



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/102-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA ALGORITMOS NUMÉRICOS, DE CARRERAS DE GRADO, 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 Programa de Estudio de Asignatura 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 **“Algoritmos Numéricos”**, 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:**

25/18/102-01 APROBAR el Programa de Estudio de la Asignatura **“Algoritmos Numéricos”**, de carreras de Grado, Sede San Lorenzo, Filial Villarrica y Filial Coronel Oviedo, detallado en el ANEXO 64 de la presente Acta.

25/18/102-02 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 25/18/102-00 Acta 1226/25/08/2025
ANEXO 64

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Nivel		Grado									
Asignatura		Algoritmos Numéricos									
Carrera		Plan		Sede/Filial		Carácter		Semestre		Prerrequisitos	
Ingeniería en Informática		2026		Sede San Lorenzo		Obligatoria		Quinto		Álgebra Lineal, Ecuaciones Diferenciales, Algoritmos y Estructura de Datos 3.	
Licenciatura en Ciencias Informáticas		2026		Sede San Lorenzo/Filial Villarrica/Filial Coronel Oviedo		Optativa		***		Ninguno.	
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 introduce al estudiante a los fundamentos teóricos y las técnicas algorítmicas necesarias para trabajar con números grandes y campos finitos, herramientas fundamentales en aplicaciones avanzadas de la informática y las ciencias de la computación. La inclusión de esta asignatura en la malla curricular responde a la necesidad de formar profesionales con competencias en el diseño de algoritmos eficientes, especialmente en áreas críticas como la criptografía, la teoría de códigos y el procesamiento de datos. Estas competencias contribuyen directamente al desarrollo del perfil profesional al capacitar al estudiante en la resolución de problemas complejos y en la implementación de soluciones de alto impacto en diversos campos de aplicación.

Los campos finitos se utilizan en la ciencia de la computación y la informática en aplicaciones que involucran números enteros. Los límites en el direccionamiento de la memoria y la ejecución de operaciones aritméticas en unidades de procesamiento hacen que la utilización de los campos finitos sea natural. Dos aplicaciones muy importantes, por nombrar algunas, son la teoría de códigos y la criptografía.

En este curso se introduce al estudiante a las técnicas de diseño de algoritmos para números grandes y campos finitos. Además de las conocidas técnicas de diseño de algoritmos como “divide y conquista”, programación dinámica y otros, aquí se suman nuevas técnicas específicas para el manejo de números enteros. Igualmente, se abordan técnicas eficientes para el manejo de matrices, como métodos iterativos y factorizaciones, con énfasis en la implementación computacional y optimización de rendimiento.

Con las técnicas algorítmicas presentadas en este curso, los estudiantes aprenderán a realizar una implementación eficiente de la transformada rápida de Fourier, el cual es uno de los algoritmos más importantes y con mayor utilidad en las ciencias y la ingeniería. La transformada de Fourier tiene aplicaciones en procesamiento digital de señales, física cuántica, criptografía, diseño de códigos, y muchas más.

El curso además presenta una introducción a la teoría de códigos correctores y detectores de errores. Estos códigos se utilizan para el diseño de algoritmos de compresión, criptografía, transmisión de datos y almacenamiento de datos. Con esta breve introducción, se pretende que el estudiante posea las bases para involucrarse en temas más avanzados que involucre manejo de datos.

La naturaleza de la asignatura es teórico-práctica, ya que combina el desarrollo de fundamentos conceptuales con la implementación de algoritmos y el análisis de su rendimiento en contextos computacionales.

La organización de la asignatura se estructura en los siguientes ejes temáticos:

- Métodos iterativos: técnicas algorítmicas para aproximaciones numéricas.
- Estructuras algebraicas: fundamentos de álgebra y su relación con la computación.
- Aritmética de campos finitos: operaciones y aplicaciones en la informática.
- Algoritmos de matrices: métodos eficientes para manipulación matricial.
- Transformación de Fourier: diseño e implementación eficiente en campos finitos.
- Códigos: introducción a códigos detectores y correctores de errores.

