

<b>Carrera INGENIERIA EN INFORMATICA</b>		
<b>Asignatura 3657 – Autómatas y Gramáticas</b>		
<b>Trayecto de Programación</b>		
<b>Año académico 2023</b>		
<b>Responsable / Jefe de cátedra</b> Lic. Mara Capuya		
<b>Carga horaria semanal</b> 4hs	<b>Carga horaria total</b> 64hs	<b>Créditos</b> -----
<b>Modalidad:</b> Presencial		
<b>Correlativas anteriores</b> PARADIGMAS DE PROGRAMACION	<b>Correlativas posteriores</b> LENGUAJES Y COMPILADORES	
<b>Conocimientos necesarios</b> -----		

### Descripción de la asignatura

Esta asignatura permite al alumno poder conocer los modelos de autómatas y de esta manera lograr que pueda 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 como son el diseño de compiladores, circuitos, interfaces, traducción entre otras. También podrá evaluar la computabilidad y la complejidad de problemas.

### Metodología de enseñanza

Se dedica el 30% del tiempo a clases teóricas expositivas, 70% a la resolución de ejercicios prácticos. Las clases teóricas se soportan en videos dinámicos y animados grabados con audios explicativos que se encuentran en la plataforma MiEL  
Dentro de la plataforma Miel, los alumnos podrán encontrar además de las clases teóricas antes mencionadas, papers de interés sobre temas novedosos y guía de trabajos prácticos, normas de cátedra, bibliografía, cronograma de clases y novedades.

### Objetivos de aprendizaje

Al finalizar el curso, el alumno será capaz de:

- Conocer los métodos formales para describir la sintaxis de todos los lenguajes estructurados gramaticalmente (gramáticas y su jerarquía, expresiones regulares)
- Conocer los distintos modelos de autómatas.
- Captar las ideas teóricas que apuntarán a la resolución de problemas reales mediante el alto poder de abstracción que poseen estos modelos.
- Entender la Máquina de Turing como modelo de cómputo de funciones
- Conocer la tesis de Church-Turing y el Halting problem.

- Descubrir los problemas que pueden resolverse con una computadora
- Medir la complejidad de algoritmos. Entender los problemas P y NP
- Descubrir los problemas que pueden resolverse eficientemente con una computadora

#### Contenidos mínimos

Gramáticas formales y autómatas. Sistema de producción y lenguajes. Gramáticas regulares y otras. Autómatas finitos, push down. Máquinas de Turing. Lenguajes naturales y regulares, autómatas determinísticos y no determinísticos. Método de Thompson. Relaciones biunívocas entre expresiones regulares y autómatas finitos. Forma normal de Chomsky. Ambigüedad. Equivalencias. Propiedades de clausura. Restricciones. Gramáticas LR, LL y de precedencia.

#### Competencias para desarrollar

##### Genéricas

- Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.
- Desempeño en equipos de trabajo.
- Comunicación efectiva.
- Actuación profesional ética y responsable.
- Aprendizaje continuo.
- Desarrollo de una actitud profesional emprendedora.
- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.

##### Específicas

- Especificación, proyecto y desarrollo de software.

Programa analítico	
Unidad 1	<b>Introducción a los lenguajes formales</b> Concepto de gramáticas formales y autómatas. Alfabeto, cadenas y lenguajes. Operaciones entre cadenas y lenguajes. Propiedades. Sistema de producción y lenguajes
Unidad 2	<b>Lenguajes regulares</b>

	Lenguajes regulares. Gramáticas regulares. Expresiones regulares. Equivalencia entre gramáticas y expresiones regulares. Leyes algebraicas de expresiones regulares. Propiedades de clausura de los lenguajes regulares.
Unidad 3	<b>Autómatas finitos</b> 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 de expresiones regulares y autómatas finitos. Método de Thompson. Relaciones biunívocas entre expresiones regulares y autómatas finitos Lema de Pumping para lenguajes regulares.
Unidad 4	Lenguajes libres del contexto. Gramáticas libres de contexto. Ambigüedad. BNF (Backus Naur Form). Base gramatical de la traducción de lenguajes. Forma normal de Chomsky. Ambigüedad. Equivalencias. Propiedades de clausura. Restricciones. Gramáticas LR, LL y de precedencia.
Unidad 5	Autómata con pila: definiciones básicas; diagrama de transición; ejemplos. Autómata con pila determinístico: reconocimiento y traducción. Autómata con pila no determinístico. Equivalencia entre gramáticas libres de contexto y autómatas con pila.
Unidad 6	Lenguajes naturales Máquina de Turing: definiciones básicas; ejemplos. Máquina de Turing determinística: reconocimiento y traducción. Máquina de Turing multicinta. Máquina de Turing no determinística. Aceptación y Rechazo en Máquinas de Turing. Los lenguajes recursivamente enumerables y la Máquina de Turing. Los lenguajes recursivos y la Máquina de Turing. Conceptos básicos de teoría de computabilidad. El enfoque funcional de Church. La Máquina de Turing como modelo general de cómputo. La Tesis de Church-Turing. Los límites de la computabilidad: visión general.
Unidad 7	Computabilidad - Límites Problemas computables y no computables. Máquina de Turing Universal. Programas autoreferentes. Problema de la detención o halting problem: su demostración con Máquinas de Turing. Problemas decidibles e indecidibles. Los límites de la computabilidad.
Unidad 8	Conceptos básicos de Complejidad Computacional. Complejidad de algoritmos. Complejidad temporal y espacial. Medición de la complejidad temporal y espacial sobre máquinas de Turing. Medición de la complejidad temporal y espacial sobre algoritmos. Teorema de Blum. Medición asintótica: asíntota superior, inferior y orden exacto. Orden asintótico y sus propiedades. Benchmarking. Complejidad en algoritmos recursivos y de ordenamiento.
Unidad 9	Problemas tratables e intratables. Clases P y NP. Los problemas P-NP y las máquinas de Turing. Verificación de problemas. Reducción

polinómica. Problemas NP-Completos. El problema de satisfacibilidad.  
Teorema de Cook. NP-completitud: prueba. Problema abierto:  $P \neq NP$ .

