

Presidencia Roque Sáenz Peña, 05 de junio de 2025

RESOLUCIÓN N° 138/2025 - C.D.C.B. y A.

VISTO:

El Expediente N° 01-2025-01894 sobre propuesta de actualización del programa de la asignatura Sintaxis y Semántica de los Lenguajes de la carrera Ingeniería en Sistemas de Información, iniciado por la Directora de Carrera Ing. ZACHMAN, Patricia; y

CONSIDERANDO:

Que la asignatura 18 Sintaxis y Semántica de los Lenguajes corresponde al Área de Programación de la Carrera Ingeniería en Sistemas de Información y se dicta en el 3° año 1er. cuatrimestre;

Que el Programa Analítico contempla los contenidos mínimos y la carga horaria propuestos en el Plan de Estudios de la Carrera aprobado por Resolución N°063/19-C.S.;

Que las asignaturas correlativas respetan lo establecido en el Sistema de Correlatividades de la Carrera aprobado por Resolución N°088/19-C.S.;

Que los objetivos planteados guardan coherencia con los contenidos, métodos pedagógicos y de evaluación propuestos, y la bibliografía propuesta es actualizada;

Que la fundamentación refleja la relevancia de la asignatura en la formación de los futuros profesionales; los Trabajos Prácticos planteados son pertinentes y adecuados y la forma de evaluación planteada se adecúa a la reglamentación vigente;

Lo aprobado en sesión de la fecha.

POR ELLO:

**EL CONSEJO DEPARTAMENTAL
DEL DEPARTAMENTO DE CIENCIAS BÁSICAS Y APLICADAS DE LA
UNIVERSIDAD NACIONAL DEL CHACO AUSTRAL
RESUELVE:**

ARTÍCULO 1°: APROBAR el programa de la asignatura Sintaxis y Semántica de los Lenguajes de la Carrera de Ingeniería en Sistemas de Información, que como Anexo Único forma parte de la presente Resolución.

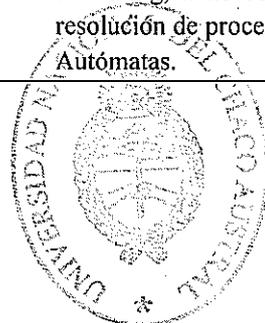
ARTÍCULO 2°: Regístrese, comuníquese, y archívese.



Nora B. Ok...
Dra. Nora B. Ok...
Directora
Dpto. de Cs. Básicas y Aplicadas

ANEXO
PROGRAMA DE ASIGNATURA

 <p>UNCAUS UNIVERSIDAD NACIONAL DEL CHACO AUSTRAL</p>		<p align="center">18 - SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES Plan de Estudios Resolución N°063/19-C.S.-</p>	
<p>Carga Horaria: 135 horas Teóricas: 65 horas Prácticas: 70 horas</p>		<p>Programa vigente desde: 2025</p>	
Carrera		Año	Cuatrimestre
Ingeniería en Sistemas de Información		3°	Primero
CORRELATIVAS PRECEDENTES		CORRELATIVAS SUBSIGUIENTES	
Asignaturas		Asignaturas	
Para cursar		Para rendir	
Regularizadas	Aprobadas	Aprobadas	
-Arquitectura de Computadoras	-Algoritmos y Estructuras de Datos	-Arquitectura de Computadoras	-Gestión de Datos -Ingeniería del Software -Redes de Información -Trabajo Final
DOCENTES:		Prof. Adjunto: Dra. Paola Daniela Budán J.T.P.: Ing. Rubén Cáceres	
FUNDAMENTACIÓN:		<p>La comprensión de los lenguajes formales y los modelos de cómputo es fundamental para la formación de profesionales en el área de la informática y los sistemas. Esta asignatura proporciona las bases teóricas necesarias para el análisis y la construcción de lenguajes de programación, así como para el diseño de autómatas y máquinas de Turing, herramientas esenciales para modelar el comportamiento de los algoritmos y los procesos de cómputo.</p> <p>Asimismo, introduce al estudiante en los conceptos y técnicas fundamentales del diseño de compiladores e intérpretes, abordando el proceso de traducción desde el análisis léxico hasta la generación y optimización del código intermedio. Estos conocimientos son clave para comprender el funcionamiento interno de los lenguajes de programación y para desarrollar software eficiente y robusto.</p>	
OBJETIVOS:		<p>Objetivo general: Desarrollar competencias para el análisis, diseño e implementación de lenguajes formales, autómatas y compiladores, mediante la aplicación de los fundamentos teóricos adecuados.</p> <p>Objetivos específicos:</p> <ol style="list-style-type: none"> Generar y reconocer gramáticas y lenguajes formales de distintos tipos (regulares, libres de contexto), aplicando los fundamentos de la Teoría de Lenguajes y Gramáticas. Definir y construir autómatas finitos, autómatas de pila y máquinas de Turing, tanto como reconocedores de lenguajes como para la resolución de procedimientos, utilizando principios de la Teoría de Autómatas. 	



