



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

PROGRAMA ANALÍTICO DE ASIGNATURA

DEPARTAMENTO: Ingeniería en Sistemas de Información

CARRERA: Analista Desarrollador Universitario de Sistemas de Información

NOMBRE DE LA ACTIVIDAD CURRICULAR: Seminario Integrador

Año Académico: 2026

Área: Sistemas de Información

Bloque: Tecnologías Aplicadas

Nivel: 3º

Tipo: Obligatoria

Modalidad: Cuatrimestral

Cargas horarias totales:

<i>Horas reloj</i>	<i>Horas cátedra</i>	<i>Horas cátedra semanales</i>
72	96	6

FUNDAMENTACIÓN

El Seminario Integrador se presenta como una instancia curricular de culminación de la carrera de Analista Desarrollador Universitario de Sistemas de Información, que consiste en un espacio de producción, reflexión y articulación. Su propósito se basa en relacionar e integrar los saberes adquiridos a lo largo del trayecto formativo, mediante el desarrollo de un proyecto que constituya una solución tecnológica integral a una problemática concreta. Dicho proyecto, deberá reunir las condiciones necesarias para su transferencia a un entorno real y factible, favoreciendo así la apropiación significativa de los aprendizajes por parte de los estudiantes. Durante la cursada, se trabajará con la aplicación de metodologías, herramientas y tecnologías utilizadas en los distintos niveles de la carrera, con el fin de fortalecer las competencias vinculadas al análisis, diseño, desarrollo, documentación y validación de sistemas de software en

contextos reales. Asimismo, se establece como requisito excluyente, la incorporación del estudio y aplicación de prácticas de peritaje informático, con el objetivo de fomentar una perspectiva profesional crítica respecto de la autenticidad, integridad y trazabilidad de los sistemas desarrollados. De esta forma, el Seminario Integrador contribuye a la formación de un perfil profesional capaz de articular las dimensiones técnicas, éticas y metodológicas de la ingeniería en sistemas, integrando teoría y práctica en un proyecto final que sintetice las competencias adquiridas a lo largo de la carrera.

COMPETENCIAS DE EGRESO ESPECÍFICAS A LAS QUE CONTRIBUYE:

Competencia	Competencias de Actividades reservadas			Competencias de Alcances
	Baja	Media	Alta	
CE1.1: Especificar, proyectar y desarrollar sistemas de información para concebir soluciones tecnológicas que permitan resolver situaciones en las organizaciones mediante el empleo de metodologías de sistemas y tecnologías asociadas a los sistemas de información.		X		(AR1) (AL1), (AL2) (AL3) (AL4) (AL7)
CE1.2: Especificar, proyectar y desarrollar sistemas de comunicación de datos, evaluando posibles soluciones tecnológicas disponibles para dar soporte a los sistemas de información en lo referido al procesamiento y comunicación de datos.		X		(AR1) (AL1) (AL3) (AL4) (AL5) (AL7)
CE1.3: Especificar, proyectar y desarrollar software para la elaboración de soluciones informáticas con el propósito de resolver problemas estratégicos y operativos, así como de servicios y de negocios, en el marco de una actividad económica que sea social y ambientalmente sustentable.		X		(AR1) (AL1) (AL3) (AL4) (AL7) (AL9)
CE2.1: Proyectar y dirigir lo referido a seguridad informática para seleccionar y aplicar técnicas, herramientas, métodos y normas, garantizando la seguridad y privacidad de la información procesada y generada por los sistemas de información			X	(AR2) (AL1) (AL3)
CE3.1: Establecer métricas y normas de calidad de software para medir, evaluar, controlar y monitorear el rendimiento, impulsando mejoras de acuerdo a técnicas y normas vigentes definidas por los organismos de estandarización.		X		(AR3) (AL3)
CE4.1: Certificar el funcionamiento, condición de uso o estado de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software para asegurar la			X	(AR4) (AL10)

generación de los resultados deseados en función de restricciones de tiempo y recursos establecidos.				
CE5.1: Dirigir y controlar la implementación, operación y mantenimiento de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software, a los fines de alcanzar los objetivos fijados por la organización.		X		(AR5) (AL2), (AL5) (AL6) (AL7)

COMPETENCIAS DE EGRESO GENÉRICAS A LAS QUE CONTRIBUYE:

Competencia	Baja	Media	Alta
CG1: Identificar, formular y resolver problemas de ingeniería en Sistemas de Información.			X
CG2: Concebir, diseñar y desarrollar proyectos de ingeniería en Sistemas de Información.		X	
CG3: Gestionar, planificar, ejecutar y controlar proyectos de ingeniería en Sistemas de Información.		X	
CG4: Utilizar de manera efectiva las técnicas y herramientas de aplicación en la ingeniería en Sistemas de Información.		X	
CG5: Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	X		
CG6: Desempeñarse de manera efectiva en equipos de trabajo.		X	
CG7: Comunicarse con efectividad.		X	
CG8: Actuar con ética, responsabilidad profesional y compromiso social, considerando el impacto económico, social y ambiental de su actividad en el contexto local y global.			X
CG9: Aprender en forma continua y autónoma.			X
CG10: Actuar con espíritu emprendedor.		X	