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 relacionados con la informática con una visión de sistema, mediante modelos teóricos validados y actualizados, capaces de abarcarlos integralmente en un contexto de incertidumbre.
3. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de la informática.
4. Aplicar, producir y difundir conocimientos técnicos y científicos en el área de la informática.
5. Planificar, proyectar, diseñar y ejecutar proyectos sostenibles e integrales para la resolución de problemas, la mejora y la innovación en el ámbito de la informática.
6. Interpretar, modelar y comunicar información referida a la informática en forma gráfica.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de Aprendizaje
1. Métodos iterativos.	1.1. Números de punto flotante y precisión. 1.2. Overflow y underflow. 1.3. Redondeo. 1.4. Error relativo y error absoluto. 1.5. Método de la bisección.	1. Analiza las limitaciones de los números de punto flotante en la construcción de algoritmos, considerando su impacto en la precisión y estabilidad numérica. 2. Calcula el error relativo y absoluto en aproximaciones de

Unidades	Contenidos	Resultados de Aprendizaje
	1.6. Método de Newton-Raphson.	números de punto flotante, evaluando su influencia en los resultados computacionales. 3. Implementa en un lenguaje de programación métodos iterativos, como bisección y Newton-Raphson, para la localización de ceros de funciones, asegurando precisión y eficiencia en su ejecución.
2. Estructuras algebraicas.	2.1. Aritmética modular. Residuos cuadráticos. 2.2. Grupos finitos. 2.3. Anillos. 2.4. Campos finitos. 2.5. Polinomios sobre campos finitos. Polinomios irreducibles. Algoritmo de división. 2.6. Algoritmos para calcular generadores de grupos finitos. 2.7. Raíces primitivas de unidad sobre campos finitos. 2.8. Polinomios ciclotómicos. 2.9. Extensiones de campos finitos.	1. Analiza la aritmética modular y sus propiedades matemáticas. 2. Explica los conceptos de grupos finitos, anillos finitos y campos finitos, relacionándolos con su estructura matemática y aplicaciones. 3. Demuestra propiedades fundamentales de grupos, anillos y campos. 4. Representa grupos y campos mediante programas de computadora. 5. Calcula generadores de grupos finitos utilizando un lenguaje de programación de bajo nivel. 6. Representa y manipula polinomios en programas de computadora. 7. Identifica polinomios irreducibles mediante técnicas computacionales. 8. Implementa el algoritmo de división de polinomios en un lenguaje de programación. 9. Desarrolla algoritmos para identificar raíces primitivas de unidad y verifica su implementación en programas de computadora. 10. Diseña y programa algoritmos para la construcción de polinomios ciclotómicos, aplicándolos en contextos algebraicos y computacionales. 11. Construye extensiones de campo mediante métodos algorítmicos.
3. Aritmética de campos finitos.	3.1. Representación de números enteros grandes en base 2^{64} . 3.2. Reducción módulo m. 3.3. Adición, multiplicación,	1. Representa enteros grandes utilizando estructuras de datos adecuadas. 2. Demuestra matemáticamente propiedades fundamentales de la

