

**Facultad de Ingeniería y Ciencias Agropecuarias**  
**Ingeniería en Sistemas de Computación e Informática**  
**ACI961 - Ingeniería de Software II**  
Período: 2018-1

**A. Identificación.-**

Número de sesiones:	48
Número de horas:	48 h presenciales + 72 h de aplicación del aprendizaje y estudio autónomo = 120 h totales
Créditos:	4.5
Profesor:	CHRISTIAN ANIBAL BASTIDAS ROMERO
Correo electrónico del docente:	christian.bastidas@udla.edu.ec
Director:	MARCO ANTONIO GALARZA CASTILLO
Campus:	Queri
Pre-requisito:	ACI920 Ingeniería de Software I
Paralelo:	1

**B. 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.

**C. Resultados de aprendizaje (RdA) del curso:**

1. Desarrolla soluciones de software de nivel empresarial con el uso de estilos arquitecturales.
2. Analiza los conceptos de mejoramiento continuo del proceso del software en las categorías de soporte y administración de procesos.

**D. Sistema y mecanismo 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:

<b>Reporte de Progreso 1</b>	<b>25%</b>
Actividades en clase:	
(Participación en Talleres, Exposiciones)	10%
Actividades autónomas:	5%
(Investigaciones, Informes de talleres)	
Evaluación continua: (Proyecto semestral, pruebas)	10%
<b>Reporte de Progreso 2</b>	<b>35%</b>
Actividades en clase:	
(Participación en Talleres, Exposiciones)	12.5%
Actividades autónomas:	10%
(Investigaciones, informes de talleres)	
Evaluación continua: (Proyecto semestral, pruebas)	12.5%

### Progreso 3

40%

Actividades en clase:

(Participación en Talleres, Exposiciones) 15%

Actividades autónomas: 10%

(Investigaciones, informes de talleres)

Evaluación continua: (Proyecto semestral, examen) 15%

### E. Asistencia

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 una evaluación anterior (el de mayor peso dentro de los componentes). 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 evaluación 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.

### F. Metodología del curso

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

#### 6.1 Escenario de aprendizaje presencial.

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

#### 6.2 Escenario de aprendizaje virtual.

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

#### 6.3 Escenario de aprendizaje autónomo.

1. Preparación para exposición de conceptos
2. Lecturas del material bibliográfico previo a cada clase
3. Desarrollo del proyecto del curso.

### G. Planificación alineada a los RdA

Planificación	Fechas	RdA 1	RdA 2
<b>Unidad o Tema</b>			
1. 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			
1.5. Microservicios			
	<b>Semanas 1-3</b>	X	

<b>Lecturas</b>			
<p>Daya, S. y otros (2015). Microservices from Theory to Practice. USA: IBM Redbooks. Páginas 3 - 38</p> <p>Llorente, C. y otros (2010). Guía de Arquitectura N-Capas orientada al dominio con .Net 4.0. España: Krasis Press. Páginas 9 – 57.</p>		X	
<b>Actividades</b>			
<p>1. Exposiciones en clases</p> <p>2. Discusiones grupales en clases</p>	Cada semana	X	
<b>Evaluaciones</b>			
1. Exposiciones	1. Cada semana	X	
<p><b>Unidad o Tema</b></p> <p>2. Arquitectura en entornos ágiles</p> <p>2.1. Patrones de arquitectura para entornos ágiles</p> <p>2.2. El rol del arquitecto en entornos ágiles</p> <p>2.3. Técnicas y Herramientas de apoyo para la arquitectura</p>	<b>Semanas 4-5</b>	X	
<b>Lecturas</b>			
Fontdevila, D y Salías, M. (2013). Software Architecture in the Agile Life Cycle. The Architecture Journal, Microsoft. Páginas 13 – 16		X	
<b>Actividades</b>			
<p>1. Exposiciones en clases</p> <p>2. Discusiones grupales en clases</p>		X	
<b>Evaluaciones</b>			
<p>1. Exposición</p> <p>2. Presentación tema del proyecto semestral</p>	1. En la semana planificada	X	
<p><b>Unidad o Tema</b></p> <p>3. Herramientas para apoyar la gestión de la calidad del software</p> <p>3.1. Áreas de Proceso de CMMi</p> <p>3.2. Área de Proceso Métricas y Análisis MA</p> <p>3.3. Aseguramiento de la Calidad de Productos y Procesos PPQA</p> <p>3.4. Agile Testing</p> <p>3.5. Construcción de Pruebas Unitarias y de componentes</p> <p>3.6. Pruebas de aceptación de usuario</p>	<b>Semanas 6-9</b>	X	
<b>Lecturas</b>			
<p>Software Engineering Institute. (2011). CMMI for Development Version 1.3. Pittsburg, USA: Carnegie Mellon University. Páginas 175 – 190, 301 – 306,</p> <p>Osherove, R. (2009). The Art of Unit Testing. Geenwich, USA: Manning Publications Co. Páginas 49 – 169.</p>		X	
<b>Actividades</b>			
<p>1. Exposiciones en clases</p> <p>2. Avance proyecto de arquitectura de software</p>		X	
<b>Evaluaciones</b>			
<p>1. Exposición</p> <p>2. Presentación de avance del proyecto</p>	1. Cada semana	X	
<b>Unidad o Tema</b>	<b>Semanas 10 - 12</b>		X

