



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 Segundo Semestre 2012
2	Horas por semana 10	Horario Lunes y Miércoles de 7:10 a 8:50 horas – Edificio T-5 Salón 305	
3	Pre-requisitos: Análisis y diseño de sistemas 1 (283)		
4	Postrequisitos: Software Avanzado (780)		
5	Secciones: A+ (Alumnos cuyo nro. De carné termina en nro. Par) y A- (Alumnos cuyo nro. De carné termina en nro. Impar)		
6	<p>I. Descripción General</p> <p>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.</p> <p>II. Objetivos</p> <ul style="list-style-type: none"> • Proveer al alumno los conceptos de administración de la configuración e integración continua para aplicarlos en proyectos reales • Presentar al alumno conceptos de arquitectura de software en el desarrollo de sistemas informáticos. • Presentar al alumno conceptos y aplicación de patrones de diseño y arquitectura. • Presentar al alumno una vista general de lo que es ITIL <p>III. Contenido</p> <p>1. Introducción a la administración de la configuración</p> <p>1.1. Conceptos</p> <p>1.1.1.1. Repositorio</p> <p>1.1.1.2. Árbol, rama, etiqueta</p> <p>1.2. Nomenclatura de versiones</p> <p>1.3. Administración del cambio</p> <p>2. Integración continua y devops</p> <p>2.1. Integración continua</p> <p>2.1.1. Componentes</p> <p>2.1.2. Características de CI</p> <p>2.1.3. Valor de CI</p> <p>2.1.4. Cuando implementarlo</p> <p>2.1.5. Reduciendo el riesgo</p> <p>2.1.6. Construyendo software en cada cambio</p> <p>2.2. Devops</p> <p>2.2.1. Definición</p> <p>2.2.2. Actividades y herramientas</p> <p>3. Arquitectura de software</p> <p>3.1. Introducción</p> <p>3.2. Definiciones</p> <p>3.2.1. Arquitectura</p> <p>3.2.2. Arquitecto</p> <p>3.3. Atributos de calidad del software</p> <p>3.4. Arquitectura middleware y tecnologías</p> <p>3.5. Proceso para definición arquitectura de software</p> <p>3.6. Documentación de arquitectura</p> <p>4. Temas de arquitectura</p> <p>4.1. SOA</p>		



- 4.2. Evolución de SOA
- 4.3. Web services y SOA
- 4.4. Principios de orientación al servicio
- 4.5. Capas de servicio
- 4.6. Estrategias de entrega
- 4.7. REST
- 4.8. Cloud computing
 - 4.8.1. Definición
 - 4.8.2. Opciones disponibles
- 4.9. Bases de datos NoSQL
- 5. Patrones de diseño**
 - 5.1. Patrones de creación
 - 5.2. Patrones de estructura
 - 5.3. Patrones de comportamiento
- 6. ITIL**
 - 6.1. Conceptos
 - 6.2. Ciclo de vida
 - 6.2.1. Estrategia del Servicio
 - 6.2.2. Diseño del Servicio
 - 6.2.3. Transición del Servicio
 - 6.2.4. Operaciones del Servicio
 - 6.2.5. Mejora Continua del Servicio

IV. Metodología:

- Clase magistral impartida 2 veces por semana
- Clase laboratorio impartida 1 vez por semana
- Desarrollo de proyecto de sistema de software
- Tareas y evaluaciones complementarias
- Lecturas y discusiones complementarias
- Autoestudio

V. Evaluación:

Aspecto	Valor
Evaluaciones parciales	40-20
1er parcial (20)	
2do parcial (20)	
Proyectos de clase (5 fases)	20
Laboratorio	10
Trabajo de investigación (temas específicos), tareas, cortos, etc.	5
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 ISE y 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

- **Subversion version control, using the subversion version control system in development projects**, William Nagel (Libro electrónico)
- **Continuos integration**, Paul M. Duvall
- **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

Lecturas obligatorias

- **Administración de la configuración**
 1. Subversion version control, using the subversion version control system in development projects, capítulo 1
 2. Git for designers, version control,



		<p>http://hoth.entp.com/output/git_for_designers.html</p> <p>3. Subversion version control, using the subversion version control system in development projects, capítulo 14</p> <p>4. Version Control for Multiple Agile Teams, http://www.infoq.com/articles/agile-version-control</p> <ul style="list-style-type: none"> • Integración continua <ol style="list-style-type: none"> 5. CI from the Trenches: Real-World Continuous Integration Challenges (and what to do about them), http://www.infoq.com/presentations/ci-from-the-trenche 6. Patterns of Agile Practice Adoption, capítulo 9, continuous integration, http://infoq.com/minibooks/agile-patterns • DevOps <ol style="list-style-type: none"> 7. From Dev To Production Through Build Pipelines and Teamwork, http://www.infoq.com/presentations/From-Development-To-Production • Arquitectura de software <ol style="list-style-type: none"> 8. Essential software architecture, Ian Gorton, capítulos 1 a 6 9. NoSQL at Twitter, http://www.infoq.com/presentations/NoSQL-at-Twitter 10. LMAX - How to Do 100K TPS at Less than 1ms Latency, http://www.infoq.com/presentations/LMAX 11. Forging ahead - Scaling the BBC into Web/2.0, http://www.infoq.com/presentations/web-20-bbc-scaling 12. Extreme transaction processing patterns: write behind caching, http://www.infoq.com/articles/write-behind-caching (opcional) • SOA <ol style="list-style-type: none"> 13. Service oriented architecture, concepts, technology and design, Thomas Erl, capítulos 3, 5, 8, 9, 10 • REST <ol style="list-style-type: none"> 14. RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision, Cesare Pautasso, Olaf Zimmermann, Frank Leyman, http://www.jopera.org/files/www2008-restws-pautassozimmermann-leymann.pdf • Cloud <ol style="list-style-type: none"> 15. Computing Strategy in the Cloud Era, http://www.infoq.com/presentations/Cloud-Computing-Strategy-Lew-Moorman 16. Clash of the clouds, The Economist <ul style="list-style-type: none"> o Cloud computing: the future of IT application architectures, http://www.cio.com/article/518113/Cloud_Computing_The_Future_of_IT_Application_Architectures?page=2&taxonomyId=3112 17. Availability, the Cloud and Everything, http://www.infoq.com/presentations/Availability-Cloud-and-Everything 18. Cloud computing for developers and architects, http://www.slideshare.net/StuC/cloud-computing-for-architectsqcon-2008-tutorial-presentation?src=related_normal&rel=2238491 (opcional)
8	No. De Secciones	2
9	Catedráticos titulares y auxiliares	Ing. William Guevara (Secc. A-), willgo021983@gmail.com Inga. Ivonne Aldana (Secc. A+), ivonnealdanal@gmail.com Auxiliares:
10	Coordinador de Área	
11	Director de Escuela	Ing. Marlon Pérez Turk