conceptuales y las clases de laboratorio y de problemas se pretende que el alumnado lleve al día la asignatura.
- Al finalizar un tema o bloque temático, proporcionar al alumnado *fichas de problemas y ejercicios* relacionados con el tema, para que puedan trabajar en ellos.
- Finalizar la clase despidiéndose cuando llegue la hora y posteriormente recoger el material que se ha llevado a clase y los CUIC elaborados por el alumnado. De este modo el alumnado podrá descansar antes de que comience la siguiente sesión. Además, se debe borrar el encerado para que el profesorado de la siguiente clase en el aula lo tenga disponible.

Es conveniente señalar aquí también que los apuntes elaborados por el alumnado con las notas tomadas en clase en los diagramas facilitados por el profesorado no pueden ser la única fuente de estudio para la preparación de una asignatura. Es necesario contar con material auxiliar, puesto que no toda la información expuesta puede ser captada por el alumnado. Además, a menudo esta información de clase puede ser insuficiente, por muy diversos motivos, para la comprensión de los conceptos por parte del alumnado. Aquí entra en juego el papel del profesorado como suministrador de información bibliográfica. Se consideran estas *fuentes de información adicional*:

- *Libros básicos*. En la literatura sobre compiladores se pueden encontrar varios libros buenos que pueden ser utilizados a modo de libro de texto. Sin embargo, es difícil encontrar uno que incluya todos los temas que el profesorado cree oportunos con el enfoque que considera más conveniente. Mi opinión es que hay que hacer un esfuerzo y proporcionar un único libro básico que sirva de referencia al alumnado y preocuparse de que esté accesible a través de la biblioteca.
- *Libros complementarios*. Existen en la literatura muchos libros de interés que profundizan en temas concretos. El profesorado debe facilitar referencias a esos libros, que pueden ser de gran utilidad para que el alumnado interesado en aclarar sus dudas o ampliar conocimientos sobre algún tema en particular obtenga más información.



- *Artículos de revistas o conferencias.* A veces es posible encontrar en revistas artículos especialmente ilustrativos sobre temas explicados en clase. Suele ser suficiente con que el propio profesorado haga un resumen de algunos temas publicados y en el caso de que alguien se muestre especialmente interesado recomendarle/proporcionarle los artículos.
- *Direcciones de sitios web.* En la actualidad, Internet es sin lugar a dudas un sitio donde se puede encontrar de forma inmediata información muy actualizada y contenidos gratuitos que pueden ser obtenidos por el alumnado y profesorado: apuntes, transparencias, ejemplos, ejercicios, software, etc. Es preciso, no obstante, saber seleccionar la información adecuadamente.

Resumiendo, el método docente para las clases teóricas en el aula estará basado en clases magistrales participativas apoyadas en el uso de la pizarra, transparencias y, ocasionalmente, en el uso de un ordenador conectado a un cañón de video para mostrar determinadas características de alguna aplicación informática. Además, para tener una retroalimentación continua se emplearán CUICs. Por otro lado, es deseable que el alumnado disponga de libros básicos y que consulten otras fuentes de información.

### Las clases de problemas en el aula

Desde la aparición de los planes estructurados en créditos, los problemas tienen su propio tiempo asignado en forma de clases reguladas, que garantizan el cumplimiento de los créditos prácticos de la asignatura. Sin embargo, no debe descartarse resolver puntualmente algún ejercicio práctico insertado dentro de una clase teórica. En este caso, es principalmente el profesorado quien presentaría el ejercicio práctico y lo resolvería. Sin embargo, en general, la solución de los problemas no la debe proponer el profesor sino que debe fluir de la discusión entre el alumnado, por lo que sería deseable que las clases de problemas fuesen muy participativas, y que fueran los propios alumnos los que expusieran y criticasen diversas soluciones. Quedaría para el profesorado la tarea de supervisar estas soluciones y sus críticas.

Los ejercicios a desarrollar en estas clases han de ser de dos tipos: *ejercicios pedagógicos* y *ejercicios prácticos*. Los primeros son aquellos cuya finalidad es ayudar al alumnado a comprender un determinado concepto, hacerle captar una determinada problemática o intuir nuevos conceptos. En ellos, en general se presentan situaciones ideales o fuera de lo habitual. Con los ejercicios prácticos, sin embargo, se busca que el alumnado vea aplicaciones de los conceptos explicados y, sobre todo, adquiera la capacidad de aplicarlos por sí mismo. Una buena fuente de problemas a desarrollar en clase pueden ser los ejercicios propuestos como examen en cursos anteriores. Esta situación permite al alumnado, además, hacerse una idea general del nivel que se espera de él en los exámenes.

Las clases de problemas son un marco muy apropiado para el fomento de la participación del alumnado en las clases. En ellas, tienen cabida *técnicas de dinámica de grupos* que se aplican en menor grado en las lecciones magistrales participativas. Así, el diálogo o debate, la consulta del alumnado, o los grupos de discusión, pueden

ser utilizados en este tipo de clases. Además, en ellas podrá apreciarse claramente el grado de aprendizaje de los conceptos impartidos mediante las clases teóricas en el aula. Indudablemente, para ello debe existir una coordinación total entre el temario de las lecciones magistrales y el de las clases prácticas, si bien puede ser conveniente que contenidos impartidos con bastante anterioridad en las clases de teoría formen también parte, de vez en cuando, de los problemas que se resuelvan para, de este modo, repasarlos y observar su grado de fijación. Por otro lado, en estas clases de problemas puede obtenerse una idea clara de los defectos con que han sido aprendidos los contenidos sobre los que versen. Las conclusiones pueden ser francamente útiles, y por ello deben considerarse dichas clases como un instrumento de vital importancia.

Resumiendo, el método docente para las clases de problemas en el aula estará basado en técnicas de dinámica de grupos. Además, para tener una retroalimentación continua se emplearán CUICs (comentados previamente).

### Las clases en el laboratorio

Los laboratorios y prácticas asociados a las asignaturas pretenden reforzar los conocimientos impartidos en las clases del aula, permitiendo al alumnado la aplicación de los mismos en un entorno real. Esto no debe suponer que la formación recibida sólo con las clases teóricas sea incompleta; está claro que la asistencia simultánea a una asignatura y su laboratorio pone al alumnado en las mejores condiciones para alcanzar los objetivos planteados por el profesorado. También es posible dotar a un laboratorio o una práctica de contenidos propios, de modo que tenga un corte eminentemente práctico pero con un interés teórico que, o bien es escaso, o bien va más allá de lo que es posible incorporar al temario.

Un laboratorio o práctica se estructurará como un conjunto de actividades que exijan una participación activa del alumnado. En ocasiones, sin embargo, el profesorado incidirá en algunos conceptos vistos superficialmente en el aula (o introducidos específicamente para el trabajo práctico) y presentará herramientas (equipos físicos, software, etc.) que el alumnado va a necesitar para la realización de su trabajo.

A la hora de diseñar los contenidos de una actividad práctica, el profesorado debe plantearse al menos dos cuestiones: qué *objetivos* tiene la práctica, qué *fundamentos teóricos* requiere, y cuál es su *nivel de complejidad*. Los objetivos son una buena medida del interés de la práctica, a nivel pedagógico o profesional. Los fundamentos teóricos influyen a la hora de decidir cuándo empezar la realización de una práctica, y si es necesario o no un soporte de clases teóricas o material complementario. El nivel de complejidad influirá básicamente en el tiempo asignado al alumnado para la realización de la práctica, y también en los criterios empleados para su evaluación.

Las actividades prácticas se realizarán, normalmente, en grupos de dos personas. Se hace este planteamiento porque de cara al futuro profesional del alumnado es importante fomentar el trabajo en equipo: la mayor parte de los proyectos a los que los/as titulados/as en Ingeniería en informática deben enfrentarse son demasiado grandes y complejos para que los realicen una única persona.

Cuando las disponibilidades de horario, de laboratorio y de profesorado lo permitan, será interesante añadir actividades voluntarias de refuerzo o ampliación de temas

que se proponen. Así, lo ideal es que un laboratorio disponga de un grupo mínimo de prácticas a realizar por todo el alumnado, más un grupo de actividades optativas para que cada alumno/a o grupo elija alguna de ellas, de manera acorde a sus inquietudes o necesidades.

La labor del profesorado durante la realización de este tipo de trabajos debe limitarse a la corrección de errores que hayan conducido a situaciones de estancamiento y a descartar, a través de la discusión con el alumnado, soluciones manifiestamente incorrectas. Es muy importante evitar cambios en los objetivos planteados, añadir nueva información durante la realización de una práctica, o modificar los plazos de realización. Si alguno de estos cambios es necesario, habrá que hacerlo, pero significará que la práctica no se diseñó de forma adecuada.

