



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/87-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA ALGORITMOS Y ESTRUCTURAS DE DATOS 2, 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 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 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 **“Algoritmos y Estructuras de Datos 2”**, 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:**

25/18/87-01 APROBAR el Programa de Estudio de la Asignatura **“Algoritmos y Estructuras de Datos 2”**, de la carrera Ingeniería en Informática – Plan 2026, Sede San Lorenzo, detallado en el ANEXO 49 de la presente Acta.

25/18/87-02 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			Algoritmos y Estructuras de Datos 2						
Carrera			Plan	Sede/Filial		Carácter	Semestre	Prerrequisitos	
Ingeniería en Informática			2026	Sede San Lorenzo		Obligatoria	Segundo	Algoritmos y Estructuras de Datos 1, Lenguajes de Programación 1.	
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

Esta asignatura es una continuación natural de la asignatura Algoritmos y Estructura de Datos 1 y presenta las bases teóricas y prácticas para el análisis de algoritmos, tema central para la evaluación analítica del rendimiento espacial y temporal de los algoritmos. En el curso se estudiarán las estructuras de datos más complejas y avanzadas, basadas en el uso de memoria dinámica, y que son usuales en todos los componentes de software presentes en las soluciones computacionales actuales.

Se abordarán dos problemas usuales en computación, el de búsqueda y el de ordenación, usualmente presentes en cualquier problema algorítmico complejo. La comprensión de las diferentes estrategias de solución de los algoritmos clásicos para estos problemas brindará al estudiante suficientes elementos para la selección de los métodos más adecuados de acuerdo a la naturaleza de los datos y al tipo de problema enfrentado.

Considerando que el estudiante ya cuenta con capacidades de pensamiento computacional, este curso presenta los rudimentos técnicos necesarios para abordar problemas más complejos, utilizando estrategias de diseño de algoritmos como *dividey vencerás* y *vuelta atrás* o *backtracking*.

La asignatura es de naturaleza teórico-práctica, enfocada en la comprensión y aplicación de algoritmos y estructuras de datos. La relación entre los ejes temáticos y las unidades es la siguiente:

- Fundamentos del análisis de algoritmos y el cálculo asintótico: relacionado con la Unidad 1. Resolución de problemas computacionales, que introduce los conceptos básicos de algoritmos y su estructura, sentando las bases para el análisis del rendimiento.

- Determinación del rendimiento temporal y espacial de algoritmos no recursivos y recursivos: vinculado a la Unidad 4. Algoritmos basados en recursión, que aborda el diseño y la implementación de algoritmos recursivos, esenciales para comprender la eficiencia temporal y espacial.
- Tipo Abstracto de Datos (TAD). Concepto y diseño de TAD sencillos: relacionado con la Unidad 3. Estructuras de datos fundamentales lineales estáticas basadas en arreglos, que introduce pilas, colas y montículos, ejemplos de TAD fundamentales.
- Estructuras de datos no lineales (árboles generales, árboles binarios y grafos): se refleja en la Unidad 2. Introducción a las estructuras de datos, ordenación y búsqueda, que establece los conceptos fundamentales de las estructuras de datos, preparando las bases para el estudio de estructuras no lineales.
- Algoritmos de búsqueda en estructuras no lineales: relacionado con las técnicas estudiadas en la Unidad 2, especialmente en algoritmos de búsqueda, aplicados luego a estructuras más complejas.
- Algoritmos de ordenamiento basados en memoria interna, basados y no basados en comparación: abordado en la Unidad 2, que cubre algoritmos de ordenación como inserción, selección y burbuja, introduciendo conceptos que se expanden a técnicas más avanzadas.
- Técnicas de diseño de algoritmos *Divide y Vencerás* y de *Vuelta Atrás (Backtracking)*: vinculado con la Unidad 5. Algoritmos combinatorios, que introduce la resolución de problemas complejos mediante técnicas específicas como la generación combinatoria y aplicaciones en optimización.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

