

SÍLABO

TEORÍA EN COMPUTACIÓN (100000S60F)

2025 - Ciclo 1 Marzo

1. DATOS GENERALES

1.1. Carrera:	Ingeniería de Software
1.2. Créditos:	3
1.3. Enseñanza de curso:	Virtual en vivo
1.4. Horas semanales:	4

2. FUNDAMENTACIÓN

En el entorno laboral actual, donde la optimización y eficiencia de sistemas son esenciales, el desarrollo de compiladores es fundamental para garantizar software de alto rendimiento. Este curso proporcionará al futuro profesional los conocimientos necesarios en teoría de la computación y diseño de compiladores, permitiéndoles crear herramientas que faciliten la traducción eficiente de lenguajes de programación a código máquina. Además, el curso contribuye al perfil de egreso al capacitar a los futuros profesionales en la construcción de compiladores robustos y adaptables, esenciales para el desarrollo de software eficiente.

3. SUMILLA

El curso es de naturaleza práctica, asimismo, introduce la teoría de lenguajes, abordando temas como el análisis lexicográfico, sintáctico y semántico, así como la generación de código que son fundamentales en el desarrollo de entornos de programación para la creación de soluciones de software.

4. LOGRO GENERAL DE APRENDIZAJE

Al finalizar el curso, el estudiante aplica los conceptos de la teoría de computación para desarrollar compiladores necesarios en la creación de soluciones de software, utilizando herramientas de programación y técnicas de optimización.

5. UNIDADES Y LOGROS ESPECÍFICOS DE APRENDIZAJE

Unidad de aprendizaje 1: Teoría de lenguajes.	Semana 1,2,3,4 y 5
Logro específico de aprendizaje: Al finalizar la unidad, el estudiante aplica los principios de la teoría de lenguajes para la construcción de compiladores, utilizando gramáticas formales, autómatas y métodos de análisis sintáctico.	
Temario: <ul style="list-style-type: none"> Introducción a la Teoría de la computación Traductores. Compiladores. Interpretes. Preprocesadores. Ensambladores. Lenguajes de programación. Introducción a los Compiladores Estructura de un compilador. Proceso de compilación. Etapas de análisis. Etapas de síntesis. Gramáticas. Definiciones previas (símbolo, vocabulario o alfabeto, cadena, lenguaje, gramática, autómata). Operaciones con cadenas. Operaciones con alfabetos. Operaciones con lenguajes. Cerradura positiva y estrella de Kleene. Gramáticas y Lenguajes Definición formal de gramática. Definición formal de lenguaje. Jerarquías de gramáticas. Correspondencia entre gramáticas y lenguajes. Expresiones regulares. Lenguajes regulares. Lenguajes libres de contexto Autómatas. Definición formal de un autómata. Autómatas Finitos Deterministas (AFD). Autómatas Finitos No Deterministas (AFND). Jerarquía de autómatas. Autómatas de Pila. 	
Unidad de aprendizaje 2: Análisis Lexicográfico y Sintáctico.	Semana 6,7,8,9 y 10

Logro específico de aprendizaje:

Al finalizar la unidad, el estudiante aplica los principios del análisis lexicográfico y sintáctico para la construcción de compiladores, utilizando métodos de análisis estructurados y estrategias de procesamiento ascendente y descendente.

Temario:

- Análisis Lexicográfico. Utilidad y funcionamiento. Token, Patrón Lexema.
- Análisis Lexicográfico. Construcción de un analizador lexicográfico usando técnicas de análisis léxico automatizado.
- Análisis Sintáctico Utilidad y funcionamiento. Tipos de análisis sintáctico. Análisis sintáctico lineal. Análisis sintáctico recursivo. Análisis sintáctico determinista.
- Análisis Sintáctico Análisis sintáctico descendente. Análisis sintáctico ascendente.
- Análisis Sintáctico. Construcción de un analizador sintáctico usando técnicas de análisis sintáctico automatizado.