Los requisitos para una correcta realización del supuesto práctico planteado deberán pasar por la entrega del trabajo realizado, comentado. Además, se realizará una pequeña *presentación de la práctica*, en la que se aportarán las conclusiones y comentarios que se crean convenientes. Todo este trabajo deberá ser evaluado de forma continua, con el fin de evitar, o detectar en el caso de que hayan ocurrido, las situaciones de enfoque erróneo o estancamiento. También es recomendable que la evaluación de las prácticas en el laboratorio tenga una ponderación importante en la evaluación final del alumnado en la asignatura. De esta forma, se consiguen dos objetivos: 1) Hacer patente la importancia de la parte práctica en la formación que hay que adquirir, y 2) *premiar* al alumnado por su trabajo en prácticas, pues verá el esfuerzo invertido reflejado en la calificación final.

### Las tutorías

Las tutorías son un complemento fundamental al método de enseñanza basado en clases teóricas. Los objetivos que se plantean para las tutorías son:

- Proporcionar al alumnado la oportunidad de exponer al profesorado dudas que hayan surgido del estudio del temario o simplemente una ocasión para profundizar en aspectos concretos.
- Determinar el aprovechamiento de las lecciones impartidas.
- Establecer una relación más personalizada con el alumnado pudiendo adaptarse a sus necesidades individuales y específicas.
- Proponer trabajos de refuerzo o ampliación a estudiantes que lo necesiten o lo soliciten.

Para ello, el horario en que se impartan las tutorías debe estar definido de acuerdo con la disponibilidad de tiempo del alumnado. Además, se debe dar la opción de concertar tutorías, previo acuerdo con el profesorado, fuera de este horario, en caso de que algún alumno/a no pudiese asistir a esas horas. También se debe animar al alumnado a acercarse a las tutorías, invitándoles repetidamente a que lo hagan. Otra estrategia que se va a aplicar, para fomentar la asistencia, es la de comentar la

revisión de los mapas conceptuales y ejercicios planteados (y no resueltos en clase) en las tutorías.

### 4.3. La evaluación

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

#### 4.3.1. La evaluación del alumnado

La tarea de la evaluación es un punto crítico y delicado de la tarea docente, por cuanto debe lograrse calificar lo más justamente posible al alumnado. En numerosas ocasiones, es difícil decidir qué criterios elegir, de entre todos los posibles, para realizar la evaluación y calificación. Esta dificultad emana, por una parte, de la diversidad de métodos docentes que se emplean y del número de alumnos/as a los que se aplican, así como de sus características individuales, y, por otra parte, de la rigurosidad que debe tenerse a la hora de evaluar los conocimientos del alumnado. Además, para realizar la evaluación es sumamente importante la coordinación de todo el profesorado implicado en la asignatura. Desde los criterios a los métodos, pasando por el contenido de las pruebas de evaluación, todo debe estar consensuado, de forma que se garantice al alumnado el derecho a la igualdad en la evaluación. Las reflexiones que se ofrecen a continuación tratan de ser una base para ese consenso.

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 serían deseables. La valoración debe hacerse sobre el grado de conocimientos adquiridos y sobre la formación global en el campo concreto de la asignatura. *Debe primarse la capacidad de razonamiento* más que la memorística y el enciclopedismo. Para evaluar según estos criterios, el diseño de pruebas y exámenes debe ser muy cuidadoso.

Proponer a lo largo del curso una serie de ejercicios sobre los contenidos del temario es una ayuda adicional que mejora el proceso de evaluación. Lamentablemente, su uso generalizado sólo es viable en grupos poco numerosos, pues el profesorado debería analizarlos y devolverlos comentados y calificados.

En cuanto a los mecanismos para proceder a la obtención de las calificaciones del alumnado, no hay fórmulas mágicas. Es imposible eliminar totalmente la subjetividad inherente a ese proceso; lo que se apunta es la conveniencia de planificar unos *criterios globales, discutidos por todo el profesorado* implicado. Desafortunadamente, nunca se alcanzará un método objetivo y universal que automatice la ponderación de tantos factores como aquí intervienen.

Conviene también considerar una serie de factores en el planteamiento de pruebas finales, de cara a mejorar en lo posible la fiabilidad de sus resultados:

- Procurar que sea *completa*. Esto supone que debe constar de un número suficiente de ejercicios y que abarque los campos y aplicaciones cuyo conocimiento se considere imprescindible.

- Buscar un *grado de dificultad consecuente con el desarrollado durante el curso y con los objetivos de la asignatura*.
- *Dar suficiente tiempo* para su resolución, acorde al grado de dificultad de cada ejercicio. Habría que procurar además que la duración total de los exámenes finales no supere las 3 horas, ya que lo contrario repercute negativamente en el rendimiento del alumnado.
- *Indicar el valor de cada ejercicio*, con el fin de orientar al alumnado durante la prueba. También es interesante *evitar el todo o nada*, por ejemplo mediante la división en apartados de las preguntas. Esto último, claro está, no siempre es aplicable.

A continuación se expone el método propuesto para la evaluación del alumnado que cursa la asignatura de Compiladores I.

#### La evaluación del alumnado en Compiladores I

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 los 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.

#### 4.3.2. 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 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 habría que reaccionar. Fundamentalmente, es la forma de realizar la explicación, el tiempo dedicado a trabajar los 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, a través de sus centros, 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 ésta. 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 opiniones negativas 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, es conveniente elaborar también nuestra propia encuesta, completamente enfocada a la asignatura en cuestión.

El resultado de las pruebas finales o las 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 resultados esperados. Además, el profesorado tratará de grabar sus clases para luego reflexionar sobre ellas o analizarlas con otros miembros del equipo docente.

## Capítulo 5

# Programa docente de Compiladores I

En este capítulo, y en base a las consideraciones realizadas a lo largo del documento, se detalla el programa docente de la asignatura de *Compiladores I*.

### 5.1. El alumnado

La asignatura de *Compiladores I* se imparte en el primer cuatrimestre de cuarto curso. Por ello, se entiende que el alumnado ha manejado suficientemente compiladores y tiene nociones de lenguajes de programación, gramáticas, y autómatas. Normalmente, se matriculan en la asignatura alrededor de 100 alumnos que se distribuyen entre el grupo de mañana y tarde, siendo el grupo de mañana mucho más numeroso (alrededor de 70 matriculados en el grupo de mañana). Alrededor del 75 % de los matriculados asiste a clase regularmente. Además se observa que muchos alumnos/as que tienen asignado el grupo de tarde asisten a clase por la mañana debido a que prefieren ese horario pero por otras razones (ofertar grupos de mañana y tarde, evitar la masificación en las aulas, etc.) se les ha asignado el grupo de tarde.

La mayoría del alumnado matriculado ha cursado el primer ciclo de la titulación de Ingeniería en Informática en el CPS, pero 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. Hay que señalar también que cada vez es más común que exista un grupo reducido de estudiantes Erasmus (sobre el 2 %) y que algunos estudiantes de la Universidad de Zaragoza matriculados en la asignatura estén realizando alguna estancia en alguna institución en el extranjero. Por último, también es conveniente indicar que alrededor del 5 % de los estudiantes compaginan sus estudios con la realización de una actividad

profesional.

## 5.2. Justificación

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.

Un elemento clave para que el estudiante aprecie la importancia de la asignatura es 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.). En [7], se destaca la importancia de hacer estas generalizaciones patentes en la clase, dada la dificultad que experimentan los estudiantes a la hora de transferir sus conocimientos a otros ámbitos de aplicación similares.

De acuerdo con [8], un proyecto de construcción de un compilador es relevante para un currículo básico incluso si el propósito no es el propio estudio de los compiladores. Así, *la construcción de un compilador no tiene por qué 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 en Informática*:

- Los estudiantes trabajan en *un proyecto completo*, y aprenden a desarrollar documentación y a depurar y realizar pruebas de funcionamiento. Así, el diseño de un compilador se puede decir que es uno de los proyectos más complejos que tiene que realizar un estudiante de la titulación, con la única excepción del Proyecto Fin de Carrera [1, 8, 9].
- 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 para los estudiantes: 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.) [12].



- 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.). Dado que el proyecto de diseñar un compilador es suficientemente complejo, puede utilizarse conjuntamente para enseñar a los estudiantes otros aspectos del diseño de proyectos. Como ejemplo, en [10] los estudiantes aprenden a manejar diversas herramientas y sistemas de control de versiones, y en [12] utilizan distintos patrones de diseño.