	<p>iii. Diseñar compiladores o intérpretes sencillos mediante la implementación de las fases del proceso de traducción: análisis léxico, sintáctico y semántico.</p> <p>iv. Aplicar técnicas de generación y optimización de código intermedio, utilizando representaciones adecuadas y considerando criterios de eficiencia.</p>
<p>CONTENIDOS MÍNIMOS:</p>	<p>Gramática y Lenguajes Formales. Jerarquía de Chomsky. Automatas Finitos. Expresiones Regulares y su aplicación al Análisis Léxico. Gramáticas Independientes del Contexto. Automatas Push Down y su Aplicación al Análisis Sintáctico. Otros Tipos de Analizadores Sintácticos. Máquinas Turing. Introducción a las Semánticas.</p>
<p>MÉTODOS PEDAGÓGICOS:</p>	<p>La asignatura propone un enfoque didáctico centrado en el estudiante, promoviendo un espacio de diálogo y trabajo constructivo en el aula, donde docentes y alumnos interactúan en un proceso de aprendizaje colaborativo. Se hace uso de recursos especialmente diseñados como videos de apoyo para el desarrollo de las actividades prácticas que se distribuyen entre los alumnos como material de consulta. Este tipo de material permite reforzar el aprendizaje y acompañar el proceso formativo fuera del aula presencial.</p> <p>Las <i>clases teóricas</i> están orientadas a presentar los contenidos de forma estructurada, utilizando presentaciones con recursos visuales (gráficos, tablas, ejemplos) y promoviendo la participación a través de preguntas, ejercicios grupales y actividades reflexivas. Al inicio y cierre de cada unidad temática se contextualiza el contenido, destacando su aplicación, y se utilizan mapas conceptuales para facilitar la integración y comprensión. Las <i>clases prácticas</i> priorizan el trabajo colaborativo como técnica metodológica principal, favoreciendo la apropiación activa del conocimiento y la socialización. Además, se incorporan herramientas como ANTLR para vincular los conceptos teóricos con aplicaciones concretas en el análisis de lenguajes.</p> <p>La asignatura también fomenta el desarrollo de habilidades de comunicación oral y escrita, fundamentales para la formación integral del estudiante. Se articula con conocimientos previos adquiridos en materias como programación, arquitectura de computadoras y teoría de conjuntos, ampliando y profundizando la comprensión de los lenguajes de programación. Asimismo, se relaciona con la asignatura de inglés, ya que permite al alumno abordar bibliografía técnica especializada escrita en este idioma.</p>
<p>MÉTODOS DE EVALUACIÓN:</p>	<p>Se llevarán a cabo dos evaluaciones parciales. La estructura y objetivos se detallan a continuación:</p> <p>PARCIAL 1</p> <p>Estructura</p> <ul style="list-style-type: none"> • Ejercicio relacionado a un lenguaje libre de contexto, la gramática correspondiente, y su respectiva comprobación utilizando JFLAP. • Ejercicio relacionado a un lenguaje regular, la gramática y la expresión regular correspondiente, y su respectiva comprobación utilizando JELAP. • Ejercicio de construcción de un autómata. <p>Objetivo: Determinar si los alumnos han adquirido habilidad para generar y reconocer gramáticas y lenguajes de distinto tipo.</p>

