



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 25/18/99-00
ACTA 1226/25/08/2025

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA INGENIERÍA DE SOFTWARE 1, 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 Programas 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 de Software 1”**, 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:**

25/18/99-01 APROBAR el Programa de Estudio de la Asignatura **“Ingeniería de Software 1”**, de la carrera Ingeniería en Informática – Plan 2026, Sede San Lorenzo, detallado en el ANEXO 61 de la presente Acta.

25/18/99-02 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

Resolución 25/18/99-00 Acta 1226/25/08/2025
ANEXO 61

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Nivel				Grado										
Asignatura				Ingeniería de Software 1										
Carrera				Plan		Sede/Filial		Carácter		Semestre		Prerrequisitos		
Ingeniería en Informática				2026		Sede San Lorenzo		Obligatoria		Cuarto		Lenguaje de Programación 3.		
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 ingeniería de software es una disciplina que incluye metodologías y técnicas para la generación de software correcto, optimizado y que cumpla con los requerimientos por los cuales nace el proyecto de desarrollo. Los profesionales de esta área deben incluir en su perfil habilidades y destrezas que le permitan liderar proyectos de desarrollo de software y conocer los fundamentos en los cuales se basan la especificación de requerimientos, el análisis y el diseño para desarrollar software con un importante grado de complejidad.

Esta asignatura otorga al estudiante una sólida preparación en los principios, metodologías y técnicas que guían la disciplina de la ingeniería de software y lo ayudará a desarrollar capacidades relacionadas a procesos, gestión, diseño e implementación de soluciones informáticas que involucren software.

La naturaleza de la asignatura esteórico-práctica y se requerirá iniciativa personal, capacidad de análisis y de síntesis, constancia para la comprensión y práctica durante el curso, además de una buena disposición tanto para el aprendizaje y desarrollo de la disciplina personal de estudio como para el trabajo en equipo. En este sentido, se requiere por parte del estudiante, predisposición y empatía para el trabajo cooperativo en equipo, fundamental en esta asignatura.

Organización en función de los ejes temáticos y las unidades

- Ingeniería de software: se aborda en la unidad 1, proporcionando una introducción a los conceptos clave y los principios fundamentales de la ingeniería de software.
- Procesos de software: se cubre en la unidad 2, en la cual se estudian los diferentes modelos de ciclo de vida y gestión de procesos en el desarrollo de software.
- Gestión de proyectos de software: se trata en la unidad 3, enfocándose en la planificación, ejecución, y control de proyectos de software, junto con la gestión de riesgos y recursos.

- Requerimientos de software: corresponde a la unidad 4, en la cual se estudian las técnicas para la adquisición, análisis y especificación de requerimientos.
- Diseño de software: se estudia en la unidad 5, abarcando la arquitectura, patrones de diseño y estructuración del código para su mantenimiento y escalabilidad.
- Implementación de software: se cubre en la unidad 6, enfocándose en las actividades de codificación, integración y pruebas del software.

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 con una visión de sistema, mediante modelos teóricos validados y actualizados, capaces de abarcarlos integralmente en un contexto de incertidumbre en el ámbito de la ingeniería informática.
8. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de la ingeniería informática.
9. Aplicar, producir y difundir conocimientos técnicos y científicos en el área de la ingeniería informática.
10. 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.
11. Interpretar, modelar y comunicar información referida a la ingeniería en informática en forma gráfica.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de aprendizaje
1. Introducción a la ingeniería de software.	1.1. Procesos de desarrollo tradicional y metodologías ágiles. 1.2. Importancia de la gestión de proyectos de software. 1.3. Responsabilidad y ética en la ingeniería de software.	1. Reconoce qué es la ingeniería de software y explica su importancia en el proceso de software. 2. Reconoce las diferencias entre el proceso tradicional y las metodologías ágiles para el desarrollo de software. 3. Explica la importancia para el éxito de un proyecto de software durante la gestión del mismo. 4. Identifica y analiza algunos conflictos éticos y profesionales en la práctica de la ingeniería de software.
2. Proceso de software.	2.1. Modelos de proceso de software. 2.1.1. Modelo en cascada (ciclo de vida clásico).	1. Interpreta conceptos y modelos sobre los procesos de software. 2. Identifica las principales actividades durante el proceso de

