



Campus de la UNA
SAN LORENZO-PARAGUAY

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
CONSEJO DIRECTIVO

RESOLUCIÓN 24/26/68-00
ACTA 1208/16/12/2024

“POR LA CUAL SE APRUEBA EL PROGRAMA DE ESTUDIO DE LA ASIGNATURA INGENIERÍA DE SOFTWARE I, DE LA CARRERA LICENCIATURA EN CIENCIAS INFORMÁTICAS – PLAN 2023 DE LA FP-UNA”

VISTO: El Memorando DA/2437/2024 del Director Académico de la FP-UNA, Prof. MSc. Felipe Santiago Uzabal Ecurra, con el cual remite el Memorando CCPTCC/036/2024 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 **“Ingeniería de Software I”**, de la carrera Licenciatura en Ciencias Informáticas – Plan 2023, cuyo plan de estudio ya fue aprobado por el Consejo Superior Universitario.

**EL CONSEJO DIRECTIVO DE LA FACULTAD POLITÉCNICA
RESUELVE:**

24/26/68-01 APROBAR el Programa de Estudio de la Asignatura **“Ingeniería de Software I”**, de la carrera Licenciatura en Ciencias Informáticas – Plan 2023 de la FP-UNA, detallado en el ANEXO 60 de la presente Acta.

24/26/68-02 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 24/26/68-00 Acta 1208/16/12/2024
ANEXO 60

DEPARTAMENTO DE ENSEÑANZA DE INFORMÁTICA
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

Asignatura	Ingeniería de Software I				
Carrera	Plan	Sede/Filial	Carácter	Semestre	Prerrequisitos
Licenciatura en Ciencias Informáticas	2023	Sede San Lorenzo/Filial Villarrica/Filial Coronel Oviedo	Obligatoria	Cuarto	Paradigmas de la Programación, Base de Datos I
Horas semanales	4				
Total de horas teóricas semestral	36				
Total de horas prácticas semestral	36				
Total de horas semestral	72				
Valor en créditos académicos	La valoración en créditos académicos será comunicada en su oportunidad ajustada al Reglamento General del Sistema de Créditos Académicos de la UNA, el cual se encuentra en proceso de elaboración conforme a las disposiciones de la Resolución CONES N° 221/2024, en su artículo N° 10.				
Responsable(s) de elaboración	Prof. Ing. Deysi Natalia Leguizamón Correa				
Actualización	Al egreso de la primera cohorte.				

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 a través de un enfoque integral lo ayudará a comprender el proceso de desarrollo, la gestión de proyecto, consideraciones de seguridad y éticas y la experiencia de los usuarios de las soluciones informáticas que involucren software.

La naturaleza de la asignatura Ingeniería de software I será teórico-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.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS

1. Liderar y trabajar en equipo con eficacia y responsabilidad tomando decisiones basadas en evidencias.
2. Aplicar en la práctica profesional los valores humanos, la ética y los mecanismos de seguridad laboral.
3. Actuar proactivamente frente a los problemas sociales y ambientales.
4. 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 las ciencias informáticas.
5. Seleccionar, utilizar y construir instrumentos innovadores asociados al ejercicio de la profesión de las ciencias informáticas.
6. Planificar, proyectar, diseñar, ejecutar proyectos sostenibles e integrales para la resolución de problemas, la mejora y la innovación en el área de ciencias informática.
7. Interpretar, modelar y comunicar información en forma gráfica referida a las ciencias informáticas.

IV. ORGANIZACIÓN DE LA ASIGNATURA

Unidades	Contenidos	Resultados de aprendizaje
1. Ingeniería de Software	1.1 Introducción a la Ingeniería de Software 1.2 Dominios de Aplicación del Software 1.3 Ciclo de vida del desarrollo de Software 1.4 Metodologías de gestión de proyecto 1.5 Estructuración de las TIC en los proyectos de software.	1. Explica conceptos fundamentales de ingeniería de software, sus principios y prácticas, estableciendo una base sólida para el trabajo en equipo en proyectos de desarrollo de software. 2. Identifica los diferentes dominios de aplicación del software e identificar propuestas de soluciones ágiles para problemas sociales y ambientales. 3. Explica el ciclo de vida del desarrollo de software 4. Analiza y selecciona metodologías de gestión de proyectos de software según características de cada proyecto. 5. Aplica técnicas de estructuración de TICs en proyectos de software, considerando concordancia con las disposiciones legales, las normativas y los mecanismos de seguridad laboral, eficiencia, escalabilidad y adaptabilidad a medida que se desarrollan soluciones tecnológicas.

