



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/44-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA ESTRUCTURA DE DATOS, 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 **“Estructura de Datos”**, 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:**

25/18/44-01 APROBAR el Programa de Estudio de la Asignatura **“Estructura de Datos”**, de la carrera Licenciatura en Ciencias Informáticas – Plan 2026, Sede San Lorenzo, Filial Villarrica y Filial Coronel Oviedo, detallado en el ANEXO 06 de la presente Acta.

25/18/44-02 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 25/18/44-00 Acta 1226/25/08/2025
ANEXO 06

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

II. Nivel				Grado						
Asignatura				Estructura de Datos						
Carrera				Plan		Sede/Filial		Carácter	Semestre	Prerrequisitos
Licenciatura en Ciencias Informáticas				2026		Sede San Lorenzo/Filial Villarrica/Filial Coronel Oviedo		Obligatoria	Segundo	Introducción a la Programación.
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

III. FUNDAMENTACIÓN

La asignatura de Estructuras de Datos es un componente obligatorio dentro de la malla curricular de la carrera debido a su impacto directo en la formación de profesionales capaces de diseñar y desarrollar soluciones informáticas eficientes y escalables. Estudiar y aprender las estructuras de datos es de suma importancia en la informática y la programación debido a su impacto en la eficiencia, la resolución de problemas, la reutilización de código, el mantenimiento de software y la preparación para desafíos técnicos.

Las estructuras de datos adecuadas permiten gestionar y acceder a los datos de manera eficiente, la comprensión de cómo funcionan y cómo se relacionan con los algoritmos permite optimizar la eficiencia de un programa y reducir el tiempo de procesamiento. Al mismo tiempo que una comprensión sólida permite reutilizar código de manera eficiente. Es posible tomar soluciones previas y aplicarlas a nuevos problemas al adaptar las estructuras de datos y algoritmos ya existentes. Esto acelera el desarrollo de software y reduce la duplicación de esfuerzo.

A medida que los proyectos de software crecen y evolucionan, el mantenimiento y la escalabilidad se vuelven críticos. Las estructuras de datos bien diseñadas facilitan la adaptación de un sistema a nuevas necesidades y cambios en los requisitos sin una reescritura completa del código. De igual forma el comprender cómo funcionan las estructuras de datos implica razonamiento lógico y pensamiento algorítmico, habilidades que son esenciales en la informática y la programación.

Esta asignatura pretende dar al estudiante los conocimientos de Estructuras de Datos, basados en un enfoque de Orientación a Objetos, para su utilización en el diseño y desarrollo de aplicaciones.

La naturaleza de esta asignatura es teórico-práctica, permitiendo a los estudiantes adquirir conocimientos sólidos sobre la representación y el manejo de datos, así como desarrollar las competencias necesarias para implementar estructuras y algoritmos en entornos reales.

La relación entre ejes temáticos y unidades es la siguiente:

- Estructuras de datos fundamentales y avanzadas: Se abordan en las Unidades 1, 3 y 4, desarrollando conceptos clave de estructuras lineales (listas, pilas, colas) y no lineales (árboles, grafos), así como su implementación en un enfoque orientado a objetos.
- Algoritmos de ordenación y búsqueda: Integrados en las Unidades 2 y 3, donde se analiza su diseño, implementación y eficiencia, mostrando su relevancia dentro de las estructuras de datos.
- Análisis de algoritmos: Trabajado en la Unidad 2, proporcionando a los estudiantes herramientas para evaluar la complejidad y optimización de las soluciones algorítmicas.
- Recursividad: Desarrollada específicamente en la Unidad 2, con énfasis en su aplicación en problemas algorítmicos y en la implementación de estructuras de datos.

