



**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

SÍLABO

1. INFORMACIÓN GENERAL

1.1 Nombre y código de la asignatura: Lenguajes y Traductores –205009

1.2 Número de créditos: 03

1.3 Número de horas: Teoría: 03 horas

1.4 Ciclo de estudio: III

1.5 Periodo Académico: 2019 - II

1.6 Requisitos:

1.7 Profesores:

Augusto Cortez Vásquez (Coordinador)

Jaime Pariona Quispe

Carlos Ruiz De La Cruz Melo

2. SUMILLA

El curso es de tipo específico y de naturaleza de teórico-práctico. Presenta las diferentes técnicas para la construcción de un compilador. Lenguajes y gramáticas. Expresiones regulares y autómatas finitos. Lenguajes libres de contexto y autómatas de pila. Análisis léxico, análisis sintáctico y análisis semántico. Generación de código intermedio, optimización de código y generación de código. Manejo de errores.

3. COMPETENCIA GENERAL

- Capacitar al alumno en los conocimientos fundamentales para definir un lenguaje formal.
- Capacitar al alumno en los conocimientos fundamentales para la construcción de Compiladores para traducción de los Lenguajes de programación de alto nivel
- Exponer y discutir las características de diseño, fundamentos teóricos, y los algoritmos que se utilizan en cada componente así como las herramientas avanzadas de desarrollo de compiladores dando énfasis en las áreas de Teoría de

4. PROGRAMACIÓN

Unidad didáctica 1: Lenguajes gramáticas y autómatas				
<p>Competencia específica: Lenguajes y gramáticas Formales: alfabeto, lenguajes, símbolos terminales y no terminales. Jerarquía de lenguajes, Clasificación de Chomsky. Forma Normal de Backus (BNF).</p> <p>Pre – requisito: es necesario repasar los conceptos estudiados en el curso de Matemáticas Discretas: Lenguajes formales, gramáticas formales.</p>				
Semana	Contenidos	Estrategias didácticas	Actividades	Evaluación
1ª	Especificación de lenguajes	Exposición del docente. Uso de diapositivas Trabajo colaborativo	Establecemos acuerdos de convivencia. Dialogamos sobre el contenido del sílabo. Resuelve problema planteado en clase.	Explica los procedimientos que siguió para diseñar el algoritmo, en un texto expositivo.
2ª	Gramáticas.	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa	Explica algoritmo y programa diseñado en un texto escrito.
3ª	Autómatas finitos y expresiones regulares y autómatas	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	Modela autómatas finitos a partir de expresiones regulares
4ª	Autómatas de pila.	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	Modela autómatas de pila a partir de gramáticas libres de contexto.

Unidad didáctica 2: Análisis lexicográfico y análisis sintáctico				
<p>Competencia específica: Especifica un analizador lexicográfico y analizador sintáctico de un lenguaje de programación y lo transforma en un programa empleando un lenguaje de programación para solucionar problema, con sentido ético.</p>				
Semana	Contenidos	Estrategias didácticas	Actividades	Evaluación
5ª	Análisis Lexicográfico Componentes léxicos. Diagramas de transiciones.	Recuperación de saberes previos.	Revisa material de clase. Discute ejemplos.	Específica y construye un analizador

	Funciones y reconocimiento de tokens. Especificación de los componentes lexicográficos. Gramáticas regulares. Tratamiento y recuperación de errores lexicográficos. Autómatas finitos y su implantación	Exposición del docente Uso de diapositivas Trabajo colaborativo	Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	lexicográfico para un lenguaje determinado.
6ª	Métodos de análisis sintáctico: descendente y ascendente. Otros tipos de parser. Analizadores sintácticos LL. Construcción de tabla de análisis sintáctico.	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	Explica algoritmo y programa diseñado en un texto escrito.
7ª	Métodos de análisis sintáctico: descendente y ascendente. Otros tipos de parser. Analizadores sintácticos LR. Construcción de tabla de análisis sintáctico.	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	Explica el funcionamiento de un analizador sintáctico descendente y lo transforma en un programa de análisis sintáctico .
8ª	EXAMEN PARCIAL			En un texto escrito presenta solución de problemas planteados.

Unidad didáctica 3: Análisis semántico				
Competencia específica: Comprende el proceso de análisis semántico y verificación de tipos. Construye un programa empleando un lenguaje de programación para solucionar problema, con sentido ético.				
Semana	Contenidos	Estrategias didácticas	Actividades	Evaluación
9ª	Funciones de la Verificación semántica. Comprobación de Tipos	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	Explica el procedimiento de análisis semántico.
10ª	Comprende los fundamentos de los arreglos para crear un algoritmo, empleando pseudocódigo e implementar un programa, muestra orden y disciplina al comunicar los resultados que arroja el algoritmo.	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal.	Explica algoritmo y programa diseñado en un texto escrito.