4





	<p>PARCIAL 2</p> <p>Estructura</p> <ul style="list-style-type: none"> • Ejercicio sobre analizadores léxicos. • Ejercicio sobre analizadores sintácticos. • Ejercicio de optimización de código. <p>Objetivo: Demostrar solvencia para reconocer los diferentes algoritmos en los que se basa el funcionamiento de un intérprete o compilador.</p> <p>Desarrollo de ejercicios con ANTLR:</p> <p>Estructura</p> <ul style="list-style-type: none"> • Realizar un Analizador Léxico con una herramienta automática. • Realizar un Analizador Sintáctico con una herramienta automática. <p>Objetivos</p> <ul style="list-style-type: none"> • Determinar si los alumnos distinguen correctamente los usos de los diferentes analizadores. • Lograr generar pequeños analizadores para comprender la utilidad de la herramienta. <p>Criterios de Evaluación en evaluaciones parciales</p> <ul style="list-style-type: none"> • Adecuada relación/aplicación de la teoría a la práctica. • Adecuada resolución de ejercicios. • Grado de concordancia con la respuesta requerida. <p>Escala de Valoración: La escala de valoración a emplear: para las evaluaciones parciales será cuantitativa politómica: (Escala: 1 al 100). El puntaje mínimo para aprobar los parciales es de sesenta (60) puntos (sobre una calificación máxima de 100). En todos los casos, y dadas las condiciones especiales de cursado, se otorga la posibilidad de rehacer los ejercicios del parcial.</p> <p>Condiciones para Lograr la Regularidad</p> <p>a) Asistir como mínimo al 75 % del total de las clases. b) Aprobar las evaluaciones parciales. c) Asistir a las clases en las que se desarrollan los ejercicios con ANTLR.</p> <p>Examen Libre: Los alumnos libres deberán cumplir las siguientes etapas, cada una de ellas eliminatória.</p> <p>1ra etapa) Presentar la carpeta de prácticos, con al menos 7 días de anticipación a la fecha de examen. La misma deberá ser aprobada por el tribunal.</p> <p>2da etapa) Realizar un ejercicio de ANTLR cuyo enunciado será solicitado al equipo de cátedra con al menos 15 días de anticipación al examen.</p> <p>2da etapa) Aprobar una evaluación escrita de tipo práctica.</p> <p>3ra etapa) Aprobar una evaluación oral de tipo teórica sobre todos los contenidos del programa.</p>
<p>PROGRAMA ANALÍTICO DE CONTENIDOS:</p>	<p>1. TEORÍA DE LENGUAJES FORMALES</p> <p>CONCEPTOS BÁSICOS: Símbolo, alfabeto, palabra. Operaciones con hileras. Lenguaje. Operaciones con lenguajes: unión, intersección, concatenación. Propiedades de las operaciones.</p> <p>GRAMÁTICAS FORMALES: definición, proceso de derivación. Jerarquía de Chomsky. Gramáticas no restringidas, sensibles al</p>

