

**FUNDAMENTOS DE TEORÍA DE LA
COMPUTACIÓN****Carrera/ Plan***Licenciatura en Sistemas Planes 2015, 2012, 2003-07***Año 2020****Año:** 4^o**Régimen de Cursada:** Semestral (1ro)**Carácter:** Obligatoria**Correlativas:** Matemática III - Conceptos y Paradigmas de
Lenguajes de Programación**Profesor/es:** Claudia Pons y Ricardo Rosenfeld**Hs. semanales:** 6 hs.**FUNDAMENTACIÓN**

Esta asignatura introduce al alumno en los fundamentos teóricos de la computación. La teoría de la computación es una rama de la matemática y la computación, que centra su interés en las limitaciones y capacidades fundamentales de las computadoras.

En este contexto, se busca que el alumno comprenda que existen paradigmas primigenios y fundamentales, cuyo conocimiento le permitirá enfrentarse con solvencia a nuevos desarrollos teóricos.

Efectos paralelos y deseables, compartidos con otras asignaturas, son contribuir a que el alumno ejercite la capacidad de correlacionar, abstraer y concretar pensamientos en el área teórica del dominio de sistemas de información.

OBJETIVOS GENERALES

Familiarizar al estudiante con la teoría de algoritmos, la teoría de máquinas abstractas, la teoría de lenguajes formales y la lógica matemática.

Introducir al estudiante en la base teórica de los sistemas inteligentes.

COMPETENCIAS

- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT4- Conocer e interpretar los conceptos, teorías y métodos matemáticos relativos a la informática, para su aplicación en problemas concretos de la disciplina

CONTENIDOS MINIMOS

Algoritmos y Recursividad

Análisis de Algoritmos

Notación $O()$

Tratabilidad, Computabilidad y Complejidad

Maquinas Matemáticas

Lenguajes Formales y Gramáticas

Lógica Matemática, Inteligencia Artificial Simbólica y No Simbólica

PROGRAMA ANALÍTICO

PARTE 1: COMPUTABILIDAD Y COMPLEJIDAD COMPUTACIONAL

Máquinas de Turing (MT). Distintos modelos de MT. Equivalencia de modelos de MT. Computabilidad y decidibilidad. Lenguajes no recursivamente numerables, recursivamente numerables y recursivos. Propiedades de dichos lenguajes. MT universal. El problema de la detención (Halting Problem) y el problema (de reconocimiento) universal. Diagonalización. Reducción de problemas. MT restringidas. Gramáticas. Jerarquía de Chomsky.

Generalidades de la complejidad computacional temporal y espacial de problemas. Representación de problemas. Jerarquía temporal. Tiempo polinomial. Clases de problemas P y NP. Reducción polinomial de problemas. NP-completitud. El problema de la satisfactibilidad de las fórmulas booleanas (SAT). Teorema de Cook. Algunas misceláneas de complejidad temporal y espacial.

PARTE 2: FUNDAMENTOS TEÓRICOS DE SISTEMAS INTELIGENTES

Fundamentos de Inteligencia Artificial Simbólica y No-Simbólica. Formalismos de representación de conocimiento. Lógica de primer orden. Enfoque sintáctico y semántico. Teoría de la demostración. Técnicas de prueba. Estructura de las pruebas formales. Principio de resolución de Robinson.

Introducción a la Inteligencia Artificial: agentes y su ambiente. Racionalidad. Agentes inteligentes, sistemas expertos. Teoría de Aprendizaje Automático, redes neuronales artificiales, algoritmos genéticos, inferencia probabilística (redes bayesianas), Aplicaciones de la inteligencia artificial. Introducción a la verificación axiomática de programas.

BIBLIOGRAFÍA

Básica

- R. Rosenfeld & J. Irazábal. 2013. *Computabilidad, Complejidad Computacional y Verificación de Programas*. EDULP.
- R. Rosenfeld & J. Irazábal. 2010. *Teoría de la Computación y Verificación de Programas*. McGraw Hill y EDULP.
- C. Pons, R. Rosenfeld & C. Smith. 2017. *Lógica para Informática*. EDULP.
- A. Hamilton. 1980. *Logic for Mathematicians*. Cambridge University Press.
- S. Russell & P. Norvig. 2010. *Artificial Intelligence. A Modern Approach*. Prentice Hall.

Complementaria

- Hopcroft & Ullman. 1979. *Introduction to Automata Theory, Language & Computation*. Prentice-Hall.
- S. Arora & B. Barak. 2007. *Computational Complexity: A Modern Approach*. Princeton Univ.
- C. Papadimitriou. 1995. *Computational Complexity*. Addison-Wesley.
- O. Goldreich. 2008. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press.
- Bovet & Crescenzi. *Introduction to the Theory of Complexity*. Prentice-Hall. 1994.
- C. Moore & S. Mertens. 2011. *The Nature of Computation*. Oxford University Press.
- N. Francez. 1992. *Program Verification*. Addison-Wesley.

METODOLOGÍA DE ENSEÑANZA

La asignatura consiste en el dictado de 14 clases de teoría y 14 clases de ejercitación (clases prácticas), ámbas estrechamente vinculadas y articuladas, distribuidas en dos partes.