Unidades	Contenidos	Resultados de aprendizaje
2. Sistemas de Información y la Ingeniería de Software	2.1 La Ingeniería de Software aplicada a Sistemas de Información 2.2 Importancia de la Ingeniería de Software para el desarrollo de sistemas de información efectivos 2.3 Gestión de Proyectos de Sistemas de Información 2.4 Especificación, priorización y factibilidad de proyectos (operacional, costo/beneficio tangibles e intangibles, cuestiones técnicas de planificación, cuestiones legales y culturales). 2.5 Abordajes para implementar sistemas de información (sistemas empaquetados, corporativos, outsourcing/in house). 2.6 Consideraciones de seguridad y éticas.	1. Aplica principios y prácticas de la Ingeniería de Software a lo que son los Sistemas de Información. 2. Valora la importancia de la Ingeniería de Software en el desarrollo de sistemas de información efectivos. 3. Gestiona proyectos de sistemas de información. 4. Analiza y prioriza requisitos de proyectos de Sistemas de Información, considerando factores operacionales, costos y beneficios tangibles e intangibles, cuestiones técnicas de planificación, así como aspectos legales y culturales. 5. Distingue diferentes abordajes para la implementación de sistemas de información, garantizando integridad, confidencialidad y disponibilidad de la información. 6. Aplica consideraciones de seguridad y éticas en el proceso de ciclo de vida del desarrollo del software.
3. Modelos de Proceso de desarrollo de Software	3.1 Modelos de proceso de Software 3.2 Tipos de modelos de proceso: 3.2.1 Modelo en Cascada (Ciclo de Vida Clásico) 3.2.2 Modelo "V" 3.2.3 Modelo de procesos basados en Prototipos 3.2.4 Modelos Evolutivos 3.2.5 Modelos Ágiles	1. Interpreta conceptos y modelos sobre los procesos de software 2. Diferencia los tipos de modelos de proceso. 3. Explica la metodología utilizada en los procesos para enfrentar los cambios en los requerimientos y el diseño.
4. Requerimientos	4.1 Definición de requerimientos. 4.2 Métodos de recolección de datos. 4.3 Métodos de estructuración de requerimientos. 4.4 Comunicación efectiva de requerimientos.	1. Explica requerimiento y su papel fundamental en el ciclo de vida del desarrollo de software. 2. Aplica métodos efectivos de recolección de datos para identificar los requerimientos del usuario y del sistema. 3. Utiliza métodos de estructuración de requerimientos. 4. Facilita la comunicación efectiva de los requerimientos, empleando técnicas apropiadas para comprender los requerimientos en el proceso de desarrollo de

Unidades	Contenidos	Resultados de aprendizaje
		software.
5. Análisis y especificación de requerimientos del negocio	5.1 Análisis de requerimientos 5.2 Importancia del análisis de requerimientos en el ciclo de vida del desarrollo de software. 5.3 Roles y responsabilidades en el proceso de análisis de requerimientos 5.4 Tipos de Requerimientos 5.4.1 Requerimientos Funcionales 5.4.2 Requerimientos No Funcionales 5.5 Especificación de requerimientos 5.6 Modelado de los procesos y de los requerimientos de información 5.7 Validación de requerimientos utilizando el modelo de proceso basado en "Prototipos": Diseño de Interfaces de usuario y Experiencia de Usuario	1. Elabora un análisis de requerimientos, aplicando técnicas y métodos para identificar y comprender las necesidades del negocio. 2. Valora la importancia del análisis de requerimientos en el ciclo de vida del desarrollo de Software. 3.Diferencia los roles y responsabilidades específicos en el proceso de análisis de requerimientos. 4. Distingue los tipos de requerimientos, clasificándolos en funcionales y no funcionales. 5. Especifica requerimientos de manera clara y detallada, utilizando métodos adecuados para documentar con precisión las características y funcionalidades del software. 6. Aplica técnicas de modelado de procesos y requerimientos de información. 7. Valida requerimientos a través de un modelo de proceso basado en Prototipos, diseñando interfaces de usuario y garantizando una experiencia de usuario efectiva.
6. Introducción al diseño de Software	6.1 Diseño de software 6.2 Importancia del diseño de software en proyectos de desarrollo de software 6.3 Introducción al Lenguaje de Modelado Unificado y su relevancia en la Ingeniería de Software 6.4 Diagramas de comportamiento: 6.4.1 Casos de Uso y Especificaciones CdU 6.4.2 Diagrama de Clases Conceptual 6.4.3 Diagrama de Estados 6.4.4 Diagrama de Actividades	1. Explica los principios fundamentales de diseño de software 2. Valora la importancia del diseño de software en proyectos de desarrollo. 3. Aplica los conceptos introductorios al UML, reconociendo su importancia para representar el diseño del software. 4. Crea y utiliza diagramas de comportamiento para modelar las interacciones y los procesos del software.

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 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 y se realiza trabajos colaborativos de aplicación de los conceptos de la asignatura.

La elección particular de la estrategia didáctica aplicada será explícita en el plan de clases, 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 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.

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

VIII. MEDIOS AUXILIARES

Aula virtual, pizarras acrílicas, proyector, marcadores, ordenadores, wifi, celulares, plataformas de videoconferencia y foros.

IX. BIBLIOGRAFÍA

- Calero, C. (Ed.) (2015). Green in software engineering. Springer.
- Jacobson, I., Grady Boock & Rumbaugh, J. (2000). El proceso unificado de desarrollo de software. Madrid: Pearson Educación
- Pressman, R. S. (2010). Ingeniería del software: un enfoque práctico. (7° ed.). México: McGraw-Hill.
- Pressman, R. (2010). Ingeniería del software: un enfoque práctico (7a. ed.). Retrieved from <http://ebookcentral.proquest.com>
- Sánchez Alonso, S., Sicilia Urbán, M. Á. & Rodríguez García (2011). Ingeniería del software: un enfoque desde la guía SWEBOK. Madrid: Ibergaceta.
- Sommerville, I. (2011). Ingeniería de software. (9° ed.). México: Addison Wesley.
- Institute of Electrical and Electronics Engineers Computer Society. (2014). Software Engineering Body of Knowledge (SWEBOK).
- Weihrich, H., Cannice, M. V. & Koontz, H. (2017). Administración: una perspectiva global, empresarial y de innovación. México: McGraw Hill Education.

