Facultad de Ingeniería - Universidad Nacional de Cuyo P1- PROGRAMA DE ASIGNATURA Adecuación a las modalidades presencial y a distancia por Pandemia COVID-19			
Asignatura:	**LENGUAJES FORMALES Y COMPUTABILIDAD**		
Docente Responsable:	**PROFESOR TITULAR: DRA. ANA CAROLINA OLIVERA**		
Carrera:	**Licenciatura en Ciencias de la Computación**		
Año: 2021	**Semestre: 2DO**	**Horas Semestre: 112**	**Horas Semana: 7**

OBJETIVOS

- Adquirir conocimientos sobre lenguajes formales, teoría de autómatas y sus relaciones.
- Distinguir conceptos y procedimientos de las gramáticas libres de contexto y gramáticas regulares para especificar la sintaxis de los lenguajes de programación.
- Identificar los distintos tipos de autómatas y tipos de notaciones gramaticales.
- Reconocer los elementos propios de la sintaxis y semántica de los lenguajes de programación.
- Analizar el procesamiento de lenguajes y en particular, el proceso de compilación e interpretación.
- Incorporar y reconocer las limitaciones de la computabilidad efectiva a través de distintos formalismos presentados en la asignatura.

CONTENIDOS

UNIDAD 1: AUTÓMATAS FINITOS Y LENGUAJES REGULARES

1.1. Reconocedores. Traductores. Diagramas de Estado. Equivalencia entre Autómatas Finitos deterministas y no deterministas. Morfismos sobre autómatas. Autómata cociente.
1.2. Propiedades de los lenguajes aceptados por autómatas finitos.
1.3. Expresiones regulares. Relación entre gramáticas regulares, lenguajes regulares y autómatas finitos.
1.4. Herramientas computacionales para reconocer lenguajes regulares, programar y simular autómatas.
1.5. Isomorfismos. Pumping Theorem para lenguajes regulares.

UNIDAD 2: LENGUAJES LIBRES DE CONTEXTO, AUTÓMATAS CON PILA Y LENGUAJES SENSIBLES AL CONTEXTO

2.1. Gramáticas libres de contexto. Relación entre gramáticas regulares y libres de contexto. Jerarquía de Chomsky. Árboles de derivación. Ambigüedad. Formas Normales: de Chomsky y BNF.
2.2. Autómata con Pila. Relación entre autómatas con pila y gramáticas sensibles al contexto.
2.3. Lenguajes Sensibles al contexto. Características procedimentales de los lenguajes de programación que se corresponden con lenguajes libres de contexto y sensibles al contexto.
2.4. Introducción a la definición de compilador e interprete: Pasos del proceso de compilación y su relación con los autómatas estudiados. Análisis léxico. Análisis Sintáctico. Análisis semántico.
2.5. Ventajas y desventajas del uso de intérpretes. compiladores e intérpretes. Representación

de datos en memoria. Manejo de memoria en ejecución.

UNIDAD 3: MÁQUINA DE TURING

3.1. Procedimiento efectivo. Funciones Turing Computables. Definición de Máquina de Turing y Autómata acotado linealmente. Configuración. Transición. Aceptación de Cadenas. 3.2. Lenguajes Turing-Decidibles.
3.3. Extensiones de la Máquina de Turing estándar: máquinas con múltiples cintas y no deterministas.

UNIDAD 4: FUNCIONES RECURSIVAS PARCIALES

5.1. Funciones recursivas primitivas sobre los naturales unido el cero. Funciones recursivas primitivas sobre cadenas. Predicados recursivos primitivos. Minimización no acotada.
5.2. Relación entre funciones recursivas parciales y máquinas de Turing. Teorema de Kleene.
5.3. Relación entre las funciones recursivas parciales y los lenguajes de programación.

UNIDAD 5: LIMITACIONES DE LA COMPUTABILIDAD

4.1. Conceptos básicos de teoría de la computabilidad y complejidad: Problemas computables y no computables.
4.2. Tesis de Turing-Church. Máquina Universal de Turing.
4.3. Lenguajes Recursivos y recursivos enumerables.
4.4. Problemas de decisión. El problema de la detención de la máquina de Turing. Teoría de la reducibilidad. Reducibilidad al problema de la detención.
4.5. Gramáticas estructuras por Frases. Poder computacional de las gramáticas estructuradas por frases en relación con el de las máquinas de Turing. Técnicas de prueba.

UNIDAD 6: INTRODUCCIÓN A LA COMPLEJIDAD COMPUTACIONAL

6.1. Problemas centrales. Análisis de algoritmos. Tratabilidad computacional.
6.2. Complejidad: P y NP. Problemas NP-Completos, NP-Difíciles.

UNIDAD 7: INTRODUCCIÓN A COMPILADORES E INTERPRETES

7.1 Conceptos de compiladores e intérpretes. Lenguajes de Programación y su relación con las gramáticas formales.

7.2. Manejo de datos en memoria y su representación. Estrategias de implementación. Manejo de memoria en ejecución.

METODOLOGÍA DE ENSEÑANZA

Por tratarse de una asignatura con importante respaldo conceptual, la teoría ocupa un lugar destacado, y para facilitar la apropiación de los saberes se aplicará la ejemplificación y demostración de manera metódica. Así, la enseñanza de la materia se apoya en una estrecha coordinación entre el dictado de los conceptos teóricos, presentación de ejemplos y la aplicación de la teoría en la resolución de ejercicios. Los ejercicios están orientados al análisis o entendimiento de las soluciones propuestas y a la concepción o diseño para responder a nuevos requerimientos. Para ello el material de práctica es seleccionado de manera de cubrir un amplio espectro temático y muestra diferentes niveles de dificultad. Algunos ejercicios son resueltos completamente en clase y otros son encaminados para ser completados por los alumnos fuera de clase.

