

4121 - FUNDAMENTOS DEL DESARROLLO DE SOFTWARE

I - Datos de identificación de la asignatura

Carrera:	Licenciatura en Análisis de Sistemas		
Código:	4121	Plan:	2024
Denominación:	Fundamentos del desarrollo de software		
Área:	Fundamentos de la informática		
Año:	Segundo		
Horas con acompañamiento docente (HTD), semanal			2
Horas de Trabajo Independiente del estudiante (HTI), semanal			2
Horas semanales (HS)			4
Cantidad de sesiones			32
Total Horas de Trabajo con el docente (THTD)			64
HTD teóricas	64	HTD prácticas	0
Total de Horas de Trabajo Independiente del estudiante (THTI)			64
Total Horas Académicas (THA)			128
Crédito académico (CA)			5,1
Pre-requisito:	Fundamentos de algorítmica y programación		

II - Fundamentación

La asignatura tiene como objetivo principal introducir a los estudiantes en las actividades técnicas y de ingeniería que se llevan a cabo en el ciclo de vida de un producto de software.

En esta asignatura, se abordan los problemas, métodos y tecnologías asociadas con la Ingeniería del Software, centrándose específicamente en el desarrollo de proyectos de software y en las áreas de gestión y control de calidad que justifican la importancia de esta asignatura.

El enfoque teórico-práctico de la asignatura permite a los estudiantes adquirir una comprensión sólida de los fundamentos del desarrollo de software y aplicar los conocimientos adquiridos en escenarios prácticos. A través de actividades prácticas, los estudiantes tendrán la oportunidad de familiarizarse con las herramientas, técnicas y metodologías utilizadas en la industria para el desarrollo efectivo de proyectos de software.

Además, la asignatura busca fomentar en los estudiantes habilidades relacionadas con la gestión y control de calidad en el desarrollo de software, destacando la importancia de la planificación, el seguimiento y la evaluación de los proyectos para garantizar la entrega de soluciones de software de alta calidad.

III - Competencias a desarrollar

Competencias genéricas

1. Conocer y saber aplicar técnicas y herramientas actualizadas en sus áreas de competencia.

Competencias específicas

1. Identificar y analizar con criterio técnico problemas informáticos y de diseño, aplicando enfoques y técnicas apropiadas para proponer soluciones mediante sistemas de información, promoviendo mejoras en los procesos organizacionales a través del uso de herramientas, metodologías y enfoques innovadores.
2. Comprender integralmente el concepto de sistemas, sus componentes e interrelaciones, para explicar las etapas del proceso de desarrollo de software y aplicar principios, buenas prácticas y criterios técnicos, organizativos y de gestión en contextos reales.
3. Identificar y describir de forma estructurada las fases del ciclo de vida del desarrollo de software (análisis de requerimientos, diseño, implementación, pruebas, despliegue), reconociendo la importancia de cada una para garantizar un desarrollo exitoso y alineado a los objetivos del proyecto.
4. Reconocer y diferenciar con fundamento crítico los principales modelos de ciclo de vida del software (ágil, cascada, en V, iterativo, espiral), analizando sus características, ventajas, limitaciones y pertinencia según el contexto del proyecto.

IV - Cuerpo de conocimientos

Unidad 1: Resolución de problemas básicos de computación y diseño utilizando metodologías apropiadas para el desarrollo de sistemas de información, herramientas de software y métodos innovadores para la mejora de procesos.

Contenidos:

- Métodos y herramientas para el uso de la información.
- Decisión de comprar versus construir un software.
- Concepto y elementos de un sistema de información.
- Posibilidades de externalización y modelos para sistemas de información.
- Herramientas de software
- Métodos computacionales.
- Metodologías de desarrollo de software
- Ciclo de vida del desarrollo de sistemas
- Técnicas de innovación y diseño.

Unidad 2: Concepto de sistemas y cómo se desarrollan

Contenidos:

- Que es un sistema
- Propósito del análisis de sistemas (lo que debe hacer el sistema)
- Propósito del diseño (cómo lograr el objetivo del sistema)
- Necesidad y valor de un enfoque formalizado paso a paso para el análisis, diseño e implementación de sistemas informáticos de información.

