



Análisis y Diseño de Sistemas 2

CÓDIGO:	785	CRÉDITOS:	5
ESCUELA:	Ciencias y Sistemas	ÁREA:	Desarrollo de Software
PRERREQUISITO:	283	POST REQUISITO:	780, 789, 735
CATEGORÍA:	Obligatorio	SECCIÓN:	A
CATEDRÁTICO:	Ing. Ricardo Morales	TUTOR ACADÉMICO:	Dodany Girón
HORAS POR SEMANA:	4	HORAS POR SEMANA LAB:	1:40
DÍAS QUE SE IMPARTE:	Martes y Jueves	DÍAS QUE SE IMPARTE LAB:	Lunes
HORARIO:	07:10 - 08:50 17:30 - 19:00	HORARIO LAB:	8:50 - 10:30

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 virtuales 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 desarrollará 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:

Frameworks	Que es un framework. Tipos de framework. Beneficios del uso de framework. Factores de elección de un framework
Elementos de Diseño	Patrones de diseño Contexto del diseño Principios de diseño
Despliegue en la nube (Cloud)	Arquitecturas Monolíticas vrs Microservicios PaaS SaaS Docker
Patrones de Arquitectura	Categoría Objetivos
Performance y Stress	Pruebas de rendimiento Pruebas de carga Pruebas de rendimiento Pruebas de stress Herramientas para las pruebas de rendimiento. Profiling Afinamiento (tunning) de aplicaciones. Métricas y Herramientas de análisis de calidad de código
DevOps	Beneficios de DevOps

	Ciclo de vida de las Aplicaciones Plan/Code/Build/Test Release/Deploy/Operate/Monitor DevOps Tools
--	-------------------------------------------------------------------------------------------------------------

EVALUACIÓN:

Actividad	Puntos
Tareas (3)	15
Cortos (3)	15
Hojas de trabajo (3)	15
Seguimiento Proyecto	40
Exámen Final	15
Total	100

INFORMACIÓN ADICIONAL

- Para aprobar el laboratorio se deberá tener una nota igual o mayor a 61/100.
- No se reciben tareas tarde.
- Solo pueden cursar el laboratorio las personas que están asignadas en el curso, no se guarda nota.
- Para reposición de actividades se debe presentar motivo por el cual no pudo realizar la actividad.
- Es obligatorio promediar 61 en las prácticas.

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