En las clases teóricas se brindan explicaciones conceptuales, con participación e intercambio con los alumnos, que servirá para que los mismos logren resolver satisfactoriamente los trabajos prácticos propuestos.

En las clases prácticas se trabaja a partir del enunciado de ejercicios que se resuelven en las mismas clases, con plena participación de los alumnos.

Para asegurar el aprendizaje de los contenidos dictados, se entrega semanalmente un trabajo práctico, a resolver por los alumnos.

Se utiliza una plataforma de gestión de cursos para la publicación de las clases, trabajos prácticos y artículos de interés. También para las consultas de los alumnos, promoviendo un foro de discusión permanente.

Particularmente, para alcanzar las competencias indicadas previamente:

CGT1- Identificar, formular y resolver problemas de Informática:

En la cátedra se pone énfasis en el proceso de identificación de problemas del mundo real, su especificación como problemas resolubles desde la informática, y el desarrollo de soluciones verificables para los mismos.

CGT4- Conocer e interpretar los conceptos, teorías y métodos matemáticos relativos a la informática, para su aplicación en problemas concretos de la disciplina:

Se intenta poner al alumno en el contexto de aplicación en el campo de la Informática de los conceptos y métodos matemáticos que se enseñan en el programa de la asignatura. Esta contextualización es informativa y se discuten diferentes casos de aplicación para mostrar la utilidad de las teorías y herramientas matemáticas para resolver diferentes problemas informáticos conocidos por el alumno. Además, se pone a disposición de los alumnos material bibliográfico para profundizar la relación entre los temas matemáticos y las soluciones informáticas.

EVALUACIÓN

La aprobación de la cursada se basa en una examinación al final de cada una de las dos partes de la materia.

La aprobación de la materia se basa en otra examinación al final de la materia, considerando sus dos partes (posibles alternativas son un trabajo final o un coloquio).

En cuanto a la evaluación de las competencias indicadas previamente:

CGT1- Identificar, formular y resolver problemas de Informática:

La evaluación de esta competencia forma parte de las evaluaciones de los trabajos prácticos y del examen final de la asignatura, reflejándose en la corrección de los entregables del alumno.

CGT4- Conocer e interpretar los conceptos, teorías y métodos matemáticos relativos a la informática, para su aplicación en problemas concretos de la disciplina:

La evaluación de esta competencia también forma parte de las evaluaciones de los trabajos prácticos y el examen final, donde se incorporan preguntas específicas del tipo: “¿dónde cree Ud. que es aplicable este conocimiento/método matemático?”. La evaluación se refleja en la corrección de los entregables del alumno.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	9/03	PARTE 1 (7 clases). Máquinas de Turing. Distintos modelos de máquinas de Turing y equivalencia entre ellos.
2	16/03	Lenguajes recursivos, recursivamente numerables y no recursivamente numerables. Propiedades. Mapa de la computabilidad.
3	30/03	Lenguajes y problemas de decisión. Máquina de Turing universal. El problema de la detención y del reconocimiento universal. Diagonalización.
4	06/04	Reducciones de problemas. MT restringidas. Gramáticas.
5	13/04	Generalidades de la complejidad computacional. Jerarquía temporal. Representación de problemas.
6	20/04	Tiempo polinomial. Las clases de problemas P y NP.
7	27/04	Reducciones polinomiales de problemas. Problemas NP-completos. El problema de la satisfactibilidad de las fórmulas booleanas (SAT). Teorema de Cook. Propiedades de los problemas NP-completos. Algunas misceláneas de complejidad temporal y espacial.
8	18/05	PARTE 2 (7 clases). Introducción a la Lógica de Enunciados. Conocimiento: definición. Adquisición del conocimiento. Formas de razonamiento. Argumentaciones
9	26/05	Sintaxis: el lenguaje simbólico de la lógica de enunciados. formas enunciativas o formulas bien formadas. Semántica: interpretación y satisfacción. tablas y funciones de verdad. Tautologías y contradicciones..
10	01/06	Lógica de enunciados. El sistema formal L. Corrección y completitud de L.
11	08/06	Lógica de predicados. Predicados y cuantificadores. Lenguajes de primer orden. Interpretaciones. Satisfacción y verdad. Introducción a la verificación axiomática de programas.
12	15/06	Introducción a la Inteligencia Artificial. Definición de IA., Pasado, presente y futuro de la IA, Usos de la IA., Agentes inteligentes y su ambiente.
13	22/06	Inteligencia artificial simbólica Definición de un agente inteligente usando lógica simbólica. Programación lógica
14	29/06	Inteligencia artificial no-simbólica Machine learning. Proceso de desarrollo de los sistemas ML. Arboles de decisión Redes neuronales artificiales.

Evaluaciones previstas	Fecha
Examen parcial parte 1	11/05
Examen parcial parte 2	14/07
Examen final parte 1	03/08
Examen final parte 2	04/08

Contactos de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

Prof. Claudia Pons (claudia.pons.33@gmail.com, claudia.pons@lifa.info.unlp.edu.ar)

Prof. Ricardo Rosenfeld (rrosenfeld@practia.global)

JTP Leandro Mendoza (leandro.mdza@gmail.com)

Ayudante Ilán Rosenfeld (ilanrosenfeld7@gmail.com)

Plataforma Ideas: Fundamentos de Teoría de la Computación

Profesor Ricardo Fabián Rosenfeld