

Facultad de Ingeniería y Ciencias Agropecuarias
Ingeniería en Sistemas de Computación e Informática
ACI961/Ingeniería de Software II
Período: 2016-1

1. Identificación.-

Número de sesiones: 48
 Número de horas: 120
 Créditos: 4.5
 Profesor: CHRISTIAN ANIBAL BASTIDAS ROMERO
 Correo electrónico del docente (Udlanet): c.bastidas@udlanet.ec
 Director: MARCO ANTONIO GALARZA CASTILLO
 Campus: Queri
 Pre-requisito: ACI920 Ingeniería de Software I
 Co-requisito:
 Paralelo: 1
 Tipo de asignatura:

| | |
|-------------|---|
| Optativa | |
| Obligatoria | X |
| Práctica | |

Organización curricular:

| | |
|---------------------------------|---|
| Unidad 1: Formación Básica | |
| Unidad 2: Formación Profesional | X |
| Unidad 3: Titulación | |

Campo de formación:

| Campo de formación | | | | |
|----------------------|--------------------|---|---|--------------------------|
| Fundamentos teóricos | Praxis profesional | Epistemología y metodología de la investigación | Integración de saberes, contextos y cultura | Comunicación y lenguajes |
| | X | | | |

2. Descripción del curso.-

La asignatura prepara a los estudiantes para afrontar con calidad y solvencia temas de arquitectura y desarrollo de software de aplicaciones empresariales.

3. Objetivo del curso.-

Al finalizar el curso el estudiante estará en la capacidad de aplicar con solvencia las mejores prácticas de arquitectura y desarrollo de software de aplicaciones empresariales priorizando la calidad en un proyecto real por medio de la construcción de un prototipo funcional de una aplicación práctica.

4. Resultados de aprendizaje deseados al finalizar el curso:

| Resultados de aprendizaje (RdA) | RdA perfil de egreso de carrera | Nivel de desarrollo (carrera) |
|--|--|---|
| <p>1. Desarrolla soluciones de software de nivel empresarial con el uso de estilos arquitecturales.</p> <p>2. Analiza los conceptos de mejoramiento continuo del proceso del software en las categorías de soporte y administración de procesos.</p> | <p>5. Lidera procesos de producción, incorporación, adaptación y transferencia de soluciones informáticas para mejorar el desempeño de las organizaciones, mostrando su capacidad de abstracción, análisis y síntesis.</p> | <p>Inicial ()</p> <p>Medio (X)</p> <p>Final ()</p> |

5. Sistema de evaluación.-

De acuerdo al Modelo Educativo de la UDLA la evaluación busca evidenciar el logro de los resultados de aprendizaje (RdA) enunciados en cada carrera y asignatura, a través de mecanismos de evaluación (MdE). Por lo tanto la evaluación debe ser continua, formativa y sumativa. La UDLA estipula la siguiente distribución porcentual para los reportes de evaluaciones previstas en cada semestre de acuerdo al calendario académico:

| | |
|-----------------------|-----|
| Reporte de progreso 1 | 35% |
| Exposiciones | 20% |
| Avance de proyecto | 15% |
| Reporte de progreso 2 | 35% |
| Exposiciones | 20% |
| Avance de proyecto | 15% |
| Evaluación final | 30% |
| Examen | 20% |
| Proyecto final | 10% |

Al finalizar el curso habrá un examen de recuperación para los estudiantes que, habiendo cumplido con más del 80% de asistencia presencial a clases, deseen reemplazar la nota de un examen anterior (ningún otro tipo de evaluación). Este examen debe integrar todos los conocimientos estudiados durante el periodo académico, por lo que será de alta exigencia y el estudiante necesitará prepararse con rigurosidad. La nota de este examen reemplazará a la del examen que sustituye. Recordar que para rendir el EXAMEN DE RECUPERACIÓN, es requisito que el estudiante haya asistido por lo menos al 80% del total de las sesiones programadas de la materia. No se podrá sustituir la nota de un examen previo en el que el estudiante haya sido sancionado por una falta grave, como copia o deshonestidad académica.

6. Metodología del curso y de mecanismos de evaluación.

Las metodologías y mecanismos de evaluación deben explicarse en los siguientes escenarios de aprendizaje:

6.1. Escenario de aprendizaje presencial.

Exposiciones de conceptos sobre arquitectura empresarial y estilos arquitecturales en las que los estudiantes deben explicar la aplicación práctica de lo aprendido.

