FUNDAMENTOS DE TEORÍA DE LA COMPUTACIÓN

Año 2026

Carrera/ Plan

Licenciatura en Sistemas Plan 2021/Plan 2015

Año: 4º
Régimen de cursada: Semestral
Carácter: Obligatoria
Correlativas: Matemática III - Conceptos y Paradigmas de Lenguajes de Programación
Profesores: Claudia Pons y Ricardo Rosenfeld
Hs semanales teoría: 2 hs
Hs semanales práctica: 4 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, y de 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.

RESULTADOS DE APRENDIZAJE

2.3. Seleccionar y utilizar los correspondientes métodos analíticos, de simulación y de modelización (Adecuado).

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
Computabilidad y Complejidad Computacional
Máquinas 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. Completitud en la jerarquía espacio-temporal. Algunas misceláneas de complejidad temporal y espacial (problemas de búsqueda, aproximaciones polinomiales, problemas de conteo, algoritmos eficientemente paralelizables, complejidad computacionales en los paradigmas probabilístico y cuántico, etc).

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. Introducción a la verificación axiomática de programas.

Introducción a la Inteligencia Artificial: agentes inteligentes, sistemas expertos. Teoría del aprendizaje automático, redes neuronales artificiales, aplicaciones de la inteligencia artificial.

BIBLIOGRAFÏA

Básica

- R. Rosenfeld & J. Irazábal. 2010. *Teoría de la Computación y Verificación de Programas*. McGraw Hill y EDULP.
- R. Rosenfeld & J. Irazábal. 2013. *Computabilidad, Complejidad Computacional y Verificación de Programas*. EDULP.
- C. Pons, R. Rosenfeld & C. Smith. 2017. Lógica para Informática. EDULP.
- R. Rosenfeld, 2024. *Verificación de Programas. Programas Secuenciales y Concurrentes*. EDULP.
- R. Rosenfeld, 2026. Computabilidad y Complejidad Computacional. EDULP.
- A. Hamilton. 1980. Logic for Mathematicians. Cambridge University Press.
- S. Russell & P. Norvig. 2021 (4ta ed). Artificial Intelligence. A Modern Approach. Pearson Education.

Complementaria

- J. Hopcroft & J. Ullman. 1979. *Introduction to Automata Theory, Language & Computation*. Prentice-Hall.
- M. Sipser. 1997. *Introduction to the Theory of Computation*. PWS Publishing.
- H. Lewis & C. Papadimitriou. 1998. *Elements of the Theory of Computation*. Prentice-Hall.
- D. Bovet & P. Crescenzi. *Introduction to the Theory of Complexity*. Prentice-Hall. 1994
- C. Papadimitriou. 1995. *Computational Complexity*. Addison-Wesley.
- S. Arora & B. Barak. 2007. *Computational Complexity: A Modern Approach*. Princeton Univ.
- O. Goldreich. 2008. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press.
- C. Moore & S. Mertens. 2011. *The Nature of Computation*. Oxford University Press.
- N. Francez. 1992. *Program Verification*. Addison-Wesley.
- K. Apt & E. Olderog, 1997. *Verification of Sequential and Concurrent Programs*. Springer.
- I. Goodfellow, Y. Bengio, A. Courville. 2016. Deep learning. The MIT Press.
- S.J.D. Prince, 2023. Understanding Deep Learning. The MIT Press.

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 sirven 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.

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	**09/03**	**PARTE 1 (7 clases).** Introducción de la materia. Máquinas de Turing (MT). Distintos modelos de MT y equivalencia entre ellos. Ejemplos y ejercicios.
2	**16/03**	Lenguajes recursivos, recursivamente numerables y no recursivamente numerables. Propiedades. Mapa de la computabilidad.

