

## SÍLABO

### TEORÍA EN COMPUTACIÓN (100000S61F)

#### 2025 - Ciclo 1 Marzo

#### 1. DATOS GENERALES

1.1. Carrera:	Ingeniería de Software
1.2. Créditos:	3
1.3. Enseñanza de curso:	Presencial
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

<b>Unidad de aprendizaje 1:</b> Teoría de lenguajes.	Semana 1,2,3,4 y 5
<b>Logro específico de aprendizaje:</b> 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.	
<b>Temario:</b> <ul style="list-style-type: none"> <li>Introducción a la Teoría de la computación Traductores. Compiladores. Interpretes. Preprocesadores. Ensambladores. Lenguajes de programación.</li> <li>Introducción a los Compiladores Estructura de un compilador. Proceso de compilación. Etapas de análisis. Etapas de síntesis.</li> <li>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.</li> <li>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</li> <li>Autómatas. Definición formal de un autómata. Autómatas Finitos Deterministas (AFD). Autómatas Finitos No Deterministas (AFND). Jerarquía de autómatas. Autómatas de Pila.</li> </ul>	
<b>Unidad de aprendizaje 2:</b> 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**

Para el desarrollo de los aprendizajes del curso, una de las estrategias que se propone es la exposición del docente que proporciona la construcción de los conocimientos a partir de ejemplos y casuísticas que faciliten la comprensión. Asimismo, se promueve la participación activa y permanente del estudiante a través del desarrollo de ejercicios, lecturas, absolución de preguntas, en forma individual y grupal (aprendizaje colaborativo) lo que permite un trabajo metacognitivo, a través de la actividad autónoma del estudiante en el desarrollo de las evaluaciones del curso (aprendizaje autónomo). Por ello es importante que el estudiante asista a las clases, habiendo leído los temas correspondientes a cada sesión. Finalmente, se utilizan otros recursos, como: pizarra, multimedia, videos (aprendizaje para la era digital) y comunicación a través de medios complementarios como correos electrónicos para fomentar una mayor interacción con el estudiante.

**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.
2. 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.
3. 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	<ul style="list-style-type: none"> <li>• Desarrollo de Software</li> </ul>

## 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.	<ul style="list-style-type: none"> <li>• Discusión de la importancia y utilidad de estos programas.</li> </ul>
		2	Introducción a la Teoría de la computación Traductores. Compiladores. Interpretes. Preprocesadores. Ensambladores. Lenguajes de programación.	<ul style="list-style-type: none"> <li>• Discusión de la importancia y utilidad de estos programas.</li> </ul>
	2	3	Introducción a los Compiladores Estructura de un compilador. Proceso de compilación. Etapas de análisis. Etapas de síntesis.	<ul style="list-style-type: none"> <li>• Análisis de la estructura de los compiladores y propone modelos básicos</li> </ul>
		4	Introducción a los Compiladores Estructura de un compilador. Proceso de compilación. Etapas de análisis. Etapas de síntesis.	<ul style="list-style-type: none"> <li>• Análisis de la estructura de los compiladores y propone modelos básicos</li> </ul>
		5	Gramáticas. Definiciones previas (símbolo, vocabulario o alfabeto, cadena, lenguaje, gramática, autómatas). Operaciones con cadenas. Operaciones	<ul style="list-style-type: none"> <li>• Desarrollo de ejercicios utilizando las operaciones con</li> </ul>

<b>Unidad 1</b> Teoría de lenguajes	3		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, autómatas). Operaciones con cadenas. Operaciones con alfabetos. Operaciones con lenguajes. Cerradura positiva y estrella de Kleene.	<ul style="list-style-type: none"> <li>Desarrollo de ejercicios utilizando las operaciones con cadenas, alfabetos y lenguajes</li> </ul>
	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"> <li>Desarrollo de ejercicios utilizando las operaciones con expresiones y lenguajes regulares</li> </ul>
		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"> <li>Desarrollo de ejercicios utilizando las operaciones con expresiones y lenguajes regulares</li> </ul>
	5	9	Autómatas. Definición formal de un autómata. Autómatas Finitos Deterministas (AFD). Autómatas Finitos No Deterministas (AFND). Jerarquía de autómatas. Autómatas de Pila.	<ul style="list-style-type: none"> <li>Exposición docente</li> </ul>
		10	Evaluación	<ul style="list-style-type: none"> <li>PRÁCTICA CALIFICADA 1</li> </ul>
<b>Unidad 2</b> Análisis Lexicográfico y Sintáctico	6	11	Análisis Lexicográfico. Utilidad y funcionamiento. Token, Patrón Lexema.	<ul style="list-style-type: none"> <li>Revisión de la utilidad y funcionamiento del analizador lexicográfico</li> </ul>
		12	Análisis Lexicográfico. Utilidad y funcionamiento. Token, Patrón Lexema.	<ul style="list-style-type: none"> <li>Revisión de la utilidad y funcionamiento del analizador lexicográfico</li> </ul>
	7	13	Análisis Lexicográfico. Construcción de un analizador lexicográfico usando técnicas de análisis léxico automatizado.	<ul style="list-style-type: none"> <li>Resolución de ejercicios utilizando técnicas de análisis léxico automatizado.</li> </ul>
		14	Análisis Lexicográfico. Construcción de un analizador lexicográfico usando técnicas de análisis léxico automatizado.	<ul style="list-style-type: none"> <li>Resolución de ejercicios utilizando técnicas de análisis léxico automatizado.</li> </ul>
		15	Análisis Sintáctico Utilidad y funcionamiento. Tipos de análisis sintáctico. Análisis sintáctico lineal. Análisis sintáctico recursivo. Análisis	<ul style="list-style-type: none"> <li>Exploración de la utilidad y funcionamiento de</li> </ul>

