

<b>Carrera INGENIERIA EN INFORMATICA</b>		
<b>Asignatura</b> 3648 - Diseño de Software		
<b>Trayecto</b> Desarrollo de Software		
<b>Año académico</b> 2023		
<b>Responsable / Jefe de cátedra</b> Magister Ing. Osvaldo Sposito		
<b>Carga horaria semanal</b> 4hs	<b>Carga horaria total</b> 64hs	<b>Créditos</b> ----
<b>Modalidad:</b> Presencial		
<b>Correlativas anteriores</b> : PRINCIPIOS DE DISEÑO DE SISTEMAS - BASES DE DATOS		<b>Correlativas posteriores</b> ARQUITECTURA DE SISTEMAS SOFTWARE
<b>Conocimientos necesarios</b> -----		

### Descripción de la asignatura

El Diseño de Software, visto como un proceso, es la actividad del ciclo del desarrollo de software en la que se define la estructura interna que tendrá el mismo, la que luego servirá de base para su construcción. El principal aporte de esta asignatura para la formación profesional de los estudiantes tiene que ver con brindarle los saberes básicos que les permitan aplicar metodologías sistemáticas, diversas técnicas, y buenas prácticas del diseño, a fin de describir y especificar en detalle todos los aspectos del software a construir. Todo lo anterior sin descuidar la evaluación continua de la calidad de los artefactos producidos mediante la aplicación de principios fundamentales de diseño.

### Metodología de enseñanza

La actividad curricular de esta asignatura consta de clases teórico-prácticas. La modalidad de cursada es presencial. El curso está dirigido a que los estudiantes incorporen capacidades, habilidades, destrezas y actitudes a partir de la resolución de problemas y la aplicación práctica de los conceptos teóricos.

Teoría y práctica no deben verse en forma separada, sino que se contempla un esquema en el que se incluye:

- el abordaje de los aspectos teóricos conceptuales sobre cada tema planteado en el programa, incluyendo el desarrollo de ejemplos prácticos en clase y aclaración de dudas,
- la presentación de las herramientas CASE que permiten operar sobre la realidad y,
- la aplicación de los conocimientos conceptuales y las herramientas a casos particulares.

Se utiliza la plataforma MIEl como repositorio de material de trabajo para los alumnos y como medio de comunicación entre los alumnos y el equipo docente.

Se proporciona una guía con trabajos prácticos que deben ser resueltos por los alumnos para asegurar la comprensión de los temas teóricos asociados.

Desarrollo de Problemas: los problemas de clase se desarrollan, en lo posible, inmediatamente después que se dicte la teoría correspondiente, con algunos ejemplos o problemas tipo y preguntas conceptuales, que el docente efectuará a la clase dando, cuando sea necesario, los fundamentos de las respuestas, para contribuir a una mejor interpretación

del tema. Se insta a los alumnos a leer el material bibliográfico y resolver los problemas propuestos, dándole la posibilidad de consultar con el equipo docente cuando sea necesario. Resolución en clase: los docentes indican a los alumnos los problemas y ejercicios de la guía a resolver en cada clase, se les asigna un tiempo para plantearlos y resolverlos. Los alumnos pueden recurrir al docente o al auxiliar para su orientación y su trabajo es individual o grupal, según lo que disponga el docente. Finalizado el tiempo, se realiza una puesta en común de la resolución, destacando los conceptos en que están basados los planteos correspondientes y los procedimientos implicados.

Se aparta momentos específicos de la clase para seguimientos generales y grupales de los estudiantes. Esto permite, por una parte, brindar apoyo a los alumnos y, por otra, establecer puntos de control, donde se llevan a cabo evaluaciones de los distintos artefactos que los alumnos obtienen a partir de sus resoluciones.

#### **Objetivos de aprendizaje**

Los alumnos que hayan aprobado el curso habrán adquirido los conceptos fundamentales sobre los principios del diseño y serán capaces de aplicarlos con el objetivo de llevar adelante diseños de software de calidad.

En consecuencia, habrán alcanzado los siguientes objetivos específicos:

- Emplear metodologías, procesos y herramientas que conforman las mejores prácticas de mercado.
- Incorporar la capacidad de modelizar requerimientos de sistemas de software. Diseñar y documentar arquitecturas, módulos, componentes e interfaces de sistemas software.
- Aplicar patrones de diseño.
- Entender lo importante de lograr una experiencia de usuario satisfactoria.

