



Universidad
de Alcalá

GUÍA DOCENTE

Compiladores

Grado en
Ingeniería de Computadores (GIC)

Universidad de Alcalá

Curso Académico 2022/2023

3^{er} Curso - 1^{er} Cuatrimestre (GIC)

GUÍA DOCENTE

Nombre de la asignatura:	Compiladores
Código:	591000 (GIC)
Titulación en la que se imparte:	Grado en Ingeniería de Computadores (GIC)
Departamento y Área de Conocimiento:	Ciencias de la Computación Ciencias de la Computación
Carácter:	Obligatoria (GIC)
Créditos ECTS:	6.0
Curso y cuatrimestre:	3^{er} Curso - 1^{er} Cuatrimestre (GIC)
Profesorado:	Por definir
Horario de Tutoría:	Consultar al comienzo de la asignatura
Idioma en el que se imparte:	Español

1a. PRESENTACIÓN

La asignatura Compiladores estudia el proceso de traducción de los lenguajes formales con carácter general, ya sean lenguajes de datos o lenguajes de programación, sus técnicas de traducción, los módulos que componen los procesadores de lenguaje clásicos y las formas en que dichos módulos se comunican entre sí. El objetivo principal es introducir a los estudiantes en la teoría y la práctica de la traducción de lenguajes.

Se organiza como una asignatura de 6 ECTS, donde se explora la teoría de la traducción de lenguajes y se muestra cómo aplicar esta teoría a la construcción de compiladores y procesadores de lenguajes formales. Abarca el diseño de gramáticas, la construcción de traductores asistida por generadores automáticos de analizadores léxicos y sintácticos, y el análisis profundo de los problemas que surgen durante el diseño de los traductores.

La asignatura presupone que el estudiante posee conocimientos de programación, específicamente en lenguaje Java o C que son los lenguajes que se necesitan para manejar adecuadamente las herramientas de generación automática de procesadores léxicos y sintácticos empleados. Es por ello por lo que resulta especialmente importante haber cursado con anterioridad asignaturas de programación que provean de estas competencias, así como competencias sobre Sistemas Operativos y Estructuras de Datos.

1b. COURSE SUMMARY

This topic studies the process of translating formal languages, data or programming oriented to executable code by building compilers. The main objective is to introduce students to the theory and practice of languages processing such as scanning, parsing, type checking and code generation. It is organized as a subject of 6 ECTS, where the theory of language translation is studied and applied to the construction of compilers and interpreters. It covers the grammar design, building parsers and scanners by making use of automatic generators, and studies the problems that arise during the design of translators. The course assumes that the student has a good knowledge of programming, specifically Java or C language. It is especially important to have attended previously a course on Fundamentals of Programming, and it is also recommended a background in Operating Systems and Data Structures.

2. COMPETENCIAS

Competencias básicas, generales y transversales.

Esta asignatura contribuye a adquirir las siguientes competencias básicas, generales y transversales:

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.

CG4 - Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5, anexo 2, de la resolución BOE-A-2009-12977.

CG5 - Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5, anexo 2, de la resolución BOE-A-2009-12977.

CG8 - 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.

CG9 - 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.

TRU1 - Capacidad de análisis y síntesis.

TRU2 - Comunicación oral y escrita.

TRU3 - Capacidad de gestión de la información.

TRU4 - Capacidad de aprendizaje autónomo.

TRU5 - Capacidad para trabajar en equipo.

Competencias Específicas

Esta asignatura proporciona la(s) siguiente(s) competencia(s) específica(s):

CI1 - 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.

CI5 - Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas.

CI7 - Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

CI8 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Resultados de aprendizaje

Al terminar con éxito esta asignatura/enseñanza, los estudiantes serán capaces de:

RA1. Diseñar e implementar analizadores léxicos sencillos que produzcan tokens que puedan ser procesados por un analizador.

RA2. Diseñar la gramática de un lenguaje y utilizarla para generar un analizador sintáctico que junto con un analizador léxico verifiquen los errores estructurales del texto fuente.