En [1], se destacan importantes retos a la hora de diseñar un curso en compiladores, desde el punto de vista del diseño de los proyectos que tienen que desarrollar los estudiantes. Así, los proyectos de compiladores son muy costosos de crear: hay que especificarlos muy bien, asegurarse de que son tratables por los estudiantes en el tiempo de que disponen, deberían ser modulares (en el sentido de que probablemente los errores acumulados en etapas previas no deberían repercutir para el estudiante en fases sucesivas e independientes del desarrollo del proyecto<sup>1</sup>: análisis léxico, análisis sintáctico, análisis semántico, generación de código, y optimización), etc. Además, los proyectos deberían poder modificarse de un año a otro con relativa facilidad, para evitar que se convierta en un problema típico y que circule demasiada información procedente de estudiantes de otros cursos.

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. 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. *Estudiar compiladores ayuda a los desarrolladores a hacer mejor su trabajo* [8]. En definitiva, 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 [12].

### 5.3. Objetivos

Se considera oportuno señalar aquí un conjunto de objetivos, tanto de carácter general o transversal (ver Sección 5.3.1) como de carácter específico (ver Sección 5.3.2), que se pretenderá que alcance el alumnado a lo largo del curso. Estos objetivos son los que regirán el modo de actuar del profesorado.

#### 5.3.1. Objetivos generales

Además de las competencias propuestas para una determinada disciplina universitaria como la Ingeniería en Informática, existen competencias generales comunes a todas las disciplinas, que se consideran fundamentales. En la misión de tratar de educar ciudadanos/as más capacitados, nos establecemos los siguientes objetivos generales:

---

<sup>1</sup>Ésta es una afirmación discutible, no compartida por ejemplo en [8].

- 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

### 5.3.2. Objetivos específicos

Los objetivos establecidos que el alumnado debe alcanzar en esta asignatura son:

- 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.
- Conocer, diferenciar y dominar las diferentes técnicas para llevar a cabo el análisis léxico, sintáctico y semántico de lenguajes.
- Dominar el manejo de herramientas para la construcción de analizadores léxicos, sintácticos y semánticos.

## 5.4. Programa detallado

A continuación se presenta la propuesta de temas a ser tratados en el marco de la asignatura de *Compiladores I*, así como su planificación temporal. El programa de la asignatura se estructura en 6 temas principales (ver Tabla 5.1):

- 0. Presentación de la asignatura
- 1. Introducción
- 2. Análisis Léxico
- 3. Fundamentos de Análisis Sintáctico
- 4. Análisis Sintáctico Descendente
- 5. Análisis Sintáctico Ascendente
- 6. Introducción al Análisis Semántico

En función de las horas de las que dispone la asignatura de *Compiladores I* (2.5 créditos teóricos, 5 créditos de problemas y 1.5 créditos prácticos), en la Tabla 5.1 se indican las horas que se le va a dedicar a cada uno de los temas. No obstante, hay que aclarar que la distribución en horas planteada se basa en la experiencia previa. Además, también hay señalar que, aunque se indica una duración en horas de cada una de las diferentes lecciones comprendidas en cada tema, se debe ser flexible y adaptarse 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. 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 se corresponden con clases de 2 horas y 50 minutos cada quincena.

Tema	Horas teóricas	Horas problemas	Horas 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

**Tabla 5.1:** Temas de la asignatura de Compiladores I

En el resto de la sección se especifican los distintos temas en más detalle, descomponiéndolos en lecciones.

#### 5.4.1. Tema 0: Presentación de la asignatura

Este tema se imparte en una única lección que se corresponde a la primera clase en el aula de teoría del curso.

##### Lección 0: Presentación

##### Objetivos y descripción general

- Presentar la asignatura al alumnado y el modelo docente elegido para su impartición.

### Contenido

El contenido de esta lección abarca la presentación de forma breve de los siguientes aspectos:

- La asignatura: dónde está situada y por qué, objetivos y requisitos previos.
- El profesorado: Horarios de tutorías y las formas de contacto.
- Temario (Contenidos).
- Planificación.
- Las diferentes clases: teoría, problemas y prácticas (método docente).
- Presentación de las prácticas.
- La evaluación: Criterios.

En esta clase también se le proporciona al alumnado la ficha de la asignatura (ver Anexo I), el plan de la asignatura (ver Anexo II), y un formulario que deberá devolver al profesorado con sus datos cubiertos. El objetivo de la ficha es poder realizar un seguimiento del alumnado.

El profesorado también repartirá una bloque de trabajo con ejercicios y problemas que debería saber resolver (conocimientos previos requeridos). Se insiste en que si tienen dificultades para comprender o resolver alguno de ellos asistan a tutorías para evaluar cuál es el problema y tratar de buscarle una solución.

La duración estimada de esta lección se de 30 minutos. El tiempo restante hasta completar los 50 minutos se empleará en resolver las dudas que tenga el alumnado.

### 5.4.2. Tema 1: Introducción

Este tema se imparte en una única lección que se corresponde a la segunda clase en el aula de teoría del curso. La duración estimada de esta lección es de 50 minutos.

#### Lección 1: introducción

##### Objetivos

- Motivar al alumnado en el aprendizaje de técnicas y métodos de construcción de compiladores y traductores.
- Identificar los elementos constitutivos de un compilador: conocerlos, diferenciarlos y comprender su función.

### Contenido

El contenido de esta lección abarca la presentación, de forma breve, de los siguientes aspectos:

- Introducción
- Las partes de un compilador
  - El analizador léxico
  - El analizador sintáctico
  - El analizador semántico
  - El optimizador
  - El generador de código
  - La tabla de símbolos
- ¿Y los intérpretes?

### Referencias

- Capítulo 1 secciones 1.1-1.5 de "Compiladores: principios, técnicas y herramientas". A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 1 secciones 1.1-1.5 de "Construcción de compiladores. Principios y práctica". K.C. Louden Thomson, 2004.
- Capítulo 1 sección 1.13 y capítulo 8 de "Construcción e intérpretes: teoría y práctica". M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006

#### 5.4.3. Tema 2: Análisis léxico

Los objetivos globales de este tema son:

- Motivar al alumnado en el aprendizaje de técnicas y métodos de construcción analizadores léxicos.
- Conocer las distintas técnicas que intervienen en la construcción de un analizador léxico.
- Utilizar el vocabulario de la forma apropiada.
- Determinar los elementos fundamentales de un lenguaje de programación.
- Dominar el manejo de herramientas para la construcción de analizadores léxicos.

Este tema se imparte en varias lecciones correspondientes a: 3 clases en el aula de tipo teórico (50'), 1 clases en el aula de problemas (50') y 2 clase en el laboratorio prácticas ( 2h 50'). El orden en que se imparten las diferentes lecciones se muestra a continuación.

**Lección 2: introducción al análisis léxico****Objetivos**

- Identificar la funcionalidad de un Analizador Léxico y los elementos con los que está interrelacionado.
- Conocer el vocabulario empleado en los diferentes métodos y técnicas de construcción de analizadores léxicos.
- Repasar términos y conceptos de expresiones regulares y autómatas.

**Contenido**

El contenido de esta lección abarca los siguientes aspectos:

- El papel de un analizador léxico (Scanner).
- Definición de conceptos: Tokens, lexemas y patrones léxicos.
- Expresiones regulares:
  - Su utilidad en compilación.
  - Definición.
  - Ejemplos.
  - Notaciones
- Autómatas Finitos:
  - Generalidades.
  - Grafo de transición asociado a un A.F.

**Referencias**

- "Teoría de autómatas y lenguajes formales". Dean Kelley. Prentice Hall, 1995.
- Capítulo 3 de "Compiladores: principios, técnicas y herramientas". A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 2 "Construcción de compiladores. Principios y práctica". K.C. Loudon Thomson, 2004.
- Capítulo 3 de "Construcción e intérpretes: teoría y práctica". M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006

**Lección 3: autómatas y analizadores léxicos****Objetivos**

- Repasar términos y conceptos de expresiones regulares y autómatas.

### Contenido

El contenido de esta lección abarca los siguientes aspectos:

- Autómatas Finitos:
  - Aceptación de un string por un A.F.
  - Generalidades de los autómatas finitos no deterministas (A.F.N.).
  - Diferencias entre autómatas finitos deterministas (A.F.D) y no deterministas (A.F.N.).
  - Aceptación de un string por un A.F.N.
- Conversion de una expresión regular en un A.F.N.
- Conversion de un A.F.N. en un A.F.D.
- Minimización de un A.F.D.
- Presentación de la práctica 1.

