

# FACULTAD DE INGENIERÍA

## ÁREA DE COMPUTACIÓN E INFORMÁTICA



**Nombre de la materia :** COMPILADORES E  
INTÉRPRETES B

**Clave de la materia:**

**Clave Facultad:** 2407

**Clave U.A.S.L.P.:**

**Clave CACEI:** CI

**Nivel del Plan de Estudios:** IC: 8 **No. de créditos:** 10

**Horas/Clase/Semana:** 5 **Horas totales/Semestre:** 80

**Horas/Práctica (y/o Laboratorio):** 0

**Prácticas complementarias:**

**Trabajo extra-clase Horas/Semana:** 5

**Carrera/Tipo de materia:** Propia de la Carrera Obligatoria

**No. de créditos aprobados:**

**Fecha última de Revisión Curricular:** 27/01/ 2016

**Materia y clave de la materia requisito:**

COMPILADORES E INTÉRPRETES A (2406)

### PROPÓSITO DEL CURSO

Dado que las computadoras sólo pueden interpretar lenguaje de máquina (binario), no hay lenguaje de programación de alto nivel que no contenga un traductor, sea compilador o intérprete. Actualmente no es común que se cree o se le dé mantenimiento a alguno. Sin embargo, esto no indica que no haya desarrollos con base

en metodologías de traductores enfocadas a la resolución de problemas particulares de software. La mayoría de implementaciones se hacen con lenguajes específicos (menos extensos) y se utilizan en la generación de herramientas para soporte de software de sistemas.

### OBJETIVO DEL CURSO

Conocer los conceptos fundamentales y técnicas utilizadas en el diseño de traductores para sistemas de software de uso general (lenguajes de programación), con el propósito de hacer aplicaciones enfocadas a la

resolución de problemas particulares de software de sistemas. Desarrollar un compilador y un intérprete para un lenguaje de propósito específico.

### CONTENIDO TEMÁTICO

#### 1.- Traducción dirigida por la sintaxis

Tiempo estimado: 26 horas

**Objetivo:** el alumno comprenderá el concepto de Traducción Dirigida por la Sintaxis, y las diferentes formas de evaluaciones de Definiciones Dirigidas por la Sintaxis y Esquemas de Traducción.

##### 1.1.- Definiciones dirigidas por la sintaxis.

##### 1.2.- Grafos de dependencias y orden de evaluación.

##### 1.3.- Aplicaciones a la construcción de árboles sintácticos

##### 1.4.- Esquemas de traducción dirigidos por la sintaxis.

##### 1.5.- Implementación de definiciones dirigidas por la sintaxis.

#### 2.- Generación de código intermedio

Tiempo estimado: 22 horas

**Objetivo:** el alumno comprenderá el concepto de la generación de código intermedio, comprobación de tipos y flujo de control.

##### 2.1.- Representaciones intermedias.

##### 2.2.- Código de tres direcciones.

##### 2.3.- Tipos y declaraciones.

##### 2.4.- Comprobación de tipos.

##### 2.5.- Flujo de control.

##### 2.6.- Relleno de retroceso.

##### 2.7.- Código intermedio para procedimientos.

#### 3.- Construcción de intérpretes

Tiempo estimado: 12 horas

**Objetivo:** el alumno comprenderá los principales mecanismos para la construcción de intérpretes

##### 3.1.- Generalidades en la construcción de intérpretes

##### 3.2.- Intérprete dirigido por la sintaxis

##### 3.3.- Intérprete basado en árbol

##### 3.4.- Intérprete bytecode basado en pila

##### 3.5.- Intérprete bytecode basado en registros

#### 4.- Generación de código.

Tiempo estimado: 20 horas

**Objetivo:** el alumno comprenderá los conceptos de la generación de código objeto.

4.1.- Cuestiones sobre el diseño de un generador de código.

4.2.- El lenguaje destino.

4.3.- Direcciones en el código destino.

4.4.- Bloques básicos y grafos de flujo.

4.5.- Optimización de los bloques básicos.

4.6.- Generador de código simple.

## METODOLOGÍA

Exposición de los temas, trabajos complementarios, análisis y discusión sobre la

implementación de métodos, conceptualización e implementación de un traductor por equipos.

## EVALUACIÓN

El examen parcial representa el 80% de la calificación y el 20% restante corresponde a tareas. Los exámenes serán teórico-prácticos dependiendo de los temas que sean evaluados. De acuerdo con el reglamento de exámenes, la calificación final es el promedio de los cinco exámenes parciales. Para presentar el último examen parcial,

examen extraordinario, examen a título y/o regularización, es requisito completar la implementación del traductor en todas las fases requeridas.

## BIBLIOGRAFÍA

### Bibliografía Básica

Compiladores : principios, técnicas y herramientas  
Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, M. Lam  
Pearson Educación, México, 2008.

The theory and practice of compiler writing  
Jean-Paul Tremblay, Paul G. Sorenson.  
McGraw-Hill, 1985

### Bibliografía Complementaria

Construcción de Compiladores: Principios y práctica  
Kenneth C. Louden  
Thomson, 2004

Advanced compiler design and implementation  
Steven S. Muchnick.  
Morgan Kaufmann, 1997.

Engineering a Compiler  
Keith Cooper, Linda Torczon  
Morgan Kaufmann, 2nd Edition, 2011

The compiler desing handbook : optimizations and  
machine code generation  
Y. N. Srikant, P. Shankar  
CRC, 2003.

Language Implementation Patterns  
Terrence Parr  
Pragmatic Bookshelf, 2010

Programming Language Prragmatics  
Michael L. Scott  
Morgan Kaufmann, 3rd edition, 2009

Introducción a la teoría de autómatas, lenguajes y  
computación  
John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman  
2a Edición, 2002  
Addison Wesley

Teoría de autómatas y lenguajes formales  
Dean Kelley  
Prentice-Hall, 1995

Crafting a compiler with C  
Charles N. Fischer, Richard J. LeBlanc  
Benjamin Cummings , 1991