Unidad de aprendizaje 3:

Análisis semántico, Generación de código.

Semana 11,12,13,14,15,16,17 y 18

Logro específico de aprendizaje:

Al finalizar la unidad, el estudiante aplica los principios del análisis semántico y la generación de código para la construcción de compiladores, utilizando estructuras de tipos, tablas de símbolos y gestionando la memoria en tiempo de ejecución.

Temario:

- Análisis Semántico. Utilidad y funcionamiento. Atributos y acciones semánticas. Notaciones. Comprobaciones semánticas.
- Tabla de Tipos y Símbolos Introducción. Tabla de tipos. Implementaciones. Tabla de símbolos. Implementaciones.
- Generación de código Visión general. Pasos para la generación de código. Generación de código intermedio. Generación de código final. Generación de código para sentencias de control.
- Gestión de memoria Gestión de memoria en tiempo de ejecución. La zona de código. La zona de datos.
- Implementaciones: La gestión de la memoria y la generación de código.
- Implementación de L-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.
- Implementación de C-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.

6. METODOLOGÍA

El curso se desarrolla a través de la plataforma virtual de aprendizaje que se usa como principal medio para el desarrollo de las sesiones sincrónicas que son complementadas con recursos y materiales que se publican a lo largo del curso para fomentar el desarrollo de aprendizajes significativos. Por otro lado, el estudiante dispone en la plataforma de un espacio de foro de consultas para resolver las dudas académicas a lo largo del curso. Finalmente, las actividades de evaluación se desarrollan de acuerdo con lo señalado en el sílabo a través de la plataforma virtual de aprendizaje (aprendizaje para la era digital).

7. SISTEMA DE EVALUACIÓN

El cálculo del promedio final se hará de la siguiente manera:

$$(20\%)PC1 + (20\%)PC2 + (20\%)PC3 + (40\%)EXFN$$

Donde:

Tipo	Descripción	Semana	Observación
PC1	PRÁCTICA CALIFICADA 1	5	Evaluación individual
PC2	PRÁCTICA CALIFICADA 2	10	Evaluación individual
PC3	PRÁCTICA CALIFICADA 3	15	Evaluación individual
EXFN	EXAMEN FINAL	18	Evaluación individual

Indicaciones sobre Fórmulas de Evaluación:

1. La nota mínima aprobatoria final es de 12.

- El estudiante que no rinde el examen final puede rendir un único examen de rezagado. La nota obtenida en este examen de rezagado reemplaza al examen final no rendido.
El estudiante rinde el examen de rezagado en la fecha programada por la Universidad, previa presentación de solicitud y pago de los derechos por examen de rezagado dispuesto en el tarifario vigente y publicado en Portal del Estudiante. Los exámenes de rezagados se aplican al final del período lectivo y abarcan todos los temas vistos en la asignatura.
- En caso un estudiante no rinda una práctica calificada (PC) y, por lo tanto, obtenga NSP, este es reemplazado por la nota obtenida en el examen final. Si también tiene NSP en el examen final, este es reemplazado por la nota obtenida en el examen rezagado. Este reemplazo de nota es automático. No es necesario que el estudiante realice trámite alguno para que proceda el remplazo de la nota. En caso de que el alumno tenga más de una práctica calificada no rendida, solo se reemplaza la práctica calificada de mayor peso.

8. FUENTES DE INFORMACIÓN

Bibliografía Base:

- José Antonio Jiménez Millán. *Compiladores y procesadores de lenguajes*. Servicio de Publicaciones de la Universidad de Cádiz. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=36068>
- Martínez López, Francisco. *Teoría, diseño e implementación de compiladores de Lenguajes*. RA-MA Editorial. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=37950>

Bibliografía Complementaria:

No hay bibliografía

9. COMPETENCIAS

Carrera	Competencias específicas
Ingeniería de Software	• Desarrollo de Software
	• Desarrollo de software

