



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/50-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA PARADIGMAS DE LA PROGRAMACIÓN, 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 **“Paradigmas de la Programación”**, 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:**

25/18/50-01 APROBAR el Programa de Estudio de la Asignatura **“Paradigmas de la Programación”**, de la carrera Licenciatura en Ciencias Informáticas – Plan 2026, Sede San Lorenzo, Filial Villarrica y Filial Coronel Oviedo, detallado en el ANEXO 12 de la presente Acta.

25/18/50-02 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 25/18/50-00 Acta 1226/25/08/2025
ANEXO 12

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Nivel			Grado							
Asignatura			Paradigmas de la Programación							
Carrera			Plan		Sede/Filial		Carácter		Semestre	Prerrequisitos
Licenciatura en Ciencias Informáticas			2026		Sede San Lorenzo/Filial Villarrica/Filial Coronel Oviedo		Obligatoria		Tercero	Estructura de Datos
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 Paradigmas de Programación proporciona una comprensión profunda de los diferentes enfoques y estilos de programación utilizables en la industria. Permite abordar conceptos fundamentales que subyacen a cada paradigma, permitiendo a los estudiantes tomar decisiones de manera objetiva sobre la selección y aplicación de paradigmas en el desarrollo de software y la solución a problemas reales. La comprensión profunda de cada paradigma de programación es más importante que el estudio de cada uno de los diferentes lenguajes.

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: Paradigmas de la programación. Tipos y características. Elección de paradigmas y diseño de programas. Tendencias actuales y futuras.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

1. Comunicarse en 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. 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.



[Handwritten signature]

[Handwritten signature]

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de aprendizaje
1. Calidad del Software	1.1. Calidad del Software. 1.2. Factores de calidad internos. 1.3. Factores de calidad externos. 1.4. Mantenimiento del software.	1. Identifica, entiende y documenta los requerimientos de sistemas de información. 2. Establece la calidad del software conforme los factores internos y externos.
2. Paradigmas y lenguajes de programación	2.1. Introducción a los paradigmas de programación. 2.2. Clasificación general de los paradigmas de programación. 2.3. Paradigmas de la programación y dominios de aplicación. 2.4. Breve historia, líneas del tiempo y evolución de los lenguajes de programación.	1. Clasifica y diferencia las técnicas de programación estructurada, orientada a objetos, guiada por eventos, funcional y lógico. 2. Enumera las características de los paradigmas de programación y los dominios de aplicación. 3. Toma decisiones en cuanto a selección de paradigmas de programación adecuado según el contexto del dominio de aplicación.
3. Programación estructurada	3.1. Programación Estructurada. 3.2. Arquitectura de Von Neumann. 3.3. Fundamentación teórica. 3.4. Recursos abstractos. 3.5. Diseño descendente. 3.6. Estructuras de control. 3.7. Programación Modular.	1. Comprende la terminología estándar de cada paradigma estudiado. 2. Implementa las diferentes técnicas de programación de los paradigmas según el contexto de aplicación de casos de estudio revisados. 3. Redacta documentaciones técnicas que acompañan las soluciones a casos planteados. 4. Trabaja en equipo para desarrollar soluciones a casos planteados.
4. Programación Orientada a Objetos	4.1. Conceptos y Características Asociadas a la Programación Orientada a Objetos. 4.2. Clases y Objetos. 4.3. Encapsulamiento. 4.4. Herencia. 4.5. Clases abstractas. 4.6. Polimorfismo. 4.7. El Modelo Orientado a Objetos. 4.8. Análisis y Diseño Orientado a Objetos. 4.9. Modelo de Objetos. 4.10. Diagrama de Clases.	1. Describe las fases del enfoque orientado a objetos. 2. Desarrolla, mantiene y evalúa sistemas y servicios basados en software cumpliendo estándares de calidad, aplicando teorías, principios, métodos y mejores prácticas de ingeniería de software. 3. Redacta documentaciones técnicas que acompañen soluciones a casos planteados. 4. Trabaja en equipo para desarrollar soluciones a casos planteados.



