

Carrera INGENIERIA EN INFORMATICA		
Asignatura 3653 -Arquitecturas de Sistemas Software		
Trayecto Desarrollo de Software		
Año académico 2023		
Responsable / Jefe de catedra Mg. Ing. Osvaldo Sposito		
Carga horaria semanal 4hs	Carga horaria total 64hs	Créditos ---
Modalidad: Presencial		
Correlativas anteriores DISEÑO DE SOFTWARE	Correlativas posteriores: GESTION APLICADA AL DESARROLLO DE SOFTWARE I	
Conocimientos necesarios -----		

Descripción de la asignatura

La arquitectura de software involucra la estructuración del sistema, la cual representa un diseño de alto nivel del mismo. A través de la arquitectura se busca, por una parte, satisfacer los atributos de calidad (desempeño, seguridad, adaptabilidad, etc.) y además proporcionar una guía en el desarrollo. No diseñar la arquitectura desde etapas tempranas del desarrollo puede limitar negativamente el producto software de modo que este no satisfaga las necesidades del cliente, elevando los costos de corrección y mantenimiento. La importancia de esta asignatura es que permitirá al alumno conocer los conceptos del diseño arquitectónico e implementación, incluyendo patrones de arquitectura, su representación, herramientas de apoyo, además, se lo capacitará para evaluar las propiedades de una arquitectura a partir del análisis del diseño estructural de un sistema software.

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

Se espera que los alumnos que hayan finalizado la cursada de esta asignatura logren los siguientes objetivos:

- Comprender la importancia de la arquitectura de software y cómo esta se relaciona con el diseño a lo largo del ciclo de vida de desarrollo.
- Entender el proceso del diseño de la arquitectura y sus distintas fases.
- Conocer las características de los principales estilos arquitecturales.
- Aplicar las mejores prácticas del diseño de la arquitectura utilizando los patrones arquitectónicos más convenientes para cada tipo de sistemas de software.
- Definir la arquitectura y su representación aplicando metodologías, procesos y herramientas que conforman las mejores prácticas del mercado.
- Capacidad de evaluar las propiedades de una arquitectura tomando en consideración los requisitos y las restricciones.

Contenidos mínimos

Conceptos de arquitectura: modularidad, escalabilidad. Géneros arquitectónicos: centrada en datos, cliente-servidor, monolíticas, en capas, flujo de datos, microservicios, jerárquicas, orientadas a objetos. Arquitecturas emergentes. Patrones y Antipatrones de Arquitecturas. Diseño a Nivel de Componentes. Diagramas Arquitectónicos.

Competencias a desarrollar

Genéricas

- Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.
- Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.
- Desempeño en equipos de trabajo.

- Comunicación efectiva.
- Actuación profesional ética y responsable.
- Aprendizaje continuo.
- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.
- Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- Gestión, planificación, ejecución y control de proyectos de ingeniería en sistemas de información / informática.

Específicas

- Especificación, proyecto y desarrollo de software.
- Especificación, proyecto y desarrollo de sistemas de información.
- Especificación, proyecto y desarrollo de sistemas de comunicación de datos.
- Establecimiento de métricas y normas de calidad de software.

Programa analítico	
Unidad 1	Arquitectura de software. Conceptos de arquitectura: modularidad, escalabilidad, Cohesión y Acoplamiento. La arquitectura de software en el proceso de desarrollo de desarrollo. Diferencia entre arquitectura de software y diseño de software. El rol del arquitecto. Arquitectura y calidad.
Unidad 2	Géneros arquitectónicos. Estilos arquitectónicos: centrado en datos, cliente-servidor, monolíticas, en capas, flujo de datos, microservicios, jerárquicas, orientada a objetos. Objetivos, características principales, escenarios de aplicación, ventajas y desventajas de cada uno. Arquitecturas emergentes.
Unidad 3	Patrones y Antipatrones de Arquitecturas. Mejores prácticas relacionadas con la arquitectura de software. Trazabilidad de los requisitos a la arquitectura.
Unidad 4	Diseño arquitectónico. El proceso de diseño de arquitectura. Arquitecturas mixtas. Despliegue arquitectónico y escalabilidad de soluciones. Herramientas tecnológicas para soportar las decisiones arquitectónicas.
Unidad 5	Representación de arquitecturas. Diagramas y Presentación, Diseño a Nivel de Componentes. Despliegue de la arquitectura. Vistas arquitectónicas. Diagramas de UML y ArchiMate.
Unidad 6	Estrategias de verificación de arquitecturas de software. Procesos de evaluación de la arquitectura y de los requisitos no funcionales. Métodos de evaluación arquitectónica. Matriz de Riesgos, Identificación y Evaluación de Riesgos.

Planificación de actividades					
Semana	Clase	Actividad	Tipo	Duración estimada	Unidad/des
Semana 1	1	Modularidad, escalabilidad, Cohesión y Acoplamiento. La arquitectura de software en el proceso de desarrollo de desarrollo. Diferencia entre arquitectura de software y diseño de software. El rol del arquitecto. Arquitectura y calidad	Teoría	180 minutos de Teoría. 60 minutos de consultas y revisión conceptual.	Unidad 1
Semana 2	2	TP 1: Arquitectura de Software	Práctica y consultas sobre el tema.	60 minutos de Teoría. 60 minutos Practica 60 minutos de Explicación TP 1 60 minutos de Consultas y Cierre de Unidad.	Unidad 1
Semana 3	3	Estilos arquitectónicos: centrado en datos, cliente-servidor, monolíticas, en capas, flujo de datos, microservicios, jerárquicas, orientada a objetos. Objetivos, características principales, escenarios de aplicación, ventajas y desventajas de cada uno. Arquitecturas emergentes.	Teoría	180 minutos de Teoría. 60 minutos de consultas y revisión conceptual.	Unidad 2
Semana 4	4	TP 2: Géneros Arquitectónicos	Práctica y consultas sobre el tema.	60 minutos de Teoría. 60 minutos Practica	Unidad 2