### Referencias

- "Teoría de autómatas y lenguajes formales". Dean Kelley. Prentice Hall, 1995.
- Capítulo 3 de "Compiladores: principios, técnicas y herramientas". A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 2 "Construcción de compiladores. Principios y práctica". K.C. Louden Thomson, 2004.
- Capítulo 3 de "Construcción e intérpretes: teoría y práctica". M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006

### Lección 1 de laboratorio: definición de un lenguaje de programación básico (Laboratorio)

#### Objetivos

- Determinar los elementos fundamentales de un lenguaje de programación.
- Definir un lenguaje propio de programación.
- Elaborar un banco de pruebas en el lenguaje de programación propio.

#### Referencias

- Manuales de diferentes lenguajes de programación.

**Lección 1 de problemas: analizadores léxicos****Objetivos**

- Repasar términos, conceptos, y su aplicación relacionados con el análisis léxico.

**Contenido**

El contenido de esta lección abarca los siguientes aspectos:

- Repaso de la teoría tratada.
- Realización de ejercicios y problemas:
  - Expresiones regulares.
  - Autómatas finitos.

**Referencias**

- "Teoría de autómatas y lenguajes formales". Dean Kelley. Prentice Hall, 1995.
- Capítulo 3 de "Compiladores: principios, técnicas y herramientas". A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 2 "Construcción de compiladores. Principios y práctica". K.C. Loudon Thomson, 2004.
- Capítulo 3 de "Construcción e intérpretes: teoría y práctica". M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006

**Lección 4: construcción de analizadores léxicos****Objetivos**

- Conocer las técnicas para la implementación de analizadores léxicos.
- Comprender las técnicas de tratamiento de errores en análisis léxico.
- Dominar el manejo de herramientas de construcción de analizadores léxicos.
- Identificar los errores de compilación que se pueden considerar errores léxicos.

**Contenido**

El contenido de esta lección abarca los siguientes aspectos:

- Implementación de analizadores léxicos.
- Recuperación de errores léxicos.
- Una herramienta de generación de analizadores léxicos: flex.
- Presentación de la práctica 2.



**Referencias**

- "Lex & Yacc". J. Levine, T. Mason, D. Brown. O'Reilly and Associates, 1992.
- Capítulo 2 y 3 de "Compiladores: principios, técnicas y herramientas". A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 2 "Construcción de compiladores. Principios y práctica". K.C. Louden Thomson, 2004.

**Lección 2 de laboratorio: construcción de un analizador léxico para "mi-Lenguaje"****Objetivos**

- Identificar y establecer los tokens fundamentales para un lenguaje de programación sencillo.
- Diseñar un analizador léxico para dicho lenguaje.
- Manejar la herramienta Flex, un generador de analizadores léxicos.

**Referencias**

- "Lex & Yacc". J. Levine, T. Mason, D. Brown. O'Reilly and Associates, 1992.

**5.4.4. Tema 3: Fundamentos de análisis sintáctico**

Los objetivos globales de este tema son:

- Motivar al alumnado en el aprendizaje de técnicas y métodos de construcción analizadores sintácticos.
- Conocer las distintas técnicas que intervienen en la construcción de un analizador sintáctico.
- Distinguir el modo de funcionamiento de analizadores sintácticos descendentes y ascendentes.
- Utilizar el vocabulario técnico de la forma apropiada.
- Dominar el manejo de herramientas de construcción de analizadores sintácticos.

Este tema se imparte en varias lecciones correspondientes a: 4 clases en el aula de tipo teórico (50') y 1 clase en el laboratorio prácticas (2h 50').

**Lección 5: introducción al análisis sintáctico****Objetivos**

- Motivar al alumnado en el aprendizaje de técnicas y métodos de construcción analizadores sintácticos.
- Conocer las distintas técnicas que intervienen en la construcción de un analizador sintáctico.
- Utilizar el vocabulario técnico de la forma apropiada.
- Repasar términos y conceptos relacionados con las gramáticas

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- El papel de un analizador Sintáctico (*Parser*).
- Gramáticas:
  - Definición.
  - Notación.
  - Clasificación.

**Referencias**

- “Teoría de autómatas y lenguajes formales”. Dean Kelley. Prentice Hall, 1995.

**Lección 6: gramáticas independientes/libres de contexto (G.I.C.)****Objetivos**

- Utilizar el vocabulario técnico de la forma apropiada.
- Repasar términos y conceptos relacionados con las gramáticas independientes de contexto.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Gramáticas Independientes de contexto:
  - Notación.
  - Derivaciones por la izquierda.
  - Derivaciones por la derecha.
  - Árboles de análisis sintáctico.

**Referencias**

- “Teoría de autómatas y lenguajes formales”. Dean Kelley. Prentice Hall, 1995.
- Capítulo 4 de ”Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 3 ”Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.

**Lección 7: la ambigüedad en gramáticas libres/independientes de contexto****Objetivos**

- Utilizar el vocabulario técnico de la forma apropiada.
- Repasar términos y conceptos relacionados con las gramáticas independientes de contexto
- Comprender cuál es la razón de que una gramática sea ambigua

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Ambigüedad en las gramáticas libres de contexto.
- Algoritmos de transformación de gramáticas libres de contexto:
  - Eliminación de símbolos inútiles.
  - Eliminación de reglas epsilon.
  - Eliminación de reglas unitarias.

**Referencias**

- “Teoría de autómatas y lenguajes formales”. Dean Kelley. Prentice Hall, 1995.
- Capítulo 4 de ”Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.

**5.4.5. Tema 4: Análisis sintáctico descendente**

Los objetivos globales de este tema son:

- Conocer las distintas técnicas que intervienen en la construcción de un analizador sintáctico descendente.
- Utilizar el vocabulario de la forma apropiada.

- Dominar el manejo de herramientas para la construcción de analizadores sintácticos descendentes recursivos y por tabla.

Este tema se imparte en varias lecciones correspondientes a: 5 clases en el aula de tipo teórico (50'), 1 clases en el aula de problemas (50') y 1 clase en el laboratorio prácticas (2h 50').

### Lección 8: introducción al análisis sintáctico descendente

#### Objetivos

- Motivar al alumnado en el aprendizaje de técnicas y métodos de construcción de analizadores sintácticos descendentes.
- Utilizar el vocabulario técnico de la forma apropiada.
- Diferenciar entre análisis sintáctica descendente y ascendente.
- Comprender que objetivo persigue la factorización a izquierda de las gramáticas.
- Dominar las técnicas para la construcción de analizadores sintácticos descendentes recursivos.
- Comprender qué objetivo persigue la eliminación de la recursividad a izquierda.

#### Contenido

El contenido de esta lección abarca la presentación de forma breve de los siguientes aspectos:

- Estrategias para el análisis sintáctico.
- Análisis sintáctico descendente.
- El *backtracking* en analizadores sintácticos descendente.
- La factorización a izquierda de gramáticas.
- Construcción de analizadores sintácticos descendentes recursivos.
- Los analizadores sintácticos descendentes y la recursividad a izquierda.
- Técnicas para eliminar la recursividad a izquierda directa en las gramáticas.
- Técnicas para eliminar la recursividad a izquierda indirecta en las gramáticas.

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 4 “Construcción de compiladores. Principios y práctica”. K.C. Loudon Thomson, 2004.
- Capítulo 4 de “Construcción e intérpretes: teoría y práctica”. M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006

**Lección 9: el análisis sintáctico descendente iterativo frente al recursivo****Objetivos**

- Utilizar el vocabulario técnico de la forma apropiada.
- Comprender la diferencia entre analizadores sintácticos descendente predictivos y no predictivos.
- Comprender el funcionamiento de los analizadores sintácticos descendente predictivos iterativos.

**Contenido**

El contenido de esta lección abarca la presentación de forma breve de los siguientes aspectos:

- Analizadores sintácticos descendentes predictivos y no predictivos.
- Construcción de analizadores Sintácticos predictivos no recursivos. Analizadores sintácticos LL(1):
  - Componentes de un analizador LL(1).
  - Análisis de cadenas en LL(1).
  - Construcción de la tabla de análisis LL(1) de forma intuitiva.

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 4 de “Construcción de compiladores. Principios y práctica”. K.C. Loudon Thomson, 2004.

**Lección 10: construcción de analizadores sintácticos iterativos LL(1)****Objetivos**

- Utilizar el vocabulario técnico de la forma apropiada.
- Comprender la diferencia entre analizadores sintácticos descendente predictivos y no predictivos.
- Dominar las técnicas para la construcción de analizadores sintácticos descendentes iterativos.

**Contenido**

El contenido de esta lección abarca la presentación de forma breve de los siguientes aspectos:

- Construcción de la tabla de analizadores sintácticos LL(1):
  - El concepto de primero de una forma de frase.
  - El concepto de siguiente de una forma de frase.
  - Construcción de la tabla de análisis LL(1) mediante técnicas formales.

