



Vicerrectoría Académica
Dirección Curricular y de Docencia
Formato para la Elaboración de Microdiseños de Cursos

1 Identificación del Curso			
1.1 Código	1.2 Nombre del Curso	1.3 Pre-Requisito	1.4 Co-Requisito
011421	Compiladores	Estructura de Datos II	N/A
1.5 No. Créditos	1.6 HAD	1.7 HTI	1.8 HAD:HTI
4	64	128	1:2
1.9 Horas presenciales aula clase	1.10 Horas presenciales laboratorio/Salida campo	1.11 Horas Virtuales Espacios	1.12 Total Horas HAD
Obligatorio	<input checked="" type="checkbox"/>	Optativo	<input type="checkbox"/>
Teórico	<input checked="" type="checkbox"/>	Practico	<input type="checkbox"/>
1.13 Unidad Académica Responsable del Curso			
	Ingeniería de Sistemas		
1.14 Área de Formación			
	Ingeniería Aplicada		
1.15 Componente			No aplica
	Algoritmos y Programación		<input type="checkbox"/>

2 Justificación del Curso
<p>Las asignaturas generan nuevos lenguajes de programación y herramientas para analizarlos en máquinas de cómputo. Esto, con el propósito de suplir las nuevas necesidades de computación del mundo moderno. mejorar la capacidad algorítmica de los estudiantes y afianzar las habilidades de desarrollo de software de calidad. El aprendizaje de la teoría de los lenguajes formales y autómatas fomenta el desarrollo de aplicaciones actuales como los sistemas de reconocimiento y síntesis de voz y los traductores.</p>

3 Competencias por Desarrollar

3.1 Competencias Genéricas

- Capacidad de abstracción, análisis y síntesis.
- Capacidad de aplicar el conocimiento en la práctica.
- Capacidad de organizar y planificar el tiempo.
- Capacidad creativa.
- Capacidad de crítica y autocrítica.
- Capacidad para identificar, plantear y resolver problemas.
- Habilidades de gestión de la información.
- Capacidad de trabajo en equipo.
- Capacidad de trabajar de forma autónoma.
- Capacidad de comunicación oral y escrita.
- Capacidad de aprendizaje.
- Compromiso ético.
- Capacidad de investigación.
- Compromiso con la calidad.
- Capacidad de adaptación a nuevas situaciones.

3.2 Competencias Específicas

- Identificar la teoría de los lenguajes formales para aplicarlos en el diseño de un lenguaje de programación basado en bloques que permita introducir sentencias de control y cómputo a una máquina.
- Identificar las técnicas y herramientas para la construcción de analizadores léxicos, sintácticos y semánticos que permitan traducir sentencias de un lenguaje de programación a código máquina.

4 Resultados de Aprendizaje del Curso

- RA-2 Habilidad de aplicar el diseño de ingeniería para generar soluciones que satisfagan necesidades específicas teniendo en cuenta la salud pública, la seguridad y el bienestar, así como factores globales, culturales, sociales, ambientales y económicos.
- RA-6 Habilidad para desarrollar y realizar la experimentación adecuada, analizar e interpretar datos y utilizar el juicio de ingeniería para sacar conclusiones.
- RAC-1 Construye un compilador, interprete o procesador del lenguaje aplicando la teoría de los lenguajes formales
- RAC-2 Crea rutinas o programas aplicados a la teoría de los lenguajes formales para soluciones de ingeniería
- RAC-3 Define la especificación de un lenguaje de propósito general y la máquina que lo interpreta

