



Universidad Nacional de Luján  
Departamento de  
Ciencias Básicas



DISPOSICION CONSEJO DIRECTIVO DEPARTAMENTAL DE CIENCIAS BÁSICAS DISPCD-CB : 30 /  
2025

LUJAN, 13 DE MARZO DE 2025

VISTO: El programa de la asignatura Teoría de la Computación I (11412) para la carrera Licenciatura en Sistemas de Información presentado por la División Computación; y

CONSIDERANDO:

Que la Comisión Plan de Estudio ha tomado intervención en el trámite.

Que se ha tratado y aprobado por el Consejo Directivo Departamental de Ciencias Básicas en su Sesión Ordinaria del día 6 de marzo de 2025.

Por ello,

EL CONSEJO DIRECTIVO DEPARTAMENTAL

DE CIENCIAS BÁSICAS

D I S P O N E :

ARTÍCULO 1°.- Aprobar el programa de la asignatura Teoría de la Computación I (11412) para la carrera Licenciatura en Sistemas de Información presentado por la División Computación que como anexo I forma parte de la presente Disposición.-

ARTICULO 2°.- Establecer que el mismo tendrá vigencia para los años 2024-2025.-

ARTÍCULO 3°.- Regístrese, comuníquese, cumplido, archívese.-

Lic. Ariel H. REAL - Secretario Académico - Departamento de Ciencias Básicas

Dr. Carlos J. DI SALVO - Vicedirector Decano - Departamento de Ciencias Básicas

UNIVERSIDAD NACIONAL DE LUJAN

DEPARTAMENTO ACADÉMICO DE CIENCIAS BÁSICAS

---

**PROGRAMA OFICIAL**

1 /6

---

DENOMINACIÓN DE LA ACTIVIDAD: 11412-Teoría de la Computación I  
TIPO DE ACTIVIDAD ACADÉMICA: Asignatura

CARRERA: Licenciatura en Sistemas de Información

PLAN DE ESTUDIOS: 17.13 ([Resolución H.C.S. N° 478/12](#) y [874/14](#)).

---

DOCENTE RESPONSABLE:

Capuya Mara Alejandra – Profesor Adjunto

EQUIPO DOCENTE:

Cuagliarelli Silvia Marcela – Jefe de Trabajos Prácticos

Céspedes Eugenia – Ayudante de Primera

---

**ACTIVIDADES CORRELATIVAS PRECEDENTES:**

PARA CURSAR: 11075 – Programación II

PARA APROBAR: 11075 – Programación II

CARGA HORARIA TOTAL: HORAS SEMANALES: 6 - HORAS TOTALES 96

DISTRIBUCIÓN INTERNA DE LA CARGA HORARIA:

TEORICO: 33% 2 horas

PRÁCTICO: 67% 4 horas

PERÍODO DE VIGENCIA DEL PRESENTE PROGRAMA: 2024-2025
------------------------------------------------------

---

**CONTENIDOS MÍNIMOS O DESCRIPTORES**

Lenguajes formales y autómatas. Minimización de autómatas. Expresiones regulares. Máquinas de Turing. Jerarquía de Chomsky. Gramáticas e isomorfismos. Lenguajes de programación: Entidades y ligaduras. Sistema de tipos, Niveles de polimorfismo. Encapsulamiento y abstracción. Conceptos de intérpretes y compiladores. Criterios de diseño y de implementación de Lenguajes de Programación. Nociones básicas de semántica formal. Verificación de algoritmos.

---

**FUNDAMENTACIÓN, OBJETIVOS, COMPETENCIAS**

Esta asignatura permitirá al alumno poder conocer los modelos de autómatas para que sea posible capturar las ideas teóricas que apuntarán a la resolución de problemas reales mediante el alto poder de abstracción que poseen estos modelos con el fin de aplicarlo a distintas áreas (compiladores, circuitos, interfaces, traducción, etc.), como así también poder conocer los lenguajes de programación desde su sintaxis y semántica para lograr entender parte de la estructura y lógica de los compiladores de los principales lenguajes.

OBJETIVOS: (específicos)

- Al finalizar el curso, el alumno será capaz de:
  1. Conocer los distintos modelos de autómatas, su alto poder de abstracción aplicado a la resolución de problemas.
  2. Conocer los métodos formales para describir la sintaxis de todos los lenguajes estructurados gramaticalmente (gramáticas y su jerarquía, expresiones regulares)
  3. Conocer los elementos de los lenguajes de programación: variables, funciones, objetos, identificadores, etc.
  4. Aprender el concepto de ligadura y su relación con los distintos modelos de ejecución.
  5. Conocer los sistemas de tipos.
  6. Construir un analizador léxico y un analizador sintáctico con el uso de herramientas o sin ellas.
  7. Integración con la construcción de código para el análisis semántico.

