



| Guía Docente | | | | |
|-----------------------|---|--------------------|-------------------------|----------|
| Datos Identificativos | | | 2014/15 | |
| Asignatura (*) | Compiladores | Código | 614111405 | |
| Titulación | Enxeñeiro en Informática | | | |
| Descriptorios | | | | |
| Ciclo | Período | Curso | Tipo | Créditos |
| 1º e 2º Ciclo | Anual | Cuarto | Troncal | 9 |
| Idioma | CastelánGalego | | | |
| Prerrequisitos | | | | |
| Departamento | Tecnoloxías da Información e as Comunicaci3ns | | | |
| Coordinaci3n | Arcay Varela, Bernardino | Correo electr3nico | bernardino.arcay@udc.es | |
| Profesorado | Arcay Varela, Bernardino | Correo electr3nico | bernardino.arcay@udc.es | |
| Web | lia2.tic.udc.es/compiladores | | | |
| Descrici3n xeral | <p>Compiladores; tradutores e intérpretes; etapas de un compilador; optimizaci3n de c3digo; macroprocesadores. O obxectivo 3 familiarizar 3 alumno co funcionamento dos compiladores, o entorno no que traballan as3 coma algunhas ferramentas software para a construcci3n dos mesmos. Asumir a caracter3stica interdisciplinar da asignatura. Adquirir os coñecementos necesarios para deseñar e implementar as diferentes etapas necesarias para o desenvolvemento dun compilador: an3lise (l3xico, sint3ctico e sem3ntico) e s3ntese (xeraci3n de c3digo intermedio, optimizaci3n de c3digo e xeraci3n de c3digo obxeto).</p> | | | |

| Competencias da titulaci3n | |
|----------------------------|--|
| C3digo | Competencias da titulaci3n |
| A1 | Aprender de maneira aut3noma novos coñecementos e t3cnicas avanzadas axeitadas para a investigaci3n, o deseño e o desenvolvemento de sistemas e servizos inform3ticos. |
| A3 | Concibir e planificar o desenvolvemento de aplicaci3ns inform3ticas complexas ou con requisitos especiais. |
| B2 | Resolver problemas de forma efectiva. |
| B4 | Aprendizaxe aut3noma. |
| B5 | Traballar de forma colaborativa. |
| B9 | Capacidade para tomar decisi3ns. |
| B11 | Razoamento cr3tico. |
| C6 | Valorar criticamente o coñecemento, a tecnolox3a e a informaci3n dispoñible para resolver os problemas cos que deben enfrontarse. |
| C7 | Asumir como profesional e cidad3n a importancia da aprendizaxe ao longo da vida. |

| Resultados da aprendizaxe | | | |
|---|---|-----------------------------|-----------------------------|
| Competencias de materia (Resultados de aprendizaxe) | Competencias da titulaci3n | | |
| | Coñecer os conceptos te3ricos b3sicos nos que se basean os compiladores | A1 A3 | B2 B4 B5 B9 B11 |
| Deseñar e implementar cada unha das fases precisas para a implementaci3n dun compilador | A1 A3 | B2 B4 B5 B9 B11 | C6 C7 |



| | | | |
|---|----|-----|----|
| Coñecer as distintas ferramentas dispoñibles para a implementación de compiladores e manexar algunhas das máis habituais. | A1 | B2 | C6 |
| | A3 | B4 | C7 |
| | | B5 | |
| | | B9 | |
| | | B11 | |

| Contidos | |
|--|--|
| Temas | Subtemas |
| Tema I. Introducción | 1.1 Estructura de un compilador. 1.2 Ejemplo de las fases de un compilador. |
| Tema II. Lenguajes y Gramáticas | 2.1 Notación de Chomsky. 2.2 Clasificación de Chomsky. 2.3 Gramáticas de contexto libre (GCL). 2.4 Diagramas de Conway. 2.5 Reglas BNF. 2.6 Problemas en las GCL. 2.7 Simplificación de gramáticas. 2.8 Gramática limpia. 2.9 Forma normal de Chomsky (FNC). 2.10 Resumen. 2.11 Ejercicios. |
| Tema III. Análisis Léxico | 3.1 Tipos de máquinas reconocedoras o autómatas. 3.2 Autómatas Finitos. 3.3 Conversión de una Gramática Regular en un Autómata finito. 3.4 Expresión regular. 3.5 Algoritmo de Thompson. 3.6 Transformación de un AFND-lambda en un AFD. 3.7 Traductores finitos (TF). 3.8 Implementación de autómatas. 3.8.1 Tabla compacta. 3.8.2 Autómata programado. 3.9 Ejemplo. Scanner para números reales sin signo en Pascal. 3.10 Acciones semánticas. 3.11 Generador LEX. |
| Tema IV. Análisis sintáctico (Parsing) | 4.1 Máquinas teóricas, mecanismos con retroceso 4.1.1 Autómatas con pila (AP). 4.1.2 Esquemas de traducción (EDT). 4.1.3 Traductores con pila (TP). 4.2 Algoritmos sin retroceso. 4.2.1 Análisis sintáctico ascendente por precedencia simple. 4.2.2 Análisis sintáctico ascendente por precedencia de operadores. 4.2.3 Analizadores descendentes LL(K). 4.2.4 Analizadores ascendentes LR(k). 4.2.5 Generador de analizadores sintácticos YACC. |
| Tema V. Análisis semántico | 5.1 Definiciones dirigidas por la sintáxis. 5.2 Esquema de traducción. 5.3 Comprobaciones en tiempo de compilación. |