OBJETIVOS

- Reconocer conocimientos teóricos y prácticos adquiridos a lo largo de la carrera para implementarlo en un proyecto de desarrollo de *software*.
- Aplicar metodologías y herramientas de ingeniería de software para la implementación de soluciones integradas.
- Desarrollar documentación técnica completa para respaldar el ciclo de vida del sistema desarrollado.
- Implementar distintos planes de prueba para garantizar la calidad y confiabilidad del *software*.
- Identificar nociones básicas de peritaje informático para analizar la validez, trazabilidad y seguridad de la información.

CONTENIDOS ANALÍTICOS

Contenidos mínimos:

- Integración de herramientas y metodologías de desarrollo.
- Tecnologías aplicadas a la integración de software.
- Elaboración de documentación técnica.
- Peritaje informático.

Contenidos analíticos:

Unidad 1: Metodologías y enfoques de desarrollo

Revisión y comparación de metodologías ágiles, tradicionales e híbridas aplicadas al desarrollo de software. Adaptación de Scrum, Kanban y XP a proyectos universitarios e interdisciplinarios. Roles, artefactos y ceremonias en equipos de trabajo colaborativos. Herramientas de gestión de proyectos y seguimiento de entregables: Jira, Trello, GitHub Projects. Indicadores y métricas de avance. Selección y ajuste del marco metodológico según el contexto del proyecto integrador. Planificación de iteraciones y administración de recursos. Comunicación efectiva y liderazgo técnico en equipos multidisciplinarios.

Unidad 2: Integración de procesos de análisis, diseño y validación

Vinculación entre los procesos de análisis de requerimientos, diseño arquitectónico, desarrollo y validación funcional. Enfoque integral del ciclo de vida del software y su relación con la ingeniería de requerimientos. Estrategias de testing temprano y revisión por pares. Aplicación de prácticas de DevOps, integración y entrega continua (CI/CD). Coordinación interdisciplinaria y gestión de calidad en el desarrollo de software. Elaboración del plan de trabajo del proyecto integrador considerando planificación, control y evaluación de resultados.

Unidad 3: Arquitecturas y tecnologías de integración

Estudio de los principales estilos arquitectónicos utilizados en sistemas modernos: arquitecturas multicapa, cliente-servidor, orientadas a servicios y microservicios. Introducción a las APIs REST, GraphQL y gRPC. Frameworks y entornos de desarrollo orientados a la interoperabilidad. Contenedores y virtualización con Docker y Kubernetes. Despliegue de aplicaciones y servicios en la nube (AWS, Azure, GCP). Estrategias de integración de componentes, interoperabilidad entre sistemas y gestión de dependencias. Seguridad, autenticación y escalabilidad en arquitecturas distribuidas.

Unidad 4: Automatización, control de versiones e integración continua

Control de versiones con Git y flujos de trabajo avanzados. Estrategias de branching, pull requests y revisión colaborativa. Automatización de pruebas y despliegues mediante Jenkins, GitHub Actions y GitLab CI. Configuración y administración de entornos de desarrollo, prueba y producción. Monitoreo, logging y registro de eventos en aplicaciones integradas. Aplicación

práctica de mecanismos de trazabilidad, mantenimiento y control de calidad en el ciclo de vida del software. Integración de bases de datos, front-end y back-end en entornos colaborativos.

Unidad 5: Documentación del ciclo de vida del software

Elaboración de documentación integral en proyectos de desarrollo de software. Especificación de requerimientos y casos de uso. Modelado del sistema mediante diagramas UML y C4. Documentación técnica de módulos, interfaces y APIs. Manuales de instalación, operación y mantenimiento. Redacción de guías de usuario bajo criterios de accesibilidad y claridad. Uso de herramientas colaborativas para documentación técnica (Confluence, Overleaf, Markdown). Revisión cruzada de documentos y estándares de presentación profesional.

Unidad 6: Estándares, trazabilidad y gestión documental

Estudio de los principales estándares de documentación técnica: IEEE 1016, ISO/IEC/IEEE 42010 y CMMI. Principios de trazabilidad entre requerimientos, diseño, código y pruebas. Control de versiones documentales, auditorías y revisiones técnicas. Registro y gestión de decisiones técnicas (ADRs). Implementación de repositorios documentales y documentación viva. Estrategias de aseguramiento de la calidad de la documentación. Evaluación de coherencia y completitud del ciclo de vida documental.

