Programa de la Asignatura:

Compiladores e Intérpretes

Código: 767

Carrera: ***Ingeniería en Computación*** Plan: ***2008*** Carácter: ***Obligatoria***

Unidad Académica: ***Secretaría Académica*** Curso: ***Cuarto Año – Primer cuatrimestre***

Departamento: ***Ingeniería*** Carga horaria total: ***60 hs.*** Carga horaria semanal: ***4 hs.***

Formación Experimental: ***00 %*** Formación teórica: ***50 %*** Formación práctica: ***50 %***

Materias Correlativas Obligatorias

- ***Lenguajes de Programación IV (cód. 763)***
- ----------
- ----------

Cuerpo Docente

Henrión, Guillermo

Índice

AÑO ACADÉMICO 2013 ÚLTIMA REVISIÓN 31/05/2013

Firma Docente Firma Coordinador

1. **FUNDAMENTACION**

Un compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar. Usualmente el segundo lenguaje es lenguaje de máquina, pero también puede ser un código intermedio, o simplemente texto. Este proceso de traducción se conoce como compilación.

Por otro lado, un paradigma de programación provee (y determina) la visión y métodos de un programador en la construcción de un programa o subprograma. Diferentes paradigmas resultan en diferentes estilos de programación y en diferentes formas de pensar la solución de problemas.

La asignatura presenta al alumno con las nociones fundamentales para la construcción de un compilador y brinda las herramientas necesarias para su construcción. Así mismo brinda las herramientas necesarias para comprender los paradigmas de programación y su adecuación de acuerdo al problema planteado.

2. **ENCUADRE Y ARTICULACIÓN DE LA ASIGNATURA**

Articulación Horizontal

Esta asignatura se articula horizontalmente con otras materias que requieren también de conocimientos avanzados de programación, de manera de brindar al alumno una visión amplia de la programación actual.

Articulación Vertical

Esta asignatura se articula verticalmente con Lenguajes de programación IV, que brinda las herramientas de desarrollo de software avanzado, necesarias para comprender los conceptos de compiladores e intérpretes.

3. **OBJETIVOS**

Objetivo General

Introducir al alumno en las nociones teóricas y prácticas de la construcción de compiladores, así como presentar los distintos paradigmas de programación, de manera que el alumno cuente con mejores herramientas al momento de evaluar cómo dar una solución computacional a un problema dado.

Objetivos Específicos

Luego de cursar esta asignatura el alumno deberá dominar los siguientes temas:

- Características de los compiladores e intérpretes
- Comprensión de la teoría de lenguajes y autómatas
- Aspectos de desarrollo de un compilador

4. **CONTENIDOS MÍNIMOS**

Análisis de compiladores. Análisis lexicográfico. Análisis sintáctico. Análisis semántico. Síntesis. Errores. Corrección de código. Intérpretes. Álgebra de clases. Grafos y lenguajes. Lenguajes y autómatas. Autómata de pila y máquina de Turing. La función de la estructura en la programación. Elementos de un lenguaje de programación. Programación con asignaciones. Activación de procedimientos. Encapsula-miento de datos. Herencia. Programación funcional. Programación lógica. Programación concurrente. Estructura sintáctica. Intérpretes de definiciones. Tipos estáticos y cálculo lambda.

5. **PROGRAMA ANALÍTICO**

Unidad 1: Fundamentos de los compiladores.

Introducción a los compiladores. Introducción a los programas LEX - YACC.

Unidad 2: Gramáticas formales.

Jerarquía de Chomsky. Gramáticas regulares y libres de contexto. Su relación con los compiladores.

Unidad 3: Autómatas.

Autómata finito. Autómata de pila. Su relación con las gramáticas.

Unidad 4: Analizadores léxicos, sintácticos y semánticos

Aplicaciones de las gramáticas formales en el análisis léxico y sintáctico. Chequeo de tipos y generación de código.

Unidad 5: Algoritmos analizadores.

Analizadores LL, LR, LALR.

