



UNIVERSIDAD NACIONAL DE ASUNCIÓN  
FACULTAD POLITÉCNICA  
CONSEJO DIRECTIVO

Campus de la UNA  
SAN LORENZO-PARAGUAY

RESOLUCIÓN 25/19/76-00  
ACTA 1227/08/09/2025

**"POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA  
FUNDAMENTOS DE PROGRAMACIÓN, DE CARRERAS DE GRADO, SEDE SAN  
LORENZO"**

**VISTO:** El Memorando DA/1796/2025 del Director Académico de la FP-UNA, Prof. MSc. Felipe Santiago Uzabal Escurra, 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 las Carreras de Grado.

**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 "**Dibujo Técnico**", la cual es común entre Carreras de Grado.

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:**

**25/19/76-01** APROBAR el Programa de Estudio de la Asignatura "**Dibujo Técnico**", de las Carreras de Grado, Sede San Lorenzo, detallado en el ANEXO 64 de la presente Acta.

**25/19/76-02** 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 POLÍTÉCNICA  
CONSEJO DIRECTIVO

Resolución 25/19/76-00 Acta 1227/08/09/2025  
ANEXO 64

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA  
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Nivel	Grado								
Asignatura	Fundamentos de Programación								
Carrera	Plan	Sede/Filial	Carácter	Semestre	Prerrequisitos				
Ingeniería Eléctrica	2026	Sede San Lorenzo	Obligatoria	Primero	Ninguno.				
Ingeniería en Electrónica	2026	Sede San Lorenzo	Obligatoria	Tercero	Ninguno.				
Semanal					Periodo				
HT	HP	HTD	HTI	HS	PA	THTD	THTI	THA	CA-PY
2	3	5	4	9	18	90	72	162	6

\*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 programación es una herramienta fundamental para el manejo y desarrollo de la tecnología moderna, y actualmente permite construir una multitud de caminos profesionales. Por ello, aprender a programar se ha transformado en un componente clave de lo que se denomina “competencias del Siglo XXI”. En este escenario, resulta esencial que el estudiante de disciplinas tecnológicas y científicas conozca los conceptos básicos de programación de manera a facilitar su interacción con otras asignaturas de la carrera; así como la extensión de este abordaje integral al ejercicio de la profesión.

Es usual emplear el paradigma de la programación estructurada en asignaturas introductorias a la programación, lo cual se hará mediante el lenguaje de programación C. Existen muchas razones del por qué usar el lenguaje C: permite ejecutar programas rápidamente, así como el manejo de bajo nivel para el control de sistemas electrónicos, mecatrónicos y eléctricos. Además, posee instrucciones que manipulan con facilidad bits, bytes y direcciones; lo cual es necesario para el control de dispositivos de E/S y para el gerenciamiento de memorias.

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: Fases de creación de un programa (análisis del problema, desarrollo, compilación y ejecución, corrección de errores, documentación). Algoritmo y pensamiento lógico. Representaciones del algoritmo. Introducción a los paradigmas de la programación. Uso de secuencia de instrucciones, bloques, variables y tipo de datos, operadores, estructuras de control condicionales y bucles, llamadas a funciones, recursividad, arreglos, punteros, preprocesador (lenguaje ANSI C).

### III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

1. Comunicarse en las lenguas oficiales del país y en una lengua extranjera.
2. Liderar y trabajar en equipo con eficacia y responsabilidad tomando decisiones basadas en evidencias.
3. Actualizarse permanentemente mediante la obtención y gestión autónoma de información de calidad, utilizando tecnología de la información y comunicación.
4. Seleccionar, construir y utilizar instrumentos innovadores asociados al ejercicio de la profesión.
5. Planificar, proyectar, diseñar y ejecutar proyectos sostenibles e integrales para la resolución de problemas, la mejora y la innovación en el ámbito profesional.

### IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de Aprendizaje
1. Introducción a la computación.	1.1 Importancia de la computación. Aplicaciones en la vida moderna. 1.2 Conceptos de computación, programas y algoritmos. 1.3 Formas de representación de algoritmos. 1.4 Pseudocódigo y lenguajes de programación. 1.5 Paradigmas de programación. 1.6 Fases de creación de un programa. 1.7 Entornos de desarrollo de programas.	1. Explica la importancia de la computación y su impacto en la vida moderna. 2. Describe los conceptos de computación, programas, y algoritmos.
2. Introducción a la programación en C.	2.1 Historia. 2.2 Esquema general de un programa en C. 2.3 Sintaxis. 2.4 Tipos de datos básicos. 2.5 Variables. Declaración de variables. 2.6 Identificadores. 2.7 Constantes. 2.8 Instrucciones. 2.9 Expresiones y operadores. 2.10 Comentarios. 2.11 Instrucciones básicas de entrada y salida de datos. 2.12 Directivas del preprocesador de C.	1. Aplica la sintaxis básica de C para escribir programas. 2. Utiliza tipos de datos, variables y operadores en la programación en C.