Unidades	Contenidos	Resultados de Aprendizaje
	resta y división módulo m. 3.4. Multiplicación de Montgomery. 3.5. Exponenciación rápida. 3.6. Algoritmo de Euclides en campos primos. 3.6.1. Algoritmo binario. 3.6.2. Algoritmo Plus-Minus. 3.6.3. Pequeño teorema de Fermat. 3.7. Algoritmos eficientes para la construcción de extensiones. 3.7.1. Torres de extensiones.	aritmética modular. 3. Implementa algoritmos de adición, multiplicación, resta y división de enteros grandes en campos finitos. 4. Diseña e implementa la multiplicación utilizando la forma de Montgomery. 5. Desarrolla algoritmos para calcular el máximo común divisor en campos finitos. 6. Construye benchmarks para evaluar el rendimiento de algoritmos en la aritmética de campos finitos. 7. Diseña e implementa algoritmos para la construcción de torres de extensiones de campo. 8. Demuestra propiedades básicas sobre extensiones de campo.
4. Algoritmos de matrices	4.1. Algoritmos para operaciones básicas: suma, multiplicación y transposición. 4.2. Resolución de sistemas de ecuaciones lineales. 4.3. Métodos para cálculo de autovalores y autovectores (Potencias, QR iterativo). 4.4. Métodos iterativos para matrices dispersas (Jacobi, Gradiente conjugado).	1. Evalúa algoritmos eficientes para realizar operaciones básicas con matrices, como suma, multiplicación y transposición, optimizando el uso de recursos computacionales. 2. Implementa algoritmos para factorizaciones LU y QR, evaluando su estabilidad numérica y eficiencia en diferentes contextos computacionales. 3. Desarrolla algoritmos para resolver sistemas de ecuaciones lineales. 4. Aplica métodos algorítmicos para el cálculo de autovalores y autovectores. 5. Diseña y analiza métodos iterativos para resolver sistemas con matrices dispersas.
5. Transformación de Fourier.	5.1 Representación de polinomios. Vector de coeficientes. Vector de punto-valor. 5.2 Evaluación versus interpolación. 5.3 Interpolación de Lagrange. 5.4 Transformación de Fourier en campos finitos. 5.5 Transformación de Fourier rápida en campos finitos. 5.6 Multiplicación rápida de polinomios. 5.7 Multiplicación rápida de	1. Representa polinomios utilizando vectores de coeficientes y vectores de punto-valor. 2. Implementa la interpolación de Lagrange para polinomios en campos finitos. 3. Explica los fundamentos matemáticos de la Transformación de Fourier en campos finitos. 4. Implementa la Transformación Rápida de Fourier (FFT) en campos finitos. 5. Desarrolla algoritmos basados

Unidades	Contenidos	Resultados de Aprendizaje
	enteros: Karatsuba y Toom-Cook 5.8 Interpolación rápida.	en FFT para evaluación, interpolación y multiplicación de polinomios en campos finitos. 6. Construye benchmarks de los algoritmos basados en FFT.
6. Códigos.	6.1 Códigos, mensajes y canales. 6.2 Códigos detectores de errores y códigos correctores de errores. 6.3 Códigos lineales. Verificación de paridad. Código de repetición. Códigos de Hamming. 6.4 Decodificación de códigos lineales. Algoritmo de líderes de clases laterales. 6.5 Códigos cíclicos. 6.6 Códigos de Reed-Solomon.	1. Define formalmente código, mensaje, alfabeto y canal. 2. Construye códigos de verificación de paridad, códigos de repetición y códigos de Hamming. 3. Demuestra propiedades matemáticas básicas sobre códigos detectores y correctores de errores. 4. Implementa algoritmos de codificación y decodificación para códigos lineales. 5. Define formalmente códigos cíclicos. 6. Define formalmente códigos de Reed-Solomon.

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:

- **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.

VII. 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.

VIII. MEDIOS AUXILIARES

Aula virtual, pizarrón, proyector, marcadores, equipo multimedia.

IX. BIBLIOGRAFÍA

- Deschamps, J. P., Imana, J. L., & Sutter, G. D. (2009). Hardware implementation of finite-field arithmetic. McGraw-Hill Education.
- Mullen, G. L., & Mummert, C. (2007). Finite Fields and Applications (Student Mathematical Library Volume 41). American Mathematical Society.
- Lidl, R., & Niederreiter, H. (1997). Finite fields (No. 20). Cambridge University Press.
- 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.
- Shoup, V. (2009). A computational introduction to number theory and algebra. Cambridge university press.
- Mullen, G. L., & Panario, D. (2013). Handbook of finite fields. CRC press.
- The MoonMath Manual. Least Authority. <https://leastauthority.com/community-matters/moonmath-manual/>
- Menezes, A. J., Blake, I. F., Gao, X., Mullin, R. C., Vanstone, S. A., & Yaghoobian, T. (2013). Applications of finite fields (Vol. 199). Springer Science & Business Media.
- Silverman, J. H., Pipher, J., & Hoffstein, J. (2008). An introduction to mathematical cryptography (Vol. 1). Springer New York.
- Burden, R. L., Faires, J. D., & Burden, A. M. (2015). Numerical analysis. Cengage learning.

