Guía docente
270030 - CL - Compiladores

Última modificación: 30/01/2024

Unidad responsable: Facultad de Informática de Barcelona
Unidad que imparte: 723 - CS - Departamento de Ciencias de la Computación.

Titulación: GRADO EN INGENIERÍA INFORMÁTICA (Plan 2010). (Asignatura optativa).

Curso: 2023 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: JOSE MIGUEL RIVERO ALMEIDA

Otros: Segon quadrimestre:
LLUIS PADRO CIRERA - 12
JOSE MIGUEL RIVERO ALMEIDA - 11, 12

CAPACIDADES PREVIAS

Para seguir la asignatura de Compiladores, es necesario que el estudiante llegue con un conocimiento adecuado de teoría de lenguajes formales (lenguajes regulares y libres de contexto) y autómatas (finitos y de pilas). También es necesario tener conocimientos de estructuras de datos (listas, árboles, grafos) y de algoritmos básicos sobre estas estructuras (recorridos, búsquedas, recursividad). Finalmente, se necesitan conocimientos básicos de lenguaje máquina y ensamblador. Por las razones antes mencionadas, conviene que el estudiante haya realizado las asignaturas de Algorítmica, Teoría de la Computación y Estructura de Computadores.

REQUISITOS

- Pre-requisito TC

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:
CCO1.2. Demostrar conocimiento de los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, el diseño y el procesamiento de lenguajes.
CCO2.3. Desarrollar y evaluar sistemas interactivos y de presentación de información compleja, y su aplicación a la resolución de problemas de diseño de interacción persona computadora.
CT1.2C. Interpretar, seleccionar y valorar conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática y su aplicación a partir de los fundamentos matemáticos, estadísticos y físicos necesarios. CEFB1: Capacidad para la resolución de los problemas matemáticos que puedan plantarse en la ingeniería. Aptitud para aplicar los conocimientos sobre: algebra, cálculo diferencial e integral i métodos numéricos; estadística y optimización.
CT4.1. Identificar las soluciones algorítmicas más adecuadas para resolver problemas de dificultad mediana.

Genéricas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

METODOLOGÍAS DOCENTES

Los contenidos teóricos de la asignatura se imparten en las clases de teoría. Estas clases se complementan con ejemplos prácticos y problemas que los estudiantes tienen que resolver en las horas de Aprendizaje Autónomo. Durante las clases de teoría y esporádicamente en algunas clases de laboratorio se resolverán algunos de los ejercicios mas representativos del curso.

En las sesiones de laboratorio se imparten aquellos conocimientos que tienen que apoyar a la realización del proyecto de la asignatura. En las primeras sesiones se revisarán ejemplos sencillos en los que los estudiantes tendrán que hacer modificaciones para familiarizarse con las soluciones propuestas. Después de unas sesiones introductorias, los estudiantes centrarán sus esfuerzos en el proyecto del curso. Durante las clases de laboratorio, el profesor irá introduciendo nuevas técnicas y dejará una parte importante de la clase para que los estudiantes trabajen con su proyecto con la ayuda del profesor cuando sea necesario. En esta fase será imprescindible la utilización de las horas de Aprendizaje Autónomo asignadas al proyecto.

El proyecto de laboratorio se hará en grupos de 2 o 3 personas, dependiendo de la complejidad del problema. Se espera que los estudiantes organicen su grupo de forma que las tareas sean distribuidas y sincronizadas. Cada estudiante tendrá que tener una responsabilidad importante e individual en el proyecto.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