- RA3.** Realizar un análisis sintáctico simple sin utilizar herramientas de generación automática para ello.
- RA4.** Utilizar herramientas de construcción automática de traductores (ANTLR, Flex/Bison, LEX/YACC, JLex-JFlex/CUP...).
- RA5.** Diseñar analizadores sintácticos para lenguajes artificiales y resolver los posibles conflictos que pudieran aparecer.
- RA6.** Identificar los tipos de errores que se pueden presentar durante el análisis de un programa, cómo se detectan, cómo se informa de estos errores y cómo un traductor se puede recuperar de ellos.
- RA7.** Incorporar a un analizador sintáctico la gestión de errores y de la tabla de símbolos.
- RA8.** Describir e interpretar cómo se aplican los sistemas de tipos en un lenguaje de programación.
- RA9.** Describir los distintos tipos de modelos de control de errores, y saber aplicar los correspondientes algoritmos de control.
- RA10.** Construir los módulos de un compilador encargados de las fases de análisis léxico, sintáctico, semántico y de generación de código intermedio.
- RA11.** Dominar los fundamentos de la generación de código final y las técnicas de optimización del código.
- RA12.** Determinar la posición que ocuparán los datos durante la ejecución de un programa y el modo de acceso a los mismos.

3. CONTENIDOS

Bloques de contenido	Total de clases, créditos u horas
Introducción a los traductores	0,5 ECTS
Análisis léxico	1,5 ECTS
Análisis sintáctico	2 ECTS
Análisis semántico	1 ECTS
Generación y optimización de código	0,5 ECTS
Fase final o back-end	0,5 ECTS

4. METODOLOGÍAS DE ENSEÑANZA APRENDIZAJE. ACTIVIDADES FORMATIVAS

4.1. Distribución de créditos (especificar en horas)

Número de horas presenciales:	58 horas (56 horas de clase presencial +2 horas de evaluación)
Número de horas del trabajo propio del estudiante:	92 (Incluye horas de estudio, elaboración de actividades, preparación de exámenes)
Total horas	150

4.2. Estrategias metodológicas, materiales y recursos didácticos

Clases magistrales y expositivas, en combinación con prácticas en el laboratorio	Exposición de contenidos en las clases teóricas presenciales y de laboratorio.
Trabajos en grupo y cooperativo	Elaboración de trabajos en equipo para presentación en laboratorio.
Trabajo y estudio personal	Elaboración de trabajos individuales para presentación en laboratorio y desarrollo de pruebas en teoría.
Actividades presenciales	- En el aula: Exposición y discusión de los conocimientos básicos de la asignatura. Planteamiento y resolución teórica de ejercicios y supuestos relacionados. Actividades (lecturas, discusiones, casos, etc.) orientadas a la enseñanza de las competencias específicas de la asignatura. - En el laboratorio: Planteamiento, desarrollo y solución de ejercicios prácticos utilizando herramientas, técnicas y métodos objeto de estudio de la asignatura, contribuyendo al desarrollo de la capacidad de análisis, razonamiento crítico y comprensión de las prácticas realizadas.
Actividades no presenciales	- Análisis y asimilación de los contenidos de la materia, resolución de problemas, consultas bibliográficas, preparación de trabajos individuales y/o grupales, realización de autoevaluaciones. Orientadas especialmente al desarrollo de métodos para la organización y planificación del trabajo individual y en equipo. - Tutorías: asesoramiento individual y en grupos durante el proceso de enseñanza-aprendizaje, bien en forma presencial o a distancia.

5. EVALUACIÓN: Procedimientos, criterios de evaluación y calificación

Preferentemente se ofrecerá a los alumnos un sistema de evaluación continua que tenga características de evaluación formativa de manera que sirva de realimentación en el proceso de enseñanza-aprendizaje por parte del alumno.

5.1. PROCEDIMIENTOS

La evaluación debe estar inspirada en los criterios de evaluación continua (Normativa de Evaluación de los Aprendizajes, NEA, art 3). No obstante, respetando la normativa de la Universidad de Alcalá se pone a disposición del alumno un proceso alternativo de evaluación final de acuerdo a la [Normativa de Evaluación de los Aprendizajes](#) según lo indicado en su Artículo 10, los alumnos tendrán un plazo de quince días desde el inicio del curso para solicitar por escrito al Director de la Escuela Politécnica

Superior su intención de acogerse al modelo de evaluación no continua aduciendo las razones que estimen convenientes. La evaluación del proceso de aprendizaje de todos los alumnos que no cursen solicitud al respecto o vean denegada la misma se realizará, por defecto, de acuerdo al modelo de evaluación continua. El estudiante dispone de dos convocatorias para superar la asignatura, una ordinaria y otra extraordinaria.