---

**CONTENIDOS**

**Unidad 1:**

Introducción a los lenguajes formales

Concepto de lenguajes formales. Alfabeto, cadenas y lenguajes. Operaciones entre cadenas y lenguajes. Propiedades. Sistema de reescritura como introducción a la definición de gramáticas. Jerarquía de Chomsky: gramáticas de tipo 0, 1, 2 y 3.

**Unidad 2:**

Lenguajes regulares y Autómata finitos

Autómata finito: definiciones básicas; diagrama de transición; ejemplos. Autómata finito determinístico: reconocimiento y traducción. Autómata finito no determinístico. Equivalencia de autómata finito determinístico y no determinístico. Minimización de autómatas. Autómatas Traductores. Equivalencia entre gramáticas regulares y autómatas finitos. Expresiones regulares.

**Unidad 3:**

Autómata con pila y lenguajes libres del contexto

Autómata con pila: definiciones básicas; diagrama de transición; ejemplos. Gramáticas libres de contexto. Equivalencia entre gramáticas libres de contexto y autómatas con pila. Ambigüedad. BNF (Backus Naur Form). Base gramatical de la traducción de lenguajes.

**Unidad 4:**

Máquina de Turing y Lenguajes no restringidos

Máquina de Turing: definiciones básicas; ejemplos. Máquina de Turing determinística: reconocimiento y traducción. Máquina de Turing multicinta. Gramáticas e Isomorfismos.

**Unidad 5:**

Lenguajes de programación.

Lenguajes de programación: Entidades. Elementos de los lenguajes de programación: identificadores, funciones, procedimientos, palabras reservadas, constantes, etc. El concepto de ligadura (binding). Binding de tipos y binding de almacenamiento. Diferencia entre binding dinámico y estático. Alcance estático y dinámico. Tiempo de vida. Criterios de diseño y de implementación de Lenguajes de Programación. Modelos de ejecución. Intérpretes y compiladores.

**Unidad 6:**

Sistemas de tipos

Tipos de datos: definición. Los distintos tipos de datos: primitivos, string, bool, etc). Conversión de tipos. Compatibilidad entre tipos. Tipos polimórficos. Niveles de polimorfismo. Encapsulamiento y abstracción. Ejemplificación en lenguaje C, C++ y Ada. Chequeo de tipos en lenguajes funcionales. Verificación de algoritmos

**Unidad 7:**

Diseño de compiladores

Estructura general de un compilador. Relación entre los tipos de gramáticas y la estructura de los compiladores.

Análisis Lexicográfico: descripción de los elementos de un lenguaje modelado a través de un autómata finito, su implementación. Tabla de símbolos. Tabla de palabras reservadas. Mecanismo de acceso.

Análisis sintáctico: Análisis sintáctico descendente. Análisis sintáctico ascendente. Retrocesos. Detección y reporte de errores. Análisis sintáctico descendente predictivo. Conflictos desplazamiento-reducción. Conflictos reducción-reducción. Nociones básicas de semántica formal. Limitaciones.

---

**METODOLOGÍA**

**Clases teóricas:**

Se muestran mediante una clase expositiva utilizando presentaciones bajo computadora y proyector y también con tiza y pizarrón. Se presenta la noción a desarrollarse en la clase, se explican los fundamentos teóricos y se procura cerrar el tema con la enunciación de conclusiones.

**Clases prácticas:**

Los alumnos reciben 8 trabajos de problemas, planificados para la ejercitación de las nociones brindadas en clase. Una vez que se ha logrado que los alumnos calibren la dificultad de los problemas ofrecidos se realizan interrupciones del trabajo en grupos y se desarrollan en el pizarrón uno o dos ejercicios, concentrándose en los que han resultado más dificultosos.

Se entregan a los alumnos trabajos prácticos para ser resueltos en computadora. Consisten en la realización de programas o fragmentos de programas de reducido tamaño (del orden de las 10 líneas de código) en las que la dificultad es netamente conceptual fortaleciendo los conceptos vistos en las clases.

---

**Trabajos Prácticos**

TP n°1: Reconocimiento de Cadenas y Lenguajes. Resolución de operaciones sobre Lenguajes.

TP n°2: Reconocimiento de Cadenas de Lenguajes Regulares. Diseño de Expresiones Regulares

TP n°3: Diseño de autómatas finitos. Conversión de autómatas no determinísticos a determinísticos, minimización. Aplicación de herramienta: JFlap. Diseño de autómatas finitos basados en expresiones regulares .Implementación de gramáticas regulares. Diseño e implementación de Autómatas Traductores.

TP n°4: Construcción de gramáticas libres de contexto. Elaboración de árboles de derivación. Implementación de gramáticas aplicadas a la sintaxis de lenguajes de programación.

