



FICHA TÉCNICA DEL CURSO: **Análisis y Diseño de Sistemas 2**

No.	Descripción		
.	Código 785	Créditos 4	
1	Escuela Ciencias y Sistemas	Área a la que pertenece Software	Vigencia Vacaciones Junio 2015
2	Horas por semana 10	Horario Lunes a Viernes 7:00-9:00 – Edificio T-3	
3	Pre-requisitos: Análisis y diseño de sistemas 1 (283)		
4	Post-requisitos: Software Avanzado (780)		
5	Sección: B		
6	I. Descripción General El curso de Análisis de Sistemas 2 es un curso especializado de la carrera de Ingeniería en Ciencias y Sistemas, comprendido en el área de Software. Es un curso que describe actividades a ser tomadas en cuenta en las fases de diseño y programación de cualquier proyecto de desarrollo de software. Específicamente se cubren los conceptos y técnicas relacionadas con la administración de la configuración, integración continua, el análisis y diseño de una solución de software basado en la definición de requerimientos, centrándose en la construcción de una arquitectura de sistema.		
II. Objetivos			
1. Dar a conocer al alumno los conceptos de administración de la configuración e integración continua para aplicarlos en proyectos reales			
2. Aplicar los conceptos de arquitectura de software en el desarrollo de sistemas informáticos.			
3. Aplicar los de patrones de diseño y arquitectura.			
4. Dar a conocer al alumno una vista general de lo que es ITIL			
III. Contenido			
1 Administración de la Configuración			
1.1 Introducción a la administración de la configuración			
1.1.1. Conceptos			
1.1.1.1 Repositorio			
1.1.1.2 Árbol, rama, etiqueta			
1.1.2 Control de versiones			
1.1.3 Administración de dependencias			
1.1.4 Administración de configuración de software			
1.1.5 Administración de ambientes			
1.2 Entrega Continua (Continuous Delivery)			
1.2.1 Principios de entrega de software			
1.2.2 Implementando estrategias de pruebas			
1.2.3 Línea de entrega (Deployment Pipeline)			
1.3 Integración Continua (Continuous Integration)			
1.3.1 Componentes			
1.3.2 Características de CI			
1.3.3 Valor de CI			
1.3.4 Cuando implementarlo			
1.3.5 Reduciendo el riesgo			
1.3.6 Construyendo software en cada cambio			
1.4 Devops			
1.4.1 Definición			
1.4.2 Actividades y herramientas			
2 Arquitectura de software			
2.1 Introducción			
2.2 Definiciones			
2.2.1 Arquitectura y Arquitecto			
2.3 Atributos de calidad del software			
2.4 Estilos de Arquitectura de Software			
2.5 Modelo 4+1 Vistas			



2.6 Tácticas

3 Diseño

3.1 Principios de diseño orientado a objetos

- 3.1.1 OCP, open closed principle
- 3.1.2 DRY, don't repeat yourself
- 3.1.3 SRP, single responsibility principle
- 3.1.4 LSP, Liskov substitution principle
- 3.1.5 DIP, dependency inversion principle
- 3.1.6 IOC, inversión of control

3.2 Patrones de diseño

3.2.1 Patrones de creación

- 3.2.1.1 Singleton
- 3.2.1.2 Factory Method

3.2.2 Patrones de estructura

- 3.2.2.1 Decorator
- 3.2.2.2 Adapter

3.2.3 Patrones de comportamiento

- 3.2.3.1 Strategy
- 3.2.3.2 Observer

3.3 Antipatrones de Diseño

4 Integración de arquitecturas

4.1 SOA

- 4.1.1 Evolución de SOA
- 4.1.2 Web services y SOA
- 4.1.3 Principios de orientación al servicio
- 4.1.4 Capas de servicio
- 4.1.5 Estrategias de entrega

4.2 REST

4.3 Cloud computing

- 4.3.1 Definición
- 4.3.2 Opciones disponibles

4.4 Bases de datos NoSQL

IV. Metodología:

1. Clase magistral impartida 1 vez por semana
2. Clase laboratorio impartida 1 vez por semana
3. Desarrollo de proyecto de sistema de software
4. Tareas y evaluaciones complementarias
5. Lecturas y discusiones complementarias

V. Evaluación:

Aspecto	Valor
Evaluaciones parciales	40
1er parcial (20)	
2do parcial (20)	
Proyectos de clase	20
Laboratorio	5
Trabajo de investigación (temas específicos), tareas, cortos, etc.	10
Total zona	75
Examen final	25

VI. Requisitos para el curso

- Es obligatorio aprobar el proyecto para tener derecho a examen final, se aprueba con 61/100
- Es obligatorio aprobar el laboratorio para tener derecho a examen final, se aprueba con 61/100
- Es obligatorio aprobar el curso SQM para tener derecho a examen final.**
- Solo se calificarán exámenes y proyectos de estudiantes asignados en el curso. NO se agregan estudiantes a actas

7	Bibliografía	<ul style="list-style-type: none"> • Subversion version control, using the subversion version control system in development projects, William Nagel (Libro electrónico) • Continuos integration, Paul M. Duvall
---	--------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



		<ul style="list-style-type: none">• Essential software architecture, Ian Gorton• Service oriented architecture, concepts, technology and design, Thomas Erl• Head first design patterns, Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra
8	No. De Secciones	1
9	Catedráticos titulares y auxiliares	Ing. Luis Alberto Arias Solórzano ariasolo_II@yahoo.com Aux. Elder Herrera
10	Coordinador de Área	N/A
11	Director de Escuela	Ing. Marlon Pérez Turk