IV. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

1. 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.
2. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de las ciencias informáticas.
3. 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. Introducción a las Estructuras de Datos y Programación Orientada a Objetos	1. Datos e Información. 2. Bit, byte, campos, registros y archivos. 3. Datos simples (entero, real, carácter, lógico). 4. Datos estáticos (arreglos, registros, archivos, cadenas). 5. Datos dinámicos (listas, pilas, colas, árboles, grafos). 6. Tipos de datos abstractos (TDA) y Estructuras de Datos. 6.1 Definición de Tipo, Tipo de Dato, TDA, Estructura de dato. 6.2 Semántica para definición de TDA. 6.3 Implementación 7. Introducción a la Programación Orientada a Objetos (POO) 7.1. Conceptos Básicos. Características y Beneficios. Clases y Objetos. Métodos y Propiedades. Herencia, Polimorfismo y Enlace Dinámico. 7.2. Diseño de estructura y comportamiento de objetos.	1. Reconoce el papel central de las estructuras de datos. 2. Identifica la taxonomía de las estructuras de datos y su relación con los algoritmos. 3. Crea y manipula tipos de datos abstractos desde el punto de vista de la orientación a objetos. 4. Identifica las características esenciales de la programación orientada a objetos.



Unidades	Contenidos	Resultados de aprendizaje
	Definición de clases, constructores, propiedades y métodos. Instanciación de objetos, selección de métodos y acceso a propiedades.	
2. Análisis de Algoritmos y Recursividad.	2.1 Introducción, conceptos. Notación asintótica. 2.2 Algoritmos de ordenamiento. 2.2.1. Definición del problema, tipos de operaciones de ordenamiento y estabilidad. 2.2.2. Ordenación por inserción, selección, mezcla, montículo (heap) y ordenación rápida (quicksort) Ventajas, desventajas y situaciones de uso de cada algoritmo 2.2.3. Comparación de algoritmos de ordenamiento. 2.2.4. Tiempo de ejecución de los algoritmos de ordenamiento 2.3 Algoritmos de búsqueda. 2.3.1. Búsqueda secuencial y Búsqueda binaria 2.3.2. Tiempo de ejecución de los algoritmos de búsqueda. 2.4. Recursividad 2.4.1. Conceptos básicos de recursión. 2.4.2. Diseño de algoritmos recursivos. 2.4.3. Algoritmos recursivos en ordenación y búsqueda. 2.4.4. Procedimientos recursivos. 2.4.5. Complejidad de algoritmos recursivos. 2.4.6. Ejemplos de aplicación.	1. Realiza análisis de complejidad asintótica de algoritmos. 2. Aprecia la importancia de la construcción de algoritmos eficientes. 3. Entiende los conceptos fundamentales de recursión. 4. Resuelve problemas mediante el diseño de algoritmos recursivos. 5. Comprende métodos de ordenamiento y búsqueda que emplean recursión. 6. Realiza algoritmos básicos de recursión, métodos de búsqueda y técnicas de organización de datos. 7. Implementa y analiza los algoritmos de ordenación básicos en un lenguaje orientado a objetos.
3. Estructuras de Datos Lineales.	3.1. Introducción a las estructuras de datos lineales. 3.2. Listas. 3.2.1. Tipos de listas enlazadas: simples, doblemente enlazadas, circulares. 3.2.2. Implementación y tiempo de ejecución de operaciones. 3.2.3. Aplicaciones 3.3. Pilas. 3.3.1. Definición y operaciones sobre pilas.	1 Implementa algoritmos y estructuras de datos lineales utilizando lenguajes de programación orientados a objetos. 2 Identifica, entiende y documenta los requerimientos de sistemas de información. 3 Diseña e implementa unidades estructurales simples que utilizan algoritmos y estructuras de datos, incluyendo las interfaces para su comunicación.