M



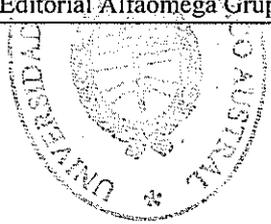


	<p>contexto, libres de contexto y regulares. Lenguajes generados por cada tipo de gramáticas.</p> <p>CARACTERÍSTICAS DE LAS GRAMÁTICAS: Gramáticas libres de contexto: árbol de derivación. Derivaciones a izquierda y a derecha. La ambigüedad. Recursividad a izquierda directa e indirecta. Factorización. Gramáticas propias.</p> <p>GRAMÁTICAS REGULARES: Expresiones regulares. Propiedades y equivalencias.</p> <p>2. TEORÍA DE AUTÓMATAS</p> <p>AUTÓMATAS FINITOS: Definición y representación gráfica. El autómata finito como reconocedor de lenguajes. Autómata finito determinista y no determinista. Equivalencia.</p> <p>AUTÓMATA DE PILA: Definición formal. Autómata de pila como reconocedor de un lenguaje. Autómata a pila determinístico y no determinístico.</p> <p>MÁQUINAS DE TURING: Definición formal. Representación. Interpretaciones de las computaciones. Configuración de una máquina de Turing. Máquina de Turing multicinta. Máquina universal de Turing. La insolubilidad del problema de la parada.</p> <p>3. DISEÑO DE COMPILADORES</p> <p>TIPO DE TRADUCTORES: compiladores e intérpretes. Características de cada uno. Fases de un compilador. Análisis y síntesis de la compilación.</p> <p>ANÁLISIS LÉXICO: Funciones. Componentes léxico, patrones y léxemas. Especificación y reconocimiento de tokens. Definiciones regulares. Diagrama de transición. Manejo de buffers de entrada. Errores léxicos. Generadores de analizadores léxicos.</p> <p>ANÁLISIS SINTÁCTICO: Funciones. Métodos descendentes y ascendentes. Manejo de errores sintácticos. Analizador sintáctico descendente: gramáticas LL(k), analizador sintáctico predictivo no recursivo. Analizador sintáctico ascendente: gramáticas LR(k). Implantación de un analizador sintáctico ascendente mediante poda. Métodos para analizador sintáctico LR. Conflictos. Generador automático de analizadores sintácticos.</p> <p>ANÁLISIS SEMÁNTICO: Traducción dirigida por la sintaxis. Definición dirigida por la sintaxis. Reglas semánticas. Árbol sintáctico para expresiones. Grafos dirigidos acíclicos para expresiones (GDA). Esquema de traducción. Acciones semánticas. Notación postfija. Comprobación de tipos.</p> <p>GENERACIÓN DE CÓDIGO: generación de código Intermedio, distinto tipo de representaciones. Optimización de Código Intermedio. Generación de Código Objeto.</p> <p>TABLA DE SÍMBOLOS: funciones, operaciones principales, estructuras de implementación: Listas lineales simple o doblemente enlazadas, Tabla de símbolos ordenada y Tablas de dispersión (hash).</p>
<p>PROGRAMA ANALÍTICO DE TRABAJOS PRÁCTICOS:</p>	<p>Trabajo Práctico N°1: Diseñado para que el estudiante adquiriera las siguientes competencias:</p> <p>Habilidad para distinguir los conceptos fundamentales de la Teoría de Lenguajes Formales y Gramáticas.</p> <p>Capacidad para formalizar y clasificar gramáticas.</p>