Convocatoria ordinaria

Evaluación continua:

La asignatura consta de pruebas individuales competitivas en clase teórica, pruebas individuales de conocimiento, pruebas individuales de laboratorio y pruebas en grupo en laboratorio.

La superación de la asignatura se tomará como la media ponderada de las distintas pruebas realizadas en función de la tabla correspondiente (sección "Evaluación").

Evaluación mediante examen final:

En el caso de evaluación mediante examen final, se exigirá la presentación de las prácticas de laboratorio de la asignatura de manera individual, lo que garantizará los conocimientos prácticos necesarios para superar la asignatura, junto con una prueba final teórica.

Convocatoria extraordinaria

El procedimiento para la convocatoria extraordinaria incluye la realización de una práctica individual y un examen teórico. Dado que los pesos y pruebas son distintos de los realizados en la convocatoria ordinaria, no se conservarán notas parciales de pruebas previas.

5.2. EVALUACIÓN

CRITERIOS DE EVALUACIÓN

Se utilizarán los siguientes criterios para la evaluación de la asignatura, relacionados con los resultados del aprendizaje:

- CE1.** El alumno reconoce los fundamentos de la traducción automática, distingue e identifica las diferentes fases y ha asimilado los conceptos subyacentes esenciales a la misma.
- CE2.** El alumno es capaz de diseñar una gramática para un lenguaje artificial.
- CE3.** El alumno demuestra aptitud en el manejo de generadores automáticos de analizadores léxicos y sintácticos.
- CE4.** El alumno sabe cómo añadir acciones semánticas a una gramática.
- CE5.** El alumno es capaz de diseñar estrategias de tratamiento de errores en la traducción de lenguajes.
- CE6.** El alumno domina desde el punto de vista teórico los diferentes enfoques para el análisis sintáctico, tanto ascendente como descendente, e identifica las diferencias, ventajas e inconvenientes entre los distintos métodos existentes.
- CE7.** El alumno es capaz de implementar y utilizar de manera óptima una tabla de símbolos.
- CE8.** El alumno reconoce la importancia del código intermedio y sabe aplicar estrategias de optimización al código generado por un traductor.
- CE9.** El alumno identifica la necesidad de relacionar el código estático de un programa con las acciones a desarrollar en ejecución, y es capaz de implementar a nivel teórico las abstracciones que aparecen en el código fuente.

INSTRUMENTOS DE EVALUACIÓN

Esta sección resume los instrumentos de calificación que serán aplicados a cada uno de los criterios de Evaluación.

- Pruebas de adquisición de conocimiento (PAC-n): Pruebas a modo de test competitivo a realizar tras las clases teóricas que tienen como fundamento fomentar la adquisición de conocimiento básico de esa sesión, actuando al mismo tiempo como refuerzo positivo activo de los elementos más importantes de la misma. Tiene un espíritu de fomento de la participación en clase y mejora de los resultados de las PEI asociadas.
- Prueba de Evaluación Intermedia (PEI-1): Consistente en la resolución de cuestiones teórico-prácticas sobre conceptos de traducción, análisis léxico y sintáctico.
- Prueba de Evaluación Intermedia (PL-1): Consistente en la elaboración y defensa de un programa capaz de evaluar una expresión regular simple a través del diseño e implementación de autómatas. Individual.
- Pruebas de Evaluación Intermedia (PL-2): Consistente en la elaboración y defensa de un analizador léxico utilizando un generador automático junto con una gramática simple. En grupo (máx. 3).
- Prueba de Evaluación Intermedia (PEI-2): Consistente en la resolución de cuestiones teórico-prácticas sobre análisis semántico, tablas de símbolos, comprobación de tipos, generación y optimización de código, y ambientes para la ejecución.
- Prueba de Evaluación Intermedia (PL-3): Consistente en la elaboración y defensa de un analizador sintáctico y semántico utilizando un generador automático, implementando una tabla de símbolos y generando código intermedio de salida. En grupo (máx. 3).
- Prueba de Evaluación Final Teórica (PEF-T): consistente en la resolución de supuestos teórico-prácticos, así como en la solución de problemas complejos, sobre conceptos de traducción, análisis léxico y sintáctico, análisis semántico, tablas de símbolos, comprobación de tipos, generación y optimización de código, y entornos para la ejecución.
- Prueba de Evaluación Final Práctica (PEF-P): consistente en la elaboración de un traductor (léxico+sintáctico+semántico) utilizando generadores automáticos, e implementando una tabla de símbolos y generando código intermedio de salida. Individual.