3	**23/03**	Lenguajes y problemas de decisión. MT universal. El problema de la detención y del reconocimiento universal. Diagonalización.
4	**30/03**	Reducciones de problemas. MT restringidas. Gramáticas.
5	**06/04**	Generalidades de la complejidad computacional. Jerarquía temporal. Representación de problemas. Tiempo polinomial. Las clases de problemas P y NP.
6	**13/04**	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.
7	**20/04**	Profundización del mapa de la complejidad temporal. Algunas misceláneas de complejidad temporal. Introducción a la complejidad espacial. Jerarquía espacio-temporal. Misceláneas (problemas de búsqueda y de conteo, aproximaciones polinomiales, circuitos booleanos, algoritmos probabilísticos y cuánticos, etc).
8	**11/05**	**PARTE 2 (7 clases).** Introducción a la Lógica de Enunciados. Conocimiento: definición. Adquisición del conocimiento. Formas de razonamiento. Argumentaciones.
9	**18/05**	Sintaxis: el lenguaje simbólico de la lógica de enunciados. Formas enunciativas. Fórmulas bien formadas. Semántica: interpretación y satisfacción. Tablas y funciones de verdad. Tautologías y contradicciones.
10	**26/05**	Lógica de enunciados. El sistema formal L. Correctitud y completitud de L.
11	**01/06**	Lógica de predicados. Predicados y cuantificadores. Lenguajes de primer orden. Interpretaciones. Satisfacción y verdad. Introducción a la verificación axiomática de programas.
12	**08/06**	Introducción a la verificación axiomática de programas.
13	**15/06**	Introducción a la Inteligencia Artificial (IA). Definición de IA. Pasado, presente y futuro de la IA. Usos de la IA. Agentes inteligentes y su ambiente.
14	**22/06**	Inteligencia artificial simbólica. Definición de un agente inteligente usando lógica simbólica. Programación lógica. Inteligencia artificial no simbólica. Machine learning. Proceso de desarrollo de los sistemas ML. Arboles de decisión. Redes neuronales artificiales.

Evaluaciones promocionales	Fecha
1er Examen parte 1	05/05
1er Examen parte 2	07/07
2do Examen parte 1	03/08
2do Examen 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)
Prof. Ricardo Rosenfeld (rosenfeldricardo@gmail.com)
JTP Leandro Mendoza (leandro.mdza@gmail.com)
Ayudante Pedro Dal Bianco (dalbianco.pedro@gmail.com)
Adscripto Matías Manzín (matias.manzin@gmail.com)
Adscripto Jeremías Salsamendi (jere4352681@gmail.com)
Plataforma Ideas: Fundamentos de Teoría de la Computación

Profesor Ricardo Rosenfeld **Profesora Claudia Pons**

FUNDAMENTOS DE TEORÍA DE LA COMPUTACIÓN (Redictado)

Año 2026

Carrera/ Plan

Licenciatura en Sistemas Plan 2021/Plan 2015

Año: 4º
Régimen de cursada: Semestral
Carácter: Obligatoria
Correlativas: Matemática III - Conceptos y Paradigmas de Lenguajes de Programación
Profesores: Claudia Pons y Ricardo Rosenfeld
Hs semanales teoría: 3 hs
Hs semanales práctica: 4 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, y de 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.

RESULTADOS DE APRENDIZAJE

2.3. Seleccionar y utilizar los correspondientes métodos analíticos, de simulación y de modelización (Adecuado).

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
Computabilidad y Complejidad Computacional
Máquinas 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. Completitud en la jerarquía espacio-temporal. Algunas misceláneas de complejidad temporal y espacial (problemas de búsqueda, aproximaciones polinomiales, problemas de conteo, algoritmos eficientemente paralelizables, complejidad computacionales en los paradigmas probabilístico y cuántico, etc).

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. Introducción a la verificación axiomática de programas.

Introducción a la Inteligencia Artificial: agentes inteligentes, sistemas expertos. Teoría del aprendizaje automático, redes neuronales artificiales, aplicaciones de la inteligencia artificial.

BIBLIOGRAFÏA

Básica

- R. Rosenfeld & J. Irazábal. 2010. *Teoría de la Computación y Verificación de Programas*. McGraw Hill y EDULP.
- R. Rosenfeld & J. Irazábal. 2013. *Computabilidad, Complejidad Computacional y Verificación de Programas*. EDULP.
- C. Pons, R. Rosenfeld & C. Smith. 2017. Lógica para Informática. EDULP.
- R. Rosenfeld, 2024. *Verificación de Programas. Programas Secuenciales y Concurrentes*. EDULP.
- R. Rosenfeld, 2026. Computabilidad y Complejidad Computacional. EDULP.
- A. Hamilton. 1980. Logic for Mathematicians. Cambridge University Press.
- S. Russell & P. Norvig. 2021 (4ta ed). Artificial Intelligence. A Modern Approach. Pearson Education.

