

PLANIFICACIÓN 2026
INGENIERÍA DE SOFTWARE I
1er Año – Técnico Superior en
Desarrollo de Software



DOCENTE REEMPLAZANTE:

Ileana Ulagnero

AÑO 2026

CAMPO DE FORMACIÓN	UBICACIÓN EN EL DISEÑO	RÉGIMEN DE CURSADO	HS. CÁTEDRAS SEMANALES	HS. CÁTEDRAS ANUALES	FORMATO PROPUESTO
Ingeniería de Software I	Primer Año	Anual	4 horas	128 horas	Asignatura

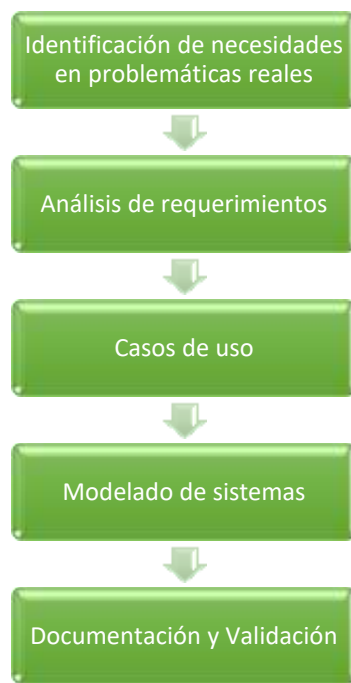
→ FUNDAMENTACIÓN:

La asignatura Ingeniería de Software I es un espacio curricular de *análisis, modelado y diseño de sistemas* que pretende desarrollar las capacidades de pensamiento sistémico y comunicación técnica.

El alumno debe analizar distintos escenarios mediante la comprensión de problemas del mundo laboral y empresarial, identificar actores y necesidades y transformarlos en modelos conceptuales de software. Esto implica trabajar con metodologías, procesos y estándares para documentar correctamente desde la lógica de la ingeniería secuenciada mediante las fases: problema, análisis, diseño y validación.

Se procura introducir al estudiante en el trabajo de Ingeniería mediante proyectos que utilicen metodología y herramientas específicas del área para que reconozcan, documenten y validen las necesidades que justifican sus resoluciones mediante modelos, diagramas y especificaciones.

→ MODELO CONCEPTUAL



→ PROPÓSITO GENERAL

- * Introducir al estudiante en el trabajo de ingeniería a través de proyectos, utilizando métodos, herramientas y documentación para identificar necesidades y diseñar soluciones.
- * Adquirir las bases metodológicas y conceptuales para diseñar proyectos de software de calidad.

→ OBJETIVOS

- * Producir diagramas de clase a partir de problemas correspondientes a diversos dominios de aplicación identificando sus entidades y atributos.
- * Analizar y evaluar críticamente los diagramas de clases generados, considerando criterios de corrección y promoviendo la discusión fundamentada sobre distintas alternativas.
- * Representar situaciones y sistemas específicos mediante el uso de diagramas UML y otras técnicas de modelado, seleccionando la notación más apropiada según el contexto y los objetivos de comunicación.
- * Examinar las representaciones obtenidas, identificando sus fortalezas, debilidades y oportunidades de mejora, con el fin de refinar los modelos y aumentar su calidad y utilidad.
- * Modelizar y especificar casos de uso a partir de descripciones de situaciones realistas, asegurando la correcta identificación y documentación de actores y escenarios.
- * Documentar requerimientos y especificaciones aplicando criterios de calidad para detectar mejoras y proponer su reelaboración mediante mejora continua y la reingeniería.

→ CONTENIDOS

UNIDAD 1: INTRODUCCIÓN Y FUNDAMENTOS A LA INGENIERÍA DE SOFTWARE

Eje curricular: introducir al marco teórico general de la disciplina y sus enfoques organizativos de desarrollo para relacionar problemas reales y con metodologías formales.

