



Ministerio de Cultura y Educación  
 Universidad Nacional de San Luis  
 Facultad de Ciencias Físico Matemáticas y Naturales  
 Departamento: Informatica  
 Area: Area V: Automatas y Lenguajes

(Programa del año 2026)  
 (Programa en trámite de aprobación)  
 (Presentado el 11/05/2026 21:46:38)

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
FUNDAMENTOS DE COMPUTACION	ING. INFORM.	OCD- 3-2/2 025	2026	1° cuatrimestre

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
APOLLONI, JAVIER MARIANO	Prof. Responsable	P.Adj Exc	40 Hs
LEGUIZAMON, MARIO GUILLERMO	Prof. Colaborador	P.Asoc Exc	40 Hs
LOOR, FABRICIO	Responsable de Práctico	JTP Simp	10 Hs
VILLEGAS, MARIA PAULA	Auxiliar de Práctico	A.1ra Semi	20 Hs

### III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	2 Hs	3 Hs	1 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
11/03/2026	23/06/2026	15	90

### IV - Fundamentación

El perfil del ingeniero debe incluir el conocimiento científico necesario que le permita analizar y comprender problemas, para los cuales pueda desarrollar y obtener una solución adecuada utilizando los beneficios de la tecnología y, en especial, de una computadora. En este contexto, una computadora puede ser vista como un dispositivo de cómputo que realiza operaciones y cálculos de manera automática, los cuales son plasmados en un algoritmo. Al intentar formalizar la idea intuitiva de algoritmo, se observa rápidamente que es necesario estudiar los modelos formales de cómputo y, con ellos, los lenguajes formales.

En un primer paso en el estudio de los fundamentos de la computación, deben analizarse los lenguajes formales, dada la estrecha relación entre estos y los problemas. De los lenguajes formales se desprende la necesidad de estudiar los modelos formales de cómputo asociados a ellos, siendo de especial importancia los modelos basados en dispositivos de reconocimiento o aceptación (autómatas o máquinas de estados) y los modelos basados en dispositivos generadores (gramáticas).

En un segundo paso, y a través de dichos modelos formales de cómputo, se puede analizar la teoría de la computabilidad.

Esta teoría estudia propiedades sobre los lenguajes y permite clasificarlos en decidibles y no decidibles. Con ello se establece un límite entre los problemas efectivamente resolubles (o computables) y aquellos que no pueden ser resueltos mediante un algoritmo (o no computables).

Por último, en un tercer paso en el estudio de los fundamentos de la computación, debe analizarse la teoría de la complejidad. En este caso, se estudian las propiedades de los lenguajes decidibles respecto al uso de los recursos, ya sea el tiempo, el espacio u otros. La finalidad de este estudio es jerarquizar la complejidad de los lenguajes decidibles -y, por ende, de sus problemas asociados- para distinguir entre aquellos cuyas soluciones utilizan eficientemente el recurso analizado y aquellos que, hasta el momento, no cuentan con una solución conocida que lo haga de manera eficiente.

Este curso aborda todos los elementos teóricos y prácticos necesarios para que las/los estudiantes puedan incorporar y comprender los conceptos mencionados anteriormente. El abordaje se realizará desde una perspectiva de aprendizaje incremental y constructivista, que propiciará la integración de los contenidos del curso con los conocimientos adquiridos en asignaturas previas de la carrera (Matemáticas Discreta, Programación I y II, Estructura de Datos y Algoritmos, etc.), considerando además el contexto social del/de la estudiante.

## V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso, se espera que la/el estudiante alcance los siguientes objetivos generales y específicos:

Objetivos generales:

- Vincular los aspectos teóricos y prácticos de la Teoría de Lenguajes, de la Teoría de la Computabilidad y de la Teoría de la Complejidad con la realidad y los problemas de la realidad.
- Fomentar el pensamiento analítico y de interpretación para resolver problemas (de la realidad) dados como enunciados en los ejercicios prácticos utilizando el marco teórico de la materia.
- Mejorar las capacidades y habilidades del/de la estudiante para su futuro laboral.