CRITERIOS DE CALIFICACIÓN (Ejemplo de tabla, modifícala según sus necesidades)

En la convocatoria **ordinaria–evaluación continua** la relación entre las competencias, resultados del aprendizaje, criterios e instrumentos de evaluación, es la siguiente.

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CG4, CG8, CG9, CI1, CI5, CB1-CB5, TRU1-TRU4	RA1, RA2, RA3	CE1, CE2, CE6	PEI-1, PAC-n	20%
CG4, CG5, CG8, CG9, CI1, CI7, CB1-CB5, TRU1-TRU4	RA1, RA2	CE1, CE2	PL-1	10%
CG4, CG5, CG8, CG9, CI1, CI7, CB1-CB5, TRU1-TRU5	RA4	CE2, CE3, CE5	PL-2	20%
CG4, CG8, CG9, CI1, CI5, CI7, CI8, CB1-CB5, TRU1-TRU4	RA2, RA5-RA12	CE1, CE2, CE3, CE4, CE6, CE7, CE8, CE9	PEI-2, PAC-n	20%
CG4, CG5, CG8, CG9, CI1, CI7, CI8, CB1-CB5, TRU1-TRU5	RA2, RA4-RA11	CE3, CE4, CE5, CE7, CE8, CE9	PL-3	30%

Se otorgará la calificación de "No presentado" al alumno que habiendo optado por el procedimiento de

evaluación continua, cumpla alguno de los siguientes requisitos:

- Cuando el alumno presente instrumentos de evaluación cuyo peso de calificación máximo combinado sea inferior al 50% del total de la asignatura.

En la convocatoria **ordinaria-evaluación final** la relación entre las competencias, resultados del aprendizaje, criterios e instrumentos de evaluación, es la siguiente.

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CG4, CG8, CG9, CI1, CI5, CI7, CI8, CB1-CB5, TRU1-TRU4	RA1-RA3, RA5-RA12	CE1-8	PEF-T	40%
CG4, CG5, CG8, CG9, CI1, CI7, CB1-CB5, TRU1-TRU4	RA1, RA2	CE1, CE2	PL-1	10%
CG4, CG5, CG8, CG9, CI1, CI7, CB1-CB5, TRU1-TRU4	RA4	CE2, CE3, CE5	PL-2	20%
CG4, CG5, CG8, CG9, CI1, CI7, CI8, CB1-CB5, TRU1-TRU4	RA2, RA4-RA11	CE3, CE4, CE5, CE7, CE8, CE9	PL-3	30%

Se otorgará la calificación de "No presentado" al alumno que habiendo optado por el procedimiento de evaluación final no presente el total de los instrumentos de evaluación establecidos.

Convocatoria **extraordinaria** la relación entre las competencias, resultados del aprendizaje, criterios e instrumentos de evaluación, es la siguiente.

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CG4, CG8, CG9, CI1, CI5, CI7, CI8, CB1-CB5, TRU1-TRU4	RA1-RA3, RA5-RA12	CE1-CE9	PEF-T	50%
CG4, CG5, CG8, CG9, CI1, CI7, CI8, CB1-CB5, TRU1-TRU4	RA2, RA4-RA11	CE3, CE4, CE5, CE7, CE9	PEF-P	50%

Se otorgará la calificación de "No presentado" al alumno que utilizando la convocatoria extraordinaria no presente el total de los instrumentos de evaluación establecidos.

Todas las pruebas podrán realizarse en las aulas de teoría o laboratorio, o a través del Aula Virtual u otros instrumentos virtuales/tecnológicos.

