



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/19/22-00
ACTA 1227/08/09/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA ESTRUCTURA DE DATOS Y ENTORNOS DE PROGRAMACIÓN, DE LA CARRERA INGENIERÍA EN ELECTRÓNICA – PLAN 2026, SEDE SAN LORENZO”

VISTO: El Memorando DA/1796/2025 del Director Académico de la FP-UNA, Prof. MSc. Felipe Santiago Uzabai Ecurra, con el cual remite el Memorando CCPTCC/032/2025 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 Electrónica.

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 y Entornos de Programación”**, de la carrera Ingeniería en Electrónica – 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:**

25/19/22-01 APROBAR el Programa de Estudio de la Asignatura **“Estructura de Datos y Entornos de Programación”**, de la carrera Ingeniería en Electrónica – Plan 2026, Sede San Lorenzo, detallado en el ANEXO 10 de la presente Acta.

25/19/22-02 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 25/19/22-00 Acta 1227/08/09/2025
ANEXO 10

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN:

Nivel			Grado							
Asignatura			Estructura de Datos y Entornos de Programación							
Carrera			Plan		Sede/Filial		Carácter		Semestre	Prerrequisitos
Ingeniería en Electrónica			2026		Sede San Lorenzo		Obligatoria		Cuarto	Fundamentos de 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.

II. FUNDAMENTACIÓN

Esta asignatura se convierte en una herramienta fundamental para el desarrollo de sistemas embebidos y aplicaciones de control. Comprender las estructuras de datos como pilas, colas y listas es esencial para la gestión eficiente de la información en dispositivos electrónicos, donde la velocidad y la memoria son recursos críticos. Además, el dominio del uso de librerías permite a los estudiantes integrar módulos preexistentes para manejar sensores, actuadores y comunicaciones sin tener que reinventar el código, acelerando el desarrollo y mejorando la fiabilidad del sistema.

La programación orientada a objetos proporciona un enfoque modular que facilita el diseño y la implementación de software para sistemas electrónicos, permitiendo la creación de clases y objetos que representan componentes del sistema y sus interacciones. Los entornos de desarrollo integrados (IDE) optimizan el proceso de codificación, depuración y prueba, cruciales en el diseño de firmware y software para dispositivos electrónicos. Finalmente, la programación de interfaces gráficas de usuario (GUI) permite la creación de interfaces intuitivas para el control y monitoreo de sistemas electrónicos, ofreciendo una experiencia de usuario clara y efectiva. Este enfoque integral no solo mejora la eficiencia en el desarrollo, sino que también facilita la integración de hardware y software, clave para la creación de soluciones innovadoras.

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: Estructura de datos (estructuras y definición de tipo de datos, pilas, colas, listas). Uso de librerías. Introducción a la programación orientada a objetos (clase, herencia, objeto, método, evento, propiedad o atributo). Entornos de desarrollo integrados (IDE). Introducción al diseño y programación de interfaces gráficas de usuario.



[Handwritten signature]

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

- 1. Liderar y trabajar en equipo con eficacia y responsabilidad tomando decisiones basadas en evidencias.
- 2. Actuar proactivamente frente a los problemas sociales y ambientales.
- 3. Adaptarse respetuosamente a contextos nuevos o adversos, así como a diversidades personales, disciplinares y culturales.
- 4. Evaluar el comportamiento de diversos fenómenos disciplinares e interdisciplinares relacionados con la ingeniería electrónica con una visión de sistema, mediante modelos teóricos validados y actualizados, capaces de abarcarlos integralmente en un contexto de incertidumbre.
- 5. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de la ingeniería electrónica.
- 6. 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.
- 7. Diseñar e implementar sistemas electrónicos utilizando componentes de vanguardia.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de Aprendizaje
1. undamentos de las Estructuras de Datos.	1.1 Concepto de estructura de datos. 1.2 Clasificación de las estructuras de datos (lineales, no lineales). 1.3 Definición y tipos de datos abstractos. 1.4 Representación de estructuras de datos en memoria.	1. Define qué es una estructura de datos y su importancia en la programación. 2. Distingue entre tipos de datos primitivos y estructuras de datos. 3. Identifica las principales operaciones sobre estructuras de datos (inserción, búsqueda, eliminación).
2. Pilas y Colas.	2.1 Concepto de pila (LIFO). 2.2 Operaciones con pilas (push, pop, peek). 2.3 Aplicaciones de pilas (evaluación de expresiones, backtracking). 2.4 Concepto de cola (FIFO). 2.5 Operaciones con colas (enqueue, dequeue). 2.6 Aplicaciones de colas (gestión de procesos, simulaciones).	1. Implementa y utiliza pilas y colas en la resolución de problemas. 2. Analiza las aplicaciones de pilas y colas en diferentes algoritmos. 3. Compara las operaciones y características de pilas y colas.



