

Presidencia Roque Sáenz Peña, 08 de mayo de 2025

**RESOLUCIÓN N° 117/2025 - C.D.C.B. y A.**

**VISTO:**

El Expediente N° 01-2025-01741 sobre propuesta de Programa actualizado de la asignatura Ingeniería de Software de la carrera Ingeniería en Sistemas de Información, iniciado por la Directora de Carrera Ing. ZACHMAN, Patricia; y

**CONSIDERANDO:**

Que la asignatura 27-Ingeniería de Software corresponde al Área de Sistemas de Información y se dicta en el 4° año 2° cuatrimestre de la Carrera Ingeniería en Sistemas de Información;

Que el Programa Analítico contempla los contenidos mínimos y la carga horaria propuestos en el Plan de estudios de la carrera, Resolución N° 063/19-C.S. y las asignaturas correlativas respetan lo establecido en el Sistema de Correlatividades de la Carrera aprobado por Resolución N° 088/19-C.S.;

Que la fundamentación refleja la relevancia de la asignatura en la formación de los futuros profesionales y los objetivos planteados guardan coherencia con los contenidos, métodos pedagógicos y de evaluación propuestos;

Que los Trabajos Prácticos planteados son pertinentes y adecuados y la bibliografía propuesta es actualizada;

Lo aprobado en sesión de la fecha.

**POR ELLO:**

**EL CONSEJO DEPARTAMENTAL  
DEL DEPARTAMENTO DE CIENCIAS BÁSICAS Y APLICADAS DE LA  
UNIVERSIDAD NACIONAL DEL CHACO AUSTRAL  
RESUELVE:**

ARTÍCULO 1°: APROBAR el Programa de la asignatura Ingeniería de Software de la carrera Ingeniería en Sistemas de Información, que como Anexo Único forma parte de la presente Resolución.

ARTÍCULO 2°: Regístrese, comuníquese, y archívese.

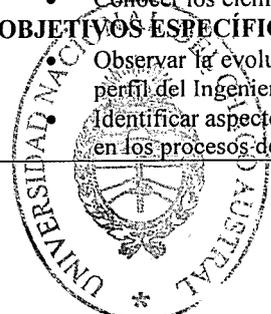


*Nora B. Okulik*  
Dra. Nora B. Okulik  
Directora  
Dpto. de Cs. Básicas y Aplicadas