#### **Contenidos mínimos**

Conceptos de diseño: abstracción, refinamiento, modularidad, cohesión, acoplamiento, rediseño, independencia funcional, reutilización. Diferencias entre diseño de sistemas y diseño de software. Atributos de calidad y restricciones del diseño. Refinamiento. Especificaciones de requisitos y su relación con el diseño de software. Patrones de diseño. Antipatrones. Interfaz Hombre-Computadora. UX vs UI. Topologías de Diseño. Web, Móvil, Escritorio, Tiempo Real. Topologías emergentes.

### Competencias a desarrollar

#### Genéricas

- Desempeño en equipos de trabajo
- Comunicación efectiva
- Actuación profesional ética y responsable
- Aprendizaje continuo
- Desarrollo de una actitud profesional emprendedora
- Utilización de técnicas y herramientas de aplicación en la ingeniería de información de información /informática
- Generación de desarrollos tecnológicos y/o innovaciones tecnológicas
- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.
- Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.

#### Específicas

- Especificación, proyecto y desarrollo de sistemas de información
- Especificación, proyecto y desarrollo de sistemas de comunicación de datos
- Especificación, proyecto y desarrollo de software
- Procedimientos y certificaciones del funcionamiento, condición de uso o estado de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.
- Dirección y control de la implementación, operación y mantenimiento de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.
- Proyecto y dirección en lo referido a seguridad informática.
- Establecimiento de métricas y normas de calidad de software.

### Programa analítico

Unidad 1	<b>Diseño de software.</b> Diferencias entre el diseño de sistemas y diseño de software. Conceptos de diseño: abstracción, refinamiento, modularidad, cohesión, acoplamiento, rediseño, independencia funcional, reutilización.
Unidad 2	<b>Del diseño de sistemas al diseño de software.</b> Procesos de negocio. Roles. Solución de software. Atributos de calidad y restricciones del diseño, Procesos de Negocio y Procesos de Software, Gobierno de Procesos, Roles, Adopción del Software.
Unidad 3	<b>Los patrones y el diseño.</b> Estudio de los objetivos, estructura y aplicación práctica de los principales patrones de diseño. Patrones creacionales, estructurales y de comportamiento. Antipatrones y malas prácticas. Mejores prácticas.
Unidad 4	<b>Diseño, refinamiento y especificación.</b> Especificaciones de requisitos y su relación con el diseño de software. Topologías de

	diseño. Diseño de sistemas Web, Móvil y de Escritorio. Topologías emergentes.
Unidad 5	<b>Diseño de sistemas de tiempo real.</b> Concepto de arquitectura de tiempo real, sensores, actuadores y administrador o núcleo del sistema. Conceptos de tareas, mecanismos de sincronización y ejecución, paso de mensajes e interfases, asignación de prioridades y mecanismos de almacenamiento de eventos. Redes de Petri para Sistemas en Tiempo Real.
Unidad 6	<b>Interfaz Hombre- Computadora.</b> UX vs UI. Las interfaces del usuario. Principios de las interfases. Simplicidad de acceso. Diseño de contenidos. Diseño de interfases del usuario. Simplicidad, características, menús, iconos, tablas. Proceso de diseño de interfases. Interfases entre procesos, seguridad y control de tráfico de información. Sistemas de Comercio electrónico. Diseño centrado en el usuario. Diseño de la interacción hombre-máquina.

**Planificación de actividades** [15/16 semanas dependiendo del calendario académico universitario – UNLaM]

Semana	Clase	Actividad	Tipo	Duración estimada	Unidad/des
Semana 1	1	Presentación del equipo docente y de la materia. Lineamientos generales de la cursada y su aprobación. Armado de grupos. Test de diagnóstico. Introducción al diseño de software. Conceptos de diseño.	Teoría	4 horas	1
Semana 2	2	Del diseño de sistemas al diseño de software	Teoría	4 horas	2
Semana 3	3	Patrones de diseño: creacionales, estructurales y	Teoría y Práctica	2 horas teoría y 2 horas práctica	3

		de comportamiento. ----- Resolución de casos prácticos.			
Semana 4	4	Antipatronos de diseño y malas prácticas. Mejores prácticas. ----- Resolución de casos prácticos.	Teoría y Práctica	2 horas teoría y 2 horas práctica	3
Semana 5	5	Clase de refuerzo, práctica de patrones de diseño.	Práctica	4 horas	3
Semana 6	6	Clase de refuerzo de temas teóricos y prácticos para el primer parcial.	Práctica	4 horas	1 a 3
Semana 7	7	<b>Primer Parcial.</b> ----- Especificaciones de requisitos y su relación con el diseño de software. Diseño de sistemas Web, Móvil y de Escritorio. Diseño de otros sistemas emergentes.	Evaluación y Teoría	4 horas	4
Semana 8	8	Resolución primer parcial – Entrega de Notas. ----- Conceptos de arquitectura para	Teoría	4 horas	5

