



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Estudios Superiores Aragón
Plan de Estudios



Ingeniería en Computación
Compiladores

Clave	Semestre	Créditos	Área	
	6	8.0	Software de Base	
Modalidad	Curso		Tipo	Teórico
Carácter	Obligatorio			
Horas				
Semana			Semestre	
Teóricas	4.0		Teóricas	64.0
Prácticas	0.0		Prácticas	0.0
Total	4.0		Total	64.0

Seriación indicativa

Asignatura antecedente	Lenguajes Formales y Autómatas
Asignatura subsecuente	Inteligencia Artificial

Objetivo general: Conocer las herramientas y técnicas que permitan diseñar y construir compiladores.

Índice temático

No.	Tema	Horas Semestre	
		Teóricas	Prácticas
1	INTRODUCCIÓN	3.0	0.0
2	ANÁLISIS LEXICOGRÁFICO	5.0	0.0
3	ANÁLISIS SINTÁCTICO	10.0	0.0
4	ANÁLISIS SEMÁNTICO	8.0	0.0
5	TABLA DE SÍMBOLOS	4.0	0.0
6	GENERACIÓN DE CÓDIGO INTERMEDIO	6.0	0.0
7	GENERACIÓN DE CÓDIGO	4.0	0.0
8	OPTIMIZACIÓN DE CÓDIGO	8.0	0.0
9	INTÉRPRETES	3.0	0.0
10	MÁQUINAS VIRTUALES	6.0	0.0
11	PROYECTO FINAL	7.0	0.0
Total		64.0	0.0
Suma total de horas		64.0	



CONSEJO ACADÉMICO DEL ÁREA DE LAS
 CIENCIAS FÍSICO MATEMÁTICAS
 Y DE LAS INGENIERÍAS

Contenido Temático	
1. INTRODUCCIÓN	
Objetivo: Conocer el concepto y funciones de un compilador.	
1.1	Lenguajes imperativos y declarativos.
1.2	Compiladores, ensambladores e intérpretes y máquinas de ejecución virtual.
1.3	Estructura de un compilador.
2. ANALISIS LEXICOGRÁFICO	
Objetivo: Conocer los elementos del análisis léxico y realizar la separación de tokens, tanto manualmente como utilizando programas propios o herramientas.	
2.1	Expresiones regulares.
2.2	Autómatas de estados finitos.
2.3	Definición de tokens para el lenguaje C++ y Java.
2.4	Uso y manejo de LEX.
3. ANÁLISIS SINTÁCTICO	
Objetivo: Conocer los elementos del análisis sintáctico.	
3.1	Principios del análisis sintáctico.
3.2	Gramáticas ambiguas.
3.3	Gramática libre de contexto: BNF. (Backus-Naur Form).
3.4	Problemas típicos en la definición de gramáticas.
3.5	Gramática de C++ y de Java.
3.6	Detección y recuperación de errores.
3.7	Uso y manejo de YACC (Yet Another Compiler-Compiler).
4. ANÁLISIS SEMÁNTICO	
Objetivo: Comprender la etapa de análisis semántico.	
4.1	Análisis semántico.
4.2	Verificación de tipos.
4.3	Verificación de parámetros.
4.4	Verificación de la compatibilidad de las expresiones con la variable a la que se asigna el resultado.
5. TABLA DE SÍMBOLOS	
Objetivo: Implementar tablas de símbolos.	
5.1	Tabla de símbolos.
5.2	Tabla de tipos.
5.3	Manejo de ámbitos.
5.4	Definiendo nuevos tipos de datos.
5.5	Implementación de la tabla de símbolos.
6. GENERACIÓN DE CÓDIGO INTERMEDIO	
Objetivo: Entender la generación de código intermedio.	
6.1	Árbol comentado.
6.2	ECMA-335. Estándares internacionales.
6.3	Proceso de traducción.

7. GENERACIÓN DE CÓDIGO

Objetivo: Conocer e implementar la etapa de generación de código para un procesador hipotético o real.

- 7.1 Traducción a partir del código intermedio.
- 7.2 Filosofías RISC y CISC.
- 7.3 Dependencia del código al procesador destino.

8. OPTIMIZACIÓN DE CÓDIGO

Objetivo: Revisar las diferentes técnicas de optimización de código.