Complementaria

- J. Hopcroft & J. Ullman. 1979. *Introduction to Automata Theory, Language & Computation*. Prentice-Hall.
- M. Sipser. 1997. *Introduction to the Theory of Computation*. PWS Publishing.
- H. Lewis & C. Papadimitriou. 1998. *Elements of the Theory of Computation*. Prentice-Hall.
- D. Bovet & P. Crescenzi. *Introduction to the Theory of Complexity*. Prentice-Hall. 1994
- C. Papadimitriou. 1995. *Computational Complexity*. Addison-Wesley.
- S. Arora & B. Barak. 2007. *Computational Complexity: A Modern Approach*. Princeton Univ.
- O. Goldreich. 2008. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press.
- C. Moore & S. Mertens. 2011. *The Nature of Computation*. Oxford University Press.
- N. Francez. 1992. *Program Verification*. Addison-Wesley.
- K. Apt & E. Olderog, 1997. *Verification of Sequential and Concurrent Programs*. Springer.
- I. Goodfellow, Y. Bengio, A. Courville. 2016. Deep learning. The MIT Press.
- S.J.D. Prince, 2023. Understanding Deep Learning. The MIT Press.

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 sirven 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.

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

FECHA DE INICIO: martes 18 /08/ 2026.

Semana	Contenidos/Actividades
1	**PARTE 1 (7 clases).** Introducción de la materia. Máquinas de Turing (MT). Distintos modelos de MT y equivalencia entre ellos. Ejemplos y ejercicios.

2	Lenguajes recursivos, recursivamente numerables y no recursivamente numerables. Propiedades. Mapa de la computabilidad.
3	Lenguajes y problemas de decisión. MT universal. El problema de la detención y del reconocimiento universal. Diagonalización.
4	Reducciones de problemas. MT restringidas. Gramáticas.
5	Generalidades de la complejidad computacional. Jerarquía temporal. Representación de problemas. Tiempo polinomial. Las clases de problemas P y NP.
6	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.
7	Profundización del mapa de la complejidad temporal. Algunas misceláneas de complejidad temporal. Introducción a la complejidad espacial. Jerarquía espacio-temporal. Misceláneas (problemas de búsqueda y de conteo, aproximaciones polinomiales, circuitos booleanos, algoritmos probabilísticos y cuánticos, etc).
8	**PARTE 2 (7 clases).** Introducción a la Lógica de Enunciados. Conocimiento: definición. Adquisición del conocimiento. Formas de razonamiento. Argumentaciones.
9	Sintaxis: el lenguaje simbólico de la lógica de enunciados. Formas enunciativas. Fórmulas bien formadas. Semántica: interpretación y satisfacción. Tablas y funciones de verdad. Tautologías y contradicciones.
10	Lógica de enunciados. El sistema formal L. Correctitud y completitud de L.
11	Lógica de predicados. Predicados y cuantificadores. Lenguajes de primer orden. Interpretaciones. Satisfacción y verdad.
12	Sistemas de primer orden. Teoría de los números.
13	Introducción a la verificación axiomática de programas.
14	Introducción a la Inteligencia Artificial (IA). Definición de IA. Pasado, presente y futuro de la IA. Usos de la IA. Agentes inteligentes y su ambiente.

Evaluaciones previstas	Fecha
1- 1er parcial	15 de octubre de 2026
2- recuperatorio	29 de octubre de 2026
3- 2do parcial	24 de Noviembre de 2026
4- recuperatorio	09 de Diciembre de 2026

Contactos de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):
Prof. Claudia Pons claudia.pons.33@gmail.com, cpons@lifia.info.unlp.edu.ar
Plataforma Ideas: Fundamentos de Teoría de la Computación

Profesora Claudia Pons