		sistemas de tiempo real.			
Semana 9	9	Redes de Petri para sistemas de tiempo real. ----- Resolución de casos prácticos.	Teoría y Práctica	2 horas teoría y 2 horas práctica	5
Semana 10	10	Clase de refuerzo, práctica de sistemas de tiempo real - Redes de Petri.	Práctica	4 horas	5
Semana 11	11	Diseño de la interfaz Hombre-Computadora. UX vs UI. Las interfaces del usuario. Principios de las interfases. ----- Aplicación a un caso práctico	Teoría y Práctica	2 horas teoría y 2 horas práctica	6
Semana 12	12	Clase de refuerzo de temas teóricos y prácticos para el segundo parcial.	Práctica	4 horas	4 a 6
Semana 13	13	<b>Segundo Parcial.</b> ----- Clase de refuerzo para el recuperatorio	Práctica	4 horas	4 a 6
Semana 14	14	Resolución segundo parcial – Entrega de Notas. ----- Clase de refuerzo para el recuperatorio.	Práctica	4 horas	

Semana 15	15	<b>Recuperatorio</b> --- ----- Evaluación de carpeta de trabajos prácticos.	Evaluación	4 horas	-
Semana 16	16	Cierre de Notas. Llenado de actas.	---	4 horas	-

### Evaluación

En este programa, se abordan un total de 6 unidades temáticas. Para la evaluación, se divide el contenido en dos partes. La primera evaluación se lleva a cabo durante la semana 7 y abarca las tres primeras unidades temáticas que se han detallado previamente en el programa (Diseño de Sistemas-Del diseño de Sistema al Diseño de software- Patrones y el Diseño). Por otro lado, la segunda evaluación, se llevará adelante en la semana 13 y se enfocará en las tres unidades temáticas restantes (Diseño, refinamiento y especificación – Diseño de Sistema de Tiempo real . Interfase hombre – Computadora). De esta manera, se asegura que los estudiantes sean evaluados de manera equitativa y abarquen todo el contenido del curso en ambas evaluaciones parciales. La evaluación consiste en un examen escrito en el cual deben responder preguntas teóricas relacionadas con el contenido del curso y, además, se presenta un caso sobre el que deberán resolver determinadas consignas referidas a la práctica del diseño de software. Con esto se busca evaluar la capacidad de los estudiantes para aplicar sus conocimientos teóricos del diseño de software en un contexto práctico y real. El caso práctico presentado en la evaluación permite evaluar su habilidad para analizar, identificar y resolver problemas relacionados con el diseño de software

Las respuestas de los estudiantes en el examen escrito, tanto para las preguntas teóricas como para el caso práctico, son evaluadas por el profesor o el equipo docente encargado del curso. Durante la evaluación, se consideran varios aspectos, que incluyen conocimiento y comprensión de cada concepto, como también la claridad de la redacción y la organización de las respuestas para asegurar una comunicación efectiva de los conceptos.

Una vez corregido el examen, se proporciona retroalimentación a los estudiantes. Esta retroalimentación puede incluir comentarios sobre los aciertos y errores cometidos, así como recomendaciones para mejorar el conocimiento y la comprensión de temas específicos.

En el caso de que un estudiante no apruebe una de las dos evaluaciones parciales, en la semana 15 habrá una instancia de recuperación.

<b>Primera evaluación</b>	Semana 7	Examen escrito teórico/práctico comprende las unidades 1 a 3 del programa.	Duración: 2 horas Horario: 2 primeras horas de cursada.
<b>Segunda evaluación</b>	Semana 13	Examen escrito teórico/práctico comprende las unidades 4 a 6 del programa.	Duración: 2 horas Horario: 2 primeras horas de cursada.

<b>Recuperatorio</b>	Semana 15	Examen escrito teórico/práctico del primer o segundo parcial según corresponda	Duración: 2 horas Horario: 2 primeras horas de cursada.
----------------------	-----------	--	--

**Bibliografía obligatoria**

Titulo	Autor	Editorial	Edición	Año
Ingeniería del software	Ian Somerville	Pearson Educación	7ma. edición	2005
Ingeniería del software: en enfoque Práctico	Roger Pressman	McGraw Hill	3ra. edición en español de la 7ma.edición en inglés	2010
Design Pattern	Erick Gamma,	Addison - Wesley		1995

**Bibliografía complementaria recomendada**

Titulo	Autor	Editorial	Edición	Año

**Otros recursos obligatorios** [Videos, enlaces, otros. Incluir una fila por cada recurso]

Nombre

**Otros recursos complementarios** [Videos, enlaces, otros. Incluir una fila por cada recurso]

Nombre