Objetivos específicos:

- Comprender los conceptos vinculados a la Teoría de los Lenguajes Formales.
- Diseñar lenguajes (formales) para representar problemas de la realidad dados en forma de enunciados.
- Comprender conceptos y funcionamientos relacionados con los modelos formales de cómputo: máquinas de estados (autómatas finitos, autómatas push-down y máquinas de Turing) y gramáticas, y la correspondencia entre ellos y con los lenguajes de programación.
- Diseñar soluciones que resuelvan un problema de la realidad dado en forma de enunciado utilizando el modelo formal de cómputos más adecuado al problema.
- Comprender y analizar los conceptos prácticos y teóricos asociados a la Teoría de la Computabilidad utilizando modelos avanzados de computación como la Máquina de Turing y a partir de ello establecer la decidibilidad o no de los lenguajes (lenguajes decidibles y no decidibles).
- Adquirir la capacidad para discernir cuándo un problema es resoluble y cuándo no lo es analizando propiedades de su lenguaje de decisión asociado
- Comprender y analizar los conceptos vinculados a la Teoría de la Complejidad utilizando modelos avanzados de computación como la Máquina de Turing y con ello establecer la tratabilidad o no de los problemas al clasificarlos en P, NP, NP-completo y otros relacionados.
- Adquirir la capacidad para detectar cuándo un problema es tratable y cuándo no lo es analizando propiedades de los lenguajes decidibles.

En el transcurso del dictado de esta asignatura se trabajarán de manera transversal los siguientes ejes:

- \* Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.
- \* Utilización de técnicas y herramientas de aplicación en ingeniería en sistemas de información/informática.
- \* Fundamentos para el desempeño en equipos de trabajo.
- \* Fundamentos para una comunicación efectiva.
- \* Fundamentos para el aprendizaje continuo.

## VI - Contenidos

### Contenidos mínimos:

Definición y especificación finita de lenguajes formales. Jerarquía de Chomsky. Autómatas y gramáticas. Lenguajes regulares, autómatas finitos y expresiones regulares. Lenguajes libres del contexto y gramáticas libres del contexto. Lenguajes dependientes del contexto y autómatas linealmente acotados. Lenguajes Recursivamente Enumerables y Máquina de Turing. Problemas y lenguajes formales. Computabilidad: problemas decidibles y no decidibles. Complejidad Computacional: clases de complejidades computacionales. Problemas tratables e intratables. Problemas P y NP.

Estos contenidos mínimos se desarrollan en las siguientes unidades:

### Unidad 1. Lenguajes Formales

Definición y especificación finita de los lenguajes formales: alfabeto, cadena, lenguaje. Representación de los lenguajes formales. Dispositivos generadores y reconocedores: Autómatas y Gramáticas. Relación general entre autómatas y gramática. Jerarquía de Chomsky.

### Unidad 2. Lenguajes Regulares

Lenguajes Regulares (Tipo 3). Expresiones Regulares (ER). Autómata Finito: Autómatas Finitos Determinísticos (AFD), Autómatas Finitos No Determinísticos (AFND), Autómata Finito No Determinístico con transiciones épsilon. Aplicaciones. Equivalencias: AFND a AFD, AFND-epsilon a AFD, ER a AFND-epsilon. Minimización de AFD. Propiedades de clausura de los lenguajes regulares.

### Unidad 3. Lenguajes Libres del Contexto

Lenguajes Libres del Contexto (Tipo 2). Gramáticas Libres de Contexto. Definición, Derivación y Árbol de derivación. Backus Naur Form (BNF) y BNF extendida. Autómata Push Down (autómatas pila): Definición, configuración y aceptación de cadenas.

