

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 Primer semestre 2015				
2	Horas por semana 4	Horario Martes de 7:10 a 8:50 y miércoles 17:20 a 19:00					
3	Pre-requisitos:283						
4	Postrequisitos:						
5	Sección: Única						
6	<p>I. Descripción General</p> <p>¿Te has preguntado cómo es posible agilizar el proceso de codificación de software? ¿Has pensado en que existe mas de una forma de definir la estructura interna de un sistema de software? En un mundo donde se busca obtener resultados mas rápido, manteniendo la calidad de un producto, los conceptos de control de versiones, integración y entrega continua, patrones de diseño y arquitectura de software son herramientas de gran utilidad para alcanzar esos objetivos</p> <p>Al final del curso los estudiantes tendrán una visión diferente de cómo agilizar los procesos de codificación y retroalimentación del código generado. Un sistema de control de versiones es un paso inicial para la automatización y agilización del proceso de codificación. Un sistema de integración continua permite la integración con el sistema de control de versiones y la generación de retroalimentación oportuna del código generado. El estudiante aplicará estos conceptos en el proyecto del curso.</p> <p>Por otro lado, al finalizar el curso, se espera que el estudiante tenga una visión integral de los elementos de diseño que se deben considerar para el desarrollo de un sistema de software. Los conceptos de patrones de diseño, arquitectura de software y patrones de arquitectura permiten tener una visión general de las estructuras propuestas para conformar un sistema. El alcanzar los diferentes atributos de calidad de un sistema se logrará a través de la aplicación de puntos de vista y perspectivas. Finalmente la integración de sistemas se plantea con la aplicación de los conceptos de una arquitectura orientada a servicios (SOA).</p> <p>II. Objetivos</p> <ul style="list-style-type: none"> • Identificar y describir los componentes de un sistema de entrega continua (continuous deployment) • Diseñar y aplicar un proceso de integración continua para el proyecto del curso • Identificar y describir los conceptos de arquitectura de software para el desarrollo de sistemas informáticos • Identificar y describir los conceptos de atributos de calidad y tácticas aplicables para alcanzarlos • Diseñar la arquitectura de software de un proyecto real, con base en diferentes puntos de vista y perspectivas para alcanzar los atributos de calidad de dicho proyecto. <p>III. Contenido</p> <table border="1"> <thead> <tr> <th>Título y objetivos de aprendizaje</th> <th>Lecturas y tareas</th> </tr> </thead> <tbody> <tr> <td> Administración de la configuración <ul style="list-style-type: none"> • Describir el concepto de administración de la configuración • Conocer y entender los elementos que integran un sistema de control de versiones • Identificar el valor de las prácticas recomendadas para implementar la administración de la </td> <td> <ul style="list-style-type: none"> • A visual guide to version control, http://betterexplained.com/articles/a-visual-guide-to-version-control/ • Pro Git, http://git-scm.com/book/en/v2 • A successful Git branching model, </td> </tr> </tbody> </table>			Título y objetivos de aprendizaje	Lecturas y tareas	Administración de la configuración <ul style="list-style-type: none"> • Describir el concepto de administración de la configuración • Conocer y entender los elementos que integran un sistema de control de versiones • Identificar el valor de las prácticas recomendadas para implementar la administración de la 	<ul style="list-style-type: none"> • A visual guide to version control, http://betterexplained.com/articles/a-visual-guide-to-version-control/ • Pro Git, http://git-scm.com/book/en/v2 • A successful Git branching model,
Título y objetivos de aprendizaje	Lecturas y tareas						
Administración de la configuración <ul style="list-style-type: none"> • Describir el concepto de administración de la configuración • Conocer y entender los elementos que integran un sistema de control de versiones • Identificar el valor de las prácticas recomendadas para implementar la administración de la 	<ul style="list-style-type: none"> • A visual guide to version control, http://betterexplained.com/articles/a-visual-guide-to-version-control/ • Pro Git, http://git-scm.com/book/en/v2 • A successful Git branching model, 						

<p>configuración</p>	<p>http://nvie.com/posts/a-successful-git-branching-model/</p> <ul style="list-style-type: none"> Tarea: identificar los pasos a seguir para implementar un sistema de control de versiones para el proyecto del curso
<p>Sistema de integración continua</p> <ul style="list-style-type: none"> Describir el concepto de integración continua Conocer y entender los elementos que integran un sistema de integración continua y su relación con un sistema de control de versiones Identificar el valor de las prácticas recomendadas para implementar la integración continua Aplicar los conceptos integración continua en el proyecto del curso 	<ul style="list-style-type: none"> Continuous integration in agile development, http://www.ibm.com/developerworks/rational/library/continuous-integration-agile-development/ Introduction to Continuous Integration with Xamarin, http://developer.xamarin.com/guides/cross-platform/ci/intro_to_ci/ Tarea: identificar los pasos a seguir para implementar un sistema de integración continua para el proyecto del curso
<p>Sistema de entrega continua</p> <ul style="list-style-type: none"> Describir el concepto de entrega continua Conocer y entender los elementos que integran un sistema de entrega continua y su relación con un sistema de control integración continua Describir y entender el concepto de deployment pipeline Identificar el valor de las prácticas recomendadas para implementar la entrega continua 	<ul style="list-style-type: none"> Continuous delivery vs continuous deployment, http://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/ 21 century software, http://www.slideshare.net/jezhumble/21-century-software Tarea: identificar los pasos a seguir para implementar un sistema de entrega continua para el proyecto del curso
<p>Principios de diseño</p> <ul style="list-style-type: none"> Identificar y comprender principios de diseño aplicados a sistemas orientados a objetos 	<ul style="list-style-type: none"> Agile design principles, http://issuu.com/todaysoftmag/docs/tsm_25_2014_en/1?e=9190753/8688902 Tarea: describir la utilidad del principio de inversión de dependencia en un sistema
<p>Patrones de diseño</p> <ul style="list-style-type: none"> Identificar y comprender patrones de diseño y su clasificación, aplicados a sistemas orientados a objetos Evaluar ejemplos de aplicación de patrones de diseño 	<ul style="list-style-type: none"> Tarea: crear un ejemplo de aplicación de 2 patrones de diseño para resolver situaciones reales en un sistema
<p>Conceptos generales de arquitectura de software</p> <ul style="list-style-type: none"> Comprender y discutir el concepto de arquitectura de software Identificar porque es importante la arquitectura de software Comprender el enfoque de un modelo dirigido por riesgo para arquitectura de software 	<ul style="list-style-type: none"> What's the Best Way to Improve Software Architectures?, http://www.infoq.com/presentations/panel-improve-software-architecture
<p>Vistas, puntos de vista y perspectivas</p> <ul style="list-style-type: none"> Comprender como describir un sistema complejo 	<ul style="list-style-type: none"> Interview with Raffi Krikorian on Twitter's Infrastructure,