- Ingeniería del Software: conceptos generales.
- Definición y contextualización.
- Objetivos de la disciplina y ubicación en la informática.
- Relación con otras disciplinas.
- Métodos y herramientas.
- Ciclo de vida del software: cascada, espiral, incremental, iterativo.
- Modelos de desarrollo y metodologías ágiles.
- Comparación y selección de procesos.
- Paradigmas. Paradigma funcional, lógico, imperativo. Características fundamentales

UNIDAD 2: INGENIERÍA Y ANÁLISIS DE REQUERIMIENTOS

Eje curricular: identificar, documentar y validar la funcionalidad del sistema: aquello que el sistema hace.

- Análisis de requerimientos.
- Análisis y especificación. Casos de uso (CU) y elaboración de especificaciones.
- Proceso y partes interesadas. Stakeholders. Técnicas de elicitación.
- Concepto de requisito funcional y no funcional.
- Requerimientos funcionales, prioridades y criterios de realización.
- Requerimientos no funcionales y su influencia en el diseño del software.
- Validación y estándares de documentación de requerimientos.
- Revisión. Detección de errores. Mejora y reingeniería.

UNIDAD 3: MODELADO DE SISTEMAS

Eje curricular: los modelos como puente conceptual hacia el diseño y la comprensión profunda de los objetos mediante la diagramación estructural y de datos.

- Herramientas de modelización. UML. Análisis orientado a objetos.
- Modelado estructural y de datos.
- Diseño centrado en el usuario. Escenarios, historias y casos de uso.
- Diagramas de clase.
- Representación del comportamiento: diagramas de secuencia, máquinas de estado, diagramas de actividad. Redes de Petri. Pre y post condiciones.
- Análisis de datos: datos de referencia y de operaciones. Datos de nivel de recursos y de volumen de actividad. Modelo Entidad-Relación. Principales formas normales. Diccionario de datos.
- Ambientes gráficos para edición. Herramientas integradas disponibles para la edición en distintos lenguajes y ambientes.

➔ **ESTRATEGIA METODOLÓGICA:**

Las clases se desarrollarán semanalmente con un abordaje de los contenidos teórico-práctico mediante la utilización de presentaciones multimedia.

Se utilizará también la plataforma virtual classroom como apoyatura digital del aula y como repositorio del material bibliográfico correspondiente al espacio curricular.

Del mismo modo, será el medio oficial utilizado para la entrega de los trabajos prácticos que sean requeridos; cuya realización y entrega en tiempo y forma, tendrá el carácter de necesario y obligatorio para mantener la condición de regularidad.

→ EVALUACIÓN

* ALUMNO REGULAR:

- ✓ Cursado presencial con asistencia del 75%
- ✓ Presentación y aprobación de los trabajos prácticos requeridos
- ✓ Realización y aprobación de las instancias de exámenes parciales
- ✓ Realización y aprobación del examen final

APROBACIÓN DIRECTA: el alumno en condición regular puede promocionar directamente la asignatura sin la instancia de evaluación final en el caso de:

- Cumplimentar la asistencia
- Cumplimentar la aprobación de la totalidad de trabajos prácticos solicitados
- Aprobar instancias parciales con nota igual o superior a 8
- Aprobar instancia integradora: coloquio

* ALUMNO LIBRE:

- ✓ Realización de por lo menos dos consultas previas y obligatorias durante el ciclo lectivo
- ✓ Realización de examen final

EXAMEN FINAL:

- Sobre el contenido: tiene carácter teórico-práctico en concordancia con los contenidos del espacio curricular
- Sobre la forma: la forma del examen será a determinar, pudiendo ser oral o escrita.

→ CRITERIOS PARA EVALUAR

- ✓ Comprensión de los contenidos esenciales de la asignatura.
- ✓ Pertinencia de las respuestas a lo solicitado en las consignas en las instancias evaluativas.
- ✓ Claridad en la exposición de sus respuestas y coherencia en la argumentación.
- ✓ Utilización de los conceptos teóricos específicos y referencia a los autores abordados.
- ✓ Transferencia teórico conceptual a la aplicación práctica

→ BIBLIOGRAFIA

* Apuntes de cátedra

* Roger S. Pressman: "Ingeniería de Software. Un enfoque práctico", Editorial Mac Graw Hill, 7ma. Edición.

* Sommerville Ian: "Ingeniería de Software", Editorial Pearson, 9na. Edición.