Unidades	Contenidos	Resultados de Aprendizaje
3. Elementos de la programación estructurada	3.1 Definición del paradigma de programación estructurada. 3.2 Estructuras condicionales. 3.2.1 Representación 3.2.2 Operadores lógicos y relacionales. 3.2.3 Expresiones booleanas. 3.2.4 Estructuras condicionales simples, alternativas y anidadas. 3.3 Estructuras repetitivas. 3.3.1 Representación. Concepto de bucle e iteración. 3.3.2 Repetición simple y condicional. 3.3.3 Contadores, acumuladores y banderas. 3.3.4 Anidamiento. 3.3.5 Interruptores y saltos. 3.4 Subprogramas 3.4.1 Abstracción. 3.4.2 Procedimientos y funciones. 3.4.3 Declaración, definición e invocación de subprogramas. 3.4.4 Parámetros y paso de parámetros. 3.4.5 Ámbito de variables. 3.4.6 Utilización de funciones externas. 3.4.7 Introducción a la recursividad.	1. Interpreta el paradigma de la programación estructurada y sus elementos. 2. Aplica estructuras condicionales y repetitivas. 3. Diseña funciones y procedimientos para la modularidad de los programas.
4. Arreglos y cadenas de caracteres.	4.1 Necesidad de estructuras de datos. 4.2 Arreglos unidimensionales (vectores) y bidimensionales (matrices) 4.2.1 Definición de arreglos. 4.2.2 Declaración en C. 4.2.3 Lectura e impresión de elementos.	1. Manipula arreglos unidimensionales y bidimensionales. 2. Realiza operaciones básicas con cadenas de caracteres.

Unidades	Contenidos	Resultados de Aprendizaje
	4.2.4 Paso de arreglos a funciones. 4.3 Cadenas de caracteres.	
5. Punteros.	5.1 Concepto. 5.2 Declaración de punteros. 5.3 Operadores asociados a punteros. 5.4 Paso de direcciones a funciones. 5.5 Aritmética de punteros. 5.6 Punteros y arreglos. 5.7 Arreglos de punteros. 5.8 Puntero doble.	1. Describe el concepto y uso básico de punteros en C. 2. Explica la relación entre punteros y arreglos en el lenguaje C.
6. Estructuras.	6.1 Variables definidas por el programador. 6.2 Estructuras. 6.2.1 Definición. 6.2.2 Inicialización 6.2.3 Asignación. 6.2.4 Punteros a estructuras. 6.2.5 Arreglos de estructuras. 6.2.6 Uso de typedef y sizeof. 6.3 Uniones. 6.4 Enumeraciones.	1. Utiliza estructuras para la creación de nuevos tipos de datos. 2. Relaciona los conocimientos previos con las estructuras para el manejo de tipos de datos complejos.
7. Archivos.	7.1 Conceptos asociados a archivos. 7.2 Apertura y cierre de archivos. 7.3 Lectura y escritura de archivos. 7.4 Acceso secuencial y directo.	1. Realiza operaciones básicas con archivos en C.



## VI. 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:

- **Exposición:** presentaciones introductorias de cada unidad por parte del docente en el laboratorio de informática, con conceptos fundamentales y ejemplos básicos a ser reproducidos por los estudiantes de manera a fomentar una mayor participación de los mismos y una mejor comprensión de lo presentado en clase.
- **Prácticas de Laboratorio:** sesiones de práctica en el laboratorio de informática para la resolución de problemas de aplicación de los principios de la unidad respectiva, con énfasis en la participación de los estudiantes.
- **Clase invertida:** disposición de materiales didácticos en aula virtual para reforzar el aprendizaje y facilitar la interacción entre docentes y estudiantes dentro y fuera del aula.

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, a excepción de las prácticas de laboratorio que tienen un carácter obligatorio en esta asignatura.

## VII. ESTRATEGIAS EVALUATIVAS

Evaluación continua de teoría y práctica, obtenida con la media de los controles realizados durante el curso a través de pruebas semanales y ejercicios de laboratorio. A criterio del docente, los estudiantes podrán desarrollar estas pruebas en grupos. Evaluación formativa a través de cuestionarios, exámenes cortos y ejercicios, para medir el progreso de los estudiantes y realizar ajustes.

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

## VIII. MEDIOS AUXILIARES

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

## IX. BIBLIOGRAFÍA

- Joyanes-Aguilar, L. (2024). *Fundamentos de programación: algoritmos, estructuras de datos y objetos* (5<sup>a</sup> ed.). McGraw-Hill.
- Kernighan, B., & Ritchie, D. (1988). *The C programming language* (2<sup>a</sup> ed.). Pearson.
- Joyanes-Aguilar, L., & Zahonero, I. (2004). *Algoritmos y estructuras de datos: Una perspectiva en C*. McGraw-Hill.
- Deitel, P., & Deitel, H. (2015). *C how to program* (8<sup>a</sup> ed.). Pearson.
- Thareja, R. (2012). *Computer fundamentals & programming in C*. Oxford University Press.
- Joyanes-Aguilar, L., Zahonero, I., Castillo, A., & Sánchez, L. (2005). *C: Algoritmos, programación y estructuras de datos*. McGraw-Hill.
- King, K. (2008). *C programming: A modern approach*. W. W. Norton & Company.
- Kochan, S. (2004). *Programming in C* (3<sup>a</sup> ed.). Pearson.



A handwritten signature in blue ink, appearing to be a stylized letter "P".