5 Programación del Curso									
Unidad Temática	Semana	Contenido de Aprendizaje	Evidencias	Actividades Aprendizaje	HAD		HTI		Total Horas
					Aula Clase	Espacio Virtual	Trabajo dirigido	Trabajo Independiente	
Introducción a la teoría de la computación y análisis léxico.	1	Introducción a la compilación, Arquitectura de compilador e interprete.	Participación en foros o entrega de informes.	Debate de conceptos	2	1	2	4	9
	2	Conceptos básicos teoría de la computación.	Entrega de actividades en medios digitales	Ejemplo, ejercicios y práctica en el computador	2	2	2	2	12
	3	Automas finito determinista y no determinista	Entrega de actividades en medios digitales	Ejemplo, ejercicios y práctica en el computador	2	2	2	2	12
	4	Autómatas de pila.	Entrega de actividades en medios digitales	Ejemplo, ejercicios y práctica en el computador	1	2	2	4	9
	5	Máquina de Turing.	Entrega de actividades en medios digitales	Ejemplo, ejercicios y práctica en el computador	1	2	2	4	9
Análisis sintáctico	6	Gramáticas, definiciones y conceptos	Entrega de actividades en medios digitales	Ejemplo, debates en clase ejercicios y práctica en el computador	1	2	2	4	9
	7	Proceso de derivación y generación de lenguajes.	Entrega de actividades en medios digitales	Ejemplo, ejercicios y práctica en el computador	1	2	2	4	9
	8	Gramáticas regulares	Entrega de actividades en medios digitales	Ejemplo, debates en clase ejercicios y práctica en el computador	1	2	2	4	9
	10	Gramáticas para generación de expresiones	Entrega de actividades en medios digitales	Ejemplo, debates en clase ejercicios y práctica en el computador	1	2	2	4	9
	11	Gramáticas libres del contexto.	Entrega de actividades en medios digitales	Ejemplo, debates en clase ejercicios y práctica en el computador	1	2	2	4	9

Formato para la Elaboración de Microdiseños de Cursos

				el computador					
	12	Ambigüedad, Recursión por izquierda, factorización de gramáticas.	Entrega de actividades en medios digitales	Ejemplo, debates en clase ejercicios y práctica en el computador	2	4	4	4	12
Tipos de Analizadores, Análisis semántico y generación de código objeto	13	Analizador descendente, Analizador ascendente.	Entrega de actividades en medios digitales	Ejemplo, debates en clase ejercicios	1	2	2	4	9
	14	Gramáticas S, Q, y LL (1), Gramáticas LR y SLR.	Entrega de actividades en medios digitales	Ejemplo, debates en clase ejercicios	2	2	4	4	12
	15	Método Shift-Identity y Método Shift-Reduce.	Entrega de actividades en medios digitales	Ejemplo, debates en clase ejercicios	2	2	4	4	12
	16	Generación de código	Entrega de actividades en medios digitales	Ejemplo, debates en clase ejercicios	1	2	2	4	9
Total					21	31	36	56	150
Créditos Académicos					4				

6 Prácticas de campo (Laboratorios y Salida de Campo)

Unidad Temática	Fundamentación Teórica	Evidencias	Actividades Aprendizaje	Recursos	Tiempo (h)	Semana
Todas las temáticas	Se aplica la teoría de los lenguajes formales.	Programas o Rutinas en lenguajes de programación definidos por el docente.	Programación o implementación de la teoría de los lenguajes formales	Computador, lenguajes de programación.	Durante el semestre	Cada semana

7 Mecanismos de Evaluación del Aprendizaje

Resultado de Aprendizaje	Mediación de Evaluación	Mecanismos, Criterios y/o Rúbricas	Semana de Evaluación
RA-2 Habilidad de aplicar el diseño de ingeniería para generar soluciones que satisfagan necesidades específicas teniendo en cuenta la salud pública, la seguridad y el bienestar, así como factores globales, culturales, sociales, ambientales y económicos.			

Formato para la Elaboración de Microdiseños de Cursos

RA-6 Habilidad para desarrollar y realizar la experimentación adecuada, analizar e interpretar datos y utilizar el juicio de ingeniería para sacar conclusiones.			
RAC-1 Construye un compilador o un intérprete para un lenguaje formal de propósito general o específico.	Se entrega software o rutinas mediante plataformas virtuales como el MS Teams, páginas web o el campus virtual	El docente define rúbricas de evaluación con los estudiantes.	Evaluación permanente te en el semestre
RAC-2 Crea rutinas o programas aplicados a la teoría de los lenguajes formales para soluciones ingeniería.	Se entrega software o rutinas mediante plataformas virtuales como el MS Teams, páginas web o el campus virtual	El docente define rúbricas de evaluación con los estudiantes.	Evaluación permanente te en el semestre
RAC-3 Define la especificación de un lenguaje de propósito general y la máquina que lo interpreta	Se entrega software o rutinas mediante plataformas virtuales como el MS Teams, páginas web o el campus virtual	El docente define rúbricas de evaluación con los estudiantes.	Evaluación permanente te en el semestre