4. Gestión de la Configuración del Software 4.1. Área de Proceso CM Configuration Manager 4.2. Fundamentos de Software Configuration Management 4.3. Uso de herramientas para Software Configuration Management (SCM)			
<b>Lecturas</b>			
Software Engineering Institute. (2011). CMMI for Development Version 1.3. Pittsburg, USA: Carnegie Mellon University. Páginas 137 – 147			X
<b>Actividades</b>			
1. Exposiciones en clases 2 2. Proyecto de arquitectura de software			X
<b>Evaluaciones</b>			
1. Exposición 2. Presentación de Avance del Proyecto	1. Cada semana		X
<b>Unidad o Tema</b> 5. 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. Fundamentos de DevOps y Continuous Delivery	<b>Semana 13 – 14</b>		X
<b>Lecturas</b>			
Duvall, P. (2007). Continuous Integration. Improving Software Quality and Reducing Risk. Boston, USA: Addison Wesley. Páginas 23 – 44.			X
<b>Actividades</b>			
1. Exposiciones en clases 2. Discusiones grupales en clases			X
<b>Evaluaciones</b>			
1. Exposición 2. Participación en discusiones en clase	1. Cada semana		X
<b>Unidad o Tema</b> 6. 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	<b>Semana 13 – 14</b>		X
<b>Lecturas</b>			
Software Engineering Institute. (2011). CMMI for Development Version 1.3. Pittsburg, USA: Carnegie Mellon University. Páginas 39 – 43.			X
<b>Actividades</b>			
1. Exposiciones en clases 2. Proyecto de arquitectura de software			X
<b>Evaluaciones</b>			
1. Exposición	1. Cada semana		X

2. Presentación Proyecto			
--------------------------	--	--	--

## H. Normas y procedimientos del aula

Solo se recibirán trabajos en el aula virtual y dentro del plazo establecido.

Se tomará lista en los primeros 10 minutos iniciada la clase si el estudiante llega después, podrá ingresar de forma silenciosa, pero no se registrará la asistencia sin excepción.

Se tomará lista en los últimos 10 minutos de la clase si el estudiante sale antes de tomar lista no se registrará la asistencia sin excepción.

Los estudiantes deberán practicar la “honestidad académica” para todas las actividades de esta asignatura. La copia de ejercicios, exámenes, proyectos, y todas las actividades de aprendizaje solicitadas por el docente, y se calificará con la mínima calificación (uno).

Se acepta el uso de cualquier dispositivo electrónico (iPad, tablets, celulares, audífonos) únicamente con fines académicos. El uso para fines no académicos equivaldrá a una inasistencia.

No se podrán ingresar alimentos al aula

En el caso de inasistencia es responsabilidad del estudiante igualarse en los contenidos de la materia dictada en dicha clase.

En el caso de que un estudiante falte a una sesión en la que se realicen pruebas o prácticas de laboratorio, no se podrán recuperar las calificaciones. Las fechas de las evaluaciones serán publicadas en el apoyo virtual de la materia.

## I. Referencias bibliográficas.

### 1. Principales

- Software Engineering Institute. (2011). *CMMI for Development Version 1.3*. Pittsburg, USA:

Carnegie Mellon University.

- Daya, S. y otros (2015). *Microservices from Theory to Practice*. USA: IBM Redbooks.

- Duvall, P. (2007). *Continuous Integration. Improving Software Quality and Reducing Risk*. Boston, USA: Addison Wesley.

### 2. Referencias complementarias

- Llorente, C. y otros (2010). *Guía de Arquitectura N-Capas orientada al dominio con .Net 4.0*. España: Krasis Press.

- Crispin, L. y Gregory, J. (2009). *Agile Testing. A Practical Guide for Testers and Agile Teams*. Boston, USA: Addison Wesley.

- Freeman, E. (2004). *Head First Design Patterns*. USA: Shroff O Reilly.

- Osheroove, R. (2009). *The Art of Unit Testing*. Geenwich, USA: Manning Publications Co.

- Fontdevila, D y Salías, M. (2013). *Software Architecture in the Agile Life Cycle*. The Architecture Journal, Microsoft.

## J. 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 2019 y CSM (Certified Scrum Master) hasta 2017, CSP (Certified Scrum Professional) hasta 2017”.

Contacto: [c.bastidas@udlanet.ec](mailto:c.bastidas@udlanet.ec) Teléfono 2506710 Ext. 4009.

Horario de atención al estudiante: lunes y viernes a partir de las 18h30