



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/47-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA ALGORITMOS AVANZADOS, DE LA CARRERA LICENCIATURA EN CIENCIAS INFORMÁTICAS – PLAN 2026, SEDE SAN LORENZO, FILIAL VILLARRICA Y FILIAL CORONEL OVIEDO”

VISTO: El Memorando DA/1707/2025 del Director Académico de la FP-UNA, Prof. MSc. Felipe Santiago Uzabal Ecurra, 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 Licenciatura en Ciencias Informáticas.

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 **“Algoritmos Avanzados”**, de la carrera Licenciatura en Ciencias Informáticas – 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:**

25/18/47-01 APROBAR el Programa de Estudio de la Asignatura **“Algoritmos Avanzados”**, de la carrera Licenciatura en Ciencias Informáticas – Plan 2026, Sede San Lorenzo, Filial Villarrica y Filial Coronel Oviedo, detallado en el ANEXO 09 de la presente Acta.

25/18/47-02 COMUNICAR, copiar y archivar.

Prof. Abg. Joel Arsenio Benítez Santacruz
Secretario

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





Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

Resolución 25/18/47-00 Acta 1226/25/08/2025
ANEXO 09

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Nivel				Grado						
Asignatura				Algoritmos Avanzados						
Carrera				Plan		Sede/Filial		Carácter	Semestre	Prerrequisitos
Licenciatura en Ciencias Informáticas				2026		Sede San Lorenzo/Filial Villarrica/Filial Coronel Oviedo		Obligatoria	Tercero	Estructura de Datos, Lógica y Matemática Discreta.
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 Algoritmos Avanzados provee a los estudiantes de conocimientos avanzados en el diseño, análisis y aplicación de algoritmos. Dado que los avances en hardware han alcanzado límites físicos significativos, la optimización y la eficiencia computacional ahora dependen principalmente de técnicas algorítmicas novedosas. Por ello, esta asignatura resulta relevante para formar profesionales capaces de abordar problemas computacionales complejos mediante soluciones eficientes e innovadoras.

El curso se enfoca en:

- Análisis asintótico: Introduce a los estudiantes en la evaluación de la eficiencia algorítmica, ayudándolos a diferenciar buenos y malos diseños de algoritmos mediante herramientas de análisis de complejidad.
- Teoría de grafos: Permite modelar y resolver problemas computacionales mediante estructuras de datos y algoritmos para grafos, con aplicaciones en áreas como redes, análisis de datos y modelado de problemas.
- Optimización computacional: Aborda problemas de maximización y minimización de recursos, esenciales en tecnologías de la información y en la toma de decisiones.
- Teoría de colas: Proporciona modelos y soluciones para sistemas que involucran líneas de espera, tiempos de procesamiento y manejo de recursos en escenarios como redes de computadoras y procesos CPU.

Esta asignatura es de naturaleza teórico-práctica. La organización de la misma en función de los ejes temáticos es la siguiente:

- Complejidad computacional: Se desarrolla en la Unidad 1, enfocándose en el análisis asintótico y la eficiencia de algoritmos.
- Teoría de grafos: Se trabaja en la Unidad 2, abordando algoritmos y estructuras de datos para modelar y resolver problemas mediante grafos.
- Optimización: Se desarrolla en la Unidad 3, con énfasis en programación lineal y técnicas para resolver problemas de optimización computacional.
- Teoría de colas: Se aborda en la Unidad 4, explorando modelos de sistemas basados en colas y tiempos de espera.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

1. Comunicarse en lenguas oficiales del país y en una lengua extranjera.
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 en el ámbito de las ciencias informáticas.
3. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de las ciencias informáticas.
4. Planificar, proyectar, diseñar y ejecutar proyectos sostenibles e integrales para la resolución de problemas, la mejora y la innovación en el ámbito de las ciencias informáticas.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de aprendizaje
1. Complejidad de los algoritmos.	1.1. Representación y costo de operaciones en los algoritmos. 1.2. Casos de análisis: peor, mejor y promedio. 1.3. Notaciones asintóticas. 1.4. Análisis asintótico de algoritmos. 1.5. Análisis de ordenamiento por inserción. 1.6. Análisis de ordenamiento de la burbuja. 1.7. Análisis de algoritmos recursivos y ordenamiento por mezcla. 1.8. Resolución de ecuaciones de recurrencia: Árbol de Recurrencia y Teorema Maestro.	1. Realiza análisis de complejidad asintótica de algoritmos utilizando herramientas matemáticas. 2. Reconoce la importancia de diseñar algoritmos eficientes en términos de tiempo y recursos. 3. Construye benchmarks para medir y estimar tiempos de ejecución de algoritmos. 4. Aplica el teorema maestro para analizar algoritmos recursivos en problemas computacionales.
2. Algoritmos de grafos.	2.1. Teoría de grafos. 2.1.1. Definición formal de grafo. 2.1.2. Caminos y ciclos. 2.1.3. Circuitos Hamiltonianos y Eulerianos. 2.1.4. Modelado de problemas como grafos. 2.2. Representaciones de grafos. 2.3. Búsqueda en profundidad.	1. Define formalmente un grafo y sus componentes fundamentales. 2. Identifica conceptos clave en grafos como vértices, aristas, grados, ciclos, caminos, circuitos hamiltonianos y eulerianos. 3. Modela problemas computacionales utilizando representaciones de grafos.

