



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/116-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA INGENIERÍA SOFTWARE 2, DE LA CARRERA INGENIERÍA EN INFORMÁTICA – PLAN 2026, SEDE SAN LORENZO”

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 las Asignaturas de la Carrera Ingeniería en Informática.

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 **“Ingeniería Software 2”**, de la Carrera Ingeniería en Informática – 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:**

25/18/116-01 APROBAR el Programa de Estudio de la Asignatura **“Ingeniería Software 2”**, de la Carrera Ingeniería en Informática – Plan 2026, Sede San Lorenzo, detallado en el ANEXO 78 de la presente Acta.

25/18/116-02 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 25/18/116-00 Acta 1226/25/08/2025
ANEXO 78

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Nivel				Grado									
Asignatura				Ingeniería de Software 2									
Carrera				Plan		Sede/Filial		Carácter		Semestre		Prerrequisitos	
Ingeniería en Informática				2026		Sede San Lorenzo		Obligatoria		Séptimo		Ingeniería de Software 1.	
Semanal						Periodo							
Hs. Teó.		Hs. Prác.		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

El software es uno de los tres fundamentos de todo programa del área de la informática y la computación; los profesionales de dichas áreas, deben incluir en su perfil habilidades y destrezas que les permitan liderar proyectos de desarrollo de software, conocer los fundamentos en los cuales se basan las técnicas y metodologías para desarrollar software a nivel industrial, administrar la configuración del software, definir el proceso para construir software, definir las herramientas y métodos para crear software, asegurar la calidad del proceso y del producto, reconocer las áreas de conocimiento relacionadas.

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 calidad de software, pruebas de software, mantenimiento y evolución de software, patrones de diseño de software, ingeniería de software orientada a objetos, métodos ágiles.

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. Aplicar en la práctica profesional los valores humanos, la ética y los mecanismos de seguridad laboral.
4. Actuar proactivamente frente a los problemas sociales y ambientales.
5. Adaptarse respetuosamente a contextos nuevos o adversos, así como a diversidades personales, disciplinares y culturales.

- 6. Actualizarse permanentemente mediante la obtención y gestión autónoma de información de calidad, utilizando tecnología de la información y comunicación.
- 7. Evaluar el comportamiento de diversos fenómenos disciplinares e interdisciplinares relacionados con la ingeniería informática con una visión de sistema, mediante modelos teóricos validados y actualizados, capaces de abarcarlos integralmente en un contexto de incertidumbre.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de Aprendizaje
1. Calidad de software.	1.1. Conceptos de calidad. 1.2. Calidad y garantía de la calidad del software. 1.3. Actividades de garantía de la calidad de software. 1.4. Administración de la calidad y el desarrollo de software. 1.5. Estándares y aseguramiento de la calidad. 1.6. Calidad basada en procesos.	1. Analiza los conceptos y principios fundamentales de la calidad y calidad de software, identificando su importancia en el desarrollo de proyectos tecnológicos. 2. Describe y evalúa las principales actividades y estrategias utilizadas en la administración de la calidad de software, relacionándolas con casos prácticos. 3. Aplica estándares de calidad de software en la organización y ejecución de proyectos, demostrando cómo contribuyen a garantizar productos de alta calidad.
2. Pruebas de software.	2.1. Características de la facilidad de prueba. 2.2. Proceso de pruebas de software. 2.3. Quienes prueban el software. 2.4. Tipos de pruebas de software. 2.5. Diseño de casos de pruebas. 2.6. Automatización de las pruebas.	1. Explica los conceptos fundamentales relacionados con las pruebas de software, destacando su rol en el aseguramiento de la calidad. 2. Identifica y clasifica los diferentes tipos de pruebas de software, analizando sus objetivos y aplicaciones en proyectos tecnológicos. 3. Diseña y automatiza pruebas de software, implementando herramientas y técnicas adecuadas para garantizar la calidad del producto.
3. Mantenimiento y evolución del software.	3.1. Cambios en el software. 3.2. Estrategias de cambio del software. 3.3. Mantenimiento de software. 3.4. Transformación de arquitectura. 3.5. Reingeniería de software. 3.6. Dinámica de evolución de programas. 3.7. Software evolutivo.	1. Explica los conceptos clave relacionados con los cambios y el mantenimiento evolutivo del software, destacando su importancia en el ciclo de vida del producto. 2. Identifica y clasifica los diferentes tipos de mantenimiento de software, analizando sus objetivos y contextos de aplicación.

Unidades	Contenidos	Resultados de Aprendizaje
		3. Evalúa las condiciones bajo las cuales es necesario aplicar la reingeniería de software, justificando su importancia para la evolución y mejora del producto.
4. Patrones de diseño de software.	4.1. Patrones de diseño. 4.2. Diseño de software basado en patrones. 4.3. Patrones arquitectónicos. 4.4. Patrones de diseño en el nivel de componentes. 4.5. Patrones de diseño de la interfaz de usuario. 4.6. Patrones de diseño de webapp.	1. Comprender la utilización de patrones en el diseño de software. 2. Conocer los tipos de patrones de software. 3. Entender la forma correcta de aplicar los patrones al diseño.
5. Ingeniería de software orientada a objetos.	5.1. Conceptos orientados a objeto. 5.2. Clases y objetos. 5.3. Atributos. 5.4. Operaciones, métodos y servicios. 5.5. Análisis y diseño orientado a objeto. 5.6. Desarrollo orientado a objeto.	1. Explica la importancia de los patrones de diseño de software y su rol en la creación de soluciones reutilizables y eficientes. 2. Identifica y clasifica los principales tipos de patrones de diseño de software, analizando sus características y casos de uso. 3. Aplica patrones de diseño de manera adecuada en la solución de problemas de desarrollo de software, demostrando su integración en contextos prácticos.
6. Métodos ágiles.	6.1. Introducción al desarrollo ágil. 6.2. Características. 6.3. Manifiesto ágil. 6.4. Principales metodologías ágiles.	1. Analiza los principios y valores del manifiesto ágil, explicando su relevancia en el desarrollo de software y cómo contribuyen a la adaptabilidad y mejora continua. 2. Identifica y compara las principales metodologías ágiles, evaluando su idoneidad y eficacia en diferentes contextos de desarrollo de software. 3. Selecciona y justifica la metodología ágil más adecuada para un proyecto específico, considerando las necesidades del cliente, las características del equipo y los objetivos del proyecto.
7. Mejora de procesos.	7.1. Introducción a la mejora de procesos. 7.2. Modelo de calidad de producto software.	1. Explica el concepto de mejoras de proceso, destacando su importancia en la optimización de operaciones organizacionales.

- Pressman, R. S. (2010). Ingeniería del software: un enfoque práctico. (7° ed.). México: McGraw-Hill.
- Alonso, S. S., Urbán, M. Á. S., & García, D. R. (2011). Ingeniería del Software: Un enfoque desde la guía SWEBOK. Garceta.
- Sommerville, I. (2011). Ingeniería de software. (9° ed.). México: Addison Wesley.
- Weitzenfeld, A. (2005). Ingeniería de software orientada a objetos con UML, Java e Internet. México: Thomson.
- IEEE Std 1016. (1998). IEEE Recommended Practice for Software Design Descriptions.
- IEEE Std 1074. (1997). IEEE Standard for Developing Software Life Cycle Processes.
- IEEE/EIA Std 12207.0. (2004). IEEE Standard for Software Life Cycle Processes.