				60 minutos de Explicación TP 2 60 minutos de Consultas y Cierre de Unidad.	
Semana 5	5	Mejores prácticas relacionadas con la arquitectura de software. Trazabilidad de los requisitos a la arquitectura.	Teoría	180 minutos de Teoría. 60 minutos de consultas y revisión conceptual.	Unidad 3
Semana 6	6	TP 3: Patrones y Antipatrones.	Trabajo de Investigación	60 minutos de Teoría. 60 minutos Practica 60 minutos de Explicación TP 3 60 minutos de Consultas y Cierre de Unidad.	Unidad 3
Semana 7	7	Primer Evaluación. Escrita y Oral.	Evaluación	120 minutos 60 minutos de planificación de tareas 60 minutos consultas post evaluación	Unidades 1, 2 y 3
Semana 8	8	Diseño Arquitectónico. El proceso de diseño	Teoría	180 minutos de Teoría.	Unidad 4

		de arquitectura. Arquitecturas mixtas. Despliegue arquitectónico y escalabilidad de soluciones. Herramientas tecnológicas para soportar las decisiones arquitectónicas.		60 minutos de consultas y revisión conceptual.	
Semana 9	9	TP 4: Diseño Arquitectónico.	Práctica y consultas sobre el tema.	60 minutos de Teoría. 60 minutos Practica 60 minutos de Explicación TP 4 60 minutos de Consultas y Cierre de Unidad.	Unidad 4
Semana 10	10	Representación de Arquitectura. Diagramas y Presentación, Diseño a Nivel de Componentes. Despliegue de la arquitectura. Vistas arquitectónicas. Diagramas de UML y ArchiMate.	Teórico / Práctico	180 minutos de Teoría. 60 minutos de consultas y revisión conceptual.	Unidad 5
Semana 11	11	TP 5: Representación de Arquitectura.	Practica	60 minutos de Teoría. 60 minutos Practica 60 minutos de Explicación TP 5	Unidad 5

				60 minutos de Consultas y Cierre de Unidad.	
Semana 12	12	Estrategias de Verificación de Arquitectura de Software. Procesos de evaluación de la arquitectura y de los requisitos no funcionales. Métodos de evaluación arquitectónica. Matriz de Riesgos, Identificación y Evaluación de Riesgos.	Teoría	180 minutos de Teoría. 60 minutos de consultas y revisión conceptual.	Unidad 6
Semana 13	13	TP 6: Estrategias de Verificación de Arquitectura de Software	Trabajo de Investigación y consultas sobre el tema.	60 minutos de Teoría. 60 minutos Practica 60 minutos de Explicación TP 6 60 minutos de Consultas y Cierre de Unidad.	Unidad 6
Semana 14	14	Segunda Evaluación. Evaluación Escrita y Oral.	Evaluación	120 minutos 60 minutos de planificación de tareas 60 minutos consultas post evaluación	Unidades 4, 5 y 6
Semana 15	15	Recuperatorio Escrito y Oral de Primera	Evaluación	120 minutos	Unidades 1, 2 y 3 o 4, 5 y 6

		Evaluación o Segunda Evaluación.		60 minutos de planificación de tareas 60 minutos consultas post evaluación	
Semana 16	16	Presentación de Carpeta de Trabajos Prácticos y cierre de notas. Actividad basada en una presentación y exposición de los temas de la materia.	Actividad integradora de conceptos teóricos y prácticos.	210 minutos 30 minutos de cierre de notas	Unidades 1, 2, 3, 4, 5 y 6

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 (Arquitectura de software - Géneros arquitectónico – Patrones y Antipatrones de Arquitectura). Por otro lado, la segunda evaluación, se llevará adelante en la semana 13 y se enfocará en las tres unidades temáticas restantes (Diseño arquitectónico – Representación de Arquitecturas – Estrategia de Verificación de Arquitectura de Software). 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 la Arquitectura 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.

Primera evaluación	Semana 7	Escrito y Oral	120 minutos, 19:00 hs.
---------------------------	----------	----------------	------------------------

Segunda evaluación	Semana 14	Escrito y Oral	120 minutos, 19:00 hs.
Recuperatorio	Semana 15	Escrito y Oral	120 minutos, 19:00 hs.

Bibliografía obligatoria [Disponibles en la Biblioteca Leopoldo Marechal, o con acceso digital]				
Título	Autor	Editorial	Edición	Año
Fundamentals of Software Architecture	Mark Richards and Neal Ford	O'Reilly	Primera Edición	2020
Software Architecture: The Hard Parts	Neal Ford, Mark Richards, Pramod Sadalage and Zhamak Dehghni.	O'Reilly	Tercera Edición	2022

Bibliografía complementaria recomendada ([disponible en la Biblioteca Leopoldo Marechal, o con acceso digital])				
Título	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	
	https://www.youtube.com/@markrichards5014 Software Architecture - Mark Richards
	https://www.developertoarchitect.com/lessons/ Training and resources for the journey from software developer to software architect Mark Richards, Software Architect and Founder
	https://www.archimatetool.com/