**ANEXO: PROGRAMA DE ASIGNATURA**

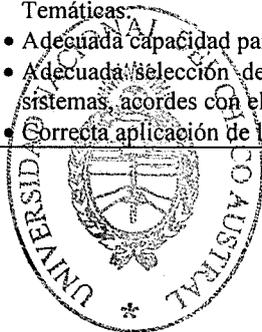
|   |   |   |   |
|---|---|---|---|
|  <p><b>UNCAUS</b><br/>UNIVERSIDAD<br/>NACIONAL DEL<br/>CHACO AUSTRAL</p> |   | <p align="center"><b>29 - INGENIERÍA DE SOFTWARE</b><br/><b>Plan de Estudios Resolución N°063/19-C.S.-</b></p>  |   |
| <p>Carga Horaria: 90 horas<br/>Teóricas: 40 horas<br/>Prácticas: 50 horas</p>   |   | <p>Programa vigente desde 2025</p>  |   |
| <p align="center">Carrera</p>   |   | <p>Año</p>  | <p>Cuatrimestre</p>   |
| <p><b>Ingeniería en Sistemas de Información</b></p>   |   | <p><b>4°</b></p>  | <p><b>Segundo</b></p>   |
| <p align="center">CORRELATIVAS PRECEDENTES</p>  |   | <p align="center">CORRELATIVAS SUBSIGUIENTES</p>  |   |
| <p align="center">Asignaturas</p>   |   | <p align="center">Asignaturas</p>   |   |
| <p align="center">Para cursar</p>   |   | <p align="center">Para rendir</p>   |   |
| Regularizadas   | Aprobadas   | Aprobadas   | <p>-Proyecto Final<br/>-Sistemas de Gestión<br/>-Práctica Profesional Supervisada</p> |
| <p>- Gestión de Datos</p>   | <p>-Análisis de Sistemas.<br/>-Paradigmas de Programación.<br/>-Probabilidad y Estadística.<br/>-Sintaxis y Semántica de los Lenguajes.</p> | <p>-Gestión de Datos</p>  |   |
| <p><b>DOCENTES:</b></p>   |   | <p>Prof. Adjunto: Dra. Ing. ZACHMAN, Patricia<br/>JTP: Ing. IZNARDO, Natanael</p>   |   |
| <p><b>FUNDAMENTACIÓN:</b></p>   |   | <p>La asignatura Ingeniería de Software constituye un componente esencial en la formación de profesionales ingenieros en sistemas de información, pues ofrece un marco teórico-práctico que permite al estudiante dominar los principios fundamentales de la disciplina, integrando herramientas técnicas, metodologías sistemáticas y una perspectiva ética para concebir soluciones de software robustas, eficientes y alineadas con las demandas de las organizaciones contemporáneas.</p> <p>En un contexto donde el software impulsa los procesos estratégicos, tácticos y operativos de instituciones de diversa índole, Ingeniería de Software se presenta como un espacio de aprendizaje que habilita al estudiante a resolver problemas complejos mediante la aplicación rigurosa de procesos y modelos. Los contenidos mínimos —Componentes de un Proyecto de Software de Sistemas de Información, Gestión de Configuración de Software, Modelos de Calidad de Software, Aseguramiento de la Calidad, Métricas de Software, y Auditoría y Peritaje de Software— sientan las bases para que el futuro profesional desarrolle habilidades de liderazgo en la gestión de proyectos tecnológicos, diseño arquitecturas innovadoras y sostenibles, y garantice productos de alta confiabilidad que respondan a estándares internacionales.</p> <p>La asignatura fomenta un aprendizaje que trasciende lo técnico, cultivando en el estudiante una visión integral que combina la planificación estratégica, el diseño creativo y la responsabilidad ética en la construcción de software. Así, se prepara a un profesional versátil, capaz de desempeñarse en equipos multidisciplinarios, adaptarse a los retos de un mercado globalizado y contribuir al éxito organizacional con soluciones tecnológicas que impactan positivamente en la sociedad.</p> |   |
| <p><b>OBJETIVOS:</b></p>  |   | <p><b>OBJETIVOS GENERALES</b></p> <ul style="list-style-type: none"> <li>Comprender la Ingeniería de Software y explicar su importancia.</li> <li>Incorporar la práctica sistemática de los principios de la Ingeniería de Software.</li> <li>Conocer los elementos de auditoría y peritaje de software.</li> </ul> <p><b>OBJETIVOS ESPECÍFICOS</b></p> <ul style="list-style-type: none"> <li>Observar la evolución del software y comprender el alcance del perfil del Ingeniero de Software en la actualidad.</li> <li>Identificar aspectos de ingeniería en la captura de requerimientos, en los procesos de desarrollo de software y en el diseño.</li> </ul>  |   |





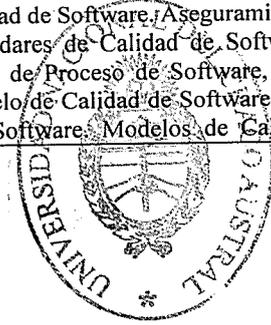
///Res. N° 117/2025-DCByA.