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 4 de “Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.

**Lección 2 de problemas: analizadores sintácticos descendentes****Objetivos**

- Repasar conceptos y técnicas relacionados con el análisis sintáctico descendente.

**Contenido**

El contenido de esta lección abarca los siguientes aspectos:

- Repaso de la teoría tratada.
- Realización de ejercicios y problemas:
  - Análisis sintáctico descendente recursivo.
  - Técnicas de factorización a izquierda.
  - Técnicas de eliminación de la recursividad a izquierda.
  - Análisis sintáctico descendente iterativo LL(1).

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 4 “Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.
- Capítulo 4 de “Construcción e intérpretes: teoría y práctica”. M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006

**Lección 11: tratamiento de errores en análisis sintáctico descendente y gramáticas de expresiones****Objetivos**

- Utilizar el vocabulario técnico de la forma apropiada.
- Comprender la diferencia entre errores léxicos y errores sintácticos.
- Dominar las técnicas para la recuperación de errores en analizadores sintácticos descendentes.
- Analizar las características de gramáticas de expresiones.

**Contenido**

El contenido de esta lección abarca la presentación de forma breve de los siguientes aspectos:

- Técnicas de recuperación de errores en análisis sintáctico descendente.
- Gramáticas de expresiones:
  - La gramática de expresiones de Pascal.
  - La gramática de expresiones de Ada.
- Presentación de la práctica 3.

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 4 “Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.

**Lección 3 de laboratorio: construcción de un analizador sintáctica descendente para “miLenguaje”****Objetivos**

- Desarrollar un analizador sintáctico descendente para “miLenguaje”.
- Integrar un analizador léxico con un analizador sintáctico.
- Emplear la utilidad “make” de *Unix*.

**Referencias**

- “Lex & Yacc”. J.Levine, T. Mason, D. Brown. O’Reilly and Associates, 1992.

**5.4.6. Tema 5: Análisis sintáctico ascendente**

Los objetivos globales de este tema son:

- Conocer las distintas técnicas que intervienen en la construcción de un analizador sintáctico ascendente.
- Utilizar el vocabulario de la forma apropiada.
- Dominar el manejo de herramientas para la construcción de analizadores sintácticos ascendentes LR.

Este tema se imparte en varias lecciones correspondientes a: 7 clases en el aula de tipo teórico (50’), 1.5 clases en el aula de problemas (50’) y 1 clase en el laboratorio prácticas ( 2h 50’).

**Lección 12: introducción al análisis sintáctico ascendente****Objetivos**

- Motivar al alumnado en el aprendizaje de técnicas y métodos de construcción de analizadores sintácticos ascendentes.
- Identificar los elementos constitutivos de un analizador sintáctico ascendente: conocerlos, diferenciarlos y comprender su función.
- Identificar las diferencias y similitudes de funcionamiento entre analizadores sintácticos descendentes y analizadores sintácticos ascendentes.
- Conocer los diferentes tipos de analizadores ascendentes.



**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Introducción al análisis sintáctico ascendente.
- Tipos de análisis sintáctico ascendente.
- Un ejemplo *intuitivo*.
- Diferencias de funcionamiento entre un analizador sintáctico descendente y un analizador sintáctico ascendente.
- Definiciones: Operaciones de desplazamiento y reducción.

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 5 “Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.
- Capítulo 4 de “Construcción e intérpretes: teoría y práctica”. M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006

**Lección 13: el análisis sintáctico ascendente SLR****Objetivos**

- Identificar los elementos constitutivos de un analizador sintáctico ascendente SLR: conocerlos, diferenciarlos y comprender su función.
- Comprender los mecanismos de análisis de cadenas de los analizadores sintácticos ascendentes.
- Dominar las técnicas para la construcción del autómata del analizador sintáctico SLR.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Análisis de cadenas mediante SLR.
- Definición de elemento de configuración LR(0).
- Construcción del autómata del analizador sintáctico SLR.

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 5 ”Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.

**Lección 14: conflictos en el análisis sintáctico ascendente SLR****Objetivos**

- Identificar los diferentes tipos de conflictos que se pueden producir en el análisis sintáctico ascendente.
- Comprender las causas que provocan los distintos tipos de conflicto del análisis sintáctico ascendente.
- Dominar las técnicas para la construcción de las tablas del analizador sintáctico SLR.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Análisis de cadenas mediante SLR.
- Definición de elemento de configuración LR(0).
- Construcción de las tablas del analizador sintáctico SLR.
- Conflictos en el análisis sintáctico ascendente: Tipos y causas.
- Políticas de resolución de conflictos.

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.

**Lección 3 de problemas: el análisis sintáctico ascendente SLR****Objetivos**

- Repasar conceptos y técnicas relacionados con el análisis sintáctico ascendente SLR.

**Contenido**

El contenido de esta lección abarca los siguientes aspectos:

- Repaso de la teoría tratada
- Realización de ejercicios y problemas:
  - Construcción de autómatas SLR.
  - Construcción de tablas SLR.
  - Análisis de cadenas en analizadores sintácticos con conflictos.

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 5 “Construcción de compiladores. Principios y práctica”. K.C. Loudon Thomson, 2004.
- Capítulo 4 de “Construcción e intérpretes: teoría y práctica”. M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006

**Lección 15: el análisis sintáctico ascendente LR canónico (LR(1))****Objetivos**

- Diferenciar entre análisis sintáctico ascendente SLR y LR canónico (o LR(1)).
- Dominar las técnicas para la construcción del autómata y las tablas del analizador sintáctico LR canónico.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Definición de elemento de configuración LR(1).
- Construcción del autómata del analizador sintáctico LR canónico (LR(1)).
- Construcción de las tablas del analizador sintáctico LR canónico (LR(1)).
- Diferencias entre analizadores ascendentes SLR y LR canónicos o LR(1).

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 5 “Construcción de compiladores. Principios y práctica”. K.C. Loudon Thomson, 2004.
- Capítulo 4 de “Construcción e intérpretes: teoría y práctica”. M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006.

**Lección 16: el análisis sintáctico ascendente LALR**

La mitad de esta clase se dedicará a la realización de ejercicios o problemas.

**Objetivos**

- Diferenciar entre análisis sintáctico ascendente SLR, LR canónico (o LR(1) y LALR).
- Dominar las técnicas para la construcción del autómata del analizador sintáctico LALR.
- Repasar conceptos y técnicas relacionados con el análisis sintáctico ascendente.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Construcción del autómata del analizador sintáctico LALR.
- Repaso de la teoría tratada.
- Realización de ejercicios y problemas:
  - Construcción de autómatas LR.
  - Construcción de tablas LR.
  - Construcción de autómatas LALR.

**Referencias**

- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 5 “Construcción de compiladores. Principios y práctica”. K.C. Loudon Thomson, 2004.

**Lección 17: introducción a Bison****Objetivos**

- Dominar el manejo de herramientas de construcción de analizadores sintácticos.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Una herramienta de generación de analizadores sintácticos: Bison.
- Presentar la primera parte de la práctica 4.

**Referencias**

- “Lex & Yacc”. J. Levine, T. Mason, D. Brown. O’Reilly and Associates, 1992.
- Capítulo 4 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 3 “Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.
- Capítulo 4 de “Construcción e intérpretes: teoría y práctica”. M. Alfonseca, M. de la Cruz, A. Ortega, E. Pulido. Pearson, 2006.

**Lección 4 de laboratorio: construcción de un analizador sintáctico ascendente para “miLenguaje”****Objetivos**

- Diseñar y desarrollar un analizador sintáctico ascendente para “miLenguaje”.
- Manejar la herramienta Bison, un generador de analizadores sintácticos LALR.

**Referencias**

- “Lex & Yacc”. J. Levine, T. Mason, D. Brown. O’Reilly and Associates, 1992.

**5.4.7. Tema 6: Análisis semántico**

Los objetivos globales de este tema son:

- Conocer las distintas técnicas que intervienen en la construcción de un analizador sintáctico semántico.
- Utilizar el vocabulario de la forma apropiada.
- Diferenciar entre los distintos tipos de analizadores semánticos distinguiendo sus ventajas e inconvenientes.

- Identificar los distintos tipos atributos de gramáticas atribuidas.
- Dominar el manejo de herramientas para la construcción de analizadores sintácticos y semánticos al vuelo.

Este tema se imparte en varias lecciones correspondientes a: 5 clases en el aula de tipo teórico (50'), 1.5 clases en el aula de problemas (50') y 1 clase en el laboratorio prácticas (2h 50').