Unidad 3: Ciclo de vida del desarrollo de software

Contenidos:

- Definición y atributos del proyecto
- Solicitud de sistemas
- Especificaciones técnicas y de requisitos de software
- Viabilidad técnica
- Análisis coste-beneficio
- Viabilidad Económica
- Documentación de requisitos comerciales
- Viabilidad organizativa

- Plan de gestión de proyectos
- Matriz de Asignación de Responsabilidades
- Requisitos funcionales y no funcionales
- Técnicas de elicitación de requisitos
- Análisis de raíz de la causa
- Casos de uso
- Plan de migración de sistemas
- Programa de mantenimiento de sistemas
- Diseño de infraestructura de TI
- Diagramas de modelado de procesos
- Diagramas de flujo de datos
- Plan de pruebas de rendimiento y seguridad.
- Plan de empleo
- Cierre de un proyecto
- Programa de mantenimiento de sistemas
- Documento de solicitud de cambio del sistema

Unidad 4: Modelos de ciclo de vida del desarrollo de software

Contenidos:

- Metodología ágil
- Cascada
- Modelo de construcción por prototipos
- Otros modelos relevantes de ciclo de vida de desarrollo de sistemas

V- Estrategias didácticas a ser implementadas en el proceso de enseñanza aprendizaje. (abarcando actividades de formación e investigación)

El curso consta de clases teóricas, explicaciones de práctica y clases prácticas.

Las explicaciones de práctica apuntan a brindar las herramientas necesarias para la realización de los trabajos prácticos.

Las actividades prácticas comprenden la realización de ejercicios de modelización de requerimientos aplicando diferentes herramientas explicadas en la teoría.

En la cátedra se pone énfasis en el proceso de identificación de problemas del mundo real, especificación de los mismos como problemas resolubles desde la informática y en el desarrollo de soluciones verificables para los mismos utilizando las técnicas de especificación de requerimientos establecidas en el programa de la materia.

Los alumnos deberán desarrollar los modelos software correspondientes a un proyecto. Se realizará a modo de portafolio en el que a medida que se vayan estudiando los distintos artefactos software en clases teóricas, se vayan desarrollando e incorporando en el portafolio del proyecto.

Dicho proyecto se realizará en grupo, y para cada fase se definirán una serie de tareas individuales a realizar (la mayoría) y otras a realizar en grupo.

VI - Estrategias de evaluación.

La evaluación será formativa y procesual, se realizará a través de pruebas (exámenes) que podrán ser escritas, orales o de ejecución que a su vez podrá ser mediante trabajos individuales o grupales. La materia consta de dos pruebas parciales, con un recuperatorio y tres oportunidades para la prueba final.

En estos parciales, así como en el examen final, se evaluarán las competencias alcanzadas a través de actividades de contenido teórico y práctico que permitan dar cuenta

del avance conceptual en los temas que se han desarrollado, se incorporan preguntas específicas tipo sobre “donde cree Ud. que es aplicable este conocimiento/método” y se refleja en la corrección de las pruebas del alumno.

En algunos temas se trabaja también con ejercitaciones de aplicación en clase, que requieren de un ejercicio de integración de conceptos y que complementan la evaluación a través de los parciales.

Los ejercicios de la práctica son evaluados teniendo en cuenta el contenido técnico, pero también la estructura, organización, sintaxis, claridad conceptual.

Además, los alumnos deben realizar una actividad grupal de elicitación de requerimientos que es evaluada de manera oral por el ayudante a cargo.

Todas las evaluaciones orales realizadas se reflejan en planillas escritas que conforman documentación de evaluación del trabajo.

Para la obtención de calificaciones parciales y finales se tendrá en cuenta el Reglamento Académico de la universidad.

VII - Actividades de extensión y de responsabilidad social universitaria.

Rige de acuerdo al reglamento de la Universidad y el reglamento interno de la facultad.

VIII - Fuentes bibliográficas

Básica

- Ingeniería de Software. 9na Edición. Ian Sommerville. Pearson. 2011.
- Software Engineering: Theory and Practice. 4th Edition. Shari Pfleeger. Prentice Hall. 2009. (Edición en castellano: Ingeniería de Software. Teoría y Práctica. Shari Pfleeger. Pearson Education. 2002)
- Ingeniería de Software. Un enfoque práctico. 7ma Edición. Roger Pressman. McGraw-Hill. 2010.
- Sistemas de Información Administrativa. Murdick R. Prentice Hall. 1988
- Systems Analysis and Design, 9/E. Kendall & Kendall. Pearson. 2013. (Edición en castellano: Análisis y diseño de sistemas. 8va Edición. Kendall & Kendall. Pearson. 2011)

Complementaria

- Systems Analysis and Design, 9/E. Kendall & Kendall. Pearson. 2013. (Edición en castellano: Análisis y diseño de sistemas. 8va Edición. Kendall & Kendall. Pearson. 2011)
- Análisis de Sistemas. Diseño y Métodos. 7ma Edición. Whitten y Bentley. 2008.