6.2. Escenario de aprendizaje virtual.

Investigación de material adicional en publicaciones en línea relacionadas con la mejora continua del proceso de software.

6.3. Escenario de aprendizaje autónomo.

Preparación para exposición de conceptos, lecturas del material bibliográfico previo a cada clase, desarrollo del proyecto del curso.

7. Temas y subtemas del curso.-

| RDA | Tema | Subtemas | |
|--|--|----------|--|
| Desarrolla soluciones de software de nivel empresarial con el uso de estilos arquitecturales. | Arquitectura de aplicaciones | 1.1 | Introducción a la Arquitectura Empresarial |
| | | 1.2 | Fundamentos de Arquitectura de Software |
| | | 1.3 | Estilos Arquitecturales |
| | | 1.4 | Arquitectura N-Capas orientada al dominio |
| | Aplicación de Patrones de diseño | 2.1 | Revisión de patrones de diseño |
| | | 2.2 | Patrones de creación |
| | | 2.3 | Patrones estructurales |
| | | 2.4 | Patrones de comportamiento |
| | Herramientas para apoyar la gestión de la calidad del software | 3.1 | Áreas de Proceso de CMMi |
| | | 3.2 | Área de Proceso Métricas y Análisis MA |
| | | 3.3 | Aseguramiento de la Calidad de Productos y Procesos PPQA |
| | | 3.4 | Construcción de Pruebas Unitarias y de componentes |
| | | 3.5 | Construcción de Pruebas de Carga |
| | | 3.6 | Pruebas de aceptación de usuario |
| Analiza los conceptos de mejoramiento continuo del proceso del software en las categorías de soporte y administración de procesos. | Gestión de la Configuración del Software | 4.1 | Área de Proceso Configuration Management CM |
| | | 4.2 | Fundamentos de Software Configuration Management |
| | | 4.3 | Uso de herramientas para Software Configuration Management (SCM) |
| | Integración Continua | 5.1 | Fundamentos de Integración Continua |
| | | 5.2 | Integración Continua de Bases de Datos |
| | | 5.3 | Herramientas para aplicar Integración Continua |
| | | 5.4 | Testing e Inspección Continua |
| | Mejoramiento Continuo del Proceso de Software | 6.1 | El proceso de mejora continua |
| | | 6.2 | Modelo de madurez de CMMI |
| | | 6.3 | Categorías de Áreas de Proceso de CMMI |
| | | 6.4 | Áreas de Proceso para Administración de Procesos |

