

FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS
INGENIERÍA EN SISTEMAS DE COMPUTACIÓN E INFORMÁTICA
ACI860/Desarrollo de software III
Periodo 2017 – 1

1. Identificación.

Número de sesiones: 48
Número total de hora de aprendizaje: 120
Créditos – malla actual: 3
Profesor: Carlos Alberto Balladares Enríquez
Correo electrónico del docente (UDLANET): c.balladares@udlanet.ec
Coordinador: Marco Antonio Galarza Castillo
Campus: Queri
Pre-requisito: Desarrollo de software II
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				
Fundamentos teóricos	Praxis profesional	Epistemología y metodología de la investigación	Integración de saberes, contextos y cultura	Comunicación y lenguajes
	X			

2. Descripción del curso.-

Proporcionar a los alumnos métodos, herramientas, técnicas y habilidades para el desarrollo de aplicaciones empresariales en concordancia con las fases de implementación y pruebas del ciclo de vida del software.

3. Objetivo del curso.-

Desarrollar aplicaciones empresariales en concordancia con las fases de implementación y pruebas del ciclo de vida del software utilizando métodos, herramientas, técnicas y habilidades para la construcción de aplicaciones empresariales.

4. Resultados de aprendizaje deseados al finalizar el curso.-

Resultados de aprendizaje (RDA)	RDA perfil de egreso de Carrera	Nivel de dominio (carrera)
1. Explica los conceptos de una metodología de desarrollo de software, en la fase de implementación y pruebas.	Aplica con destreza las fases del ciclo de vida de desarrollo de soluciones informáticas (concepción, diseño, desarrollo, operación y evaluación), basándose en principios de ingeniería y estándares de calidad..	I ____ M ____ X ____ F ____
2. Utiliza los artefactos generados en la fase de diseño de una metodología de desarrollo, para realizar la implantación de una aplicación empresarial.		I ____ M ____ X ____ F ____
3. Aplica las buenas prácticas y estándares en la fase de implementación y pruebas de una metodología para el proceso de desarrollo de software.		I ____ M ____ X ____ F ____

1. 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 Sub componentes	35%
Reporte de progreso 2 Sub componentes	35%
Evaluación final Sub componentes (si los hubiese)	30%

PROGRESO 1:

20 %: Control de lecturas académicas. Avances caso de estudio. Tareas de Investigación y laboratorios

5 %: Control de lecturas de tecnología y exposición libre

10 %: Prueba

35 %

PROGRESO 2:

20 %: Control de lecturas académicas. Avances caso de estudio. Tareas de Investigación y laboratorios

5 %: Control de lecturas de tecnología y exposición libre

10 %: Prueba

35 %

EVALUACION FINAL:

20 %: Caso de estudio final

10 %: Prueba

30 %

Para la calificación se generarán las siguientes rubricas:

- Rúbrica de control de trabajo de investigación
- Rúbrica de control de lecturas académicas.
- Rúbrica de Avances caso de estudio.
- Rúbrica de culminación de laboratorio.
- Rubrica de trabajo final.

(Se adjunta documento de rúbricas)

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.

2. Metodología del curso y de mecanismos de evaluación.-

A través del planteamiento de un caso de estudio real, se analizarán los diferentes aspectos de ingeniería de software en cada fase.

2.1. Escenario de aprendizaje presencial.

El docente realizará la explicación de los temas principales ejemplificando con situaciones cotidianas o comunes al caso de la teoría que se estudie. Se exhortará a los alumnos para que realicen sus observaciones, aporten con ideas, y de manera grupal obtener conclusiones.

El estudiante realizará el desarrollo de las actividades definidas en cada clase, y deberá presentar el resultado del avance del problema planteado. Se realizarán presentaciones de defensa de investigaciones y avance de los casos de estudio, además, se realizarán laboratorios en clase con el manejo de herramientas útiles para la gestión de la ingeniería de Software.