1. Evaluar el comportamiento de diversos fenómenos disciplinares e interdisciplinares relacionados con la ingeniería en informática con una visión de sistema, mediante modelos teóricos validados y actualizados, capaces de abarcarlos integralmente en un contexto de incertidumbre.
2. Planificar, proyectar, diseñar y ejecutar proyectos sostenibles e integrales para la resolución de problemas, la mejora y la innovación en el ámbito de la ingeniería informática.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de aprendizaje
1. Análisis de algoritmos.	1.1. Motivación. 1.2. Representación general de algoritmos. 1.3. Descripción del modelo de computación RAM. Costo de operaciones en los algoritmos. 1.4. Casos de análisis: peor, mejor y promedio. 1.5. Análisis de tiempo y espacio de algoritmos secuenciales.	1. Describe el modelo de computación RAM aplicado para el análisis de algoritmos. 2. Describe las características de los casos (peor, mejor y promedio) de análisis de algoritmos. 3. Utiliza la técnica de análisis de algoritmo para la caracterización de algoritmos secuenciales y recursivos. 4. Calcula correctamente el costo asintótico temporal y espacial de

Unidades	Contenidos		Resultados de aprendizaje
	1.6.	Análisis de tiempo y espacio de algoritmos recursivos.	algoritmos secuenciales y recursivos.
	1.6.1.	Ecuación de recurrencia.	5. Resuelve correctamente las ecuaciones de recurrencia utilizando el árbol de recurrencia o el teorema maestro.
	1.6.2.	Resolución de ecuaciones de recurrencia: Árbol de Recurrencia y Teorema Maestro.	6. Aplica las notaciones asintóticas superior, inferior y exacta de acuerdo a cada algoritmo o situación.
	1.7.	Notaciones asintóticas.	
	1.7.1.	Definiciones y uso.	
	1.7.2.	Determinación de cotas superior (O), inferior (Omega) y exacta (Zeta).	
2. Tipo abstracto de datos.	2.1	Definición de Tipo Abstracto de Datos (TAD).	1. Define las características de un TAD.
	2.2	Semántica para definición de TAD.	2. Describe la importancia de la definición de TAD en el contexto del diseño de las estructuras de datos.
	2.3	Diseño de TAD sencillos: Pilas, Listas, Colas, Arreglos entre otros.	3. Utiliza la estrategia de diseño y representación semántica para la definición de TAD.
			4. Aplica la técnica de representación de TAD a las estructuras de datos sencillas como pilas, listas, colas y arreglos.
			5. Desarrolla TAD basados en los TAD conocidos y de acuerdo al problema planteado.
3. Estructuras de datos no lineales.	3.1	Estructura de datos no lineales y consideraciones de implementación.	1. Explica las características, estructura y operaciones fundamentales de las estructuras de datos no lineales, incluyendo listas, pilas, colas, árboles y grafos.
	3.2	Listas.	2. Describe las características y diferencias de diseño entre árboles binarios, árboles binarios de búsqueda balanceados y no balanceados, montículos y colas de prioridad.
	3.2.1	Representación y operaciones con memoria dinámica.	3. Describe los algoritmos asociados a las estructuras de datos no lineales, incluyendo barridos de grafos y operaciones en árboles y montículos.
	3.2.2	Pila y cola basada en lista.	4. Aplica estructuras de datos no lineales, como árboles, montículos y grafos, para resolver problemas algorítmicos específicos.
	3.3.	Árboles generales.	
	3.3.1	Representaciones con vectores y con memoria dinámica.	
	3.3.2.	Algoritmos de recorridos y mantenimiento.	
	3.3.3.	Aplicaciones con árboles.	
	3.4	Árboles binarios	
	3.4.1	Árboles binarios generales.	
	3.4.2	Diccionario. Concepto.	
	3.4.3	Problema de búsqueda.	