10. CRONOGRAMA DE ACTIVIDADES

Unidad de aprendizaje	Semana	Sesión	Tema	Actividades y evaluaciones
	1	1	Introducción a la Teoría de la computación Traductores. Compiladores. Interpretes. Preprocesadores. Ensambladores. Lenguajes de programación.	• Discutir sobre la importancia y utilidad de estos programas.
		2	Introducción a la Teoría de la computación Traductores. Compiladores. Interpretes. Preprocesadores. Ensambladores. Lenguajes de programación.	• Discutir sobre la importancia y utilidad de estos programas.
	2	3	Introducción a los Compiladores Estructura de un compilador. Proceso de compilación. Etapas de análisis. Etapas de síntesis.	• Analiza la estructura de los compiladores y propone modelos básicos
		4	Introducción a los Compiladores Estructura de un compilador. Proceso de compilación. Etapas de análisis. Etapas de síntesis.	• Analiza la estructura de los compiladores y propone modelos básicos
			Gramáticas. Definiciones previas (símbolo, vocabulario o alfabeto, cadena, lenguaje, gramática, autómatas).	• Desarrolla ejercicios utilizando las operaciones con

Unidad 1 Teoría de lenguajes	3	5	Operaciones con cadenas. Operaciones con alfabetos. Operaciones con lenguajes. Cerradura positiva y estrella de Kleene.	cadenas, alfabetos y lenguajes.
		6	Gramáticas. Definiciones previas (símbolo, vocabulario o alfabeto, cadena, lenguaje, gramática, autómeta). Operaciones con cadenas. Operaciones con alfabetos. Operaciones con lenguajes. Cerradura positiva y estrella de Kleene.	<ul style="list-style-type: none"> Desarrolla ejercicios utilizando las operaciones con cadenas, alfabetos y lenguajes.
	4	7	Gramáticas y Lenguajes Definición formal de gramática. Definición formal de lenguaje. Jerarquías de gramáticas. Correspondencia entre gramáticas y lenguajes. Expresiones regulares. Lenguajes regulares. Lenguajes libres de contexto	<ul style="list-style-type: none"> Desarrolla ejercicios utilizando las operaciones con expresiones y lenguajes regulares.
		8	Gramáticas y Lenguajes Definición formal de gramática. Definición formal de lenguaje. Jerarquías de gramáticas. Correspondencia entre gramáticas y lenguajes. Expresiones regulares. Lenguajes regulares. Lenguajes libres de contexto	<ul style="list-style-type: none"> Desarrolla ejercicios utilizando las operaciones con expresiones y lenguajes regulares.
	5	9	Autómatas. Definición formal de un autómeta. Autómatas Finitos Deterministas (AFD). Autómatas Finitos No Deterministas (AFND). Jerarquía de autómatas. Autómatas de Pila.	<ul style="list-style-type: none"> Exposición del docente
		10	Evaluación	<ul style="list-style-type: none"> PRÁCTICA CALIFICADA 1
Unidad 2 Análisis Lexicográfico y Sintáctico	6	11	Análisis Lexicográfico. Utilidad y funcionamiento. Token, Patrón Lexema.	<ul style="list-style-type: none"> Revisa la utilidad y funcionamiento del analizador lexicográfico.
		12	Análisis Lexicográfico. Utilidad y funcionamiento. Token, Patrón Lexema.	<ul style="list-style-type: none"> Revisa la utilidad y funcionamiento del analizador lexicográfico.
	7	13	Análisis Lexicográfico. Construcción de un analizador lexicográfico usando técnicas de análisis léxico automatizado.	<ul style="list-style-type: none"> Resolución de ejercicios utilizando técnicas de análisis léxico automatizado.
		14	Análisis Lexicográfico. Construcción de un analizador lexicográfico usando técnicas de análisis léxico automatizado.	<ul style="list-style-type: none"> Resolución de ejercicios utilizando técnicas de análisis léxico automatizado.
		15	Análisis Sintáctico Utilidad y funcionamiento. Tipos de análisis sintáctico. Análisis sintáctico lineal. Análisis sintáctico recursivo. Análisis sintáctico determinista.	<ul style="list-style-type: none"> Explora la utilidad y funcionamiento de los diferentes tipos de analizador