### Unidad 4. Lenguajes Recursivamente Enumerables

Lenguajes Recursivamente enumerables (Tipo 0). Máquinas de Turing (MT): Máquina de Turing determinística o estándar (MTD), MT como modelo de cómputo: reconocedor y computador de funciones. Extensiones a la MTD: Máquina de Turing No Determinísticas, Máquina de Turing con k-cintas. Tesis de Church-Turing. Simplificaciones a la MTD: Lenguajes dependientes del Contexto y Autómatas Linealmente Acotados.

### Unidad 5. Teoría de la Computabilidad

Problemas y Lenguajes Formales: Relación. Computabilidad: Idea intuitiva de Algoritmo. Lenguajes recursivos y recursivamente enumerables: Propiedades. Problemas decidibles: definición y ejemplos. Problemas no decidibles: Problema asociado al lenguaje Universal, Problema de Parada o Detención (Halting), reducción de un problema a otro, implicaciones de la no decidibilidad de un problema.

### Unidad 6. Teoría de la Complejidad Computacional

Complejidad Computacional: Complejidad Temporal y Espacial. Problemas tratables e intratables: jerarquía de complejidades. Clases de Complejidades computacionales: Clases P y NP. NP-Complejidad y Problemas NP-Complejos: Problema de Satisfactibilidad (SAT) y reducción. Otros ejemplos de problemas NP-Complejos estándares.

## VII - Plan de Trabajos Prácticos

Durante las semanas que correspondan al dictado se abordarán las 6 unidades del curso tanto desde el contenido teórico como desde las respectivas actividades prácticas. El material de cada unidad estará disponible en el repositorio del curso para que la/el estudiante lo pueda acceder y consultar durante todo el curso.

Las actividades prácticas permitirán el desarrollo y profundización de los contenidos teóricos y serán el ámbito donde la/el estudiante alcance los objetivos propuestos para el curso. Las clases prácticas consistirán principalmente en la resolución de ejercicios poniendo en práctica los conocimientos logrados en este curso y aquellos previos utilizando el material didáctico accesible desde el repositorio. En los casos apropiados la/el estudiante podrá hacer uso de las herramientas informáticas que

implementan los conceptos teóricos desarrollados en el curso. Se espera que la/el estudiante utilice estas clases prácticas también como ámbito para contrastar ideas y confraternizar con sus compañeros generando y propiciando un espacio de reflexión e intercambio de ideas.

A continuación se desarrolla el plan de trabajos prácticos apuntando en cada caso las habilidades que se esperan que el alumno desarrolle en cada caso.

Plan de Trabajos Prácticos de Aula:

Práctico 1: Alfabetos - Cadenas - Lenguajes.

- Entender el significado de los términos alfabetos, cadenas y utilizarlos para definir lenguajes.
- Operar con cadenas y lenguajes.
- Describir lenguajes mediante una representación finita.
- Vincular los conceptos generales con el de los lenguajes de programación.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal.

Práctico 2: Lenguajes Regulares - Expresiones Regulares - Autómatas Finitos.

- Entender la definición de lenguajes regulares.
- Entender las expresiones regulares (ER) reconociendo operando y operadores.
- Describir lenguajes denotados por las ER.
- Construir ER para que denoten lenguajes específicos.
- Representar problemáticas de la realidad utilizando las ER.
- Analizar y diferenciar las componentes de Autómatas Finitos Determinísticos (AFD).
- Entender el funcionamiento de un AFD.
- Ejecutar un AFD.
- Determinar el lenguaje reconocido por un AFD.
- Diseñar y construir AFD que se ajusten a un lenguaje específico.
- Entender el funcionamiento de un Autómata Finito No Determinístico (AFND) y de un Autómata Finito No Determinístico con transiciones epsilon (AFND-e).
- Ejecutar un AFND y AFND-e.
- Determinar el lenguaje reconocido por un AFND y por un AFND-e.
- Diseñar y construir AFND y AFND-e que se ajusten a un lenguaje específico.
- Discernir las facilidades descriptivas de los 3 modelos de acuerdo al problema o lenguaje específico.
- Representar problemáticas de la realidad utilizando los Autómatas Finitos.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal, y la corroboración de lo realizado manualmente utilizando la herramienta JFLAP o similar.

