

Guía Docente

Modalidad A Distancia

Compiladores

Curso 2017/18

Grado en Ingeniería de
Sistemas de Información



UCAV

www.ucavila.es



Nombre:	COMPILADORES
Carácter:	OPTATIVA
Código:	40303GH
Curso:	4º
Duración (Semestral/Anual):	SEMESTRAL
Nº Créditos ECTS:	6
Prerrequisitos:	NINGUNO
Responsable docente:	MIGUEL ÁNGEL GUTIÉRREZ GARCÍA DOCTOR EN INFORMÁTICA ESPECIALIZADO EN LENGUAJES Y SISTEMAS INFORMÁTICOS.
Email:	miguel.gutierrez@ucavila.es
Departamento (Área Departamental):	TECNOLÓGICO
Lengua en la que se imparte:	CASTELLANO
Módulo:	CONTENIDOS COMUNES A LA INGENIERÍA INFORMÁTICA
Materia:	COMPILADORES



A lo largo de toda la carrera habéis desarrollado programas en distintos lenguajes. Tras escribir el código fuente lo habéis compilado para generar un ejecutable. Sin embargo, pese a los conocimientos adquiridos sobre lenguajes de programación, el compilador sigue siendo un gran desconocido. ¿Cómo funciona? ¿Qué partes lo componen? En esta asignatura aprenderemos esto y empezaremos a comprender lo importante y difícil que es crear un buen compilador para un lenguaje de programación.

2.1. COMPETENCIAS BÁSICAS Y GENERALES

- CB1. Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.
- CB2. Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
- CB3. Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
- CB4. Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- CB5. Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
- G1. Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto la

concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.

- G8. Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- G9. Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

2.2. COMPETENCIAS ESPECÍFICAS

- E4. Conocimientos básicos sobre el uso y programación de los ordenadores, los sistemas operativos, las bases de datos y, en general, los programas informáticos con aplicación en ingeniería.
- E7. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
- E18. Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.
- E33. Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación.

2.3. RESULTADOS DE APRENDIZAJE

- Distinguir las distintas fases de la compilación de un programa y comprender su funcionamiento.
- Diseñar e implementar un compilador sencillo.



3.1. PROGRAMA

UNIDAD 1. INTRODUCCIÓN

- 1.1. Traductores
- 1.2. Estructura de un compilador
- 1.3. Análisis léxico
- 1.4. Análisis sintáctico
- 1.5. Análisis semántico
- 1.6. Tabla de símbolos
- 1.7. Gestión de errores
- 1.8. Generación de código

UNIDAD 2. ANÁLISIS LÉXICO

- 2.1. Definiciones
- 2.2. Gramáticas lineales
- 2.3. Expresiones regulares
- 2.4. Autómatas finitos deterministas
- 2.5. Conversión desde gramáticas lineales
 - 2.5.1. A expresiones regulares
- 2.6. Conversión desde expresiones regulares
 - 2.6.1. A gramáticas lineales
 - 2.6.2. A autómatas finitos
- 2.7. Conversión desde autómatas finitos
 - 2.7.1. A gramáticas lineales
 - 2.7.2. A expresiones regulares
- 2.8. Implementación de analizadores léxicos

UNIDAD 3. ANÁLISIS SINTÁCTICO

- 3.1. Gramáticas libres de contexto
 - 3.1.1. Árboles de análisis sintáctico
 - 3.1.2. Eliminación de problemas gramaticales

- 3.2. Diagramas de sintaxis
- 3.3. Autómatas de pila deterministas
- 3.4. Tipos de analizadores sintácticos
- 3.5. Análisis sintáctico descendente con retroceso
- 3.6. Análisis sintáctico descendente predictivo
 - 3.6.1. Cálculo de los conjuntos de primeros
 - 3.6.2. Cálculo de los conjuntos de siguientes
 - 3.6.3. Cálculo de los conjuntos de predicción
 - 3.6.4. Uso del análisis predictivo LL(1)
 - 3.6.5. Implementación mediante funciones recursivas
 - 3.6.6. Implementación mediante tabla
- 3.7. Análisis sintáctico ascendente por reducción - desplazamiento
- 3.8. Análisis sintáctico ascendente general
 - 3.8.1. Analizadores LR(0)
 - 3.8.2. Analizadores SLR
 - 3.8.3. Analizadores LR(1)
 - 3.8.4. Analizadores LALR
- 3.9. Gestión de errores sintácticos

UNIDAD 4. ANÁLISIS SEMÁNTICO

- 4.1. Definiciones dirigidas por la sintaxis
- 4.2. Esquemas de traducción dirigidos por la sintaxis
- 4.3. Comprobación de tipos

UNIDAD 5. GENERACIÓN DE CÓDIGO

- 5.1. Generación de código intermedio
 - 5.1.1. Lenguajes intermedios
 - 5.1.2. Llamadas a subrutinas
 - 5.1.3. Enlazado de nombres
 - 5.1.4. Paso de parámetros a subrutinas
- 5.2. Generación de código final
- 5.3. Optimización de código

3.2. BIBLIOGRAFÍA

- Kenneth C. Louden. Construcción de compiladores. Thomson.
- Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman. Compiladores. Principios, técnicas y herramientas. Addison-Wesley.
- Sergio Gálvez Rojas, Miguel Ángel Mora Mata. Traductores y compiladores con Lex/Yacc, JFlex/Cup y JavaCC. Universidad de Málaga.
- John R. Levine, Tony Mason, Doug Brown. Lex & Yacc. O'reilly.
- J. P. Bennett. Introduction to Compiling Techniques: First Course Using ANSI C, LEX and YACC. Alfred Waller Ltd.
- R. M. Kaplan. Constructing Language Processors for Little Languages. John Wiley & Sons.
- S. S. Muchnick. Advanced Compiler Design and Implementation. Morgan Kaufmann.
- Proyecto LUCAS. Manuales sobre Flex y Bison. Enlace: <http://es.tldp.org/htmls/manuales.html>
- John R. Levine, Tony Mason, Doug Brown. Lex & Yacc. O'reilly.