Unidad 7: Fundamentos del peritaje informático

Introducción al peritaje informático aplicado al desarrollo de software. Conceptos de evidencia digital, cadena de custodia y preservación de la información. Tipos de evidencia en sistemas de información: registros, logs, archivos y bases de datos. Aspectos legales y éticos del peritaje informático. Marco normativo y rol del ingeniero perito en el contexto judicial. Evaluación de la integridad, autenticidad y trazabilidad de sistemas de software. Análisis de casos reales de peritaje sobre aplicaciones informáticas y entornos empresariales.

Unidad 8: Herramientas y prácticas forenses en entornos de desarrollo

Aplicación práctica de herramientas forenses en el análisis de sistemas de software. Uso de Autopsy, FTK, Wireshark, HashCalc y utilitarios especializados para la verificación de integridad, hash y firmas digitales. Análisis de metadatos, commits y logs de código fuente. Técnicas de recolección, preservación y análisis de evidencias digitales. Elaboración de informes técnicos y periciales. Validación de autenticidad e integridad en los proyectos desarrollados por los estudiantes. Simulación de escenarios periciales sobre sistemas reales.

DISTRIBUCIÓN DE CARGA HORARIA ENTRE ACTIVIDADES TEÓRICAS Y PRÁCTICAS

Modalidad organizativa de las clases	Horas Reloj totales presenciales	Horas reloj virtuales totales	Horas totales
Teórica	24	0	24
Formación práctica	48	0	48

Tipo de prácticas	Horas Reloj totales presenciales	Horas reloj totales virtuales	Lugar donde se desarrolla la práctica
Formación experimental	0	0	-
Problemas abiertos de Ingeniería (ABP)	20	0	Laboratorio de Sistemas
Proyecto y diseño	28	0	Laboratorio de Sistemas
Otras:	0	0	-
Práctica supervisada	0	0	-
Total de horas	48	0	48

ESTRATEGIAS DE ENSEÑANZA Y ACTIVIDADES DE APRENDIZAJE

En el marco de la asignatura, se implementan diversas estrategias pedagógicas orientadas a consolidar los saberes adquiridos a lo largo de la carrera y a favorecer su aplicación en el desarrollo del proyecto final. La propuesta didáctica articula instancias teóricas expositivas y prácticas que promueven un aprendizaje activo, colaborativo y reflexivo, permitiendo al estudiante integrar conocimientos previos y transferirlos a contextos reales de la práctica profesional.

Las actividades teóricas se desarrollan mediante clases expositivas con actividades de intercambio grupal, análisis de casos, debates guiados y el uso de recursos digitales y audiovisuales como complemento. Estas instancias tienen como propósito reforzar los conceptos vinculados con metodologías, tecnologías de integración, documentación técnica y fundamentos del peritaje informático, fomentando la participación colaborativa, crítica y el abordaje interdisciplinario.

En el ámbito práctico, el estudiante aplicará los contenidos en la elaboración de un proyecto integrador, que abarca las etapas de análisis, diseño, implementación, documentación, prueba y defensa de un sistema de software. El proyecto se desarrollará en equipos, siguiendo metodologías ágiles e incorporando herramientas de integración continua y control de versiones. Se prevén instancias de revisión entre pares, presentaciones parciales de avance y una exposición oral final del trabajo completo.

Como actividades complementarias, se promoverá la búsqueda, lectura y análisis de documentación técnica y normativa actualizada, elaboración de informes parciales y la participación en simulaciones de procesos periciales aplicados a los sistemas desarrollados. Estas estrategias tienen como finalidad fortalecer la autonomía, la capacidad de comunicación técnica y la responsabilidad profesional del futuro egresado.

MODALIDAD DE EVALUACIÓN

Se propone un plan de evaluación continua, formativa e integral contemplando el desarrollo de un proyecto que constituya una solución tecnológica integral a una problemática concreta. Dicho proyecto, deberá reunir las condiciones necesarias para su transferencia a un entorno real y factible, favoreciendo así la apropiación significativa de los aprendizajes por parte de los estudiantes. Dentro de los criterios de evaluación, se considerará la participación activa, la calidad de los entregables y el avance del proyecto integrador. Se evaluarán conocimientos teóricos, competencias prácticas, trabajo en equipo, documentación técnica y la correspondiente defensa del proyecto final. Al culminar cada estudiante deberá exponer grupalmente y realizar una defensa individual de la arquitectura propuesta. Se hará una ronda de preguntas y se espera que los estudiantes puedan desenvolverse como futuros profesionales defendiendo las decisiones tomadas (situación que se les presenta frecuentemente en el campo laboral).