	<p>. Destreza para determinar qué tipo de gramática genera las hileras de un lenguaje particular.</p> <p>Cantidad de semanas asignadas: 3 semanas.</p> <p>Trabajo Práctico N°2: Diseñado para que el estudiante adquiera las siguientes competencias:</p> <ul style="list-style-type: none"> . Habilidad para distinguir los conceptos fundamentales de la Teoría de Automatas. . Capacidad para reconocer y utilizar autómatas. . Destreza para integrar conocimientos adquiridos. <p>Cantidad de semanas asignadas: 3 semanas.</p> <p>Trabajo Práctico N°3: Actividades relacionadas con la primera fase de un compilador. Las técnicas utilizadas para construir analizadores léxicos también se pueden aplicar a otras áreas, como, por ejemplo, a lenguajes de consulta y sistemas de recuperación de información. En cada aplicación, el problema de fondo es la especificación y diseño de programas que ejecuten las acciones activadas por palabras que siguen ciertos patrones dentro de las cadenas a reconocer. La programación dirigida por patrones es de mucha utilidad, por lo que se introduce una herramienta denominada ANTLR – considerada de última generación – que permite generar analizadores léxicos, sintácticos y semánticos a partir de una gramática.</p> <p>Cantidad de semanas asignadas: 2 semanas.</p> <p>Trabajo Práctico N°4: Este trabajo práctico aborda en profundidad el análisis sintáctico en el contexto de compiladores, abarcando tanto analizadores sintácticos descendentes como ascendentes. Se ejercita el cálculo de los conjuntos PRIM y SIG, la verificación de propiedades LL(1), el reconocimiento de cadenas válidas por diversas gramáticas, y la construcción de tablas de análisis. Además, incluye una sección dedicada a la generación y optimización de código intermedio, donde se solicita traducir expresiones aritméticas y programas a árboles sintácticos, notación postfija, código de tres direcciones, triplas, cuádruplas y triplas indirectas, aplicando técnicas de optimización.</p> <p>Cantidad de semanas asignadas: 3 semanas.</p>
<p>BIBLIOGRAFÍA:</p>	<p>Unidad 1 a 3</p> <p>Alfred V. Aho - Ravi Sethi : AT&T Bell Laboratories. Murray Hill, New Jersey - Jeffrey D. Ullman : Stanford University, (aut.), "Compiladores: Principios, Técnicas y Herramientas", Pearson Educación, 1990</p> <p>Barchini, Graciela y Alvarez Margarita. Fundamentos Teóricos de la Ciencia de la Computación. Departamento de Informática. FCEyT. UNSE: Santiago del Estero.1998</p> <p>Michael A. Harrison. Introduction to Formal Languages. Theory Addison-Wesley Publishing. Estados Unidos. 1978.</p> <p>Ruiz- Catalán Jacinto, "Compiladores: Teoría E Implementación", Editorial Alfaomega Grupo Editor, 2010</p>

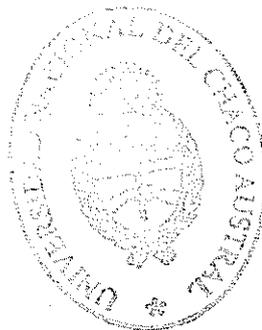
(Handwritten mark)





///Res. N° 138/2025-DCByA.

	<p>Grune Dick, Bal Henri E , Jacobs Cerial J. H. , Longendoen Koen G., "Diseño de Compiladores Modernos", Editorial MCGRAW-HILL,2007</p> <p>Louden Kenneth C., "Construcción de Compiladores", Editorial Cengage Learning / Thomson Internacional, 2004</p> <p>Isasi, Martinez y Borrajo, "Lenguajes, Gramáticas y Autómatas, un enfoque práctico", Editorial Addison-Wesley., 1997</p> <p>Hopcroft, Motwani y Ullman, "Introducción a la Teoría de Autómatas, Lenguajes y Computación",Ed. Pearson, 2ª edición, 2002.</p> <p>Juan Giró, "Introducción a la Informática Teórica", Ed. Ciudad Gráfica 2001</p> <p>Enrique Alfonseca, Manuel Alfonseca y Roberto Moriyon, "Teoría de Autómatas y Lenguajes Formales" , Editorial Mc Graw Hill, 2007.</p> <p>PAPERS DE ACTUALIZACIÓN:</p> <p>Palmkvist, V., Castegren, E., Haller, P., & Broman, D. (2021, March). <i>Resolvable ambiguity: principled resolution of syntactically ambiguous programs</i>. In Proceedings of the 30th ACM SIGPLAN International Conference on Compiler Construction (pp. 153-164).</p> <p>Pai T, V., & Aithal, P. S. (2020). <i>A Systematic Literature Review of Lexical Analyzer Implementation Techniques in Compiler Design</i>. International Journal of Applied Engineering and Management Letters (IJAEML), 4(2), 285-301.</p> <p>Vázquez, J. C., Constable, L., Jornet, W., & Meloni, B. (2015). <i>Enseñanzas de la Implementación de un Analizador Léxico</i>.</p> <p>Stanier, J., & Watson, D. (2013). <i>Intermediate representations in imperative compilers: A survey</i>. ACM Computing Surveys (CSUR), 45(3), 1-27.</p>
--	---




Dra. Nora B. Okada
Directora
Dpto. de Cs. Básicas y Aplicadas