	Estructuras Básicas: Arreglo Bidimensional Definición, declaración, manejo de filas y columnas. Ejemplos.		Resuelve problema planteado en clase de manera colaborativa.	
11 ^a	Comprende los fundamentos de los registros para crear un algoritmo, empleando pseudocódigo e implementar un programa, muestra orden y disciplina al comunicar los resultados que arroja el algoritmo. Estructuras Básicas: Registros Definición. Vector de registros, operaciones leer y mostrar, buscar, editar, etc. Ejemplos. Registros anidados. Ejemplos.	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	Explica algoritmo y programa diseñado en un texto escrito.

Unidad didáctica 4: Generación de código				
Competencia específica: Crea código intermedio y código objeto optimizándolo y lo transforma en un programa empleando un lenguaje de programación para solucionar problema, con sentido ético.				
Semana	Contenidos	Estrategias didácticas	Actividades	Evaluación
12 ^a	Código intermedio. Optimización de código. Generación de código.	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	Explica el procedimiento de generación de código intermedio.
13 ^a	Optimización de código intermedio	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	Explica el procedimiento de optimización de código intermedio.
14 ^a	Generación de código objeto	Recuperación de saberes previos. Exposición del docente Uso de diapositivas Trabajo colaborativo	Revisa material de clase. Discute ejemplos. Resuelve problema planteado en clase de manera personal. Resuelve problema planteado en clase de manera colaborativa.	Explica el procedimiento de generación de código objeto

15ª	ENTREGA DE PROYECTO DEL CURSO. Presentación y sustentación.	Explicación de orden de exposiciones de trabajos. Retroalimentación luego de cada exposición.	Explica solución planteada a caso propuesto de manera colaborativa.	Explica algoritmo y programa que soluciona caso propuesto, en un texto escrito y de manera oral.
16ª	EXAMEN FINAL			En un texto escrito presenta solución de problemas planteados.

5. ESTRATEGIA DIDÁCTICA

Por parte del docente se desarrollará la asignatura siguiendo los criterios inductivos, flexibles, participativos y constructivos con la participación del alumno; en este sentido se utilizará las técnicas participativas, laboratorios guiados, desarrollo de proyectos.

Por parte del estudiante, participará activamente a través de desarrollo de un proyecto en la que transfiere los conocimientos adquiridos en un caso práctico

6. EVALUACIÓN DEL APRENDIZAJE

CRITERIO	PESO
PF = (N1 +N2+N3)/3 Donde:	
Examen Parcial (N1)	1
Promedio Evaluaciones Calificadas y Proyecto del curso=0.5PromLab+0.5Proyecto (N2)	1
Examen Final (N3)	1

7. REFERENCIAS BIBLIOGRÁFICAS

A. BÁSICA

[AHO 1990] Aho A., Sethi R., Ullman J. - "Compiladores, Principios, Técnicas y herramientas". Addison-Wesley. 1990. Wilmington Delaware EUA

[BROOKSHEAR 1989] Brookshear J. "Teoría de la computación, lenguajes formales, autómatas y complejidad". Addison-Wesley. 1989. Wilmington Delaware EUA

[CORTEZ 2012] Augusto Cortez Vásquez. "Lenguajes y traductores". CEUCSS Lima 2012

[GRASSMANN 1997] Grassmann W., Tremblay J. "Matemática discreta y lógica". 1997. Prentice Hall Hispanoamericana

[KOLMAN 1989] Kolman B. Busby R. "Estructuras de matemáticas discretas para la computación". 1989. Prentice Hall Hispanoamericana

B. COMPLEMENTARIA

[PRATT 1998] Pratt, T. - Zelkowitz, M. "Lenguajes de programación, diseño e Implementación". 1998. Prentice Hall Hispanoamericana

[SETHI 1992] Sethi R. - "Lenguajes de programación, Conceptos y constructores". Addison-Wesley. 1992. Wilmington Delaware EUA

[TEUFEL 1995] Teufel – Schmidt – Teufel "Compiladores Conceptos fundamentales". Addison-Wesley. 1995. Wilmington Delaware EUA

[HOPCROFT 1993] Hopcroft J., Ullman J. "Introducción a la teoría de autómatas, lenguajes y computación". CECSA 1993 EUA

•