



UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

## SÍLABO

### INFORMACIÓN GENERAL

ASIGNATURA	:	COMPILADORES
CÓDIGO	:	CC461
CRÉDITOS	:	04 (CUATRO)
PRE-REQUISITO	:	CC302 LENGUAJE PARALELO DISTRIBUIDO ORIENTADO A OBJETOS CC342 TEORIA DE LA COMPUTACION
CONDICIÓN	:	OBLIGATORIO
HORAS POR SEMANA	:	06 (TEORÍA: 03, LABORATORIO: 03)
SISTEMA DE EVALUACIÓN	:	G

### OBJETIVO

Que el alumno conozca y comprenda los conceptos y principios fundamentales de la teoría de compilación para realizar la construcción de un compilador. Conocer las técnicas básicas empleadas durante el proceso de generación intermedio, optimización y generación de código. Aprender a implementar pequeños compiladores.

### PROGRAMA ANALÍTICO

#### 1. Visión General de los Lenguajes de Programación

I: Historia de los lenguajes de programación. II: Breve revisión de los paradigmas de programación. III: Lenguajes procedurales. IV: Lenguajes orientados a objetos. V: Lenguajes funcionales. VI: Lenguajes declarativos y no algorítmicos. VII: Lenguajes de *scripts*. VIII: Los efectos de la escalabilidad en las metodologías de programación.

#### 2. Introducción a la Traducción de Lenguajes

I: Comparación entre intérpretes y compiladores. II: Fases de traducción del lenguaje (análisis léxico, análisis sintáctico, generación de código, optimización). III: Aspectos de traducción dependientes e independientes de la máquina.

#### 3. Sistemas de Traducción del Lenguaje

I: Aplicación de las expresiones regulares en analizadores léxicos. II: Análisis sintáctico (sintaxis concreta y abstracta, árboles de sintaxis abstracta). III: Aplicación de las gramáticas libres de

contexto en un parseo dirigido por tablas o recursivo descendente. IV: Administración de tablas de símbolos. V: Generación de código por seguimiento de un árbol. VI: Operaciones específicas de la arquitectura: selección de instrucciones y asignación de registros. VII: Técnicas de optimización. VIII: El uso de herramientas como soporte en el proceso de traducción y las ventajas de éste. IX: Librerías de programas y compilación separada. X: Construcción de herramientas dirigidas por la sintaxis.

#### **4. Paralelismo a nivel de instrucción**

I: Arquitectura de procesadores. II: Restricciones de programación de código. III: Programación de bloques básicos. IV: Programación de código global. V: Canalización por software.

#### **5. Optimización para el paralelismo y la localidad**

I: Conceptos básicos. II: Multiplicación de matrices. III: Espacios de iteraciones. IV: Índices de arreglos afines. V: Análisis de dependencias de datos de arreglos. VI: Búsqueda del paralelismo sin sincronización. VII: Sincronización entre ciclos paralelos.

### **BIBLIOGRAFÍA**

1. Aho, A. Compiladores Principios, técnicas y herramientas. Addison Wesley. 1990.
2. Aho, A., Lam, M., Sethi, R., and Ullman, J. D. Compiladores. Principios, técnicas y herramientas. Addison Wesley, 2nd edition. ISBN:10-970-26-1133-4. 2008.
3. A.Lemone, K. Fundamentos de Compiladores. CECSA-Mexico. 1996.
4. Appel, A. W. Modern compiler implementation in Java. Cambridge University Press, 2.a edición. 2002.
5. Loudon, K. C. Construcción de Compiladores Principios y Práctica. Publicado por Thomson. 2004.
6. Loudon, K. C. Lenguajes de Programación. Publicado por Thomson. 2004.
7. Pratt, T. W. and V.Zelkowitz, M. Lenguajes de Programación Diseño e Implementación. Prentice-Hall Hispanoamericana S.A. 1998.
8. Teufel, B. and Schmidt, S. Fundamentos de Compiladores. Addison Wesley Iberoamericana. 1998