

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN

FACULTAD DE INGENIERIA

ESCUELA PROFESIONAL DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS

S I L A B O

COMPILADORES Y TEORÍA DE LENGUAJES

I. DATOS GENERALES

1.1. Facultad	: Ingeniería
1.2. Escuela Académica Profesional	: Ingeniería en Informática y Sistemas
1.3. Asignatura	: Compiladores y teoría de lenguajes
1.4. Código	: IIS.0363
1.5. Horas Semanales	: 05 Horas [T: 03 Hrs. L: 02 Hrs]
1.6. Año Académico	: 2016-II
1.7. Régimen	: Semestral
1.8. Área Curricular	: Formación especializada
1.9. Duración del Curso	: 17 Semanas
1.10. Docente	: Msc. Ana Silvia Cori Morón

II. FUNDAMENTOS DE LA ASIGNATURA

2.1. Aporte de la asignatura al perfil profesional

El curso de compiladores corresponde al área de informática. La asignatura presenta un carácter teórico – práctico que pretende dar a conocer a los estudiantes de Ingeniería en Informática y Sistemas el estudio de las fases de todo compilador, estas fases se enseñan para comprenderlas y aplicarlas mediante la creación de Software que cumpla tales funciones.

2.2. Sumilla

El desarrollo de la asignatura comprende Teoría de Lenguajes y la construcción de compiladores por fases, conceptos, definiciones formales, Técnicas, Analizador Léxico, Analizador Sintáctico, Tipos, contexto de desarrollo, Análisis Semántico, Tabla de Símbolos, Generación de Código, tratamiento y recuperación de errores.

III. OBJETIVOS DE LA ASIGNATURA

- El principal objetivo es el de formar al alumno en el diseño de procesadores. Se pretende que el alumno comprenda los principales métodos de diseño de las distintas fases que componen el desarrollo de un compilador. Todo ello mediante una actividad eminentemente práctica que consta de numerosos ejemplos que completan la fase teórica y el desarrollo de todas las fases, por parte del alumno, de un compilador sencillo.

- Interpretar los diferentes métodos y algoritmos del cual se basa la construcción de un compilador, valorando el uso fundamental de la programación como fuente de información sustentable en el diseño de este.

IV. CONTENIDO

UNIDAD DE APRENDIZAJE I: LENGUAJES DE PROGRAMACION Y TRADUCCION DE LENGUAJES			
Contenidos conceptuales	Contenidos procedimentales	Semana	Contenido actitudinal
Historia de los lenguajes de programación. Breve revisión de los paradigmas de programación.	Reconoce el concepto de lenguajes de programación.	01	Desarrolla su capacidad de síntesis y análisis. Participa activamente con responsabilidad y respeto
Lenguajes procedurales. Lenguajes orientados a objetos. Lenguajes funcionales. Lenguajes declarativos y no algorítmicos. Lenguajes de scripts.	Realiza el análisis comparativo entre los lenguajes de programación.	02	Exposición dialogada - Trabajo aplicativo en aula.
Los efectos de la escalabilidad en las metodologías de programación. Comparación entre intérpretes y compiladores.	Realiza el análisis comparativo entre intérpretes y compiladores.	03	Trabajo aplicativo en aula (ejercicios). Demostrar el trabajo grupal y respetar la opinión de sus compañeros.
Fases de traducción del lenguaje (análisis léxico, análisis sintáctico, generación de código, optimización). Aspectos de traducción dependientes e independientes de la máquina.	Reconocer las características esenciales de las 3 fases de un compilador.	04	Demostrar la responsabilidad frente al curso, y asistir puntualmente. Pone en práctica en los temas desarrollados en casos prácticos.
Análisis Léxico. Gramáticas Regulares. Especificación de los componentes léxicos (tokens).	Reconocer las técnicas algorítmicas para la construcción de una tabla tokenizada.	05	
Descripción del Analizador Léxico. Autómata Finito Determinista y Acciones Semánticas	Elaborar ejercicios mediante algoritmos que entiendan el computador respecto al lenguaje de programación que deseamos compilar.	06	
Tabla de Símbolos. Función de la Tabla de Símbolos. Estructura de la Tabla de Símbolos. Desarrollo de la Tabla de Símbolos.		07 Y 08	
EXAMEN PARCIAL		09	I UNIDAD

UNIDAD DE APRENDIZAJE II: SISTEMAS DE TRADUCCION DE LENGUAJES			
Contenidos conceptuales	Contenidos procedimentales	Semana	Sesión - Actividades
Análisis Sintáctico. Análisis Sintáctico Ascendente y Descendente. Gramáticas de contexto libre.	Reconocer los diagramas libres de Contexto formulando las reglas a seguir en la construcción de nuestro Lenguaje De Programación.	10	Desarrolla su capacidad de síntesis y análisis. Participa activamente con responsabilidad y respeto
Analizador Sintáctico Descendente Recursivo predictivo.	Construir diagramas respetando la derivación de su gramática y los posibles caminos para crear dicha fase del compilador.	11	Exposición dialogada - Trabajo aplicativo en aula.
Analizador Sintáctico Descendente con tablas (LL).		12	Trabajo aplicativo en aula (ejercicios).
Analizador Sintáctico. Ascendente LR	Diseñar algoritmos que evalúen el análisis por derecha, por izquierda y la diferencia de parseo: Top-down vs. Bottomup.	13	Demostrar el trabajo grupal y respetar la opinión de sus compañeros.
Análisis Semántico. Gramáticas de atributos. Traducción Dirigida por la Sintaxis. Comprobaciones semánticas		14	Demostrar la responsabilidad frente al curso, y asistir puntualmente.
Gestión de Errores. Tipos de errores. Detección y Recuperación de errores. Mensajes de error	Sustentar los algoritmos para Cálculo de atributos, Tabla de símbolos y verificación de análisis de tipos.	15 y 16	Pone en práctica en los temas desarrollados en casos prácticos.
EXAMEN PARCIAL		17	II UNIDAD

V. ESTRATEGIAS METODOLOGICAS

Las clases se realizarán utilizando medios audiovisuales como ayuda para fomentar un mejor entendimiento, así como también se proporcionarán trabajos prácticos y de investigación para completar lo explicado en clase.

MÉTODOS: Inductivo-Deductivo

TÉCNICAS: Exposiciones, trabajos de investigación, conferencias, simulación de soluciones, casos prácticos.

MEDIOS DIDÁCTICOS: Pizarra, multimedia, Ecran, dispositivos, etc. Separatas, direcciones electrónicas, power point.

VI. SISTEMA DE EVALUACIÓN

$$0.25*EP1 + 0.25*EP2 + 0.25*TI + 0.10*PC + 0.15*LAB$$

Donde:

EP1 = Examen parcial unidad 1

EP2 = Examen parcial unidad 2

TI = Promedio Trabajo de Investigación

PC = Prácticas calificadas

LAB = Laboratorio

REQUISITOS DE APROBACION

- Asistencia más de 70% de clases teóricas y prácticas.
- El estudiante que no rinde un examen tendrá Cero (00), salvo justificación con certificado médico de acuerdo a las normas

VII. BIBLIOGRAFÍA

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