### **Lección 18: introducción al análisis semántico**

#### **Objetivos**

- Identificar la funcionalidad de un analizador semántico y los elementos con los que está interrelacionado.
- Utilizar el vocabulario técnico de la forma apropiada.
- Comprender la diferencia entre errores léxicos, errores sintácticos y errores semánticos.
- Definir gramáticas atribuidas.
- Diferenciar entre atributos heredados y atributos sintetizados.

#### **Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Introducción al análisis semántico.
- Definición de gramáticas atribuidas.
- Definición de árbol de sintaxis decorado.
- Tipos de atributos: sintetizados y heredados.
- Tipos de evaluación de atributos:
  - Por recorrido del árbol de sintaxis.
  - Al vuelo.

#### **Referencias**

- Capítulo 5 de "Compiladores: principios, técnicas y herramientas". A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 6 "Construcción de compiladores. Principios y práctica". K.C. Loudon Thomson, 2004.

**Lección 19: la evaluación de atributos****Objetivos**

- Diferenciar entre evaluación de atributos por recorrido de árbol de sintaxis y la evaluación de atributos al vuelo.
- Identificar y valorar la características de los distintos tipos de métodos para realizar la evaluación de atributos.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- La evaluación de atributos por recorrido del árbol de sintaxis.
- La evaluación de atributos al vuelo.
- Definición de gramáticas L-atribuidas.

**Referencias**

- Capítulo 5 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 6 ”Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.

**Lección 20: la evaluación de atributos en Bison****Objetivos**

- Dominar el manejo de herramientas para la construcción de analizadores semánticos al vuelo.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- Tratamiento de atributos en Bison.
- La evaluación ascendente de atributos sintetizados.
- Presentación de la segunda parte de la práctica 4.

**Referencias**

- “Lex & Yacc”. J.Levine, T. Mason, D. Brown. O’Reilly and Associates, 1992.

**Lección 21: la evaluación de atributos heredados (I)****Objetivos**

- Dominar el manejo de herramientas para la construcción de analizadores semánticos al vuelo.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- La evaluación ascendente de atributos heredados de hermanos izquierdos.

**Referencias**

- Capítulo 5 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 6 “Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.

**Lección 5 de laboratorio: construcción de un traductor al vuelo para “miLenguaje”****Objetivos**

- Diseñar y desarrollar un traductor dirigido por la sintaxis basado en LALR para “miLenguaje”.
- Manejo de atributos semánticos en la herramienta Bison.

**Referencias**

- “Lex & Yacc”. J.Levine, T. Mason, D. Brown. O’Reilly and Associates, 1992.

**Lección 22: la evaluación de atributos heredados (II)****Objetivos**

- Dominar el manejo de herramientas para la construcción de analizadores semánticos al vuelo.

**Contenido**

El contenido de esta lección abarca la presentación de los siguientes aspectos:

- La evaluación ascendente de atributos heredados de hermanos derechos



**Referencias**

- Capítulo 5 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.

**Lección 4 de problemas: el análisis semántico en analizadores ascendentes**

La mitad de esta clase se dedicará a la realización de problemas

**Objetivos**

- Dominar el manejo de herramientas para la construcción de analizadores semánticos al vuelo.
- Repasar conceptos y técnicas relacionados con el análisis semántico.

**Contenido**

El contenido de esta lección abarca los siguientes aspectos:

- La evaluación ascendente de atributos heredados de hermanos derechos y de hermanos izquierdos a la vez.
- Repaso de la teoría tratada.
- Realización de ejercicios y problemas.

**Referencias**

- Capítulo 5 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 6 “Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.

**Lección 5 de problemas: el análisis semántico en analizadores ascendentes****Objetivos**

- Repasar conceptos y técnicas relacionados con el análisis semántico.
- Resolución de dudas de cara al examen final de teoría.

**Contenido**

El contenido de esta lección abarca los siguientes aspectos:

- Repaso de la teoría tratada.
- Realización de ejercicios y problemas.

**Referencias**

- Capítulo 5 de “Compiladores: principios, técnicas y herramientas”. A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1990.
- Capítulo 6 “Construcción de compiladores. Principios y práctica”. K.C. Louden Thomson, 2004.

**5.4.8. Tema 7: Repaso**

Los objetivos globales de este tema son:

- Repasar el contenido de la asignatura y resolver dudas del alumnado.

Este tema se imparte en varias lecciones correspondientes a: 2 clases en el aula de tipo teórico (50'). La experiencia demuestra que debido a festivos y pérdidas de clase por otros motivos (cierre del centro, huelgas, etc.) estas dos lecciones no se suelen impartir.

**5.5. Estimación de la carga de trabajo del alumnado**

En esta sección, se pretende hacer una estimación, a priori, del tiempo dedicado por el alumnado a la asignatura. Esta estimación será contrastada, en el próximo curso, con datos recogidos del alumnado; para ello, la candidata a la plaza está elaborando guías para la recogida de datos (en el Anexo III se puede ver como ejemplo la Guía de Estudio del Tema 2). A continuación, mostramos nuestras estimaciones desglosadas por temas y lecciones. A modo de resumen, se totalizan luego las horas estimadas en la Sección 5.5.9.

**5.5.1. Tema 0: Presentación de la asignatura**

La duración estimada de dedicación del tiempo del alumnado a este tema es de 1 hora y cuarto.

**Lección 0: Presentación**

Se pretende que el alumnado, una vez haya asistido a la presentación de la asignatura (50 minutos a lo sumo), acuda a la biblioteca del campus para localizar los libros presentes en la bibliografía recomendada (se estima que esta tarea debería suponer unos 25 minutos).

**5.5.2. Tema 1: Introducción**

La duración estimada de dedicación del tiempo del alumnado a este tema es de 2 horas.

**Lección 1: introducción**

Se pretende que el alumnado, una vez haya asistido a la lección de introducción a los compiladores (50 minutos), dedique 1 hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección. Una vez haya hecho esto, se le aconseja que elabore un mapa conceptual de la lección en unos 10 minutos y lo entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado.

**5.5.3. Tema 2: Análisis léxico**

La duración estimada de dedicación del tiempo del alumnado a este tema es de 18 horas, considerando que 4 horas y media se dedicarán a teoría, 2 h y media se dedicarán a realización de ejercicios, y 11 horas a la realización de prácticas de laboratorio.

**Lección 2: introducción al análisis léxico**

Se pretende que el alumnado, una vez haya asistido a esta lección de introducción al análisis léxico (50 minutos) dedique 1 hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

**Lección 3: autómatas y analizadores léxicos**

Es recomendable que el alumnado haya dedicado media hora, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que el alumnado, una vez haya asistido a la lección (50 minutos), dedique 1 hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección. Una vez haya hecho esto, se le aconseja que elabore un mapa conceptual del tema de análisis léxico completo en unos 30 minutos y lo entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado.

**Lección 1 de laboratorio: definición de un lenguaje de programación básico (Laboratorio)**

Antes del inicio de esta sesión de laboratorio el alumnado debe haber dedicado 1 hora a recoger material sobre las características de los lenguajes de programación, puesto que tiene que definir su propio lenguaje. Durante la sesión de prácticas, se pretende que dedique 1 h y media a la definición de su propio lenguaje. Una vez hecho esto discutirá su definición del lenguaje con el profesorado durante unos 15 minutos, y posteriormente redactará el documento a entregar con la definición de su lenguaje de programación propio (1 hora). Al final de la clase entregará, a través del sistema establecido, la práctica (15 minutos).

**Lección 1 de problemas: analizadores léxicos**

Antes del inicio de esta lección de problemas el alumnado debe haber dedicado media hora a repasar conceptos de la teoría del tema de análisis léxico. Durante la lección se realizarán y se corregirán diversos ejercicios de exámenes de cursos anteriores (50 minutos). Se pretende que el alumnado, una vez haya asistido a la lección de problemas de análisis léxico, realice una ficha de ejercicios y la entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado. Se estima que la realización de la ficha de ejercicios y su entrega supondrá una carga de 1 hora y 10 minutos.

**Lección 4: construcción de analizadores léxicos**

Es recomendable que el alumnado haya dedicado media hora antes de comenzar esta sesión a repasar las lecciones anteriores de este tema. Además, se pretende que el alumnado una vez haya asistido a la lección (50 minutos), dedique una hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección. Una vez haya hecho esto, se le aconseja que elabore un mapa conceptual del tema de análisis léxico y que lo entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado (30 minutos).

**Lección 2 de laboratorio: construcción de un analizador léxico para "mi-Lenguaje"**

Antes del inicio de esta sesión de laboratorio, el alumnado debe haber dedicado 1 hora y media a repasar conceptos acerca de flex y análisis léxico. Durante la sesión de prácticas se trabajará en la implementación del analizador léxico y el profesorado hará una serie de preguntas a cada uno de los grupos (20 minutos por grupo). La finalización de la implementación y la entrega se hará una vez haya finalizado la sesión de prácticas, y se estima que supondrá alrededor de 1 hora y media.

