

NOMBRE DEL CURSO: Análisis y Diseño de sistemas 2

CODIGO:	785	CRÉDITOS:	5
ESCUELA:	Ciencias y Sistemas	AREA:	Desarrollo de Software
PRERREQUISITO:	283	POSTREQUISITO:	780, 789, 735
CATEGORÍA:	Obligatorio	SECCIÓN:	A+
CATEDRATICO:	Ing. William Guevara	TUTOR ACADEMICO:	Alex Montufar
HORAS POR SEMANA DEL CURSO:	4	HORAS POR SEMANA DE LABORATORIO:	2
DÍAS QUE SE IMPARTE EL CURSO:	Martes y Jueves	DÍAS QUE SE IMPARTE LABORATORIO:	Sábado
HORARIO DEL CURSO:	7:10 - 8:50	HORARIO DE LABORATORIO:	7:10 – 8:50

DESCRIPCIÓN DEL LABORATORIO:

El laboratorio del curso de Análisis y Diseño de Sistemas 2, trata sobre la parte práctica del curso. Se encarga de brindar a los estudiantes herramientas para la aplicación de principios y metodologías para utilizar en el proceso de construcción de software. Se guía a los estudiantes en la aplicación de patrones de diseño de software.

OBJETIVO GENERAL:

Diseñar la arquitectura de software de un proyecto real, con base en diferentes puntos de vista y perspectivas para alcanzar los atributos de calidad.

OBJETIVOS ESPECÍFICOS:

- Identificar y describir conceptos de arquitectura de software para el desarrollo de sistemas informáticos.
- Identificar y describir conceptos de atributos de calidad y tácticas aplicables para alcanzarlos.

- Evaluar y adaptar el diseño de un sistema de software de acuerdo con los diferentes tipos de requerimientos de un proyecto.

HABILIDADES:

1. Conocimiento en la toma de requerimientos y atributos de calidad para la construcción de un proyecto de software.
2. Conocimiento sobre la parte teórica de patrones y principios de diseño para aplicarlos en la construcción de software.
3. Conocimiento en vistas, puntos de vista y perspectivas que se deben desarrollar para un proyecto de software.

COMPETENCIAS:

1. Dominar los conceptos básicos de la fase de análisis de la construcción de un proyecto de software.
2. Aplicar diferentes tipos de frameworks en el diseño de arquitecturas de proyectos de software.

METODOLOGÍA:

- Se impartirán clases presenciales para fortalecer el conocimiento de la implementación de patrones y principios de diseño.
- Elaboración de tareas para conocer los diferentes frameworks de trabajo que se utilizan para el diseño de la arquitectura de un proyecto de software.
- Se realizarán prácticas para evaluar el correcto análisis y diseño de un sistema enfocado a los diferentes puntos de vista involucradas.
- Se realizarán exámenes cortos para evaluar el conocimiento adquirido por los estudiantes en las prácticas y proyectos aplicados en laboratorio.
- Se un proyecto para poder evaluar los conceptos adquiridos en clase sobre la implementación de frameworks, patrones de diseño, vistas y perspectivas en el desarrollo de software.

CONTENIDO DEL CURSO:

Tópico	Descripción	Recursos
Fundamentos de diseño de software	Conceptos generales de diseño de software Contexto del diseño de software El proceso de diseño de software	Libro: Software Design, David Budgen Capítulos 1, 2 y 3
	Principios de diseño de software	Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 6, 7 y 21
Aspectos clave en el diseño de software	Concurrencia Event Handling and Control Persistencia de datos	Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 9, 18 y 21
	Distribución de componentes Error and Exception Handling Tolerancia a fallos Seguridad	Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 12, 16, 18
Introducción a la arquitectura de software	Definición Importancia de arquitectura de software Conceptos de arquitectura	Libro: Software systems Architecture, Working with Stakeholders Using viewpoints and Perspectives, Nick Rozanski, Eoin Woods Capítulos 1 y 2
	Puntos de Vista y Vistas Stakeholders	Libro: Software systems Architecture, Working with Stakeholders Using viewpoints and Perspectives, Nick Rozanski, Eoin Woods Capítulo3: 3
	Patrones (Estilos) de Arquitectura PoEAA (Martin Fowler)	Libro: Documenting Software Architecture, Clemments et al Capítulos: 1, 2, 3, 4, 5
Diseño de bajo nivel	Principios de diseño (SOLID)	http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod
	Patrones de diseño (GoF)	Libro: Head First Design Patterns, Eric Freeman & Elisabeth Freeman
Diseño de software, análisis de calidad y evaluación	Atributos de Calidad (puntos de vista, vistas, perspectivas)	Libro: Software Design, David Budgen Capítulo 4
	Análisis de calidad y técnicas de evaluación	Libro: Software systems Architecture, Working with Stakeholders Using viewpoints and Perspectives, Nick Rozanski, Eoin Woods Part 3: The Viewpoint Catalog Part 4: The Perspective Catalog
	Descripciones Estructurales (Vistas Estáticas) Descripciones de Comportamiento (Vistas Dinámicas)	

	Proceso para definir una arquitectura	Libro: Essential software architecture, Ian Gorton
Arquitectura de Software	Descripciones Estructurales (Vistas Estáticas) Descripciones de Comportamiento (Vistas Dinámicas)	Libro: Software systems Architecture, Working with Stakeholders Using viewpoints and Perspectives, Nick Rozanski, Eoin Woods Part 3: The Viewpoint Catalog Part 4: The Perspective Catalog
Arquitectura de Software Patrones/Estilos	Documentando la arquitectura de software	Libro: Documenting Software Architecture, Clemments et al
	Arquitectura de software en ambientes ágiles Toward Agile Architecture: Insights from 15 Years of ATAM Data, http://www.infoq.com/articles/atam-quality-attributes	http://martinfowler.com/articles/microservices.html Measure and improve code quality, http://www.infoq.com/news/2016/01/measure-improve-code-quality?utm_source=infoq&utm_medium=popular_widget&utm_campaign=popular_content_list&utm_content=homepage
Estrategias y métodos de diseño de Software	Estrategias generales Diseño orientado a funcionalidad (Estructurado) Diseño orientado a objetos Diseño centrado en la estructura de los datos Diseño basado en componentes	Libro: Software Design, David Budgen Capítulos 8,10, 13, 14, 15, 16, 17
	El arquitecto de software	

OBSERVACIONES:

- La calificación de los proyectos de laboratorio es en grupos formados en clase acoplándose al día y horario que se indique previamente.
- Copias parciales o totales de los proyectos tendrán una nota de 0 puntos y los responsables serán reportados a la Escuela de Ingeniería en Ciencias y Sistemas.
- Se debe de mandar los archivos entregables en fechas establecidas para tener derecho a calificación.

BIBLIOGRAFÍA:

1. Head first design patterns, Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra
2. Software systems architecture, working with stakeholders using viewpoints and perspectives, Nick Rozanski, Eoin Woods
3. Essential software architecture, Ian Gorton
4. Software Engineering 9th Edition, Ian Sommerville
5. Documenting Software Architecture, Clemments
6. Patterns of Enterprise Application Architecture - Martin Fowler