|                                      |   |
|--------------------------------------|---|
|                                      | <ul style="list-style-type: none"> <li>• Ejercitarse en la gestión y dirección de proyectos, en lo que se refiere a recursos humanos, tiempos, calidad, alcances, comunicaciones, riesgos y adquisiciones.</li> <li>• Diferenciar tipos de revisiones en el software, cubriendo el alcance de verificación, validación y testing.</li> <li>• Planificar la calidad de software desde modelos y estándares</li> <li>• Interpretar la auditoría informática y el marco jurídico respecto de programas, datos, personas, delitos e intercambios de datos.</li> </ul>   |
| <p><b>CONTENIDOS MÍNIMOS:</b></p>    | <p>Componentes de un Proyecto de Software de Sistemas de Información. Gestión de Configuración de Software. Modelos de Calidad de Software. Aseguramiento de la Calidad. Métricas de Software. Auditoria y Peritaje de Software.</p>  |
| <p><b>MÉTODOS PEDAGÓGICOS:</b></p>   | <p>Se propone:<br/>El desarrollo del pensamiento crítico, reflexivo, práctico y autónomo sobre la práctica educativa.<br/>El desarrollo de habilidades y competencias propias del modelado de sistemas de información para su aplicación en problemas computacionales reales.<br/>En esta propuesta, el aula se entiende como un espacio de taller para la construcción, en el que se trabaja interactuando permanentemente. La retroalimentación se concreta con una estructura bidireccional, donde tanto los alumnos como el docente se consideran fuente de información. Se han considerado las siguientes técnicas metodológicas:</p> <ul style="list-style-type: none"> <li>_ Exposiciones magistrales.</li> <li>_ Trabajo en grupo e individual</li> <li>_ Exposiciones abiertas orales</li> <li>_ Estudio dirigido mediante guías orientadoras</li> <li>_ Investigación Bibliográfica.</li> </ul> <p>La exposición magistral, las exposiciones abiertas y el estudio dirigido se emplearán por lo general para presentar temas nuevos, mientras que el trabajo en grupo, la investigación bibliográfica y en Internet, se emplearán para su aplicación en ejercitaciones reales y profundizar en un tema ya presentado.<br/>En las clases prácticas la técnica metodológica por excelencia será el trabajo taller grupal que permite promover la construcción compartida del conocimiento y lograr así no sólo la apropiación activa del mismo por parte de los miembros del grupo, sino también la indispensable socialización del estudiante, ya que toda su vida deberá transcurrir en contacto y en cooperación con sus semejantes.</p> |
| <p><b>MÉTODOS DE EVALUACIÓN:</b></p> | <p><b>Condiciones para lograr la Regularidad de la Asignatura</b></p> <ul style="list-style-type: none"> <li>• Reunir un 75% de asistencia en las clases teóricas y 75% en clases prácticas.</li> <li>• Aprobación del 100% de Trabajos Prácticos.</li> <li>• Aprobar los 3 (tres) parciales teórico - prácticos</li> </ul> <p><b>Examen Final Regular:</b> La evaluación final será escrita u oral sobre los temas incluidos en la programación analítica de la asignatura.<br/><b>Examen Libre:</b> Los alumnos libres deberán cumplir las siguientes etapas, cada una de ellas eliminatória.</p> <p>1ra etapa) Aprobar una evaluación escrita de tipo práctica.<br/>2da etapa) Aprobar una evaluación oral de tipo teórica sobre todos los contenidos del programa.</p> <p><b>Sobre evaluaciones parciales:</b><br/>Parcial I: Unidades 1, 2 y 6. Escrita, individual, y teórico-práctica.<br/>Parcial II: Unidad 3. Software, individual, y teórico-práctica.<br/>Parcial III: Unidades 4, 5 y 7. Escrita, individual, y teórico-práctica.<br/>Los criterios de evaluación a aplicar en las evaluaciones parciales son:</p> <ul style="list-style-type: none"> <li>• Correcto manejo de conceptos teóricos vinculados a las Unidades Temáticas</li> <li>• Adecuada capacidad para sintetizar e integrar informaciones e ideas.</li> <li>• Adecuada selección de las metodologías y técnicas de desarrollo de sistemas, acordes con el problema a resolver.</li> <li>• Correcta aplicación de las técnicas seleccionadas.</li> </ul>   |