**5.5.4. Tema 3: Fundamentos de análisis sintáctico**

La duración estimada de dedicación del tiempo del alumnado a este tema es de 5 horas y 45 minutos, considerando que todo este tiempo se dedicará a teoría.

**Lección 5: introducción al análisis sintáctico**

Se pretende que el alumnado, una vez haya asistido a esta lección de introducción al análisis sintáctico (50 minutos), dedique media hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

**Lección 6: gramáticas independientes/libres de contexto (G.I.C.)**

Es recomendable que el alumnado haya dedicado media hora, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que

el alumnado, una vez haya asistido a la lección (50 minutos) dedique media hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

### **Lección 7: la ambigüedad en gramáticas libres/independientes de contexto**

Es recomendable que el alumnado haya dedicado media hora antes de comenzar esta sesión a repasar las lecciones anteriores de este tema. Además, se pretende que el alumnado, una vez haya asistido a la lección (50 minutos), dedique media hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección. Una vez haya hecho esto, se le aconseja que elabore un mapa conceptual del tema de fundamentos de análisis sintáctico completo, en unos 45 minutos, y lo entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado.

#### **5.5.5. Tema 4: Análisis sintáctico descendente**

La duración estimada de dedicación del tiempo del alumnado a este tema es de 18 horas, considerando que 7 horas se dedicarán a teoría, 2 horas y media se dedicarán a la realización de ejercicios, y 6 horas y media a la realización de prácticas de laboratorio.

### **Lección 8: introducción al análisis sintáctico descendente**

Se pretende que el alumnado, una vez haya asistido a esta lección de introducción al análisis sintáctico descendente (50 minutos), dedique media hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

### **Lección 9: el análisis sintáctico descendente iterativo frente al recursivo**

Es recomendable que el alumnado haya dedicado 15 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que el alumnado, una vez haya asistido a la lección (50 minutos), dedique media hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

### **Lección 10: construcción de analizadores sintácticos iterativos LL(1)**

Es recomendable que el alumnado haya dedicado 15 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que el alumnado, una vez haya asistido a la lección (50 minutos), dedique media hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

**Lección 2 de problemas: analizadores sintácticos descendentes**

Antes del inicio de esta lección de problemas, el alumnado debe haber dedicado media hora a repasar conceptos de la teoría del tema de fundamentos de análisis sintáctico y del tema de análisis sintáctico descendente. Durante la lección se realizarán y se corregirán diversos ejercicios de exámenes de cursos anteriores (50 minutos). Se pretende que el alumnado, una vez haya asistido a la lección de problemas de análisis sintáctico descendente, realice una ficha de ejercicios y la entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado. Se estima que la realización de la ficha de ejercicios y su entrega supondrá una carga de 1 hora y 10 minutos.

**Lección 11: tratamiento de errores en análisis sintáctico descendente y gramáticas de expresiones**

Es recomendable que el alumnado haya dedicado media hora, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que el alumnado, una vez haya asistido a la lección (50 minutos), dedique media hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección. Una vez haya hecho esto, se aconseja al alumnado que elabore un mapa conceptual del tema de análisis sintáctico descendente completo, en unos 40 minutos, y lo entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado.

**Lección 3 de laboratorio: construcción de un analizador sintáctico descendente para “miLenguaje”**

Antes del inicio de esta sesión de laboratorio, el alumnado debe haber dedicado 2 horas y media a repasar conceptos acerca de análisis descendente y a elaborar una gramática adecuada para análisis descendente. Durante la sesión de prácticas se trabajará en la implementación del analizador sintáctico y el profesorado hará una serie de preguntas a cada uno de los grupos y también les comentará aspectos de la práctica anterior (20 minutos por grupo). La finalización de la implementación y la entrega se hará una vez haya finalizado la sesión de prácticas, y se estima que supondrá alrededor de 1 hora.

**5.5.6. Tema 5: Análisis sintáctico ascendente**

La duración estimada de dedicación del tiempo del alumnado a este tema es de 25 horas, considerando que 15 horas se dedicarán a teoría, 6 horas se dedicarán a la realización de ejercicios, y 4 horas a la realización de prácticas de laboratorio.

**Lección 12: introducción al análisis sintáctico ascendente**

Es recomendable que el alumnado haya dedicado una hora, antes de comenzar esta sesión, a repasar los temas de fundamentos de análisis sintáctico y análisis sintáctico descendente. Además, se pretende que el alumnado una vez haya asistido a la lección

(50 minutos), dedique otra media hora a completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

### **Lección 13: el análisis sintáctico ascendente SLR**

Es recomendable que el alumnado haya dedicado 30 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que el alumnado, una vez haya asistido a la lección (50 minutos), dedique unos 45 minutos a completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

### **Lección 14: conflictos en el análisis sintáctico ascendente SLR**

Es recomendable que el alumnado haya dedicado 30 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que el alumnado, una vez haya asistido a la lección (50 minutos), dedique unos 45 minutos a completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

### **Lección 3 de problemas: el análisis sintáctico ascendente SLR**

Antes del inicio de esta lección de problemas, el alumnado debe haber dedicado 1 hora a repasar conceptos de la teoría de los temas de fundamentos de análisis sintáctico y análisis sintáctico ascendente. Durante la lección se realizarán y se corregirán diversos ejercicios de exámenes de cursos anteriores (50 minutos). Se pretende que el alumnado, una vez haya asistido a la lección de problemas de análisis sintáctico ascendente, realice una ficha de ejercicios y la entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado. Se estima que la realización de la ficha de ejercicios y su entrega supondrá una carga de 2 hora y 10 minutos.

### **Lección 15: el análisis sintáctico ascendente LR canónico (LR(1))**

Es recomendable que el alumnado haya dedicado 30 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que el alumnado, una vez haya asistido a la lección (50 minutos), dedique 1 hora a completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección. Además, una vez haya asistido a esta lección, el alumnado debe realizar una ficha de ejercicios y entregarla en la siguiente sesión de teoría o en horario de tutorías al profesorado. Se estima que la realización de la ficha de ejercicios y su entrega supondrá una carga de 1 hora.

### **Lección 16: el análisis sintáctico ascendente LALR**

Es recomendable que el alumnado haya dedicado 30 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que

el alumnado, una vez haya asistido a la lección (50 minutos), dedique 1 hora a completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección. Además, una vez haya asistido a esta lección, debe realizar una ficha de ejercicios y entregarla en la siguiente sesión de teoría o en horario de tutorías al profesorado. Se estima que la realización de la ficha de ejercicios y su entrega supondrá una carga de 1 hora.

### **Lección 17: introducción a Bison**

Es recomendable que el alumnado haya dedicado 30 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que, una vez haya asistido a la lección (50 minutos), dedique unos 45 minutos a completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección. Una vez haya hecho esto, se le aconseja que elabore un mapa conceptual del tema de análisis sintáctico ascendente completo en unos 45 minutos y lo entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado.

### **Lección 4 de laboratorio: construcción de un analizador sintáctico ascendente para “miLenguaje”**

Antes del inicio de esta sesión de laboratorio, el alumnado debe haber dedicado 1 hora a repasar el modo de funcionamiento de Bison y su sintaxis. Durante la sesión de prácticas se trabajará en la implementación del analizador sintáctico ascendente adaptando la gramática realizada en la práctica anterior. Además, el profesorado hará una serie de preguntas a cada uno de los grupos y también les comentará aspectos de la práctica anterior (20 minutos por grupo). La entrega se hará una vez haya finalizado la sesión de prácticas.

### **5.5.7. Tema 6: Análisis semántico**

La duración estimada de dedicación del tiempo del alumnado a este tema es de 25 horas, considerando que 10 horas se dedicarán a teoría, 6 horas se dedicarán a la realización de ejercicios, y 9 horas a la realización de prácticas de laboratorio.

### **Lección 18: introducción al análisis semántico**

Se pretende que el alumnado, una vez haya asistido a esta lección de introducción al análisis semántico (50 minutos), dedique media hora para completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

### **Lección 19: la evaluación de atributos**

Es recomendable que el alumnado haya dedicado 30 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que, una vez haya asistido a la lección (50 minutos), dedique media hora a completar los



apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

#### **Lección 20: la evaluación de atributos en Bison**

Es recomendable que el alumnado haya dedicado una hora al repaso de las lecciones anteriores de este tema y al tema de análisis sintáctico ascendente, antes de comenzar esta sesión. Además, se pretende que, una vez haya asistido a la lección (50 minutos), dedique otra media hora a completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección.