	8		sintáctico determinista.	los diferentes tipos de analizador 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"> <li>• Exploración de la utilidad y funcionamiento de los diferentes tipos de analizador sintáctico</li> </ul>
	9	17	Análisis Sintáctico Análisis sintáctico descendente. Análisis sintáctico ascendente.	<ul style="list-style-type: none"> <li>• Resolución de ejercicios de análisis sintácticos</li> </ul>
		18	Análisis Sintáctico Análisis sintáctico descendente. Análisis sintáctico ascendente.	<ul style="list-style-type: none"> <li>• Resolución de ejercicios de análisis sintácticos</li> </ul>
	10	19	Análisis Sintáctico. Construcción de un analizador sintáctico usando técnicas de análisis sintáctico automatizado.	<ul style="list-style-type: none"> <li>• Exposición docente</li> </ul>
		20	Evaluación	<ul style="list-style-type: none"> <li>• PRÁCTICA CALIFICADA 2</li> </ul>
		11	Análisis Semántico. Utilidad y funcionamiento. Atributos y acciones semánticas. Notaciones. Comprobaciones semánticas.	<ul style="list-style-type: none"> <li>• Exploración de la utilidad y funcionamiento del analizador semántico.</li> </ul>
			Análisis Semántico. Utilidad y funcionamiento. Atributos y acciones semánticas. Notaciones. Comprobaciones semánticas.	<ul style="list-style-type: none"> <li>• Exploración de la utilidad y funcionamiento del analizador semántico.</li> </ul>
		12	Tabla de Tipos y Símbolos Introducción. Tabla de tipos. Implementaciones. Tabla de símbolos. Implementaciones.	<ul style="list-style-type: none"> <li>• Análisis de la utilidad y funcionamiento de las Tablas de Tipos y Símbolos</li> </ul>
			Tabla de Tipos y Símbolos Introducción. Tabla de tipos. Implementaciones. Tabla de símbolos. Implementaciones.	<ul style="list-style-type: none"> <li>• Análisis de la utilidad y funcionamiento de las Tablas de Tipos y Símbolos</li> </ul>
		13	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"> <li>• Desarrollo de ejercicios de generación de código.</li> </ul>
			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"> <li>• Desarrollo de ejercicios de generación de código.</li> </ul>

<b>Unidad 3</b> Análisis semántico, Generación de código	14	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"> <li>• Exploración de la gestión de la memoria de los programas compilados</li> </ul>
		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"> <li>• Exploración de la gestión de la memoria de los programas compilados</li> </ul>
	15	29	Implementaciones: La gestión de la memoria y la generación de código.	<ul style="list-style-type: none"> <li>• Exposición docente</li> </ul>
		30	Evaluación	<ul style="list-style-type: none"> <li>• PRÁCTICA CALIFICADA 3</li> </ul>
	16	31	Implementación de L-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.	<ul style="list-style-type: none"> <li>• Implementación de un caso completo basado en L-0.</li> </ul>
		32	Implementación de L-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.	<ul style="list-style-type: none"> <li>• Implementación de un caso completo basado en L-0.</li> </ul>
	17	33	Implementación de C-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.	<ul style="list-style-type: none"> <li>• Implementación de un caso completo basado en C-0.</li> </ul>
		34	Implementación de C-0 Especificaciones. Análisis Léxico. Análisis Sintáctico. Análisis Semántico. Generación de Código.	<ul style="list-style-type: none"> <li>• Implementación de un caso completo basado en C-0.</li> </ul>
	18	35	Evaluación	<ul style="list-style-type: none"> <li>• Examen Final</li> </ul>