En el contexto de pandemia COVID 19, la modalidad a distancia adoptada utiliza la plataforma de Aula Abierta, meet y/o jitsi para el desarrollo de las clases teóricas en vivo en los horarios habituales de cursada sumado a herramientas como diapositivas (PowerPoint/PDF) para ayudar al alumno en el seguimiento de la clase. Además, se utiliza una herramienta para escribir sobre una pizarra virtual que permite simular el encuentro en el aula lo que facilita la interacción docente-alumnos.

DISTRIBUCIÓN DE LA CARGA HORARIA

Actividad	Carga horaria por semestre
Teoría y resolución de ejercicios simples	70
Formación práctica	
Formación Experimental – Laboratorio	0
Formación Experimental - Trabajo de campo	0
Resolución de problemas de ingeniería	42
Proyecto y diseño	0
Total	**112**

Porcentaje de Horas Presenciales	0 % del Total
Porcentaje de Horas a Distancia	100% del Total

BIBLIOGRAFÍA

Bibliografía básica

Autor	Título	Editorial	Año	Ejemplares en biblioteca
H. R. Lewis, C. Papadimitriou	Elements of theory computation	Prentice Hall	1998	
J. Hopcroft, R. Motwani, J. Ullman	Teoría de autómatas, lenguajes y computacion. 3ra Edicion	Pearson	2008	
M. Sipser	Introduction to the theory of computation 2°.ed.	PWS	2006	
D. Cohen	Introduction to computer theory	Wiley	1992	
AHO, A. et. al	Compiladores. Principios, técnicas y herramientas	Pearson Educación	2008	

Bibliografía complementaria

Autor	Título	Editorial	Año	Ejemplares en biblioteca
N. CHOMSKY	Three models for the description of language. IEEE Transactions on Information Theory. 3(2): 113-124	IEEE	1956	
A.M. TURING	On computable numbers, with an application to the entsheidungs problem. Proccedings of the London Mathematical Society. 42:230-265	London Mathematical Society	1937	

EVALUACIONES (S/ Ord. 108-10_CS)

Se propone una evaluación continua de alumno de acuerdo al desarrollo ejercicios entregables y

un examen parcial a distancia. Se presenta a continuación las normas para regularizar y aprobar la materia.

Evaluaciones durante el cursado:

- Examen parcial. Se evaluarán dos exámenes parciales aquellos conceptos teóricos-prácticos que sean relevantes para alcanzar la regularidad de la asignatura.

Evaluación Recuperatoria y Global:

- Los exámenes parciales tendrán su correspondiente recuperatorio global.

Condición de Regularidad

En base a los resultados de las evaluaciones el alumno quedará como Alumno Regular, cuando haya aprobado los correspondientes exámenes parciales o su recuperatorio global.

Alumnos recursantes. No hay régimen especial para alumnos recursantes.

Criterios de evaluación:

En cuanto a alumnos regulares: se valorará la precisión del alumno en las respuestas a las preguntas realizadas durante el examen oral. La coherencia de sus respuestas entre sí junto con el grado de conocimiento demostrado en la capacidad de relacionar los conceptos vistos durante la cursada. La madurez conceptual evidenciada en la capacidad de respuestas rápidas concisas y precisas a consultas realizadas. El conocimiento de los modelos y su potencialidad.

En cuanto a alumnos libres: se evaluará la capacidad del alumno para desarrollar gramáticas, autómatas y diseñar funciones recursivas parciales. Se valorará la precisión del alumno en las respuestas a las preguntas realizadas durante el examen oral. La coherencia de sus respuestas entre sí junto con el grado de conocimiento demostrado en la capacidad de relacionar los conceptos de la asignatura. La madurez conceptual evidenciada en la capacidad de respuestas rápidas, concisas y precisas a consultas realizadas. El conocimiento de las fases del desarrollo de un compilador tanto a nivel teórico como las dificultades a nivel práctico.

Condiciones para la acreditación de la asignatura:

Examen Final alumnos regulares. El examen final es de tipo integrador teórico-práctico, de forma oral o escrita, sobre cualquiera de los temas desarrollados en la asignatura. El examen final se rinde a programa completo, exigiéndose el nivel superior correspondiente. independientemente que se hayan tomado o no en las evaluaciones parciales. Todos los temas evaluados deben conocerse en al menos un 60% del alcance desarrollado en la materia. Podrán rendir examen final aquellos alumnos regulares.

Examen final para alumnos libres. El examen final es de tipo integrador teórico-práctico, de forma escrita, sobre cualquiera de los temas desarrollados en la asignatura. En caso de que los ejercicios estén aprobados con una nota superior o igual a 6 (SEIS) el alumno libre deberá rendir un examen teórico-práctico oral en el mismo momento y lugar según los criterios de evaluación mencionados previamente. El examen final libre se rinde a programa completo, exigiéndose el nivel superior correspondiente.

Programa de examen:

Contempla la totalidad de los temas del presente programa.

FECHA, FIRMA Y ACLARACIÓN TITULAR DE CÁTEDRA