Unidades	Contenidos	Resultados de aprendizaje
	3.4.4 Árboles binarios de búsqueda. 3.4.5 Árboles binarios de búsqueda balanceados. AVL y rojinegros. 3.5 Colas de prioridad. Definición y operaciones. 3.6 Grafos. 3.6.1. Definición. Terminología y representaciones: matriz de adyacencia y listas de adyacencia. 3.6.2. Algoritmos de barrido de grafos: en amplitud, en profundidad y orden topológico. 3.6.3. Problemas aplicando recorrido de grafos.	5. Utiliza el concepto de diccionario y su implementación en problemas computacionales. 6. Analiza el rendimiento espacial y temporal de las operaciones realizadas con estructuras de datos no lineales, como listas, árboles y grafos.
4. Ordenación.	4.1. Definición formal del problema de ordenación. Inversiones y análisis de casos. 4.2. Algoritmos de ordenación basados en comparación 4.2.1. Revisión: Inserción y montículo. 4.2.2. Mezcla. 4.2.3. Ordenación rápida 4.3. Cota mínima de los algoritmos de ordenación basados en comparación. 4.4. Algoritmos de ordenación no basados en comparación. 4.4.1. Por conteo (<i>CountingSort</i>). 4.4.2. Por cubetas (<i>BinSort</i>) 4.4.3. Por radix (<i>RadixSort</i>).	1. Explica las características de los algoritmos de ordenación basados y no basados en comparación, describiendo sus aplicaciones y relación con problemas como el de las inversiones. 2. Describe y compara los principales algoritmos de ordenación, incluyendo algoritmos basados en comparación (mezcla, inserción, <i>QuickSort</i>, montículos) y no basados en comparación (<i>CountingSort</i>, <i>RadixSort</i>, <i>BinSort</i>). 3. Analiza el rendimiento espacial y temporal de algoritmos de ordenación utilizando técnicas de análisis de algoritmos. 4. Selecciona el algoritmo de ordenación más adecuado de acuerdo con las características del problema y la naturaleza de los datos manipulados. 5. Desarrolla soluciones algorítmicas utilizando diferentes métodos de ordenación en problemas computacionales específicos.
5. Técnica de diseño de algoritmos.	5.1. Introducción a las técnicas de diseño de algoritmos. 5.2. <i>Divide y Vencerás</i> 5.3. <i>Vuelta Atrás</i>	1. Describe la importancia de las técnicas de diseño de algoritmos. 2. Describe las características y uso de las técnicas de diseño <i>Divide y Vencerás</i> y de <i>Vuelta Atrás</i> o

Unidades	Contenidos	Resultados de aprendizaje
	(Backtracking). 5.4. Problemas clásicos de las técnicas.	Backtracking. 3. Desarrolla soluciones algorítmicas aplicando la técnica de diseño <i>Divide y Vencerás</i> en problemas computacionales. 4. Desarrolla soluciones algorítmicas aplicando la técnica de diseño <i>Vuelta Atrás</i> en problemas computacionales. 5. Utiliza el análisis de algoritmos para evaluar el rendimiento espacial y temporal de los algoritmos desarrollados con las técnicas de <i>Divide y Vencerás</i> o de <i>Vuelta Atrás</i> .

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 la materia. 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 escritas o de forma de cuestionario durante el desarrollo de las unidades, implementación de las ideas de los algoritmos en un lenguaje de programación orientado a objetos como tareas entregables. 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, equipo de audio.

VIII. BIBLIOGRAFÍA

- Weiss, M. A. (2013). Estructuras de datos en Java. 4ta. Edición. Pearson.
- 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.
- Clifford, Shaffer (2013). Data Structures and Algorithm Analysis. Java version. Accesible en <http://people.cs.vt.edu/~shaffer/Book/JAVA3elatest.pdf>.
- Guerequeta, R.; Vallecillo, A. (1998). Técnicas de Diseño de Algoritmos. Servicio de Publicaciones de la Universidad de Málaga. Accesible en: <http://www.lcc.uma.es/~av/Libro/indice.html>
- Joyanes A., Luis; Zahonero Martínez, Ignacio. (2011). Programación en Java 6. Algoritmos y Programación Orientada a Objetos.
- Brassard, G.; Bratley, P. (1996) Fundamentos de Algoritmia. Prentice Hall.
- Erickson, J. (2019) Algorithms. Accesible en: <https://jeffe.cs.illinois.edu/teaching/algorithms>
- Steven S. Skiena (2020). The Algorithm Design Manual. 3ra Edición. Springer.