<p>a través de un conjunto de vistas interrelacionadas</p> <ul style="list-style-type: none"> • Identificar los beneficios de un enfoque vistas y puntos de vista para describir un sistema • Comprender el uso de perspectivas para alcanzar un conjunto relacionado de propiedades de calidad en un sistema • Aplicar los conceptos de vista, punto de vista y perspectiva en el proyecto del curso 	<p>http://www.infoq.com/resource/minibooks/emag-scalability/en/pdf/Scalability-eMag.pdf</p>
<p>Proceso de arquitectura</p> <ul style="list-style-type: none"> • Comprender las actividades de un proceso de definición de arquitectura • Aplicar el proceso de arquitectura en el proyecto del curso 	<ul style="list-style-type: none"> • Interview: Adrian Cockcroft on High Availability, Best Practices, and Lessons Learned in the Cloud, http://www.infoq.com/resource/minibooks/emag-scalability/en/pdf/Scalability-eMag.pdf
<p>Estilos de arquitectura de software</p> <ul style="list-style-type: none"> • Comprender el concepto de estilos de arquitectura • Evaluar estilos de arquitectura existentes y comprender escenarios para su aplicación 	<ul style="list-style-type: none"> • Wix architecture at scale, http://www.infoq.com/presentations/wix-architecture
<p>Catálogo de puntos de vista</p> <ul style="list-style-type: none"> • Comprender el concepto de punto de vista y su aplicación • Comprender los elementos que cubre cada punto de vista y los modelos para presentarla • 	<ul style="list-style-type: none"> • 10 Billion a Day, 100 Milliseconds Per: Monitoring Real-Time Bidding at AdRoll, http://www.infoq.com/presentations/erlang-bidding-system
<p>Catálogo de perspectivas</p> <ul style="list-style-type: none"> • Comprender el concepto de perspectiva y su aplicación • Comprender los elementos que cubre cada perspectiva y las tácticas asociadas • Aplicar el uso de perspectivas en el proyecto del curso 	<ul style="list-style-type: none"> • Web & database load testing with Jmeter, http://www.infoq.com/presentations/jmeter-data-driven-methodology
<p>Conceptos de arquitectura orientada a servicios</p> <ul style="list-style-type: none"> • Comprender el concepto y principios de orientación al servicio • Comprender y evaluar estrategias de entrega de arquitectura orientada a servicios 	<ul style="list-style-type: none"> • Updated Principles of Service Orientation, http://www.infoq.com/articles/updated-soa-principles • Microservices: Decomposing Applications for Deployability and Scalability, http://www.infoq.com/articles/microservices-intro

IV. Metodología:

El curso se basa principalmente en clases magistrales, pero también se desarrollarán otras actividades. Las discusiones en clase, de conceptos y ejemplos, buscarán complementar los conceptos teóricos desarrollados. También se desarrollará un proyecto grupal de clase y el desarrollo de un artículo que será calificado a través de revisiones de pares (peer review).

V. Evaluación:

Aspecto	Valor
Evaluaciones parciales	38
1er parcial, 10 marzo (19)	
2do parcial, 21 abril (19)	
Proyecto	23
Fase 1, 6 febrero (5)	
Fase 2, 6 marzo (6)	
Fase 3, 10 abril (6)	
Fase 4, 1 mayo (6)	
Artículo, 24 marzo	5
Tareas y cortos	4
Laboratorio	5
Total zona	75
Examen final	25

VI. Observaciones:

- 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
- Solo se calificarán exámenes y proyectos de estudiantes asignados en el curso. NO se agregan estudiantes a actas
- Los artículos del curso son elegibles para la siguiente edición de la revista de la escuela

7	Bibliografía	<ul style="list-style-type: none"> • Continuous delivery, Jez Humble, David Farley • Continuos integration, Paul M. Duvall • Head first design patterns, Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra • Software systems architecture, working with stakeholders using viewpoints and perspectives, Nick Rozanski, Eoin Woods • Essential software architecture, Ian Gorton • Just enough software architecture, George Fairbanks • Service oriented architecture, concepts, technology and design, Thomas Erl
8	No. De Secciones	1
9	Catedráticos titulares y auxiliares	Ing. Ricardo Morales Robson Cruz
11	Director de Escuela	Ing. Marlon Pérez