Práctico 3: Equivalencias entre Autómatas Finitos y Expresiones Regulares. Minimización de Autómatas Finitos.

- Entender y aplicar el algoritmo que obtiene un AFD equivalente a un AFND
- Entender y aplicar el algoritmo que obtiene un AFD equivalente desde un AFND-e
- Entender y aplicar el algoritmo que obtiene un AFND-e que acepta el mismo lenguaje que el denotado por una ER.
- Entender y aplicar el algoritmo que obtiene el AFD mínimo equivalente a un AFD.
- Entender la utilización de todos los algoritmos de transformación en las problemáticas de la realidad.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal, y la corroboración de lo realizado manualmente utilizando la herramienta JFLAP o similar.

Práctico 4: Gramáticas Libres del Contexto - Autómatas Push-Down.

- Comprender notación, formas y estructura, ejecución y usos de las Gramáticas Libres del Contexto (GLC).
- Pensar, diseñar y construir GLC que deriven las cadenas de lenguajes específicos.
- Representar problemáticas de la realidad y asociadas a los lenguajes de programación utilizando GLC.
- Entender la notación, formas y estructuras y usos de la notación Backus Nour Form (BNF) para GLC.
- Entender la notación, formas y estructuras y usos de la notación BNF-extendida para GLC.
- Analizar, entender y determinar diferencias y similitudes entre ambas notaciones.
- Analizar, entender y determinar diferencias y similitudes entre ambas notaciones y las GLC.
- Representar problemáticas de la realidad y asociadas a los lenguajes de programación utilizando GLC en notación BNF y BNF-extendida.
- Entender notación, forma y estructura, ejecución y uso de los Autómatas Push-Down (APD).
- Aprender a discernir entre los Lenguajes Libres del Contexto, aquellos que son inherentemente no determinísticos de los determinísticos para poder diseñar el APD adecuado.
- Pensar, diseñar y construir APD Determinísticos y APD No Determinísticos (según sea el caso) que deriven las cadenas de lenguajes específicos.
- Aprender a determinar la conveniencia de utilizar uno u otro de los 2 tipos de aceptación en los APD (aceptación por estado final o pila vacía) de acuerdo al lenguaje.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal, y la corroboración de lo realizado manualmente utilizando la herramienta JFLAP o similar.

Práctico 5: Máquina de Turing estándar - Extensiones a la MT,

- Entender notación, forma y estructura, ejecución y uso de la Máquina de Turing estándar (MTD).
- Pensar, diseñar y construir MTD que reconozcan y acepten las cadenas de lenguajes específicos.
- Pensar, diseñar y construir MTD que computen funciones numéricas y/o realicen procedimientos varios.
- Modelizar problemáticas de la realidad utilizando MTD.
- Entender notación, forma y estructura, ejecución y uso de la extensión a la MTD: Máquina de Turing con k cintas (MTD-kcintas).
- Pensar, diseñar y construir MTD-kcintas que resuelvan problemas específicos.
- Entender notación, forma y estructura, ejecución y uso de la extensión a la MTD: Máquina de Turing No Determinística (MTND).
- Pensar, diseñar y construir MTND que resuelvan problemas específicos.
- Aprender a discernir las facilidades descriptivas de los 3 modelos (MTD, MTD-kcintas, MTND) para diseñar y construir el modelo adecuado al problema a resolver.
- Analizar y comparar las similitudes y diferencias del modelo de cómputo más general existente (Máquina de Turing en cualquiera de sus variantes) y de una computadora tanto desde el poder computacional como el descriptivo (o representacional) para resolver problemas de la realidad.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal, y la corroboración de lo realizado manualmente utilizando la herramienta JFLAP o similar, e implementar una versión simplificada y restringida del modelo de MT en una computadora haciendo uso de un lenguaje de programación.