1.Conocer la estructura general de los compiladores e intérpretes de lenguajes de programación.
2.Conocer las técnicas de análisis léxico y los métodos para generar analizadores léxicos
3.Conocer las técnicas de análisis sintáctico y los métodos para generar analizadores sintácticos.
4.Conocer las técnicas de análisis semántico de un lenguaje de programación.
5.Conocer la estructura de un intérprete y los conceptos básicos de máquinas virtuales.
6.Conocer las técnicas de generación de código de un compilador.
7.Conocer las técnicas de optimización de código de un compilador.
8.Aprender a organizar un grupo de trabajo para realizar una tarea de mediana complejidad.
9.Aprender a diseñar lenguajes de programación para aplicaciones en diversos ámbitos de la Informática.
10.Aprender a desarrollar un proyecto de mediana complejidad de diseño y traducción/interpretación de un lenguaje de programación.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas aprendizaje autónomo	84,0	56.00
Horas grupo grande	30,0	20.00
Horas actividades dirigidas	6,0	4.00
Horas grupo pequeño	30,0	20.00

Dedicación total: 150 h

CONTENIDOS

Introducción.

Descripción:
Elementos de un lenguaje de programación. Compiladores e intérpretes: concepto y diferencias. Estructura y fases de un compilador y un intérprete.

Análisis léxico.

Descripción:
Objetivos del análisis léxico. Revisión de la teoría de lenguajes regulares y autómatas finitos. Construcción de analizadores léxicos a partir de autómatas indeterministas. Herramientas de análisis léxico.

Análisis sintáctico.

Descripción:
Objetivos del análisis sintáctico. Revisión de la teoría de lenguajes libres de contexto y autómatas de pila. Gramáticas ambiguas y recursivitat por la derecha y por la izquierda. Construcción de analizadores sintácticos descendentes. Construcción de analizadores sintácticos ascendentes.

Árboles sintácticos.

Descripción:
Árboles sintácticos concretos y abstractos. Construcción de árboles sintácticos. Información almacenada en los árboles sintácticos.

Análisis semántico.

Descripción:
Nombres y objetos en un lenguaje de programación. Vida de los objetos. Visibilidad de nombres estática y dinámica. Bloc de activación de una función.Tablas de símbolos. Comprobaciones de tipos.

Intérpretes y máquinas virtuales.

Descripción:
Interpretación de un lenguaje de programación. Concepto de máquina virtual. Tipos de intérpretes. Fases de un intérprete. Representaciones internas e intermedias: en árbol, código de tres direcciones, código para máquinas de pilas. Acceso a nombres locales y no locales. Ejemplos de máquinas virtuales.

Generación de código.

Descripción:
Código intermedio. Generación de código para expresiones aritméticas y booleanas. Generación de código para instrucciones: asignación, instrucción condicional, instrucción iterativa. Funciones: bloques de activación, de parámetros y retorno de resultados. Acceso a estructuras de datos.

Optimización de código.

Descripción:
Bloques básicos y optimizaciones locales. Grafo de control de flujo. Análisis de flujo para optimizaciones globales. Vida y uso/definición de variables. Optimizaciones globales: subexpressions comunes, código muerto, propagación de constantes y propagación de copias. Optimizaciones en bucles: invariantes y variables de inducción.

ACTIVIDADES

Prueba de laboratorio

Objetivos específicos:
3, 7

Dedicación: 5h
Actividades dirigidas: 3h
Aprendizaje autónomo: 2h

Proyecto

Objetivos específicos:
8, 9, 10

Competencias relacionadas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 9h
Actividades dirigidas: 1h
Aprendizaje autónomo: 8h

Aprendizaje de conceptos generales sobre lenguajes, compiladores e intérpretes.

Descripción:
La actividad básica es la de assitir a clase para adquirir los conocimientos de este tema y tener una visión global del curso. El estudiante podrá consultar libros especializados en el supuesto de que quiera aprofondir en algún aspecto determinado.

Objetivos específicos:
3, 7, 9

Competencias relacionadas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 4h
Grupo grande/Teoría: 2h
Aprendizaje autónomo: 2h

Examen de teoría

Objetivos específicos:
3, 7, 9

Competencias relacionadas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 9h
Actividades dirigidas: 3h
Aprendizaje autónomo: 6h

Aprendizaje sobre la teoría y técnicas de análisis léxico.

Descripción:
El estudiante asistirá en las clases teóricas y dedicará tiempos a estudiar la teoría sobre análisis léxico y realizar ejercicios propuestos por el profesor. Una parte de los conocimientos adquiridos tendrán que ser aplicados al proyecto de la asignatura.

