



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/130-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA ASPECTOS DE TEORÍA DE LA COMPUTACIÓN, DE LA CARRERA INGENIERÍA EN INFORMÁTICA – PLAN 2026, SEDE SAN LORENZO”

VISTO: El Memorando DA/1707/2025 del Director Académico de la FP-UNA, Prof. MSc. Felipe Santiago Uzabal Escurra, con el cual remite el Memorando CCPTCC/030/2025 de la Comisión Coordinadora del Proyecto de Transformación Curricular de Carreras de Grado de la FP-UNA, en el que presenta la propuesta de Programas de Estudio de las Asignaturas de la Carrera Ingeniería en Informática.

CONSIDERANDO: La Ley 4995/2013 de Educación Superior, el Estatuto de la Universidad Nacional de Asunción y las deliberaciones sobre el tema.

Que la Comisión Coordinadora del Proyecto de Transformación Curricular de Carreras de Grado, solicita la aprobación del Programa de Estudio de la asignatura **“Aspectos de Teoría de la Computación”**, de la carrera Ingeniería en Informática – Plan 2026.

Que los programas fueron elaborados conforme a las disposiciones establecidas por el Consejo Nacional de Educación Superior (CONES) en materia de **créditos académicos**, según lo dispuesto en la Resolución CONES N.º 221/2024, que regula el *Sistema de Créditos Académicos – Paraguay* y los criterios para su publicación en las carreras de grado.

**EL CONSEJO DIRECTIVO DE LA FACULTAD POLITÉCNICA
RESUELVE:**

25/18/130-01 APROBAR el Programa de Estudio de la Asignatura **“Aspectos de Teoría de la Computación”**, de la carrera Ingeniería en Informática – Plan 2026, Sede San Lorenzo, detallado en el ANEXO 92 de la presente Acta.

25/18/130-02 COMUNICAR, copiar y archivar

Prof. Abg. Joel Arsenio Benítez Santacruz
Secretario



Prof. Ing. Silvia Teresa Leiva León, MSc.
Presidenta

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Nivel				Grado									
Asignatura				Aspectos de Teoría de la Computación									
Carrera				Plan		Sede/Filial		Carácter		Semestre		Prerrequisitos	
Ingeniería en Informática				2026		Sede San Lorenzo		Electiva 1		Séptimo		Ingeniería de Software 1, Bases de Datos 2, Programación de Aplicaciones 2, Seguridad Informática.	
Semanal					Periodo								
HT		HP		HTD	HTI	HS	PA	THTD		THTI	THA	CA-PY	
2		2		4	4	8	18	72		72	144	5	

- *HT: Horas Teóricas semanales.
- *HP: Horas Prácticas semanales.
- *HTD: Horas semanales de Trabajo académico con acompañamiento Docente.
- *HTI: Horas semanales de Trabajo académico Independiente del estudiante.
- *HS: Horas Semanales (HTD+HTI).
- *PA: Periodo Académico en semanas.
- *THTD: Total de Horas de Trabajo académico con acompañamiento Docente (HTD*PA).
- *THTI: Total de Horas de Trabajo académico Independiente del estudiante (HTI*PA).
- *THA: Total de Horas de trabajo Académico (THTD+THTI).
- *CA-PY: Créditos académicos de la asignatura.

II. FUNDAMENTACIÓN

La asignatura Aspectos de Teoría de la Computación, ofrecida como electiva, se presenta como una ampliación de los conceptos fundamentales que ya han sido abordados en otras materias del plan de estudios para el desarrollo de una comprensión profunda de los fundamentos teóricos de la computación. Esta materia tiene un carácter teórico-práctico, ya que abarca tanto la conceptualización de modelos computacionales abstractos como la implementación de estos modelos en aplicaciones reales.

El curso proporciona a los estudiantes herramientas para analizar la complejidad computacional, la decidibilidad, y la relación entre diversos modelos de cómputo como los autómatas, las máquinas de Turing, y los lenguajes formales. Estos conocimientos son útiles para abordar problemas computacionales complejos y evaluar la viabilidad de algoritmos en diferentes contextos.

En relación a la naturaleza de la asignatura, se aborda de manera teórico-práctica, se combinarán conceptos teóricos con ejercicios prácticos. La organización de la asignatura se basa en los ejes temáticos, se incluyen conceptos fundamentales como: autómatas, lenguajes formales, máquinas de Turing, complejidad computacional.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

1. Actualizarse permanentemente mediante la obtención y gestión autónoma de información de calidad utilizando tecnología de la información y comunicación.