- 8.1 Tipos de optimización.
- 8.2 Movimiento de código.

9. INTÉRPRETES

Objetivo: Entender e implementar intérpretes.

- 9.1 Intérpretes.
- 9.2 Lenguajes interpretados vs lenguajes compilados.
- 9.3 Mejorando la eficiencia de lenguajes interpretados.

10. MÁQUINAS VIRTUALES

Objetivo: Comprender el concepto de máquina virtual.

- 10.1 Máquinas virtuales.
- 10.2 La máquina virtual de Java.
- 10.3 Ventajas de las máquinas virtuales.

11. PROYECTO FINAL

Objetivo: Construir un compilador para una gramática sencilla.

- 11.1 Definiendo la gramática.
- 11.2 Implementación de las etapas de análisis.
- 11.3 Implementación de las etapas de generación y optimización de código.
- 11.4 Uniendo todas las piezas.

Estrategias didácticas		Evaluación del aprendizaje		Recursos	
Exposición	(X)	Exámenes parciales	(X)	Aula interactiva	()
Trabajo en equipo	(X)	Examen final	(X)	Computadora	(X)
Lecturas	(X)	Trabajos y tareas	(X)	Plataforma tecnológica	(X)
Trabajo de investigación	(X)	Presentación de tema	(X)	Proyector o Pantalla LCD	(X)
Prácticas (taller o laboratorio)	()	Participación en clase	(X)	Internet	(X)
Prácticas de campo	()	Asistencia	(X)		
Aprendizaje por proyectos	(X)	Rúbricas	()		
Aprendizaje basado en problemas	()	Portafolios	()		
Casos de enseñanza	()	Listas de cotejo	()		
Otras (especificar)		Otras (especificar)		Otros (especificar)	
Aprendizaje colaborativo					

Perfil profesiográfico	
Título o grado	<ul style="list-style-type: none"> • Poseer un título a nivel licenciatura en Ingeniería en Computación, Ciencias de la Computación, Matemáticas Aplicadas a la Computación o carreras cuyo perfil sea afín al área de Software de Base.
Experiencia docente	<ul style="list-style-type: none"> • Poseer conocimientos y experiencia profesional relacionados con los contenidos de la asignación a impartir. • Tener la vocación para la docencia y una actitud permanentemente educativa a fin de formar íntegramente al alumno: <ul style="list-style-type: none"> ○ Para aplicar recursos didácticos. ○ Para motivar al alumno. ○ Para evaluar el aprendizaje del alumno, con equidad y objetividad.
Otra característica	<ul style="list-style-type: none"> • Poseer conocimientos y experiencia pedagógica referentes al proceso de enseñanza-aprendizaje. • Tener disposición para su formación y actualización, tanto en los conocimientos de su área profesional, como en las pedagógicas. • Identificarse con los objetivos educativos de la institución y hacerlos propios. • Tener disposición para ejercer su función docente con ética profesional: <ul style="list-style-type: none"> ○ Para observar una conducta ejemplar fuera y dentro del aula. ○ Para asistir con puntualidad y constancia a sus cursos. ○ Para cumplir con los programas vigentes de sus asignaturas.

Bibliografía básica	Temas para los que se recomienda
Aho, A. (2008). <i>Compiladores: principios, técnicas y herramientas</i> . México: Addison Wesley.	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11
Kakde, O. G. (2005). <i>Comprehensive Compiler Design</i> . India: Laxmi Publications.	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11
Lemone, K. (1996). <i>Fundamentos de compiladores</i> . México: Cecsa.	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11
Martínez, F. J. (2014). <i>Teoría, diseño e implementación de compiladores de lenguajes</i> . España: Ra-Ma Editorial.	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11
Mohan, H. S. (2014). <i>Compiler Design</i> . U. K.: Alpha Science International.	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11



Ruíz, C. J. (2010). <i>Compiladores. Teoría e implementación.</i> España: Alfaomega.	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11
--	------------------------------------

Bibliografía complementaria	Temas para los que se recomienda
Kernighan, B. (1991). <i>The C programming language.</i> EU: Prentice Hall.	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11
Raya, C. J. L. (2010). <i>Guía de campo máquinas virtuales.</i> España: Alfaomega.	10