Unidades	Contenidos	Resultados de aprendizaje
	2.4. Búsqueda en anchura. 2.5. Aplicaciones de la búsqueda en profundidad. Componentes conexos, grafos bipartitos. 2.6. Caminos cortos. Colas de prioridad y algoritmo de Dijkstra. 2.7. Árboles de recubrimiento. Algoritmo de Kruskal y estructura Union-Find.	4. Representa grafos con estructuras de datos adecuadas como listas de adyacencia o matrices de adyacencia. 5. Implementa los algoritmos de búsqueda en profundidad (DFS) y búsqueda en anchura (BFS). 6. Aplica la búsqueda en profundidad para identificar componentes conexos y verificar si un grafo es bipartito. 7. Implementa colas de prioridad para resolver problemas con el algoritmo de Dijkstra. 8. Utiliza la estructura Union-Find en la implementación del algoritmo de Kruskal. 9. Resuelve problemas computacionales aplicando algoritmos de grafos.
3. Optimización computacional.	3.1. Optimización matemática. 3.1.1. Función objetivo. 3.1.2. Restricciones. 3.1.3. Modelado de problemas. 3.2. Programación lineal. 3.3. Formulación de problemas como programas lineales. 3.4. Problemas de maximización y minimización. 3.5. Forma estándar de un programa lineal.	1. Define un problema de optimización a través de su función objetivo y restricciones utilizando matemáticas. 2. Modela problemas computacionales como problemas de optimización. 3. Modela problemas de programación lineal. 4. Representa un problema de programación lineal en su forma estándar. 5. Implementa problemas de optimización utilizando librerías de código abierto o propietario.
4. Teoría de colas.	4.1. Modelo de colas. Población de entrada y tiempo entre llegadas. Mecanismo de servicio. 4.2. Terminología y notación. Notación de Kendall. 4.3. La distribución exponencial. 4.4. Modelos de colas basados en el proceso de nacimiento y muerte. Modelo M/M/S.	1. Define un problema de colas utilizando la terminología de la teoría de colas. 2. Modela problemas computacionales como problemas de teoría de colas. 3. Implementa problemas de colas utilizando librerías de código abierto o propietario.

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:

- **Aprendizaje basado en problemas:** exposición por parte del docente de los conceptos básicos por unidad, con materiales de lectura y ejemplos orientados a la enseñanza de las

competencias específicas de la asignatura. El estudiante buscará resolver un problema a través del conocimiento que adquirió en el aula.

- **Aprendizaje basado en proyectos:** el docente propondrá la realización de un proyecto que involucre todos los resultados de aprendizaje de asignatura. De esta forma el estudiante participa activamente en su aprendizaje, desarrollando diferentes habilidades para solucionar un problema a través de este proyecto.

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 orales y/o escritas durante el desarrollo de las unidades, implementación de algoritmos utilizando sistemas de control de versiones, redacción de informes de resultados experimentales de algoritmos. Todos estos serán valorados y que en su conjunto aportarán para la calificación y promoción, las que serán aplicadas según normativas institucionales.

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

Aula virtual, pizarrón, proyector, marcadores, ordenadores, conexión a internet.

VIII. BIBLIOGRAFÍA

- Hillier F.S., Lieberman, G.J. (2010) Introducción a la investigación de operaciones, 9na. Edición. McGrawHill.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms. MIT press.
- Dasgupta, S., Papadimitriou, C. H., & Algorithms, U. V. (2006). McGraw-Hill Science. Engineering/Math.
- Brassard, G. y Bratley, P. (2006). Fundamentos de Algoritmia. Prentice Hall.
- Erickson, J. (2019). Algorithms <https://jeffe.cs.illinois.edu/teaching/algorithms/>
- Aho, A. V. (2001). Data Structures and Algorithms. Stillwaters.
- Shaffer, C. A. (2010). A Practical Introduction to Data Structures and Algorithm Analysis Third Edition.
- Weiss, M. A. (2012). Data structures and algorithm analysis in Java. Pearson Education, Inc.
- Weiss, M. A. (2010). Data Structures and Problem Solving Using Java, 4th edition Pearson Education, Inc.



[Handwritten signature in blue ink]