2.2. Escenario de aprendizaje virtual

El estudiante realizará lecturas, e investigación relacionados a los temas tratados, de las cuales se podrán realizar actividades de control de lectura a través de la plataforma del aula virtual.

2.3. Escenario de aprendizaje autónomo.

Adicional a las lecturas de los temas relacionados con la materia, y con la finalidad de impulsar el espíritu de investigación, para cada progreso, cada estudiante deberá presentar un tema de investigación novedoso concerniente a Tecnologías de la Información.

Además, con la finalidad de impulsar la lectura, en todas las clases se realizarán preguntas respecto al segmento de tecnología de los días domingo del diario el Comercio u otras lecturas que se indique en clase.

Finalmente, el desarrollo de un caso de estudio a lo largo de toda la materia.

7. Temas y subtemas del curso.-

RDA	Temas	Subtemas
1. Explica los conceptos de una metodología de desarrollo de software, en la fase de implementación y pruebas.	1. El papel de la implementación y de las pruebas en el ciclo de desarrollo de software	1.1 UML 1.2 Casos de Uso 1.3 Proceso unificado de desarrollo de software
3. Aplica las buenas prácticas y estándares en la fase de implementación y pruebas de una metodología para el proceso de desarrollo de software.	2. Implementación en proceso de desarrollo de software	2.1 Introducción 2.2 El papel de la implementación en el ciclo de vida del software 2.3 Artefactos 2.4 Trabajadores 2.5 Flujo de trabajo
	3. Diseño de la arquitectura	3.1 Arquitectura del software 3.2 Géneros arquitectónicos 3.3 Estilos Arquitectónicos 3.4 Diseño arquitectónico 3.5 Evaluación de los diseños alternativos para la arquitectura
	4. Diseño en el nivel de componentes	4.1 Componentes 4.2 Diseño de componentes basados en clase 4.3 Realización del diseño en el nivel de componentes 4.4 Diseño de componentes para webapps 4.5 Diseño de componentes tradicionales 4.6 Desarrollo basado en componentes
	5. Diseño de la interfaz de usuario	5.1 Las reglas doradas 5.2 Análisis y diseño de la interfaz de usuario 5.3 Análisis de interfaz 5.4 Etapas del diseño de la interfaz 5.5 Diseño de una interfaz para webapps 5.6 Evaluación del diseño.

	6. Diseño basado en patrones	6.1 Patrones de diseño 6.2 Diseño de software basado en patrones 6.3 Patrones arquitectónicos 6.4 Patrones de diseño en el nivel de componentes 6.5 Patrones de diseño de la interfaz de usuario 6.6 Patrones de diseño de webapps
2. Utiliza los artefactos generados en la fase de diseño de una metodología de desarrollo, para realizar la implantación de una aplicación empresarial.	7. Aplicaciones empresariales	7.1 Introducción a Java EE 7.2 Enterprise Java Beans 7.3 Java Persistence API 7.4 Servlets y JSPs 7.5 Java Server Face 7.6 Seguridad JEE
3. Aplica las buenas prácticas y estándares en la fase de implementación y pruebas de una metodología para el proceso de desarrollo de software.	8. Pruebas en el proceso de desarrollo	8.1 Introducción 8.2 El papel de las pruebas en el ciclo de vida del software 8.3 Artefactos 8.4 Trabajadores 8.5 Flujo de trabajo
	9. Pruebas de software	9.1 Pruebas de desarrollo 9.2 Desarrollo dirigido por pruebas 9.3 Pruebas de versión 9.4 Pruebas de usuario
2. Utiliza los artefactos generados en la fase de diseño de una metodología de desarrollo, para realizar la implantación de una aplicación empresarial.	10. Aplicaciones empresariales	10.1 Desarrollo de una aplicación empresarial.