<b>Planificación de actividades</b>					
Semana	Clase	Actividad	Tipo	Duración estimada	Unidad
Semana 1	Presencial	Clase	Teoría-Práctica	4	1-2
Semana 2	Presencial	Clase y ejercicios		4	3
Semana 3	Presencial	Clase y ejercicios		4	4
Semana 4	Presencial	Clase y ejercicios		4	5
Semana 5	Presencial	Clase de Consultas	Práctica	4	1-2-3-4-5
Semana 6	Presencial	FERIADO			
Semana 7	Presencial	Primer Parcial			
Semana 8	Presencial	Clase	Teórica	4	6
Semana 9	Presencial	Clase y ejercicios		4	6
Semana 10	Presencial	Clase y ejercicios		4	7
Semana 11	Presencial	Clase y ejercicios		4	8-9
Semana 12	Presencial	Clase de Consultas	Práctica	4	6-7-8-9
Semana 13	Presencial	FERIADO			
Semana 14	Presencial	Segundo Parcial			
Semana 15	Presencial	Recuperatorio			
Semana 16	Presencial	Cierre			

**Evaluación**

Se evaluará a través de dos exámenes parciales.

Habrá un solo examen recuperatorio para el primer o segundo examen parcial.

Los exámenes parciales y el examen recuperatorio serán escritos conteniendo preguntas y ejercicios que permitan enunciar, describir, aplicar y relacionar algunos de los conceptos presentados a lo largo del curso.

Primer examen parcial:

- abarcará los contenidos de las unidades 1 a 5
- la modalidad será escrita con posibilidad de varios temas

Segundo examen parcial

- abarcará los contenidos de las unidades 6 a 9
- se dividirá en dos partes, la primera consistirá en reunir grupos aleatorios de no más de 5 alumnos a los que se les dará un problema de resolución en máquina (unidad 6) y tendrá una puntuación del 60% del total del segundo parcial.

La segunda parte consistirá en un set de preguntas sobre las unidades 7 a 9 y conformará el 40% de la puntuación del total del segundo parcial

Los exámenes finales podrán ser escritos u orales, conteniendo preguntas y ejercicios que permitan enunciar, describir, aplicar y relacionar todos los conceptos presentados a lo largo del curso.

*Promoción:*

Se obtiene aprobando los parciales o su recuperatorio con 7 (siete) o más puntos

*Examen libre:*

Con tres meses de anticipación a la fecha de sustanciación de mesa, el alumno deberá entregar la resolución de todas las actividades prácticas (incluidos los archivos desarrollados en las herramientas de software).

El alumno deberá presentarse tres meses antes para obtener el enunciado de un trabajo práctico por cada unidad. Los mismos se deberán entregar y defender el día del examen.

El examen libre abarcará los contenidos de todas las unidades temáticas. El alumno será evaluado en primer lugar con un examen escrito. De estar aprobada la instancia escrita, deberá rendir un examen de forma oral.

Exámenes Parciales

<b>Primera evaluación</b>	Semana 7	Examen escrito	2
<b>Segunda evaluación</b>	Semana 14	Examen escrito	2
<b>Recuperatorio</b>	Semana 15	Examen escrito	2

**Bibliografía obligatoria**

Titulo	Autor	Editorial	Edición	Año
Compiladores, Principios, Técnicas y Herramientas	Aho A. V., Lam, M., SEthi, R. Ullman J.	Pearson Addison-Wesley Iberoamericana		2008

**Bibliografía complementaria recomendada**

Titulo	Autor	Editorial	Edición	Año
<u>Teoría de la computación: lenguajes formales, autómatas y complejidad</u>	Brookshear, J. Glenn	Addison-Wesley Iberoamericana		1993
Teoría de Autómatas, Lenguajes y computación	J.Hopcroft, R. Motwani, J.Ullman	Person- Addison Wesley Iberoamericana	11	2010
An introduction to Formal Languages and Automata	Peter Linz	Jones & Bartlett Publ.	5	2012
Computability, Complexity and Languages, fundamentals of theoretical computer science	Davis Martin , Sigal Ron, Weyuker Elaine	Academic Press	2	1994
Lenguajes Formales y Teoría de la Computación	Martin John	McGraw-Hill	3	2004
<u>Languages and machines: an introduction to the theory of computer science</u>	Michael L. Scott	Sudkamp, Thomas A.	2	1998
Construcción de Compiladores: Principios y Práctica	Louden,K.	Thomson		2004
Programming Language Concepts	Ghezzi Carlo	John Wiley & Sons		1997

**Otros recursos obligatorios** (videos, enlaces, otros) Incluir una fila por cada recurso

Nombre

**Otros recursos complementarios** (videos, enlaces, otros) Incluir una fila por cada recurso

Nombre