Unidades	Contenidos	Resultados de aprendizaje
	4.11. Identificación de Clases. 4.12. Relaciones entre Clases. 4.13. Generalización / Especialización. 4.14. Agregación / Composición. 4.15. Asociación. 4.16. Estructuras que relacionan tiempo de ejecución y de compilación. 4.17. Herencia frente a composición. 4.18. Persistencia de objetos.	
5. Programación Guiada por Eventos	5.1. Conceptos de la programación orientada a eventos. 5.2. El modelo de eventos. 5.3. Manipulación de eventos.	1. Comprende la terminología estándar de cada paradigma estudiado. 2. Desarrolla, mantiene y evalúa sistemas y servicios basados en software cumpliendo estándares de calidad, aplicando teorías, principios, métodos y mejores prácticas de ingeniería de software. 3. Redacta documentaciones técnicas que acompañan las soluciones a casos planteados. 4. Trabaja en equipo para desarrollar soluciones a casos planteados.
6. Programación Funcional	6.1. Programación Funcional. 6.2. Las funciones y el cálculo lambda. 6.3. Conceptos de la programación funcional. 6.4. Funciones de Orden Superior. 6.5. Transparencia referencial. 6.6. Evaluación perezosa. 6.7. Lenguajes funcionales puros e impuros.	1. Comprende la terminología estándar de cada paradigma estudiado. 2. Desarrolla, mantiene y evalúa sistemas y servicios basados en software cumpliendo estándares de calidad, aplicando teorías, principios, métodos y mejores prácticas de ingeniería de software. 3. Redactar documentaciones técnicas que acompañan las soluciones a casos planteados. 4. Trabaja en equipo para desarrollar soluciones a casos planteados.
7. Programación Lógica	7.1. Programación Lógica. 7.2. Lógica, predicados y cláusula de Horn. 7.3. Conceptos básicos de la programación lógica.	1. Comprende la terminología estándar de cada paradigma estudiado. 2. Desarrolla, mantiene y evalúa sistemas y servicios basados en

Unidades	Contenidos	Resultados de aprendizaje
	7.4. Campos de aplicación. 7.5. El lenguaje Prolog. 7.6. Hechos y reglas. 7.7. Líneas generales de programación y consultas.	software cumpliendo estándares de calidad, aplicando teorías, principios, métodos y mejores prácticas de ingeniería de software. 3. Redacta documentaciones técnicas que acompañan las soluciones a casos planteados. 4. Trabaja en equipo para desarrollar soluciones a casos planteados.

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:

- **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.
- **Clase invertida:** con materiales didácticos dispuestos en el aula virtual previamente y aplicar en clases presenciales, analizando y respondiendo a planteamientos con estudio de casos a través de trabajos grupales, orientadas especialmente al desarrollo de soluciones a problemas de programación utilizando paradigmas de programación adecuados.
- **Desarrollo de Proyectos Prácticos:** Los estudiantes trabajarán en proyectos prácticos que requieran la aplicación de múltiples paradigmas. Esto les permitirá experimentar de manera concreta cómo diferentes enfoques abordan un problema específico.

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

Diagnóstica, formativa y sumativa (exámenes parciales y finales), participación en clase, presentación y exposición de trabajos, trabajos de investigación y laboratorios

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, notebook, equipos de laboratorio de informática y software de lenguajes de programación y para las prácticas.

VIII. BIBLIOGRAFÍA

- Cristiá, M. (2007). Programación en Lógica con Prolog.
- Deitel, P., & Deitel, H. (2016). Java: cómo programar.

- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Patrones de Diseño: Elementos de Reutilización de Objetos Software.
- Louden, K. C. (2004). Lenguajes de Programación: Principios y Prácticas
- Martin, R. C. (2008). Código Limpio: Manual de Artesanía de Software Ágil.
- Meyer, O. (1998). Construcción De Software O/O (2nd ed.). Prentice-Hall.
- Scott, M. L. (2000). Pragmática de los Lenguajes de Programación.
- Serrano Mena, A. (2017). Programación Funcional en Haskell.
- Tucker, A., & Noonan, R. (2003). Lenguajes de programación: principios y paradigmas.
- Watt, D. A. (2004). Programming Language Design Concepts. John Wiley & Sons