Práctico 6: Problemas decidibles y no decidibles.

- Entender el vínculo entre problemas y lenguajes formales.
- Analizar, entender y aprender a discernir la diferencia entre lenguajes recursivos y recursivamente enumerables.
- Entender la relación entre lenguajes recursivos y algoritmos.
- Analizar, entender y aprender a diferenciar los problemas decidibles de los problemas no decidibles.
- Pensar y diseñar codificaciones de instancias de los problemas de la realidad y que sean válidas y adecuadas para el modelo de cómputo general de Máquina de Turing.
- Analizar y diseñar soluciones a problemas decidibles utilizando el modelo general de cómputo de la Máquina de Turing.
- Analizar y diseñar soluciones utilizando el modelo general de cómputo de la Máquina de Turing y aplicando el concepto de reducción de un problema a otro para determinar la no decidibilidad de ciertos problemas .

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo

grupal.

#### Práctico 7: Complejidad Temporal

- Entender el cálculo de la complejidad temporal en una Máquina de Turing de acuerdo al tamaño de la instancia de entrada (longitud de la cadena de entrada).
- Entender la manera en la que se construye la ecuación para la función de complejidad temporal asociada a una Máquina de Turing M para cadenas de una longitud n ( $tc_M(n)$ ).
- Analizar una Máquina de Turing M (en cualquiera de sus variantes) y construir la ecuación para la función de complejidad temporal asociada a esa Máquina de Turing M para cadenas de una longitud n ( $tc_M(n)$ ).
- Analizar la afectación que tiene alguna de las variantes de Máquina de Turing sobre la complejidad temporal asociada a ese problema.
- Analizar y entender la jerarquía de velocidades de crecimiento de la función de complejidad temporal.
- Analizar y entender las clases de lenguajes o problemas P, NP y NP-Completo.
- Analizar cuándo un problema es NP-Completo utilizando ejemplos teóricos.
- Analizar y entender el efecto que un problema NP-Completo causa a un/a ingeniero/a que está resolviendo un problema de la realidad con esas características.
- Vincular los desarrollos para ejemplos teóricos con los necesarios para ejemplos prácticos de la realidad.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal.

#### Plan de Trabajos Prácticos de Laboratorio:

##### Práctico de Laboratorio:

- Realizar un proceso investigativo mínimo acerca de un problema asociado a alguna de las temáticas de la materia.
- Analizar una posible solución al problema.
- Diseñar e implementar un algoritmo en un lenguaje de programación que resuelva ese problema.

Metodología: Resolver en forma grupal el práctico utilizando internet, computadora y software adecuado.

A continuación se describe cómo se abordan y cómo se evalúan los ejes transversales trabajados en la asignatura:

\*Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática:

¿Cómo se aborda?

Todos los prácticos abordan problemas de la realidad y/o académicos con características desafiantes para la/el estudiante, y que deben resolverse utilizando los conceptos y modelos analizados en la teoría.

¿Cómo se evalúa?

Este eje transversal será evaluado de dos maneras. Por un lado, se realizará una evaluación formativa durante la cursada, solicitándole al/a la estudiante la entrega de ejercicios prácticos, algunos realizados de forma individual y otros de manera grupal. Todas las entregas contarán con una corrección grupal y/o individual a lo largo del proceso en el que se desarrolla cada práctico y su correspondiente entrega. El objetivo de esta corrección es que las/los estudiantes desarrollen de manera progresiva su habilidad para resolver problemas, puedan medir su nivel de apropiación de los conocimientos y, a su vez, que las/los docentes puedan guiar el proceso de aprendizaje de cada uno de ellos.