- 2. Evaluar el comportamiento de diversos fenómenos disciplinares e interdisciplinares con una visión de sistema mediante modelos teóricos validados y actualizados capaces de abarcarlos integralmente en un contexto de incertidumbre.
- 3. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de la ingeniería en informática.
- 4. Aplicar, producir y difundir conocimientos técnicos y científicos en el área de ingeniería en informática.
- 5. Planificar, proyectar, diseñar, ejecutar proyectos sostenibles e integrales para la resolución de problemas, la mejora y la innovación en el área de tecnología.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de aprendizaje
1. Introducción a la teoría de la computación.	1.1. Historia y evolución de los lenguajes formales y autómatas. 1.2. Jerarquía de Chomsky y clasificación de lenguajes formales.	1. Conoce y contextualiza la evolución histórica de los modelos formales y su impacto en la computación moderna.
2. Autómatas finitos y lenguajes regulares.	2.1. Autómatas finitos deterministas (AFD) y no deterministas (AFN). 2.2. Expresiones regulares y su relación con los autómatas. 2.3. Aplicaciones de lenguajes regulares.	1. Analiza las propiedades de los autómatas finitos y su aplicación en lenguajes regulares.
3. Gramáticas libres de contexto y autómatas de pila.	3.1. Autómatas de pila y su relación con lenguajes libres de contexto. 3.2. Aplicaciones de los autómatas de pila.	1. Aplica modelos de autómatas de pila para resolver problemas en lenguajes libres de contexto.
4. Máquinas de Turing y computabilidad.	4.1. Definición de máquinas de Turing y sus variantes (determinista, no determinista, con múltiples cintas). 4.2. Problemas decidibles y semidecidibles.	1. Evalúa problemas de decidibilidad e indecidibilidad mediante máquinas de Turing y técnicas de reducción.
5. Introducción a la complejidad computacional.	5.1. Clases de complejidad temporal: P, NP, y NP-completo. 5.2. Problemas NP-completos: Definición y ejemplos clásicos (SAT, 3-SAT, mochila). 5.3. Reducciones y teorema de Cook.	1. Evalúa la complejidad computacional de problemas mediante clases de complejidad temporal y espacial.
6. Complejidad espacial y	6.1. Complejidad espacial: PSPACE, L, NL.	1. Analiza problemas de complejidad espacial y su relación con las

Unidades	Contenidos	Resultados de aprendizaje
alternancia.	6.2. Relación entre clases de complejidad temporal y espacial (Teorema de Savitch).	clases de complejidad temporal.
7. Proyectos y casos de estudio.	7.1. Proyectos integradores sobre autómatas, máquinas de Turing o complejidad computacional. 7.2. Estudios de caso en computabilidad y complejidad.	1. Diseña soluciones a problemas computacionales mediante el uso de modelos formales y la evaluación de estudios de casos.

V. ESTRATEGIAS DIDÁCTICAS

En el desarrollo del programa se aplicarán estrategias didácticas conducentes a la apropiación teórica y la ejecución práctica de procesos y procedimientos, a saber:

- **Aula invertida:** Los estudiantes recibirán lecturas y materiales introductorios. Se espera que estudien estos materiales antes de la clase. Durante la clase, se dedicará tiempo a aclarar conceptos y a realizar ejercicios guiados, enfocándose en preguntas clave y en la discusión de ejemplos.
- **Lluvia de ideas:** Al inicio de la clase, los estudiantes participarán en una lluvia de ideas para expresar sus conocimientos previos y percepciones. Esto ayudará a activar conocimientos previos y a introducir los nuevos conceptos.
- **Aprendizaje basado en problemas:** Se presentarán a los estudiantes problemas prácticos. Los estudiantes, en grupos o de manera individual, deberán analizar el problema y diseñar la solución.

La elección particular de la estrategia didáctica aplicada será explícita en el Planeamiento de la Asignatura, de acuerdo con el perfil de los estudiantes, los recursos disponibles y el contexto educativo.

VI. ESTRATEGIAS EVALUATIVAS

Procesos de producción grupales e individuales, pruebas individuales escritas o de forma de cuestionario durante el desarrollo de las unidades, implementación de soluciones seguras a problemas específicos de desarrollo de software.

Con fines de calificación y promoción se aplicará el Reglamento Académico vigente en la institución que prevé valoraciones de proceso y final.

VII. MEDIOS AUXILIARES

Pizarras, marcadores, computadoras, proyector, parlantes para multimedia, plataforma virtual, sala de laboratorio equipada para prácticas con sistemas operativos Linux o Windows y acceso a Internet.

VIII. BIBLIOGRAFÍA

- Rich, E. (2019). Automata, Computability and Complexity: Theory and Applications (Rev. ed.). Pearson Education.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms (4th ed.). MIT Press.
- Sipser, M. (2013). Introduction to the Theory of Computation (3rd ed.). Cengage Learning.
- Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2007). Introduction to Automata Theory, Languages, and Computation (3rd ed.). Pearson.
- Kozen, D. C. (1997). Automata and Computability. Springer.

- Arora, S., & Barak, B. (2009). Computational Complexity: A Modern Approach. Cambridge University Press