Objetivos específicos:
2

Dedicación: 8h
Grupo grande/Teoría: 3h
Aprendizaje autónomo: 5h

Aprendizaje sobre la teoría y técnicas de diseño de analizadores sintácticos descendentes.

Descripción:
El alumno asistirá en clase para adquirir los conocimientos teóricos. Addicionalement tendrá que consolidar estos conceptos con su estudio personal y con la resolución de los problemas propuestos a clase. Una parte de los conocimientos adquiridos serán utilizados en el proyecto.

Objetivos específicos:
3

Dedicación: 9h
Grupo grande/Teoría: 3h
Aprendizaje autónomo: 6h

Aprendizaje sobre la teoría y técnicas de diseño de analizadores sintácticos ascendentes.

Descripción:
El alumno asistirá en clase para adquirir los conocimientos teóricos. Addicionalement tendrá que consolidar estos conceptos con su estudio personal y con la resolución de los problemas propuestos a clase.

Objetivos específicos:
3

Dedicación: 5h
Grupo grande/Teoría: 2h
Aprendizaje autónomo: 3h

Aprendizaje sobre la construcción de árboles de sintaxis abstractos.

Descripción:
El alumno asistirá en clase para adquirir los conocimientos teóricos. Además tendrá que consolidar estos conceptos con su estudio personal y la resolución de los problemas propuestos a clase.

Objetivos específicos:
3, 4

Dedicación: 5h
Grupo grande/Teoría: 2h
Aprendizaje autónomo: 3h

Aprendizaje de teoría y técnicas de análisis semántico

Descripción:
El alumno asistirá en clase para adquirir los conocimientos teóricos. Adicionalmente tendrá que consolidar estos conceptos con su estudio personal y la resolución de los problemas propuestos a clase.

Objetivos específicos:
4

Dedicación: 7h
Grupo grande/Teoría: 4h
Aprendizaje autónomo: 3h

Aprendizaje sobre el diseño de intérpretes y máquinas virtuales

Descripción:
El alumno asistirá en clase para adquirir los conocimientos sobre el tema. Addicionalement tendrá que consolidar estos conceptos con su estudio personal y la resolución de los problemas propuestos a clase. Los conocimientos adquiridos a clase serán posteriormente utilizados en el proyecto del curso.

Objetivos específicos:
5

Dedicación: 6h
Grupo grande/Teoría: 2h
Aprendizaje autónomo: 4h

Aprendizaje sobre la teoría y técnicas de generación de código

Descripción:
El alumno asistirá en clase para adquirir los conocimientos sobre el tema. Adicionalmente tendrá que consolidar estos conceptos con su estudio personal y la resolución de los problemas propuestos a clase.

Objetivos específicos:
6

Dedicación: 10h
Grupo grande/Teoría: 4h
Aprendizaje autónomo: 6h

Aprendizaje sobre la teoría y técnicas de optimización de código

Descripción:
El alumno asistirá en clase para adquirir los conocimientos del tema. Addicionalement tendrá que consolidar estos conceptos con su estudio personal y la resolución de los problemas propuestos a clase.

Objetivos específicos:
7

Dedicación: 10h
Grupo grande/Teoría: 4h
Aprendizaje autónomo: 6h

Introducción a la herramienta ANTLR: diseño de un evaluador de expresiones.

Descripción:
El alumno realizará una modificación del ejemplo propuesto por el profesor a la clase de laboratorio. Durante la clase, utilizará la herramienta ANTLR, que será la misma con la que se realizará el proyecto del curso.

Objetivos específicos:
3, 9

Competencias relacionadas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 3h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 1h

Ampliación de un intérprete de un lenguaje de programación.

Descripción:
El alumno tendrá que realizar la modificación del intérprete propuesto por el profesor durante las horas de laboratorio y de estudio personal. Finalmente tendrá que construir un juegos de pruebas que validen la correctesa de las modificaciones realizadas.