[Handwritten signature]

Unidades	Contenidos	Resultados de Aprendizaje
3 Listas Enlazadas.	3.1 Concepto de lista enlazada. 3.2 Nodos de una lista enlazada. 3.3 Tipos de listas enlazadas (simples, doblemente enlazadas, circulares). 3.4 Operaciones con listas enlazadas (inserción, búsqueda, eliminación).	1. Implementa listas enlazadas simples enlazadas. 2. Analiza las ventajas y desventajas de las listas enlazadas frente a los arreglos. 3. Utiliza listas enlazadas para resolver problemas de inserción y eliminación de elementos.
4 Introducción a la Programación Orientada a Objetos.	4.1 Concepto de clase y objeto. 4.2 Encapsulación, herencia y polimorfismo. 4.3 Métodos, atributos y constructores. 4.4 Relaciones entre clases (asociación, agregación, composición).	1. Describe los conceptos fundamentales de la programación orientada a objetos. 2. Modela problemas del mundo real utilizando clases y objetos. 3. Utiliza herencia y polimorfismo para crear jerarquías de clases.
5 Uso de Librerías y Entornos de Desarrollo Integrados.	5.1 Concepto de librería. 5.2 Gestión de paquetes y dependencias. 5.3 Introducción a un IDE popular (por ejemplo, Visual Studio, Eclipse, PyCharm). 5.4 Entorno de trabajo, depuración y perfiles.	1. Utiliza librerías estándar y de terceros para agilizar el desarrollo de software. 2. Selecciona y utiliza un IDE adecuado para el desarrollo de aplicaciones. 3. Configura y personaliza un IDE para mejorar la productividad.
6 Diseño y Programación de Interfaces Gráficas de Usuario.	6.1 Conceptos básicos de diseño de interfaces de usuario. 6.2 Elementos de una interfaz gráfica (ventanas o pestañas, botones, etiquetas, etc.). 6.3 Manejo de eventos (clics, teclado, etc.). 6.4 Diseño de diálogos y formularios.	1. Diseña interfaces de usuario intuitivas y amigables. 2. Utiliza herramientas de diseño de interfaces gráficas. 3. Implementa interfaces gráficas utilizando un lenguaje de programación y una librería gráfica.



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:** estrategia de enseñanza donde se busca resolver un problema a través del conocimiento que adquirió en el aula, el estudiante toma liderazgo de su aprendizaje e identifica la importancia de su aprendizaje y el conocimiento.
- **Aprendizaje basado en proyectos:** metodología donde el estudiante participa activamente en su aprendizaje, desarrollando diferentes habilidades para solucionar un problema a través de un proyecto, y que pueda implementarse para la mejora del contexto.
- **Aprendizaje cooperativo colaborativo:** conjunto de métodos de instrucción para la aplicación en grupos pequeños, de entrenamiento y desarrollo de habilidades mixtas (aprendizaje, desarrollo personal y social) donde cada componente del grupo es responsable tanto de su aprendizaje como del de los restantes.
- **Debate:** 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 docente asume el rol de expositor y buscará generar el debate a través de preguntas sobre lo expuesto y desde la participación de los estudiantes.

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.

VII. MEDIOS AUXILIARES

Aula o plataforma virtual, pizarras acrílicas, proyector, marcadores, borrador de pizarra acrílica, equipo de audio, ordenadores, wifi, celulares, plataformas de videoconferencia y foros.

VIII. BIBLIOGRAFÍA

- Sznajdleder, P. (2017). Programación orientada a objetos y estructura de datos a fondo. Alpha Editorial.
- López, L. (2013). Metodología de la programación orientada a objetos. Alpha Editorial.
- Shaffer, C. A. (2010). A Practical Introduction to Data Structures and Algorithm Analysis Third Edition.
- Aho, A. V. (2001). Data Structures and Algorithms. stillwaters.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms. MIT press.
- Artículos científicos que serán provistos por el profesor.