Por otro lado, se contempla una evaluación sumativa en la que las/los estudiantes deberán resolver ejercicios al estilo de los prácticos, en el marco de instancias formales de evaluación con sus respectivas recuperaciones, con el fin de acreditar los conocimientos adquiridos durante las actividades prácticas.

\*Utilización de técnicas y herramientas de aplicación de ingeniería en sistemas de información/informática:

¿Cómo se aborda?

Para realizar los prácticos 2 al 5 se utilizará la herramienta de diseño de modelos formales JFlap o similar y de libre disposición en internet;

¿Cómo se evalúa?

Se evaluará de manera formativa, incremental y continua, verificando la habilidad adquirida por la/el estudiante durante las actividades prácticas en la utilización y manipulación de las herramientas.

\* Fundamentos para el desempeño en equipos de trabajo:

¿Cómo se aborda?

El desempeño en equipos de trabajo se aborda principalmente en las entregas de ejercicios prácticos, donde, a partir de pautas establecidas por las/los docentes del curso, las/los estudiantes deben organizarse como equipo para seleccionar el problema, distribuir las actividades y resolver la consigna propuesta.

¿Cómo se evalúa?

El desempeño será evaluado considerando la capacidad y habilidad de las/los estudiantes para seguir las pautas establecidas por las/los responsables del curso, llevar a cabo los actividades en un clima de colaboración y coordinación entre todos los integrantes del equipo, y obtener una resolución del práctico que evidencie, tanto en el informe escrito como en la exposición oral, una comprensión compartida de la solución propuesta y el cumplimiento adecuado del rol de cada integrante.

\*Fundamentos para una comunicación efectiva.:

¿Cómo se aborda?

Este eje se aborda considerando dos dimensiones. Por un lado, la expresión oral, a partir de la participación de los/las estudiantes en cada una de las actividades del curso (clases teóricas, prácticas, entregas, debates, entre otras), tanto con las/los docentes como con las/los demás estudiantes. Por otro lado, se considera la expresión escrita, presente en los informes de las actividades a entregar, así como en las evaluaciones prácticas y teóricas.

¿Cómo se evalúa?

La expresión oral se evalúa formativamente, observando la capacidad y habilidad de las/los estudiantes para adquirir y utilizar el vocabulario y las expresiones propias del temario del curso de una manera efectiva y eficaz durante sus participaciones orales. La expresión escrita, por su parte, se evaluará a partir de la capacidad de la/del estudiante para emplear un vocabulario riguroso, científico y adecuado, tanto desde la perspectiva del área disciplinar como del temario propio del curso, en los informes a entregar y en las evaluaciones prácticas y teóricas.

\*Fundamentos para el aprendizaje continuo:

¿Cómo se aborda?

Los conceptos abordados en las unidades se relacionan entre sí y con conceptos aprendidos en actividades curriculares de años anteriores; por lo tanto, las/los estudiantes realizarán un proceso de aprendizaje continuo e incremental a medida que avanza el desarrollo de las unidades, tanto desde el punto de vista teórico como práctico. Este proceso se llevará a cabo mediante la participación activa en las actividades teóricas, analizando los apuntes propios del curso y la bibliografía propuesta, así como el acceso a material disponible públicamente en internet que la/el estudiante considere pertinente. Dicho proceso se concretará también a través de las entregas que deban realizarse durante el curso.

¿Cómo se evalúa?

La evaluación del aprendizaje continuo se realizará mediante la observación de la participación e interacción de las/los estudiantes en cada actividad áulica, la corrección de las actividades prácticas y el posterior debate grupal, instancia en la que cada estudiante podrá analizar y determinar su propio nivel de apropiación de los contenidos teóricos y prácticos.