8. Planificación secuencial del curso

| # RDA | Tema | Subtemas | | Actividad/ metodología/clase | Tarea/ trabajo autónomo | MdE/Producto/fec ha de entrega |
|----------------|---|----------|--|---|--|--|
| Semana 1 a 3 | | | | | | |
| #1 | 1. Arquitectura de aplicaciones | 1.1 | Introducción a la Arquitectura Empresarial | (1) Clases magistrales (1) Análisis de casos | - Lectura Capítulos 1,2: (Llorente, 2010) - Exposiciones | - Presentación Fecha de entrega: 21-sep-2015. |
| | | 1.2 | Fundamentos de Arquitectura de Aplicaciones | | | |
| | | 1.3 | Estilos Arquitecturales | | | |
| | | 1.4 | Arquitectura N-Capas orientada al dominio | | | |
| Semana 4,5 | | | | | | |
| #1 | 2. Aplicación de patrones de diseño | 2.1 | Que son los patrones de diseño | (1) Clases magistrales (1) Análisis de casos - Guía en proyecto | - Lectura capítulo 1 (Freeman, 2004) - Proyecto de programación. | - Presentación proyecto. Fecha de entrega: 12-oct-2015 |
| | | 2.2 | Patrones de creación | | | |
| | | 2.3 | Patrones estructurales | | | |
| | | 2.4 | Patrones de comportamiento | | | |
| Semana 6 a 9 | | | | | | |
| #1 | 3. Herramientas para apoyar la gestión de la calidad del software | 3.1 | Áreas de Proceso de CMMi | (1) Clases magistrales (1) Análisis de casos (1) Guía en proyecto (1) Prueba | - Lectura Áreas de Proceso: (SEI, 2011) - Lectura Capítulos 1 a 5: (Oshero, 2009) - Proyecto de programación | - Presentación - Presentación proyecto Fecha de entrega: 9-nov-2015 |
| | | 3.2 | Área de Proceso Métricas y Análisis MA | | | |
| | | 3.3 | Aseguramiento de la Calidad de Productos y Procesos PPQA | | | |
| | | 3.4 | Construcción de Pruebas Unitarias y de componentes | | | |
| | | 3.5 | Construcción de Pruebas de Carga | | | |
| | | 3.6 | Pruebas de aceptación de usuario | | | |
| Semana 10 a 12 | | | | | | |
| #2 | 4. Gestión de la Configuración del Software | 4.1 | Área de Proceso CM Configuration Manager | (1) Clases magistrales (1) Presentación de herramientas | - Lectura Área de Proceso CM (SEI, 2011) - Exposiciones | - Presentación. Fecha de entrega: 16-nov-2015 |
| | | 4.2 | Fundamentos de Software Configuration Management | | | |
| | | 4.3 | Uso de herramientas para Software Configuration Management (SCM) | | | |
| Semana 13, 14 | | | | | | |
| #2 | 5. Integración Continua | 5.1 | Fundamentos de Integración Continua | (1) Clases magistrales (1) Presentación de herramientas | - Lectura capítulos 1,2, 5 a 7 (Duvall, 2007) - Exposiciones | - Presentación. Fecha de entrega: 14-dic-2015 |
| | | 5.2 | Integración Continua de Bases de Datos | | | |
| | | 5.3 | Herramientas para aplicar Integración Continua | | | |
| | | 5.4 | Testing e Inspección Continua | | | |
| Semana 15, 16 | | | | | | |
| #2 | Mejoramiento Continuo del Proceso de Software | 6.1 | El proceso de mejora continua | (1) Clases magistrales (1) Análisis de casos | - Lectura Área de Proceso de Administración de Procesos (SEI, 2011) | - Presentación. - Presentación Proyecto Fecha de entrega: |
| | | 6.2 | Modelo de madurez de CMMI | | | |

| | | | | | | |
|--|--|-----|--|--|--|------------|
| | | | | | | 4-ene-2016 |
| | | 6.3 | Categorías de Áreas de Proceso de CMMI | | | |
| | | 6.4 | Áreas de Proceso para Administración de Procesos | | | |

9. Normas y procedimientos del aula

Ingeniería de software II, es una materia que permite al estudiante desarrollar competencias para diseñar, implementar, certificar, desplegar mantener y mejorar arquitecturas de software de nivel empresarial.

- Cualquier caso que esté en contra de la honestidad académica será reportado a las autoridades de la universidad.
- No se recibirán trabajos fuera de la plataforma virtual.
- Se recibirán trabajos retrasados hasta 48 horas luego del plazo con una penalidad en la nota.
- En el caso de inasistencia, es responsabilidad del estudiante estar atento a las tareas para la siguiente clase.

10. Referencias bibliográficas.

10.1 Principales

- Llorente, C. y otros (2010). *Guía de Arquitectura N-Capas orientada al dominio con .Net* 4.0. España: Krasis Press.
- Software Engineering Institute. (2010). *CMMI for Development Version 1.3*. Pittsburg, USA: Carnegie Mellon University.

10.2 Referencias complementarias

- Freeman, E. (2004). *Head First Design Patterns*. USA: Shroff O Reilly.
- Osherove, R. (2009). *The Art of Unit Testing*. Geenwich, USA: Manning Publications Co.
- Duvall, P. (2007). *Continuous Integration. Improving Software Quality and Reducing Risk*. Boston, USA: Addison Wesley.

11. Perfil del docente

Nombre de docente: Christian Bastidas Romero

“Maestría en Dirección de Empresas MBA 2014, Universidad Andina Simón Bolívar, Ing. en Sistemas de Computación e Informática en Escuela Politécnica Nacional, 1998. Experiencia en la industria del desarrollo de software y la definición de estilos arquitecturales principalmente en software financiero. Certificado PMP (Project Management Professional) hasta 2016 y CSM (Certified Scrum Master) hasta 2016”.

Contacto: c.bastidas@udlanet.ec. Teléfono 2506710 Ext. 138.

Horario de atención al estudiante: lunes, martes a partir de las 18h30