Como resultado del proceso de evaluación el alumno obtendrá una calificación que dependerá de su actividad en las distintas pruebas de la asignatura. El resultado de cada prueba arrojará información bien mediante indicadores cuantitativos de adquisición de competencias, bien mediante una calificación cualitativa, que a modo de orientación podrá determinarse en función del grado de dominio mostrado en las tareas propuestas por los profesores responsables de la asignatura:

Sobresaliente	Notable	Aprobado	Suspense
Excelente dominio de los conocimientos básicos.	Dominio de los conocimientos básicos.	Domina los conocimientos básicos.	Bajo nivel de comprensión y aplicación de ideas.
Elaboración de ideas a partir de la reflexión y aplicación de los conocimientos adquiridos.	Alto nivel de reflexión.	Nivel medio de reflexión.	Nivel bajo de reflexión.
Cumplimiento de todas las tareas programadas.	Cumplimiento adecuado de la mayoría de las tareas programadas.	Cumplimiento de un número suficiente de las tareas programadas.	Falta de implicación en las tareas propuestas por el profesor.

6. BIBLIOGRAFÍA

6.1. Bibliografía básica

- AHO, A.V., SETHI, R. y ULLMAN, J.D. (2008) Compiladores: Principios, técnicas y herramientas, 2ª edición. Addison–Wesley. Clásico de la materia que trata con mucha profusión y profundidad la mayoría de los temas.
- LOUDEN, K. C. 1997. Construcción de compiladores: Principios y práctica. Uno de los libros más claros y didácticos sobre el tema, aborda todas las fases de la traducción con la profundidad exigida a un curso de grado pero sin perder por ello la claridad y la visión didáctica. Es posiblemente el mejor libro para hacer un estudio poco dirigido por un profesor, por lo que se recomienda especialmente dada la naturaleza de las nuevas titulaciones en el EEES
- PARR, T, 2014. Language Implementation Patterns: Patrones de uso para la implementación de lenguajes de programación. Hace uso de casos prácticos con la herramienta ANTLR, mostrando las distintas fases del proceso, desde el análisis léxico hasta la generación de código.

6.2. Bibliografía complementaria

- BENNETT, J. P. (1996). Introduction to Compiling Techniques. McGraw-Hill, 1996. Texto fácil de leer y ameno de nivel introductorio. Puede ser un complemento como texto de introducción para aquellos que buscan un libro fácil y didáctico con una fuerte componente práctica al mismo tiempo.
- DOMINGUEZ, J. (2008). Compiladores: Teoría Y práctica Con Java, Jlex, Cup Y Ens2001. Lulu.com. Describe completamente el proceso de creación de un compilador en Java con las herramientas JLex, Cup y Ens2001. Es un excelente complemento para el estudio del análisis sintáctico y una imprescindible referencia para los trabajos prácticos de las Pruebas de Evaluación Continua.
- GRUNE, D y Jacobs, C.J.H. (2008). Parsing techniques: a practical guide, 2nd edition. New York: Springer. Se dedica íntegramente a una de las partes más complejas para los alumnos, el análisis sintáctico, que constituye en la práctica el primer escalón verdaderamente fuerte en su curva de aprendizaje en la asignatura.
- MAK, R. (2009). Writing Compilers and Interpreters: A Software Engineering Approach, 3rd edition.

Wiley. Desarrolla muy bien las técnicas para construir un analizador, un intérprete, un depurador a nivel de código fuente y un compilador para la máquina virtual Java Virtual Machine. El libro es más para el que quiere aplicar los compiladores que para el teórico del compilador, por lo que resulta especialmente recomendable como compañero a la hora de implementar las partes más prácticas de la asignatura.

- MUCHNICK, S. (1997). Advanced Compiler Design and Implementation. Morgan Kaufmann. Cubre temas avanzados en todas las áreas fundamentales del diseño de compiladores, pero es especialmente interesante como guía de estudio en los temas relacionados con la optimización de código pues describe una amplia gama de posibles algoritmos de optimización y determina la importancia relativa de las optimizaciones. Excelente complemento para estudiar la fase de optimización.
- WIRTH, N. 1996. Compiler construction. Addison–Wesley. Presenta un ejemplo completo de compilador con el lenguaje Oberon para una arquitectura RISC lo cual resulta muy útil para el alumno cuando se enfrenta a la tarea de realizar su propia implementación pues tiene ejemplos similares de los que aprender. Interesante complemento para el estudio de la asignatura.

NOTA INFORMATIVA

La Universidad de Alcalá garantiza a sus estudiantes que, si por exigencias sanitarias las autoridades competentes impidieran la presencialidad total o parcial de la actividad docente, los planes docentes alcanzarían sus objetivos a través de una metodología de enseñanza-aprendizaje y evaluación en formato online, que retornaría a la modalidad presencial en cuanto cesaran dichos impedimentos.