Unidad 6: LEX-YACC

Conceptos avanzados de estos programas para la creación de compiladores.

Unidad 7: Complejidad computacional.

Problemas P y NP. Orden de complejidad.

Unidad 8: Paradigmas de programación: funcional

Cálculo lambda. LISP.

Unidad 9: Paradigmas de programación: lógico

Algoritmo de resolución. PROLOG.

6. BIBLIOGRAFÍA BÁSICA

- Compilers: Principles, Techniques, and Tools – Alfred V Aho
- Lenguajes de Programación Principios y Paradigmas - A. Tucker

7. BIBLIOGRAFÍA DE CONSULTA

- Computational complexity - Christos H. Papadimitriou

8. METODOLOGÍA DEL APRENDIZAJE

8.a DESARROLLO DE LA ASIGNATURA

En la primer parte de la asignatura el hilo conductor de la misma es la construcción de un prototipo de compilador utilizando las herramientas LEX-YACC. Para eso se le brindará al alumno del conocimiento teórico y práctico necesario. En la segunda parte se expondrán temas referidos a los distintos paradigmas de programación, realizando prácticas en cada uno de ellos, de manera de poder evaluar qué paradigma es más adecuado utiliza de acuerdo al problema planteado.

8.b DINÁMICA DEL DICTADO DE LAS CLASES

La clase de dividirá en dos partes: una teórica, en donde se explicarán los conceptos teóricos correspondientes, y otra práctica en donde se aplicarán y programarán los temas vistos en la primera parte.

8.c TRABAJOS PRÁCTICOS

Los prácticos realizados en la segunda parte de cada clase serán completados como tarea y entregados completos a las dos clases siguientes de comenzados. Para ello el alumno dispondrá del conocimiento y de todo el software necesario para poder realizarlo.

9. METODOLOGÍA DE EVALUACIÓN

9.a NORMAS DE EVALUACIÓN.

- El criterio es que la evaluación del alumno es permanente.
- Se tomarán dos exámenes parciales teórico/prácticos pudiendo acceder a un recuperatorio.
- Las notas de los parciales representan los resultados de la evaluación teórico/práctica.
- Los exámenes parciales y sus recuperatorios pueden ser orales o escritos.

9.b RÉGIMEN DE APROBACIÓN DE LA MATERIA.

- Para la aprobación de la materia los alumnos deberán tener los dos parciales aprobados, teniendo la posibilidad de recuperar cada UNO de ellos en dos oportunidades adicionales, en la fecha acordada con los docentes.
- Además los alumnos deberán aprobar los trabajos prácticos, como condición para la aprobación de la materia.
- Los alumnos que obtengan una nota inferior a cuatro puntos se les asignará la nota insuficiente y deberán recursar la materia.

10. PLANIFICACIÓN

CALENDARIO DE CLASES Y EVALUACIONES	
Semana 1	Unidad 1
Semana 2	Unidad 2
Semana 3	Unidad 3
Semana 4	Unidad 4 parte I
Semana 5	Unidad 4 parte II
Semana 6	Consultas y repaso
Semana 7	Primer parcial
Semana 8	Unidad 4 parte III
Semana 9	Unidad 5
Semana 10	Unidad 6
Semana 11	Unidad 7
Semana 12	Unidad 8
Semana 13	Unidad 9
Semana 14	Consultas y repaso
Semana 15	Segundo Parcial
Semana 16	Recuperatorios
Del al de	FINAL

Información de Versiones	
Nombre del Documento:	Ficha Académica de la asignatura Compiladores e Intérpretes
Nombre del Archivo	Compiladores e Intérpretes – Plan 2008
Documento origen:	Compiladores _e_ Intérpretes – Plan 2008
Elaborado por:	Guillermo Henrión
Revisado por:	Aníbal Romandetta
Aprobado por:	Alejandro Oliveros
Fecha de Elaboración:	31/05/2013
Fecha de Revisión:	04-06-2013
Fecha de aprobación	
Versión:	