Unidades	Contenidos	Resultados de aprendizaje
	3.3.2. Representación. 3.3.3. Implementación y tiempo de ejecución de operaciones. 3.3.4. Aplicaciones. 3.4. Colas. 3.4.1. Definición y operaciones sobre colas. 3.4.2. Tipos de Colas: simple, bicolos o doble cola, de prioridad. 3.4.3. Representación. 3.4.4. Implementación y tiempo de ejecución de operaciones. 3.4.5. Aplicaciones.	4 Desarrolla, mantiene y evalúa sistemas y servicios basados en software cumpliendo estándares de calidad y aplicando teorías, principios, métodos y mejores prácticas de ingeniería de software. 5 Identifica, diseña e implementa los tipos abstractos de datos adecuados a una aplicación concreta.
4. Estructuras de datos no lineales.	4.1. Árboles. 4.1.1. Terminología. Aplicaciones. 4.1.2. Árboles etiquetados y de expresión. 4.1.3. Operaciones sobre árboles. 4.1.4. Recorrido de un árbol: en amplitud y en profundidad y por niveles (anchura o amplitud) 4.1.5. Implementación con arreglos. 4.1.6. Implementación mediante lista enlazadas. 4.2. Árboles binarios. 4.2.1. Árboles binarios de búsqueda. 4.2.2. Operaciones insertar y suprimir. 4.2.3. Búsqueda en el árbol. 4.2.4. Recorridos del árbol. 4.2.5. Árboles binarios balanceados AVL: definición, operaciones, aplicaciones e implementación 4.2.6. Árboles B: definición, operaciones y aplicaciones 4.3. Grafos. 4.3.1. Terminología. 4.3.2. Tipos de grafos. 4.3.3. Representación de grafos mediante listas y matrices. 4.3.4. Recorrido. 4.3.5. Aplicaciones: Algoritmos sobre grafos: Ordenación topológica, Búsqueda de camino mínimo, Árbol de expansión mínima.	1. Diseña e implementa unidades estructurales complejas que utilizan algoritmos y estructuras de datos, incluyendo las interfaces para su comunicación. 2. Diseña, implementa, integra, administra y evalúa soluciones informáticas de arquitectura empresarial, de datos y de aplicaciones. 3. Aplica los principios de interacción hombre-máquina para la evaluación y construcción de una amplia gama de componentes, incluyendo interfaces de usuario, considerando las diferencias culturales. 4. Demuestra la reutilización de software y adaptación, realiza mantenimiento, integración, migración de productos software y componentes, prepara elementos de software para su reutilización potencial y crea interfaces técnicas para componentes y servicios. 5. Identifica las oportunidades creadas por las innovaciones tecnológicas. 6. Integra eficazmente soluciones basadas en Tecnologías de Información, considerando el entorno del usuario, estrategias, estándares y tecnologías disponibles. 7. Implementa algoritmos y estructuras de datos no lineales utilizando lenguajes de programación orientados a objetos.



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 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 con diálogos e interpretaciones que los estudiantes realicen sobre los contenidos, debates, retroalimentación en casos necesarios y actividades que amplíen el conocimiento, que 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, dispositivos de cómputo, Internet, celulares, salas de chats, correo electrónico.

VIII. BIBLIOGRAFÍA

- Sznajdleder, P. (2017). Programación orientada a objetos y estructura de datos a fondo. Alpha Editorial. ISBN: 9789587783384
- López, L. (2013). Metodología de la programación orientada a objetos. Alpha Editorial. ISBN: 9786077077961
- Weiss, M. A. (2013). Estructuras de Datos en Java 4ta Ed. México: Pearson. ISBN: 9788415552222
- Shaffer, C. A. (2010). A Practical Introduction to Data Structures and Algorithm Analysis Third Edition.
- Sedgewick, R., Wayne, K. (2011). Algorithms. Reino Unido: Addison-Wesley. ISBN: 9780321573513
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms. MIT press. ISBN: 9780262367509
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1999). Estructuras de datos y algoritmos. México: Pearson Educación. ISBN: 9789684443457.
- Goytia, L. L., & González, Á. G. (2014). Programación orientada a objetos C++ y Java. Grupo Editorial Patria. ISBN: 9786074389333
- Weiss, M. A. (2012). Data structures and algorithm analysis in Java. Pearson Education, Inc. ISBN: 9780132576277



[Handwritten signature]

[Handwritten signature]

- Goodrich, M. T., Tamassia, R., Goldwasser, M. H. (2013). Data Structures and Algorithms in Python. Reino Unido: Wiley. ISBN: 9781118290279
- Wirth, N. (2012). Algoritmos + estructuras de datos. España: Ediciones del Castillo. Ed. original 1980. ISBN: 9788421901724.



A blue ink signature, consisting of a stylized, cursive letter 'A' followed by a horizontal stroke.