Objetivos específicos:
1, 2, 3, 5, 9, 10

Competencias relacionadas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 7h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 3h

Diseño de un lenguaje de programación y su analizador sintáctico.

Descripción:
La actividad consistirá al diseñar un lenguaje de programación sencillo orientado a un tipo específico de aplicaciones. En esta actividad se permitirá que el propio estudiante proponga el lenguaje y el ámbito de aplicación. De lo contrario, el estudiante podrá optar para realizar el proyecto propuesto por el profesor de la asignatura. Algunos ejemplos de lenguajes de programación que se podrían proponer: (1) un lenguaje para la creación y visualización de animaciones de objetos sencillos, (2) un lenguaje de programación del control de un Lego-Robot, (3) un lenguaje para generar patrones Escher, (4) un lenguaje para generar partituras de música, (5) un lenguaje para definir y manipular funciones matemáticas, etc. Una vez elegido el lenguaje, el alumno tendrá que realizar el analizador léxico y sintáctico utilitzants las herramientas del curso.

Objetivos específicos:
3, 9

Competencias relacionadas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 20h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 16h

Diseño de la fase de interpretación de un lenguaje de programación.

Descripción:
El alumno tendrá que construir el analizador semántico e intérprete del lenguaje utilizando las herramientas del curso durante las clases de laboratorio y las horas de trabajo personal. Finalmente tendrá que construir unos juegos de prueba y una documentación a nivel de usuario para que el proyecto pueda ser evaluado.

Objetivos específicos:
9, 10

Competencias relacionadas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 33h
Grupo pequeño/Laboratorio: 17h
Aprendizaje autónomo: 16h

SISTEMA DE CALIFICACIÓN

El método tiene como objetivo evaluar los conocimientos tanto teóricos como prácticos del estudiante.

La evaluación consistirá en tres actos evaluativos. El primero de ellos (L1) tiene como objetivo evaluar conocimientos prácticos del curso y consistirá en una prueba de laboratorio donde el estudiante tendrá que hacer pequeñas modificaciones a un compilador existente para añadir nueva funcionalidad. Con esta prueba se evaluará el conocimiento que el estudiante tiene de la estructura de un compilador y su capacidad de implementar nuevos conceptos de lenguajes de programación y expresarlos en una gramática. La evaluación se realizará mediante la validación de la implementación con juegos de pruebas previamente preparados.

El segundo acto evaluativo (L2) se basará en el desarrollo del análisis semántico y generación de código del compilador propuesto en la asignatura. Este trabajo se hará en grupo. El acto evaluativo consistirá en una prueba de laboratorio donde habrá que ejecutar unos juegos de pruebas para validar el compilador y las extensiones propuestas a la misma prueba. En este acto se evaluará la capacidad del estudiante para aplicar los conceptos adquiridos durante el curso y el conocimiento del trabajo realizado en grupo.

Finalmente, el tercer acto consistirá en una prueba escrita donde se evaluarán los conocimientos teóricos impartidos durante el curso.

La nota final del curso (F) se calculará con una suma ponderada de las dos pruebas de laboratorio (L1 y L2) y la prueba de conocimientos teóricos (T):

BIBLIOGRAFÍA

Básica:
- Aho, A.V.; Lam, M.S.; Sethi, R.; Ullman, J.D. Compilers: principles, techniques, and tools. 2nd ed. Addison-Wesley, 2007. ISBN 9780321491695.
- Parr, T. The definitive ANTLR 4 reference. The Pragmatic Programmers, 2012. ISBN 9781934356999.
- Cortadella, J. Compilers: lecture notes. 2015.

Complementaria:
- Appel, A.W.; Ginsburg, M. Modern compiler implementation in C. Cambridge University Press, 2004. ISBN 0521607655.
- Appel, A.W.; Palsberg, J. Modern compiler implementation in Java. 2nd ed. Cambridge University Press, 2002. ISBN 052182060X.
- Arnold, K.; Gosling, J.; Holmes, D. The Java programming language. 4th ed. Addison-Wesley, 2006. ISBN 0321349806.