4

|   |  |
|---|--|
|   | <ul style="list-style-type: none"> <li>• Completos y correctos desarrollos prácticos y teóricos.</li> <li>• Calidad de la presentación: la documentación entregada deberá ser clara, libre de errores de ortografía, ordenada, concisa y acotada a lo que se le solicita.</li> </ul> <p>Se aplica la normativa vigente.</p>  |
| <p><b>PROGRAMA ANALÍTICO DE CONTENIDOS:</b></p> | <p><b>UNIDAD 1: SOFTWARE E INGENIERIA DE SOFTWARE</b><br/>El software su evolución y naturaleza cambiante. El software heredado. La teoría unificada para la evolución del software. Los mitos del software. Ingeniería de Software. Visión estratificada de la I.S. Ramas de la Ingeniería de Software. Similitudes y diferencias con otras Ingenierías. Producto y Proceso. Perfil del Ingeniero en Software. Ética en la Ingeniería de Software. El código de ética de la IEEE para la Ingeniería de Software.</p> <p><b>UNIDAD 2: INGENIERIA DE REQUERIMIENTOS</b><br/>El proceso de requerimientos. Beneficios e inconvenientes. Tipos de requerimientos. Características de los requerimientos. Documentación. Participantes en el proceso de especificación de requerimientos. El proceso de Ingeniería de requerimientos. Técnicas de Análisis y Especificación de requerimientos. Validación y medición de requerimientos.</p> <p><b>UNIDAD 3: INGENIERIA DE PROCESOS DE DESARROLLO SOFTWARE</b><br/>Modelos de Procesos de Software. Modelo lineal secuencial. Modelo de construcción de prototipos. Modelo DRA. Modelos evolutivos. Modelo incremental. Modelo espiral. Modelo de desarrollo concurrente. Desarrollo basado en componentes. Desarrollo ágil. Herramientas y técnicas para modelado de procesos. Mapa de actividades de un proyecto. Definición de un proceso software. Madurez del proceso software. Herramientas para el proceso de software.</p> <p><b>UNIDAD 4: GESTIÓN Y DIRECCIÓN DE PROYECTOS</b><br/>Fundamentos y finalidad del PMBOK, definición de proyecto, análisis de dirección de, relación entre: dirección de proyecto, dirección de programas y gestión del portafolio. Ciclo de vida del Proyecto y Organización. Procesos de Dirección de Proyectos para un proyecto: Gestión de la Integración del proyecto, Gestión de Alcance del proyecto, gestión del tiempo, gestión de costos, gestión de calidad, gestión de recursos humanos, gestión de las comunicaciones del proyecto, gestión del riesgo y gestión de las adquisiciones del proyecto.</p> <p><b>UNIDAD 5: INGENIERÍA DE DISEÑO</b><br/>Arquitectura de software: Definición. Objetivos. Conflictos. Principios del diseño de software. Propiedades del diseño de software. Descomposición. Arquitectura de software y diseño detallado. Diseño arquitectónico: Componentes, propiedades y relaciones. Estilos arquitectónicos (cliente-servidor, capas objetos, tubos, centralizado) sus ventajas y desventajas. Patrones. Estrategias generales para el diseño de software. Lenguajes de Descripción de Arquitecturas. Arquitecturas CASE. UML. Alternativas opensource para UML. Rational Rose. Arquitectura MVC. Reutilización de componentes.</p> <p><b>UNIDAD 6: VERIFICACION Y VALIDACION DE SOFTWARE</b><br/>Verificación, Validación y Testing: diferencias y momentos. Revisiones de requerimientos, Revisiones de Diseño, Inspecciones de Código y Pruebas del Producto. Inspecciones y Pruebas. VV estática y dinámica. Pruebas de Validación y Pruebas de defectos. Pruebas de integración y Pruebas de entregas. Prueba de Caja Negra. Pruebas de rendimiento. Pruebas de Unidad. Pruebas de Interfases, Diseño de Casos de Prueba. Pruebas basadas en requerimientos, Pruebas de particiones y Pruebas estructurales. Automatización de pruebas.</p> <p><b>UNIDAD 7: CALIDAD DE SOFTWARE</b><br/>Calidad. Gestión de la Calidad. Planificación de la Calidad. Control de la Calidad de Software. Aseguramiento de la Calidad de Software. Modelos y Estándares de Calidad de Software. Calidad a Nivel Organizacional, a Nivel de Proceso de Software, a Nivel de Software. Calidad de Datos. Modelo de Calidad de Software: CMM yCMMi, Tickit, PSP y Six Sigma For Software. Modelos de Calidad de Software a Nivel de Producto.</p> |