Unidades	Contenidos	Resultados de aprendizaje
	2.1.2. Modelo “V”. 2.1.3. Modelo de construcción de prototipos. 2.1.4. Modelos evolutivos. 2.1.5. Modelos ágiles. 2.2. Actividades del proceso. 2.3. Cambios en los requerimientos y diseño.	software. 3. Explica la metodología utilizada en los procesos para enfrentar los cambios en los requerimientos y el diseño.
3. Gestión de proyectos de software.	3.1. Actividades de gestión. 3.1.1. Gestión del alcance. 3.1.2. Planificación del proyecto. 3.1.3. Estimación del proyecto. 3.1.4. Seguimiento y control. 3.1.5. Gestión de riesgos.	1. Identifica las principales tareas de los administradores en los proyectos de software.
4. Requerimientos de software.	4.1. Adquisición y análisis de requerimientos. 4.2. Tipos de requerimientos de software. 4.3. Métodos de recolección de datos, de especificación y de comunicación de requerimientos.	1. Explica las actividades y técnicas principales de la ingeniería de requerimientos: adquisición y análisis durante el proceso de software. 2. Identifica las diferencias entre los tipos de requerimientos en los proyectos de software. 3. Comunica los requerimientos del sistema, de manera oral y escrita, por medio de un documento de especificación de requerimientos.
5. Diseño de software.	5.1. Lenguaje Unificado de Modelado UML. 5.2. Diagramas de diseño de software. 5.2.1. Interacciones en el sistema. 5.2.1.1. Casos de uso. 5.2.1.2. Diagrama de secuencia. 5.2.2. Comportamiento del sistema. 5.2.2.1. Diagramas de estados. 5.2.2.2. Diagramas de actividad. 5.2.3. Estructura de sistema. 5.2.3.1. Diagrama de clases. 5.2.3.2. Diagrama de objetos.	1. Aplica el UML para el diseño y modelado durante el proceso de software. 2. Identifica tipos de diagramas en el Lenguaje de Modelado Unificado (UML) y entiende cómo se utilizan dichos diagramas en el modelado del sistema.
6. Implementación de software.	6.1. Introducción a los patrones de diseño. 6.2. Actividades en la etapa de implementación de software. 6.3. Conflictos en la implementación de software. 6.4. Desarrollo de código abierto.	1. Reconoce los patrones de diseño como una solución general, reutilizable y aplicable a diferentes problemas de diseño de software. 2. Identifica las actividades y conflictos a considerar durante la implementación de software, incluidos la reutilización de software y el desarrollo de código abierto.

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 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.
- **Juego de roles:** se realizará la simulación de relevamiento y preparación de requerimientos funcionales y no funcionales, donde el docente toma el rol de cliente o interesado en un software y los estudiantes se dividen en grupos de trabajo, representando al equipo de analistas y desarrolladores, por cada área eligen a un líder de equipo que actuará de scrum master; se realizarán trabajos colaborativos de aplicación de los conceptos de la asignatura.

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.

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

VIII. BIBLIOGRAFÍA

- Pressman, R. S., & Maxim, B. R. (2019). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill.
- Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.
- Freeman, E., Robson, E., & Sierra, K. (2020). Head First Design Patterns: A Brain-Friendly Guide (2nd ed.). O'Reilly Media.
- Pantaleo, G., & Rinaudo, L. (2021). Ingeniería de software. Editorial Universitaria de Buenos Aires.
- Bass, L., Clements, P., & Kazman, R. (2021). Software Architecture in Practice (4th ed.). Addison-Wesley.
- Fowler, M. (2018). Refactoring: Improving the Design of Existing Code (2nd ed.). Addison-Wesley.
- Piatinni, M., & Cárdenas, L. (2018). Gestión de Proyectos de Software. Editorial McGraw-

Hill.

- IEEE Std 29148-2018. (2018). IEEE Standard for Requirements Engineering.
- IEEE/EIA Std 12207-2017. (2017). ISO/IEC/IEEE International Standard - Systems and Software Engineering - Software Life Cycle Processes.
- IEEE Std 24765-2017. (2017). IEEE Standard Glossary of Software Engineering Terminology. Institute of Electrical and Electronics Engineers.