Requisitos de regularidad

Condiciones para Aprobación Directa

Un alumno podrá aprobar la asignatura en forma directa (es decir, aprobar sin necesidad de rendir examen final) si cumple con las siguientes condiciones:

- Cumplir con los prerequisites de inscripción dispuestos por la Facultad.
- El alumno debe estar presente en clase, según las normas de la Universidad. En caso de presentismo menor al 75 % perderá toda posibilidad de aprobar la cursada en forma directa.
- Aprobar el Trabajo Práctico Integrador con calificación de 8 (ocho) o superior.
- Aprobar su presentación final grupal (que incluye una defensa oral individual) y la entrega de documentación completa.

Condiciones para Aprobación No Directa

- Un alumno podrá aprobar la asignatura en forma directa (es decir, aprobar sin necesidad de rendir examen final) si cumple con las siguientes condiciones:
- Cumplir con los prerequisites de inscripción dispuestos por la Facultad.
- El alumno debe estar presente en clase, según las normas de la Universidad. En caso de presentismo menor al 75 % perderá toda posibilidad de aprobar la cursada en forma directa.
- Aprobar el Trabajo Práctico Integrador con calificación de 6 (seis) o superior.

ARTICULACIÓN HORIZONTAL Y VERTICAL CON OTRAS MATERIAS

Dado que el *Seminario Integrador* constituye la instancia final de la carrera, su articulación horizontal y vertical resulta esencial para consolidar la coherencia del trayecto formativo y

garantizar que los estudiantes logren integrar los conocimientos, habilidades y actitudes adquiridas a lo largo de los tres niveles de formación.

En relación con la **articulación horizontal**, la asignatura se vincula con **Diseño de Sistemas de Información, Desarrollo de Software, Bases de Datos y Comunicación de Datos**, integrando en un mismo proyecto los aspectos conceptuales, metodológicos y tecnológicos abordados en esas materias. Asimismo, mantiene una relación directa con **Economía**, en tanto el proyecto integrador requiere considerar la viabilidad técnica y económica de las soluciones propuestas, y comprender el contexto organizacional en el que se insertan.

Respecto de la **articulación vertical**, el Seminario se apoya en las bases formativas desarrolladas en **Análisis de Sistemas de Información, Arquitectura de Computadoras, Paradigmas de Programación y Sistemas Operativos**, las cuales brindan los fundamentos teóricos y técnicos que sustentan la construcción de sistemas de software robustos y eficientes.

En conjunto, esta articulación permite que el Seminario Integrador funcione como un espacio de convergencia de saberes, donde el estudiante no sólo aplica conocimientos técnicos, sino que también desarrolla competencias profesionales, éticas y comunicacionales, propias del perfil de egreso de la UTN.

CRONOGRAMA ESTIMADO DE CLASES

Clase	Tema	Modalidad de dictado
1	Presentación de la Asignatura Unidad 1: Metodologías y enfoques de desarrollo	Presencial
2	Unidad 2: Integración de procesos de análisis, diseño y validación	Presencial
3	Trabajo Integrador: ETAPA 1	Presencial
4	Unidad 3: Arquitecturas y tecnologías de integración	Presencial
5	Unidad 4: Automatización, control de versiones e integración continua	Presencial
6	Trabajo Integrador: ETAPA 2	Presencial
7	Unidad 5: Documentación del ciclo de vida del software	Presencial
8	Unidad 6: Estándares, trazabilidad y gestión documental	Presencial

9	Trabajo Integrador: ETAPA 3	Presencial
10	Unidad 7: Fundamentos del peritaje informático	Presencial
11	Unidad 8: Herramientas y prácticas forenses en entornos de desarrollo	Presencial
12	Trabajo Integrador: ETAPA 4	Presencial
13	Trabajo Integrador. Revisión.	Presencial
14	Trabajo Integrador: Presentación y coloquios	Presencial
15	Trabajo Integrador: Presentación y coloquios	Presencial
16	Cierre de la asignatura	Presencial

BIBLIOGRAFÍA OBLIGATORIA

- Pressman, R. (2020). Ingeniería del Software: Un enfoque práctico. McGraw-Hill.
- Sommerville, I. (2020). Software Engineering. Pearson.
- Casey, E., & Ferraro, M. (2019). Digital Evidence and Computer Crime. Academic Press.
- IEEE Std 1016-2017. IEEE Standard for Information Technology—System Design—Software Design Descriptions.

BIBLIOGRAFÍA COMPLEMENTARIA

- Beizer, B. (2009). Software Testing Techniques. Dreamtech Press.
- Carrier, B. (2016). File System Forensic Analysis. Addison-Wesley.
- Fowler, M. (2018). Patterns of Enterprise Application Architecture. Addison-Wesley.
- Bass, L., Clements, P., & Kazman, R. (2021). Software Architecture in Practice. Addison-Wesley.