## VIII - Regimen de Aprobación

La propuesta de trabajo, en lo que respecta al régimen de aprobación, está basada en la evaluación continua o formativa, con el objetivo de relacionar la información sobre la evolución del proceso de aprendizaje de los/las estudiantes con las

características de la acción didáctica, a medida que se desarrollan las actividades de enseñanza y aprendizaje. En este sentido, la/el estudiante puede regularizar -para luego rendir el examen final- o promocionar. Las condiciones y la metodología propuesta para la aprobación del curso son las siguientes:

**A. Régimen de Regularidad:**

- 1- Los/las estudiantes deben alcanzar un porcentaje mínimo del } 60% de asistencia y participación activa en las clases y/o actividades prácticas propuestas por las/los docentes durante el cursado de la asignatura.
- 2- Entregar en tiempo y forma y aprobar el 100% de las evaluaciones periódicas propuestas por la cátedra, las cuales consistirán en ejercicios de los prácticos de aula y, de ser necesario, ejercicios adicionales referidos a las temáticas abordadas en los prácticos. Como parte del proceso de aprendizaje y con la finalidad de promover un trabajo colaborativo de comprensión, los docentes realizarán devoluciones evaluativas de las entregas. Asimismo, estas evaluaciones podrán incluir una instancia de defensa oral posterior a la entrega, en la que la/el estudiante exponga, explique y responda las dudas o inquietudes de sus compañeras/os y/o docentes acerca del trabajo realizado.
- 3- Entregar en tiempo y forma y aprobar el práctico de laboratorio.
- 4- Cada evaluación y el práctico de laboratorio se aprueban con un porcentaje mínimo de setenta por ciento (70%) sobre el puntaje total.
- 5- Aprobar 1 examen parcial práctico o alguna de sus dos respectivas recuperaciones, conforme a lo establecido por la reglamentación vigente.
- 6- El examen parcial se aprueba alcanzando un mínimo de setenta por ciento (70%) sobre el puntaje total, y además, desarrollando correctamente al menos el 50% de cada uno de los ejercicios involucrados.
- 7- Si cualquiera de los puntos anteriores no fuera cumplimentado, la/el estudiante pasa a condición de libre.

**B. Régimen de Aprobación por Promoción sin Examen Final:**

- 1- Ídem a lo requerido para estudiantes regulares, con excepción del porcentaje de asistencia, que debe alcanzar el mínimo establecido por la reglamentación vigente (80%).
- 2- La/El estudiante deberá rendir y aprobar un coloquio integrador, oral o escrito, en el que demuestre su capacidad de vinculación, integración y análisis de los conceptos teóricos desarrollados durante el curso.
- 3- El examen integrador se aprueba alcanzando un mínimo del setenta por ciento (70%) sobre el puntaje total y habiendo desarrollado correctamente al menos el 50% de cada uno de los ejercicios involucrados.
- 4- La nota final de promoción se calculará promediando la nota promedio de las evaluaciones del punto A y la nota obtenida en el coloquio integrador del punto B. Cada instancia de evaluación (examen parcial, evaluaciones periódicas, práctico de laboratorio, coloquio integrador de promoción, entre otras) se aprueba con una nota igual o superior a 7, conforme a la normativa vigente.

**C. Régimen de Aprobación por Examen Final:**

- 1- El examen final puede ser oral y/o escrito. Para su aprobación, será necesario desarrollar correctamente al menos el 70% de cada punto involucrado en el examen.

**D. El curso no admite alumnos en condición de Libre.**

## **IX - Bibliografía Básica**

- [1] Hopcroft J.; Ullman J.; Motwani R. – “Introduction to automata theory, languages and computation”. 2nd ed. Addison-Wesley, 2001, ISBN: 0201441241
- [2] Sudkamp, T.A. - “Languages and Machines (An Introduction to the Theory of Computer Science)”. 2nd ed. Addison Wesley. 1997 ISBN: 0201821362
- [3] Sipser, M. – “Introduction to the Theory of Computation”. 1st ed. PWS Publishing Company. 1997. ISBN: 053494728X
- [4] Linz Peter – “An introduction to formal languages and automata”. 6th ed. Jones & Bartlett Learning, 2017. ISBN: 9781284077247
- [5] Gutú, Olivia - "Lenguajes formales y autómatas". Universidad de Sonora. 2019. URL: [https://www.mat.uson.mx/~oliviagutu/LIBRO\\_AUTOMATAS.pdf](https://www.mat.uson.mx/~oliviagutu/LIBRO_AUTOMATAS.pdf)
- [6] Notas de Clase de la Cátedra.