8 Valoración de los Resultados de Aprendizaje

Valoración	Sobresaliente	Destacado	Satisfactorio	Básico	No Cumplimiento
Fundamentos Cualitativos					
Construye un compilador, intérprete o procesador de lenguaje teniendo en cuenta la teoría de los lenguajes formales.	Crea un procesador de lenguaje aplicando la teoría de los lenguajes formales, los principios de la ingeniería de software y lo aplica en un problema específico de cómputo.	Crea un procesador de lenguaje aplicando la teoría de los lenguajes formales y los principios de la ingeniería de software.	Crea un procesador de lenguaje aplicando la teoría de los lenguajes formales.	Comprende la teoría de los lenguajes formales, pero no los aplica en la construcción del procesador del lenguaje.	No Construye un compilador o un intérprete teniendo con los principios de los lenguajes formales.
Crea rutinas o programas teniendo en cuenta los principios de la ingeniería de Software	Crea rutinas o programas teniendo en cuenta los principios de la ingeniería de Software y	Crea rutinas o programas teniendo en cuenta los principios de la ingeniería de Software y	Crea rutinas o programas teniendo en cuenta los principios de la ingeniería de Software	Comprende los principios de la ingeniería de software, pero no los aplica en el desarrollo de rutinas.	No crea rutinas o programas aplicando los principios de la ingeniería del Software.

Formato para la Elaboración de Microdiseños de Cursos

	la teoría de los lenguajes formales y resuelve problemas específicos de cómputo.	la teoría de los lenguajes formales.			
Define la especificación de un lenguaje de propósito general y la máquina que lo interpreta teniendo en cuenta la teoría de los lenguajes formales.	Define la especificación de un lenguaje de propósito general y la máquina que lo interpreta teniendo en cuenta la teoría de los lenguajes formales, los principios de la ingeniería de software y las necesidades de cómputo específicas al problema.	Define la especificación de un lenguaje de propósito general y la máquina que lo interpreta teniendo en cuenta la teoría de los lenguajes formales y los principios de la ingeniería de software.	Define la especificación de un lenguaje de propósito general y la máquina que lo interpreta teniendo en cuenta la teoría de los lenguajes formales.	Comprende como especificar un lenguaje de propósito general y su máquina de interpretación, pero no lo aplica.	No redacta la especificación de un lenguaje de propósito general y la máquina que lo interpreta teniendo en cuenta la teoría de los lenguajes formales.
Construye un compilador, intérprete o procesador de lenguaje teniendo en cuenta la teoría de los lenguajes formales.	Crea un procesador de lenguaje aplicando la teoría de los lenguajes formales, los principios de la ingeniería de software y lo aplica en un problema específico de cómputo.	Crea un procesador de lenguaje aplicando la teoría de los lenguajes formales y los principios de la ingeniería de software.	Crea un procesador de lenguaje aplicando la teoría de los lenguajes formales.	Comprende la teoría de los lenguajes formales, pero no los aplica en la construcción del procesador del lenguaje.	No Construye un compilador o un intérprete teniendo en cuenta los principios de los lenguajes formales.

9 Recursos Educativos y Herramientas TIC

N	Nombre	Justificación	Contenido de Aprendizaje
1	Laboratorio de cómputo	Se usará para programar y ejecutar programas en clase	
2	Proyector de video	Requerido para las actividades de presentación por parte del profesor y la socialización de los trabajos	

Formato para la Elaboración de Microdiseños de Cursos

3	Herramientas de Desarrollo de software.	Requerido para implementar los algoritmos y técnicas aprendidas durante el curso.	
4	Software JFLAP para experimentación	Requerido para afianzar los tópicos de las asignaturas.	
5	Herramientas PCLex o JLex, Jflex, PCYACC o CUP	Requerido para implementar algunos tópicos de las asignaturas.	
6	Base de datos por suscripción.	Requeridos para lecturas complementarias de los temas.	
7	Libros impresos	Requeridos para lecturas complementarias de los temas.	

10 Referencias Bibliográficas

[1] Enrique Alfonseca Moreno, Manuel Mariyón Salomón, Roberto , 2007. Teoría de autómatas y lenguajes formales. McGraw-Hill España .España.

[2] JACINTO RUIZ CATALAN, 2009. Compiladores teoría e implementación. Alfaomega. España.

[3] Jiménez Millán, José Antonio, 2009. Compiladores y procesadores de lenguajes. Servicio de Publicaciones de la Universidad de Cádiz. España.

[4] HOLUB, Allen. Compiler Design in C. Primera Edición. 1990. Prentice Hall.

[5] AHO, SETHI y ULLMAN. Compiladores. Principios, técnicas y herramientas. Edición única. 1986. Addison Wesley.

[6] LOUDEN C. Kenneth. Construcción de Compiladores, Primera edición. 2004. Thomson Editorial.

Director de Programa

Decano Facultad