|   |   |
|---|---|
|   | <p>Estándares de Calidad del Software. ISO 90003, ISO/IEC 9001, IEEE/EIA, COBIT, ITIL ISO 20000, otros. Comparación entre modelos.</p> <p><b>UNIDAD 8: AUDITORIA DE SISTEMAS Y PERITAJE.</b></p> <p>Conceptos de auditoría informática, Diversos tipos de auditoría y su relación con la informática. Ventajas de la informática como herramienta de la auditoría financiera. Control interno, definición y metodologías de controles internos. El informe de auditoría: normas, evidencia, irregularidades, documentación y conclusiones. Marco jurídico de la auditoría informática: protección de datos personal, protección jurídica de los programas, base de datos, delitos informáticos, intercambio electrónico de datos. Principales áreas de la auditoría informática: auditoría física, ofimática, auditoría de la dirección, de explotación, de desarrollo, auditoría del mantenimiento y auditoría de bases de datos. Introducción al Peritaje. La investigación forense: etapas de una investigación forense, requisitos de la investigación, valoración jurídica de la prueba digital.</p>   |
| <p><b>PROGRAMA ANALÍTICO DE TRABAJOS PRÁCTICOS:</b></p> | <p>TP1: Ingeniería de Software y su evolución. Atributos del software. Productos genéricos y personalizados.</p> <p>TP2: Ingeniería de Requerimientos / Ingeniería de Procesos / Ingeniería de Diseño.</p> <p>TP3: Pruebas de Software. Complejidad ciclomática, testing, tipos de verificación y validación.</p> <p>TP4: Proyectos de Sistemas. PMBOOK.</p> <p>TP5: Métricas y Calidad de Software.</p>  |
| <p><b>BIBLIOGRAFÍA:</b></p>                             | <p><b>Básica:</b></p> <ol style="list-style-type: none"> <li>Somerville I (2005). Ingeniería de Software – 9na. Edición, Editorial Pearson - (Unidad I, Unidad 2, Unidad 3, Unidad 5, Unidad 6 y Unidad 7)</li> <li>Pressman R. (2021) Ingeniería de Software – Un práctico -9na. Edición, , Editorial Mc Graw Hill (Unidad I, Unidad 2, Unidad 3, Unidad 5, Unidad 6 y Unidad 7)</li> <li>Roda García J.L., Brito Santana J., Canarias (2001) Introducción a la Ingeniería de Software. Dirección General de Universidades e Investigación (Unidad 1, Unidad 2, Unidad 3, Unidad 5, Unidad 6 y Unidad 7)</li> <li>Cerrada M., Collado M. (2000) Introducción a la Ingeniería de Software, Editorial Centro de Estudios Ramón Araces SA -(Unidad 1, Unidad 2, Unidad 3, Unidad 5, Unidad 6 y Unidad 7)</li> <li>Desarrollo de software dirigido por modelo: Conceptos, métodos y herramientas. ISBN 978-84-9964-215-4 Editorial RA-MA (Unidad 5 y Unidad 6)</li> <li>Sarmiento Rojas A. (2024) Gestión de Proyectos aplicada con PMBOOK. Editorial UPTC (Unidad 4)</li> <li>Abraham S. (2021) Calidad y sostenibilidad de sistemas de información en la práctica Editorial RA-MA (Unidad 7)</li> <li>Rosales M. (2022) Guía de recomendaciones para la preservación de la prueba sobre el uso y acceso a los Sistemas de Información en un entorno corporativo. Editorial Universidad Fasta.</li> <li>Piattini M., García Rubio F. Rodríguez de Guzmán I. (2012) Calidad de los Sistemas de Información,, Editorial Alfaomega Ra-Ma</li> <li>Piattini M., García Rubio F. Rodríguez de Guzmán I (2008) Medición y Estimación de Software. Editorial Alfaomega Ra-Ma</li> </ol> <p><b>Complementaria:</b></p> <ul style="list-style-type: none"> <li>Ingeniería de Software Benet Campderrich Falgueras, Ramon Maspons Editorial Universidad Abierta de Cataluña -2003</li> <li>Fábricas de Software: experiencias, tecnologías y organización, Piattini M. , Garzás Parra J., Editorial Alfaomega – Ra-Ma, 2007</li> <li>Informe COSO (Committee of Sponsoring Organizations of the Treadway Commission)<br/><a href="http://www.erm.coso.org/Coso/coserm.nsf/frmWebCOSOExecSum?ReadForm">http://www.erm.coso.org/Coso/coserm.nsf/frmWebCOSOExecSum?ReadForm</a>.</li> <li>COBIT (Control Objectives for Information and related Technology).<br/><a href="http://www.isaca.org">www.isaca.org</a> SAC (Systems Auditability and Control Report)</li> </ul> |



*Nora B. Okulik*  
Dra. Nora B. Okulik  
Directora  
Dpto. de Cs. Básicas y Aplicadas