| | |
|-------------------------------------|---|
| Tema VI. Generación de código | 6.1 Lenguajes intermedios. 6.2 Generación de código intermedio. 6.3 Generación de código desde lenguaje intermedio. |
| Tema VII. Optimización de código | 7.1 Algoritmo de Nakata. 7.2 Un ejemplo de optimización manual. 7.3 Lazos en los grafos de flujo. 7.4 Análisis global del flujo de datos. 7.5 Solución iterativa de las ecuaciones de flujo de datos. |
| Tema VIII. Errores | 8.1 Tipos de errores. 8.2 Recuperación de errores léxico-gráficos. |
| Tema IX. Intérpretes y complementos | 9.1 Estructura de un intérprete actual. 9.2 Arquitectura neutral de java. |

| Planificación | | | |
|--------------------------|-------------------|---|--------------|
| Metodoloxías / probas | Horas presenciais | Horas non presenciais / traballo autónomo | Horas totais |
| Sesión maxistral | 40 | 80 | 120 |
| Proba obxectiva | 3 | 3 | 6 |
| Prácticas de laboratorio | 15 | 60 | 75 |
| Atención personalizada | 24 | 0 | 24 |

*Os datos que aparecen na táboa de planificación son de carácter orientativo, considerando a heteroxeneidade do alumnado

| Metodoloxías | |
|--------------------------|--|
| Metodoloxías | Descrición |
| Sesión maxistral | Sesións maxistras coincidentes entre a asignatura do Master e en Enxeñería Informática. O axuste de horas non é exacto pero, de cara a facilitar o desenvolvemento do proxecto, no segundo cuatrimestre dispoñeráse dunha redución de horas maxistras a tal efecto, igualando o esquema de docencia entre ambas. |
| Proba obxectiva | Faranse dous exames, un en febreiro/marzo sobre o contido do primeiro cuatrimestre e outro en xuño do contido do segundo cuatrimestre. Neste último tamén haberá unha segunda oportunidade para recuperar, se é necesario, a materia do primeiro cuatrimestre. |
| Prácticas de laboratorio | Realizaranse varias pequenas prácticas hasta xaneiro e a partir de ahí haberá que desenvolver un proxecto que involucre a maior parte das fases dun compilador. Este proxecto será proposto polo estudante. |

| Atención personalizada | |
|--------------------------|---|
| Metodoloxías | Descrición |
| Prácticas de laboratorio | Especialmente no caso do proxecto a desenvolver polo alumno, realizarase un seguimento semanal dos traballos. |

| Avaliación | | |
|--------------------------|---|---------------|
| Metodoloxías | Descrición | Cualificación |
| Prácticas de laboratorio | Proxecto a propoñer e desenvolver polo alumno durante o segundo cuatrimestre | 30 |
| Proba obxectiva | Duas probas (febreiro/marzo e xuño) que aportarán cada unha un 50% da nota neste apartado (será condición imprescindible ter presentadas as pequenas prácticas iniciais). | 70 |
| Outros | | |

| Observacións avaliación |
|--|
| En calquera caso, é preciso aprobar as dúas parte. No caso contrario, a máxima nota que se poderá acadar é un 4.5. |



Fontes de información

Bibliografía básica

Bibliografía complementaria

Recomendacións

Materias que se recomenda ter cursado previamente

Materias que se recomenda cursar simultaneamente

Deseño de Sistemas Operativos/614407114

Enxeñería do Software/614407115

Materias que continúan o temario

Linguaxes Naturais/614407220

Observacións

A asignatura troncal de Enxeñería Informática e Enxeñería Técnica en Informática de Sistemas "Teoría de autómatas e linguaxes formais" é de gran utilidade para a comprensión da asignatura de Compiladores.

(*A Guía docente é o documento onde se visualiza a proposta académica da UDC. Este documento é público e non se pode modificar, salvo casos excepcionais baixo a revisión do órgano competente dacordo coa normativa vixente que establece o proceso de elaboración de guías