8. Planificación secuencial del curso.-

RdA	Temas	Subtemas	Actividad/ metodología/clase	Tarea/ trabajo autónomo	MdE/Producto/fecha de entrega
Semana 1 – 6					
1	1. El papel de la implementación y de las pruebas en el ciclo de desarrollo de software	1.1 UML 1.2 Casos de Uso 1.3 Proceso unificado de desarrollo de software		-A: Investigación Implementación del proceso unificado de Software -B: Lectura capítulo 1,2, 3 Graig Larman -C: Definición del alcance del caso de estudio del proyecto de desarrollo de software	Rúbrica trabajo investigación progreso 1/ A / 16/09/2016 Rúbrica control de lecturas B / 16/09/2016 Rúbrica de avance caso de estudio C / 16/09/2016
3	2.Implementación en proceso de desarrollo de software	2.1 Introducción 2.2 El papel de la implementación en el ciclo de vida del software 2.3 Artefactos 2.4 Trabajadores 2.5 Flujo de trabajo		-A: Lectura capítulo 9 Pressman - B: Lectura capítulo 7 tema 7.1 Sommerville - C: Invesetigación UML	Rúbrica control de lecturas A / 23/09/2016 B / 23/09/2016 Rúbrica trabajo investigación progreso 1 C / 23/09/2016
	3. Diseño de arquitectura	3.1 Arquitectura del software 3.2 Géneros arquitectónicos 3.3 Estilos Arquitectónicos 3.4 Diseño aruitectónico 3.5 Evaluación de los diseños alternativos para la arquitectura		- A: Lectura capítulo 10 Pressman B: -Lectura capítulo 7 tema 7.3, 7.4 Sommerville - C: Presentación técnica del caso de estudio (depende de desarrollo de software II)	Rúbrica control de lecturas A / 30/09/2016 B / 30/09/2016 Rúbrica de avance caso de estudio C / 30/09/2016

4. Diseño en el nivel de componentes	<p>4.1 Componentes</p> <p>4.2 Diseño de componentes basados en clase</p> <p>4.3 Realización del diseño en el nivel de componentes</p> <p>4.4 Diseño de componentes para webapps</p> <p>4.5 Diseño de componentes tradicionales</p> <p>4.6 Desarrollo basado en componentes</p>	<p>- A: Lectura capítulo 11 Pressman</p> <p>B: -Lectura capítulo 7 tema 7.2 Sommerville</p> <p>-</p>	<p>Rúbrica control de lecturas</p> <p>A / 07/10/2016</p> <p>B / 07/10/2016</p> <p>C / 07/10/2016</p>
5. Diseño de la interfaz de usuario	<p>5.1 Las reglas doradas</p> <p>5.2 Analisis y diseño de la intrfaz de usuario</p> <p>5.3 Análisis de interfaz</p> <p>5.4 Etapas del siseño de la interfaziones</p> <p>5.5 Diseño de una interfaz para webapps</p> <p>5.6 Evaluación del diseño.</p>	<p>- A: Lectura capítulo 12 Pressman</p> <p>B: -Lectura capítulo 7 tema 7.2 Sommerville</p> <p>C: Lectura capítulo 16 Graig Larman</p> <p>- D: Investigación Maven</p> <p>- E: Investigación Glasfish</p>	<p>Rúbrica control de lecturas</p> <p>A / 14/10/2016</p> <p>B / 14/10/2016</p> <p>C / 14/10/2016</p> <p>Rúbrica trabajo investigación progreso 1/</p> <p>D / 14/10/2016</p> <p>E / 14/10/2016</p>
6. Diseño Basado en Patrones	<p>6.1 Patrones de diseño</p> <p>6.2 Diseño de software basado en patrones</p> <p>6.3 Patrones arquitectónicos</p> <p>6.4 Patrones de diseño en el nivel de componentes</p> <p>6.5 Ptrones de diseño de la interfaz de usuario</p> <p>6.6 Patrones de diseño de webapps</p>	<p>- A : Lectura capítulo 20 Graig Larman</p> <p>- B: Investigación de aplicaciones empresariales</p> <p>- C: Investigación MySql</p>	<p>Rúbrica control de lecturas</p> <p>A / 21/10/2016</p> <p>Rúbrica trabajo investigación progreso 1/</p> <p>B / 21/10/2016</p> <p>C/ 21/10/2016</p> <p>Exposición tema libre progreso1:</p> <p>C / 21/10/2016</p> <p>Control de lecturas de tecnología progreso 1</p> <p>D/ 21/10/2016</p> <p>Prueba progreso 1 / 21/10/2016</p>