#### **Lección 21: la evaluación de atributos heredados (I)**

Es recomendable que el alumnado haya dedicado 30 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que tras la clase (50 minutos) dedique media hora a completar los apuntes que ha tomado con alguno de los libros de la bibliografía recomendada en esta lección.

#### **Lección 5 de laboratorio: construcción de un traductor al vuelo para “mi-Lenguaje”**

Antes del inicio de esta sesión de laboratorio, el alumnado debe haber dedicado 2 horas y media a repasar conceptos acerca de análisis semántico y uso de gramáticas atribuidas en Bison. Durante la sesión de prácticas se trabajará en la implementación de un traductor dirigido por la sintaxis del analizador sintáctico realizado en la práctica anterior. En esta sesión, el profesorado también hará una serie de preguntas a cada uno de los grupos y les comentará aspectos de la práctica anterior (20 minutos por grupo). La finalización de la implementación y la entrega se hará una vez haya finalizado la sesión de prácticas, y se estima que supondrá alrededor de 2 horas y media.

#### **Lección 22: la evaluación de atributos heredados (II)**

Es recomendable que el alumnado haya dedicado 30 minutos, antes de comenzar esta sesión, a repasar las lecciones anteriores de este tema. Además, se pretende que tras la clase (50 minutos) dedique media hora a completar los apuntes que ha tomado en clase con alguno de los libros de la bibliografía recomendada en esta lección. Una vez haya hecho esto, se le aconseja que elabore un mapa conceptual del tema de análisis semántico completo en unos 50 minutos y que lo entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado.

#### **Lección 4 de problemas: el análisis semántico en analizadores ascendentes**

Antes del inicio de esta lección de problemas, el alumnado debe haber dedicado media hora a repasar conceptos de la teoría de los temas de análisis sintáctico ascendente y análisis semántico. Durante la lección se realizarán y se corregirán diversos

ejercicios de exámenes de cursos anteriores (50 minutos). Se pretende que, una vez haya asistido a la lección de problemas de análisis semántico, realice una ficha de ejercicios y la entregue en la siguiente sesión de teoría o en horario de tutorías al profesorado. Se estima que la realización de la ficha de ejercicios y su entrega supondrán una carga de 2 horas y 40 minutos.

### Lección 5 de problemas: el análisis semántico en analizadores ascendentes

Antes del inicio de esta lección de problemas, el alumnado debe haber dedicado media hora a repasar los problemas de análisis semántico de anteriores clases. Durante la lección se realizarán y se corregirán diversos ejercicios de exámenes de cursos anteriores (50 minutos). Se pretende que, una vez haya asistido a la lección de problemas de análisis semántico, los repase y realice algún ejercicio más. Se estima que la carga de trabajo de esto debe ser de 40 minutos.

#### 5.5.8. Tema 7: Repaso

La duración estimada de dedicación del tiempo del alumnado a este tema es de 6 horas, considerando que 3 de esas horas se dedicarán a la realización del examen teórico-práctico final y las demás serán de repaso del curso.

#### 5.5.9. Conclusiones

En la siguiente tabla se muestra un resumen de las horas presenciales y no presenciales estimadas de dedicación del alumnado. Nótese que se hace un ligero redondeo para trabajar con horas en lugar de con minutos.

	Horas presenciales	Horas no presenciales	Total
0. Presentación de la asignatura	1	0	1
1. Introducción	1	1	2
2. Análisis Léxico	10	8	18
3. Fundamentos de Análisis Sintáctico	3	3	6
4. Análisis Sintáctico Descendente	8	9	17
5. Análisis Sintáctico Ascendente	10	15	25
6. Introducción al Análisis Semántico	10	15	25
7. Repaso	3	3	6
TOTAL	46	54	100

**Tabla 5.2:** Carga de trabajo del estudiante

Se estima que el tiempo total en carga de trabajo del alumnado dedicado a la asignatura de *Compiladores I* debería estar alrededor de 100 horas. Esto supondría 4

créditos ECTS.

## 5.6. Problemas detectados en el alumnado

Para concluir el Proyecto Docente, se indican en esta sección algunos problemas detectados que surgen debido a las infraestructuras actuales y a la rica heterogeneidad del alumnado.

### 5.6.1. Relacionados con la movilidad del alumnado

El alumnado procedente de otros centros puede no tener ciertos conocimientos base que se consideran necesarios para que puedan cursar la asignatura, como sucede con el alumnado procedente de Teruel, pues en su plan de estudios no está incluida ninguna materia donde se traten. Debido a esto, desde el centro se les exige que se matriculen en la asignatura *Lenguajes, gramáticas y autómatas*.

Algunos estudiantes Erasmus tienen ciertos problemas con el lenguaje empleado, debido a que las clases se imparten en castellano. Para facilitarles el estudio, se ha optado por proporcionarles en el siguiente curso la documentación de la asignatura en inglés, y por permitirles realizar las diferentes pruebas también en este idioma.

Este curso se ha planteado un nuevo problema, ya que existían alumnos matriculados en la asignatura que se encontraban de estancia en centros extranjeros. Como se espera que esta sea una práctica cada vez más habitual, se están estudiando diferentes posibilidades para que puedan cursar la asignatura (plan de tutorías personalizadas cuando visiten Zaragoza, defensas de prácticas a través de sistemas de videoconferencia, etc.).

### 5.6.2. Relacionados con preconcepciones erróneas de conceptos

Con respecto a los conocimientos adquiridos por parte de los estudiantes, se identifican algunas preconcepciones erróneas: “que la factorización a la izquierda y la eliminación de la recursión a la izquierda son suficientes para transformar cualquier gramática en una gramática LL(1)”, “que programar es un arte más que una ciencia, y que el código de un programa sólo necesita entenderlo el que lo escribe”, no está clara la diferencia entre *tiempo de compilación* y *tiempo de ejecución* (gracias en parte a los entornos de desarrollo rápido, que ofrecen una visión unificada), que “el análisis sintáctico es simplemente una comprobación de consistencia previa”, y a menudo se confunde “la ambigüedad de un lenguaje con los conflictos del análisis sintáctico”. Se va a tratar de replantear la forma de realizar la explicación, el tiempo dedicado a trabajar esos conceptos y los métodos y medios utilizados, con objeto de reducir estas deficiencias de comprensión.

## 5.7. Bibliografía Recomendada

- “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. Louden. Thomson, 2004 [Loud04]
- “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.

# Referencias

- [1] Alexander Aiken. Cool: A portable project for teaching compiler construction. *SIG-PLAN Notices*, 31(7):19–24, 1996.
- [2] Andrew W. Appel. *Modern Compiler Implementation in C*. Cambridge University Press, 1998.
- [3] Andrew W. Appel. *Modern Compiler Implementation in Java*. Cambridge University Press, 1998.
- [4] Andrew W. Appel. *Modern Compiler Implementation in ML*. Cambridge University Press, 1998.
- [5] Profesorado de la asignatura. *Página oficial de la asignatura de Compiladores de la Universidad Politécnica de Catalunya*. 2006. <http://www.lsi.upc.es/ggodoy/cl.html>.
- [6] Profesorado de la asignatura. *Página oficial de la asignatura de Compiladores de la Universidad Politécnica de Madrid*. 2006. <http://www-lt.ls.fi.upm.es/compiladores/index.html>.
- [7] Saumya K. Debray. Making compiler design relevant for students who will (most likely) never design a compiler. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education, 2002, Cincinnati, Kentucky, USA, February 27 - March 3, 2002*, pages 341–345, 2002.
- [8] Akim Demaille. Making compiler construction projects relevant to core curriculums. In *Proceedings of the Tenth Annual Conference on Innovation and Technology in Computer Science Education (ITICSE'05)*, pages 266–270, Universidade Nova de Lisboa, Monte da Pacarita, Portugal, June 2005.
- [9] M.L. Forcada F. Moreno-Seco. Learning compiler design as a research activity. *Computer Science Education*, 7(1):73–98, 1996.
- [10] Jeremy D. Frens and Andrew Meneely. Fifteen compilers in fifteen days. In *37th SIGCSE technical symposium on Computer science education (SIGCSE'06)*, pages 92–96, New York, NY, USA, 2006. ACM Press.
- [11] Allen B. Tucker. Computing curricula 1991. *Communications of the ACM*, 34(6):68–84, 1991.
- [12] Li Xu and Fred G. Martin. Chirp on crickets: teaching compilers using an embedded robot controller. In *37th SIGCSE technical symposium on Computer science education (SIGCSE'06)*, pages 82–86, New York, NY, USA, 2006. ACM Press.