La asignatura se desarrollará a través de los siguientes métodos y técnicas generales, que se aplicarán diferencialmente según las características propias de la asignatura:

- **Estudio personal dirigido:** el alumno acometerá de forma individual el estudio de la asignatura de modo que le permita adquirir las competencias de la misma. Para ello contará con la tutorización personalizada del profesor de la asignatura, como principal responsable docente.

- **Prácticas con ordenador:** el alumno realizará individualmente las diferentes prácticas según las competencias y actividades correspondientes que deba trabajar durante el semestre. Para ello contará con la tutorización personalizada del profesor de la asignatura, como principal responsable docente.
- **Realización de test de autoevaluación:** El alumno contará con test al término de cada unidad para repasar los contenidos y poner en práctica los conocimientos adquiridos.
- **Acceso a clases en directo:** el alumno tendrá la posibilidad de participar en directo en las clases de la asignatura. Además, estas clases quedarán grabadas y estarán a disposición del alumno en la plataforma virtual de la asignatura.
- **Tutorías personalizadas:** El profesor pondrá a disposición del alumno un tiempo para que éste pueda plantear cuantas dudas le surjan en el estudio de la materia, pudiendo el docente ilustrar sus explicaciones por medio de ejemplos y cualquier otra orientación de interés para el alumno.
- **Actividades de evaluación**



La evaluación es un componente fundamental de la formación del alumno. Está compuesta por un examen final escrito y la evaluación continua, que consta de ejercicios y actividades evaluables.

La evaluación de esta asignatura se realiza mediante la media del examen (valorado en un 60%) y la realización de un trabajo obligatorio individual (con valor del 40%).

➤ Examen (60 % de la nota final)

La superación de dicho examen constituye un requisito indispensable para la superación de la asignatura. El alumno deberá tener en el examen al menos un 5

para poder realizar la ponderación de notas. El alumno con nota inferior se considerará suspenso. El alumno dispondrá de dos convocatorias de examen por curso académico.

No se guardará la nota del examen, si éste estuviera aprobado, para una convocatoria posterior.

➤ Trabajo obligatorio (40% de la nota final)

La superación del trabajo constituye un requisito indispensable para la superación de la asignatura. El alumno deberá tener en el trabajo al menos un 5 para poder realizar la ponderación de notas. El alumno con nota inferior se considerará suspenso. En el caso de tener el trabajo obligatorio superado y no aprobar el examen, se guardará su nota hasta la segunda convocatoria de examen perteneciente al curso académico actual.

No se admitirán trabajos fuera de la fecha límite de entrega, que será comunicada al alumno con suficiente antelación. Con la no presentación del trabajo obligatorio se considerará suspensa la asignatura, independientemente de la nota obtenida en el examen.

Criterios de calificación de la evaluación continua

Los criterios para la evaluación del trabajo obligatorio se presentan en la siguiente tabla, donde se resumen los aspectos a valorar y el porcentaje que representa cada uno de los mismos:

COMPONENTES EVALUABLES	PROPORCIÓN
Implementación correcta	30%
Complejidad del lenguaje	50%
Documentación	20%
TOTAL	100%

Para el apoyo tutorial, el alumno tendrá a su disposición un equipo docente encargado de acompañar al alumno durante toda su andadura en el proceso formativo, prestando una atención personalizada al alumno. Sus funciones están claramente diferenciadas complementándose al mismo tiempo. Las dos personas principales de este acompañamiento tutorial son:

- **Orientador Académico Personal:** encargado de planificar al alumno el estudio de la asignatura en función del tiempo disponible, incluso realiza nuevas planificaciones ajustándose a nuevos periodos marcados por el alumno según sus circunstancias personales y familiares. Otra de sus funciones es la de realizar un seguimiento del estudio del alumno, así como de dar al alumno información de carácter general necesaria en su proceso formativo.
- **Profesor docente:** encargado de resolver todas las dudas específicas de la asignatura y de informar al alumno de todas las pautas que debe seguir para realizar el estudio de la asignatura.

El alumno dispondrá de un horario de tutorías para contactar con estas figuras durante toda su formación académica. La información sobre el horario la encontrará el alumno en la plataforma virtual.

Horario de tutorías de la asignatura: martes de 20:00 a 21:00 y jueves de 19:00 a 21:00.

El peso de cada unidad formativa dentro de cada asignatura queda determinado en el cronograma por el tiempo dedicado a la misma. El alumno deberá acometer el estudio marcado por la herramienta de planificación utilizada en el campus virtual, después de la planificación realizada con su tutor. A continuación se muestra una tabla con las unidades

didácticas que componen la asignatura y las unidades de tiempo que se requieren para su estudio.

UNIDADES DIDÁCTICAS	HORAS DEDICACIÓN
Unidad 1 – Introducción	15 HORAS
Unidad 2 – Análisis Léxico	45 HORAS
Unidad 3 – Análisis Sintáctico	60 HORAS
Unidad 4 – Análisis Semántico	15 HORAS
Unidad 5 – Generación de Código	15 HORAS
TOTAL	150