Semana 7 - 12					
2	7. Aplicaciones empresariales	7.1 Introducción a Java EE	<ul style="list-style-type: none"> - (1) Instrucción directa - (1) Foros y Debates - (1) Prácticas - (2) Portafolio del estudiante - (3) Proyectos y exposiciones - (3) Lecturas y videos - (3) Evaluaciones 	-A: Laboratorio 1: Preparación del ambiente de desarrollo. - B: Laboratorio 2 : Primer ejercicio con Java EE y Maven C: Documentación técnica del caso de estudio (análisis y diseño) D: Investigación Enterprise Jaba Beans EJB	Rúbrica de culminación de laboratorios: A / 28/10/2016 B / 28/10/2016 Rúbrica trabajo investigación progreso 2 D / 28/10/2016
		7.2 Enterprise Java Beans		A: Laboratorio Enterprise Java Beans B: Laboratorio Enterprise java Beans Sesión C: Investigación Persistencia java, JPA D: Implementación de EJB con caso de estudio	Rúbrica de culminación de laboratorios: A / 11/11/2016 B / 11/11/2016 Rúbrica trabajo investigación progreso 2 C / 11/11/2016 Rúbrica avance caso de estudio D / 11/11/2016
		7.3 Java Persistencia API		A: Laboratorio Persistencia con JPA B: Laboratorio Instalación de Hibernate Tools C: Laboratorio Ingeniería Inversa JPA D: Investigación Servlets y JSPs E: Implementación JPA caso de estudio	Rúbrica de culminación de laboratorios: A / 18/11/2016 B / 18/11/2016 C / 18/11/2016 Rúbrica trabajo investigación progreso 2 D / 18/11/2016 Rúbrica avance caso de estudio E / 18/11/2016
		7.4 Servlets y JPSs		A: Laboratorio con Servlets, EJB y JPA. B: Investigación de Java Server Faces C: Implementación de Servlets caso de estudio	Rúbrica de culminación de laboratorios: A / 25/11/2016 Rúbrica trabajo investigación progreso 2 B / 25/11/2016 Rúbrica avance caso de estudio C / 25/11/2016

		7.5 Java Server Face		A: Laboratorios con JSF/Prime Faces, EJB y JPA B: Investigación Seguridad en Java C: Implementación de JSF caso de estudio	Rúbrica de culminación de laboratorios: A / 02/12/2016 Rúbrica trabajo investigación progreso 2 B / 02/12/2016 Rúbrica avance caso de estudio C / 02/12/2016
		7.6 Seguridad Java		A: Laboratorios con seguridad en JEE B: Implementación de seguridad caso de estudio	Rúbrica de culminación de laboratorios: A / 09/12/2016 Rúbrica avance caso de estudio B / 09/12/2016 Exposición tema libre progreso 2: 09/12/2016 Control de lecturas de tecnología progreso 2 D / 09/12/2016 Prueba progreso 2 / 09/12/2016015
Semana 13 - 14					
3	8. Pruebas en el proceso de desarrollo	8.1 Introducción 8.2 El papel de las pruebas en el ciclo de vida del software 8.3 Artefactos 8.4 Trabajadores 8.5 Flujo de trabajo	- (1) Instrucción directa - (1) Foros y Debates - (2) Portafolio del estudiante - (3) Proyectos y exposiciones - (3) Lecturas y videos	A: Lectura de capítulo 8 Sommerfiel B: Presentación de primer borrador de l proyecto del caso de estudio	Rúbrica de lecturas A / 16/02/2016 Rúbrica avance caso de estudio B / 16/02/2016
	9. Pruebas de Software	9.1 Pruebas de desarrollo 9.2 Desarrollo dirigido por pruebas 9.3 Pruebas de versión 9.4 Pruebas de usuario		A: Investigación Aplicaciones empresariales .NET B: Presentación de segundo borrador de l proyecto del caso de estudio	Rúbrica de lecturas A / 06/01/2017 Rúbrica avance caso de estudio B / 06/01/2017