TP n°5: Diseño de autómatas con pila. Construcción de autómatas con pila a partir de gramáticas libres de contexto.

TP n°6: JFlex y JCup. Parsing.

Trabajo Práctico Compilador

Cada alumno o grupo de alumnos deberá realizar un trabajo práctico que consisten en el desarrollo de un analizador léxico, y un analizador sintáctico. La modalidad de desarrollo de este trabajo práctico consiste en presentación de las consignas en una breve clase inicial y la atención de consultas y verificación del grado de avance en puntos de control establecidos a principios del curso. Las entregas involucradas en estos puntos de control son:

Entrega de la propuesta de gramática para el lenguaje.

Entrega del Analizador Léxico

Entrega de integración del Analizador Léxico y Analizador Sintáctico

Entrega de código final integrado.

Cada alumno recibirá las consignas específicas para su trabajo práctico que hace que sea diferente de los restantes. Estas consignas personalizadas contienen elementos léxicos, elementos sintácticos y elementos semánticos.

---

**REQUISITOS DE APROBACION Y CRITERIOS DE CALIFICACIÓN:**

**Regularización:**

- a) Estar en condición de regular en las actividades correlativas al momento de su inscripción al cursado de la asignatura.
- b) Cumplir con un mínimo del 75 % de asistencia a las clases
- c) Aprobar 2 (dos) evaluaciones parciales con un promedio no inferior a cuatro (4) puntos, pudiendo recuperar sólo una de ellas. Cada evaluación solo podrá recuperarse en una oportunidad.
- d) Aprobar el trabajo práctico obligatorio con nota superior a 6-seis- puntos

**Promoción:**

- a) Tener aprobadas las actividades correlativas al finalizar el turno de examen extraordinario de ese cuatrimestre.

---

**PROGRAMA OFICIAL**

5 /6

- b) Cumplir con un mínimo del 75% de asistencia a las clases.
- c) Aprobar 2 (dos) evaluaciones parciales con un promedio no inferior a siete (7) puntos sin recuperar ninguna.
- d) Aprobar el trabajo práctico obligatorio con nota superior a 6-seis- puntos

**Exámenes para estudiantes en condición de libres**

Para aquellos estudiantes que, habiéndose inscriptos oportunamente en la presente actividad hayan quedado en condición de libres por aplicación de los artículos 29 o 32 del Régimen General de Estudios. Quedará en condición de libre el alumno que incurra en alguna/s de estas situaciones:

- No haber realizado el trabajo práctico obligatorio
- Haber desaprobado los dos parciales
- Haber desaprobado alguno de los parciales y su correspondiente recuperatorio.

Hasta quince días antes de la fecha de sustanciación de mesa, el alumno deberá entregar el trabajo práctico integrador correspondiente y todos los trabajos prácticos de problemas (TP 1 a TP8).

- 1) Para aquellos estudiantes que, habiéndose inscriptos oportunamente en la presente actividad hayan quedado en condición de libres por aplicación de los artículos 29 o 32 del Régimen General de Estudios, podrán rendir en tal condición la presente actividad..
- 2) Para aquellos estudiantes que no cursaron la asignatura y se presenten en condición de alumnos libres en la Carrera, por aplicación de los artículos 10 o 19 del Régimen General de Estudios, no podrán rendir en tal condición la presente actividad.
- 3) Las características del examen libre son las siguientes:
  - Hasta treinta días antes de la fecha de inscripción al examen, el alumno deberá entregar:
    - i. el trabajo práctico obligatorio correspondiente
    - ii. todos los trabajos prácticos de problemas (TP 1 a TP8).

---

**BIBLIOGRAFÍA**

**OBLIGATORIA**

- Aho A. V., Lam, M., SEthi, R. Ullman J. "Compiladores, Principios, Técnicas y Herramientas" (segunda edición). Pearson Addison-Wesley Iberoamericana, 2008

**COMPLEMENTARIA**

- Grune D., Bal H, Jacobs C., Langendoen K. "Diseño de Compiladores Modernos". Mc Graw Hill Interamericana, 2007
- Hopcroft J., Ullman J. "Introduction to Automata Theory, languages and computation". USA, Addison Wesley, 1979
- Kelley, D. "Teoría de Autómatas y Lenguajes Formales". Prentice-Hall, 1995.

UNIVERSIDAD NACIONAL DE LUJAN

DEPARTAMENTO ACADEMICO DE CIENCIAS BÁSICAS

---

**PROGRAMA OFICIAL**

6 /6

- Martin, John “Lenguajes Formales y Teoría de la Computación”. 3ra. Edición. McGraw-Hill. 2004.
- Sebesta Robert. “Concepts of programming languages”. 10° Edición. . Pearson. 2010

---

DISPOSICIÓN CD

## Hoja de firmas