	8			sintáctico.
		16	Análisis Sintáctico Utilidad y funcionamiento. Tipos de análisis sintáctico. Análisis sintáctico lineal. Análisis sintáctico recursivo. Análisis sintáctico determinista.	<ul style="list-style-type: none"> Explora la utilidad y funcionamiento de los diferentes tipos de analizador sintáctico.
	9	17	Análisis Sintáctico Análisis sintáctico descendente. Análisis sintáctico ascendente.	<ul style="list-style-type: none"> Resuelve ejercicios de análisis sintácticos.
		18	Análisis Sintáctico Análisis sintáctico descendente. Análisis sintáctico ascendente.	<ul style="list-style-type: none"> Resuelve ejercicios de análisis sintácticos.
	10	19	Análisis Sintáctico. Construcción de un analizador sintáctico usando técnicas de análisis sintáctico automatizado.	<ul style="list-style-type: none"> Exposición del docente
		20	Evaluación	<ul style="list-style-type: none"> PRÁCTICA CALIFICADA 2
	11	21	Análisis Semántico. Utilidad y funcionamiento. Atributos y acciones semánticas. Notaciones. Comprobaciones semánticas.	<ul style="list-style-type: none"> Explora la utilidad y funcionamiento del analizador semántico.
		22	Análisis Semántico. Utilidad y funcionamiento. Atributos y acciones semánticas. Notaciones. Comprobaciones semánticas.	<ul style="list-style-type: none"> Explora la utilidad y funcionamiento del analizador semántico.
	12	23	Tabla de Tipos y Símbolos Introducción. Tabla de tipos. Implementaciones. Tabla de símbolos. Implementaciones.	<ul style="list-style-type: none"> Analiza la utilidad y funcionamiento de las Tablas de Tipos y Símbolos.
		24	Tabla de Tipos y Símbolos Introducción. Tabla de tipos. Implementaciones. Tabla de símbolos. Implementaciones.	<ul style="list-style-type: none"> Analiza la utilidad y funcionamiento de las Tablas de Tipos y Símbolos.
Unidad 3 Análisis semántico, Generación de código	13	25	Generación de código Visión general. Pasos para la generación de código. Generación de código intermedio. Generación de código final. Generación de código para sentencias de control.	<ul style="list-style-type: none"> Desarrolla ejercicios de generación de código.
		26	Generación de código Visión general. Pasos para la generación de código. Generación de código intermedio. Generación de código final. Generación de código para sentencias de control.	<ul style="list-style-type: none"> Desarrolla ejercicios de generación de código.
		27	Gestión de memoria Gestión de memoria en tiempo de ejecución. La zona de código. La zona de datos.	<ul style="list-style-type: none"> Explora cómo se gestiona la memoria de los programas compilados

	14	28	Gestión de memoria Gestión de memoria en tiempo de ejecución. La zona de código. La zona de datos.	<ul style="list-style-type: none"> • Explora cómo se gestiona la memoria de los programas compilados
	15	29	Implementaciones: La gestión de la memoria y la generación de código.	<ul style="list-style-type: none"> • Exposición del docente
		30	Evaluación	<ul style="list-style-type: none"> • PRÁCTICA CALIFICADA 3
	16	31	Implementación de L-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.	<ul style="list-style-type: none"> • Implementa un caso completo basado en L-0.
		32	Implementación de L-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.	<ul style="list-style-type: none"> • Implementa un caso completo basado en L-0.
	17	33	Implementación de C-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.	<ul style="list-style-type: none"> • Implementa un caso completo basado en C-0.
		34	Implementación de C-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.	<ul style="list-style-type: none"> • Implementa un caso completo basado en C-0.
	18	35	Evaluación	<ul style="list-style-type: none"> • Examen Final