Semana 15					
2	10. Aplicaciones empresariales	10. Desarrollo de una aplicación empresarial	<ul style="list-style-type: none"> - (1) Instrucción directa - (1) Foros y Debates - (3) Evaluaciones 	A: Entrega 1 y defensa de la implementación del caso de estudio	Rubrica de proyecto Final A / 13/01/2017
2	10. Aplicaciones empresariales	10. Desarrollo de una aplicación empresarial	<ul style="list-style-type: none"> - (1) Instrucción directa - (1) Foros y Debates - (3) Evaluaciones 	A: Entrega 2 y defensa de la implementación del caso de estudio	Rubrica de proyecto Final A / 20/01/2017 Examen final 03/02/2017

9. Normas y procedimientos de clase.-

1. Se permitirá entregar una tarea hasta con 48 horas de retraso con una penalidad del 50% de la nota asignada. Salvo casos de fuerza mayor debidamente comprobados.
2. Se tomará lista dentro de los primeros 10 minutos luego de iniciado cada módulo, si el estudiante llega después, podrá ingresar de forma silenciosa, pero no se registrará la asistencia.
3. Los estudiantes deberán practicar la “honestidad académica” para todas las actividades de esta asignatura
4. El uso de cualquier dispositivo electrónico se aceptará en la clase solo para fines académicos. El uso para fines no académicos equivaldrá a una inasistencia
5. Todos los trabajos que realiza el estudiante debe contar con la evidencia correspondiente
6. El estudiante debe mantener la limpieza del aula evitando ingresar con alimentos.
7. El estudiante puede acceder a tutoría personal en los horarios establecidos por el docente.
8. En el caso de inasistencia es responsabilidad del estudiante igualarse en los contenidos de la materia dictada en dicha clase.
9. 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.

10. Referencias bibliográficas.-

10. 1 Principal:

Pressman, R. (2010). Ingeniería de Software (7ma edición), México: McGraw-Hill

10.2 Secundarias:

Sommerville, I. (2011). Ingeniería de software (9 na edición), Madrid: Pearson Educación

Larman, C. (2010). UML y Patrones (2 da edición), Madrid: Pearson Educación

McGovern, James, Adatia, Rahim, Fain, Yakov, (2011). Java2 Enterprise Edition 1. 4 (J2EE 1.4) Bible, Nueva York: John Wiley & Sons

Roman, E., Patel, R., Brose G. (2010). Mastering Enterprise JavaBeans (3 ra edición), John Wiley & Sons

11. Perfil del docente

Nombre de Docente: Ing. Carlos Balladares E., Msc.

“Ingeniero de Sistemas Informáticos y de Computación, Master en Gestión de la Comunicaciones y Tecnologías de la Información”

Trayectoria profesional con una amplia experiencia en la gestión de proyectos de tecnología. Ha tenido a cargo proyectos de desarrollo de software, proyectos de implantación de Infraestructura, y proyectos que han involucrado las dos áreas, y otras como son: levantamiento y rediseño de

procesos, levantamiento de carga laboral, etc. Además ha participado en la investigación, implantación y asesoría para la contratación de sistemas informáticos a la medida y de plataformas tecnológicas.

Actualmente, a parte de la docencia, es funcionarios público, que tiene a su cargo el equipo de desarrollo de software de la Agencia Nacional de Regulación y Control de Telecomunicaciones.

Contacto: c.balladares@udlanet.ec

Teléfono: 0998025475.