## X - Bibliografía Complementaria

- [1] Rodger, S. y Finley, T. - JFLAP – An Interactive Formal Languages and Automata Package (2008). (URL <https://www.jflap.org/jflapbook/jflapbook2006.pdf>)
- [2] Gómez, D. y Pardo, L. M. y Tirnauca, C. - "Lenguajes Formales (para Ingenieros Informáticos)". Universidad de Cantabria (2015). URL: [https://ocw.unican.es/pluginfile.php/2209/course/section/2081/material\\_teorico\\_del\\_curso\\_lenguajes\\_formales.pdf](https://ocw.unican.es/pluginfile.php/2209/course/section/2081/material_teorico_del_curso_lenguajes_formales.pdf)
- [3] Elena Jurado Málaga - "Teoría de autómatas y lenguajes formales". Universidad de Extremadura (2008). URL: <https://dehesa.unex.es/bitstreams/727aecb0-af02-4534-8b4c-c04ce6028601/download>
- [4] Martin, J. - Introduction to languages and the theory of computation (2010).
- [5] Wood, D. – "Theory of computation". John Wiley & Sons, Inc. (1988).
- [6] Brena, Ramón - "Autómatas y Lenguajes: Un enfoque de diseño", ITESM, México, (2003) URL: <https://lenguajesformalesyautomatas.wordpress.com/wp-content/uploads/2017/11/automatas-y-lenguajes.pdf>

## XI - Resumen de Objetivos

Al finalizar el curso se espera que el alumno sea capaz de: (a) comprender la teoría de los lenguajes formales y su jerarquización; (b) discernir las diferencias entre las clases de lenguajes formales; (c) entender el funcionamiento de los modelos formales de cómputo (autómatas finitos, expresiones regulares, gramáticas libres del contexto, autómatas push-down, máquina de turing, autómata linealmente acotado) y sus relaciones con los lenguajes formales; (d) relacionar los lenguajes formales con la estructura y formas de los lenguajes de programación; (e) conocer la relación entre problemas, problemas decidibles y lenguajes formales, lenguajes decidibles; (f) comprender con cierta profundidad la teoría de la computabilidad y sus implicancias en la posibilidad de resolver o no problemas (lenguajes decidibles y no decidibles); y (g) comprender también con cierta profundidad la teoría de la complejidad computacional y a través de ella categorizar a los problemas de acuerdo a su propiedades y dificultades para resolverlos (problemas de la clases P, NP y NP-Completo).

## XII - Resumen del Programa

Unidad 1: Alfabetos. Cadenas. Lenguajes. Representación de Lenguajes. Jerarquía de Chomsky.

Unidad 2: Lenguajes Regulares. Expresiones Regulares. Autómatas Finitos. Equivalencias. Minimización. Propiedades.

Unidad 3: Gramáticas Libres del Contexto. Autómatas Push-Down. BNF. BNF-extendida

Unidad 4: Máquinas de Turing. Extensiones a la Máquinas de Turing.

Unidad 5: Teoría de la Computabilidad. Relación entre problemas y lenguajes. Lenguajes Recursivos y Recursivamente Enumerables. Problemas decidibles y no decidibles.

Unidad 6: Teoría de la Complejidad Computacional. Clases de complejidad temporal y espacial. Problemas P y NP. Problema NP-completos.

## XIII - Imprevistos

Correo de contacto: javierma@email.unsl.edu.ar (Javier Apolloni)

## XIV - Otros

**ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA****Profesor Responsable**

Firma:

Aclaración:

Fecha: