

FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS
INGENIERÍA EN SISTEMAS DE COMPUTACIÓN E INFORMÁTICA
ACI760/ DESARROLLO DE SOFTWARE II
 Período 2017-2

1. Identificación Sílabo Maestro

Número de sesiones: 48 sesiones
 Número total de horas de aprendizaje: 120 horas
 Créditos – malla actual: 3 créditos
 Profesor: Hugo Fernando Chimbo Acosta
 Correo electrónico del docente (Udlanet): hugo.chimbo@udlanet.ec
 Coordinador: Marco Antonio Galarza Castillo
 Campus: Queri
 Pre-requisito: Análisis de requerimientos (ACI480) Co-requisito: N/A
 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

Se estudiará los conceptos de desarrollo tradicional de software utilizando una metodología orientada a objetos y las fases para desarrollar sistemas de este tipo, partiendo del análisis de requerimientos hasta el diseño con los diagramas de casos, actividad, estado, clases, interacción, componentes y despliegue. Durante el curso se planteará la aplicación de toda la metodología hasta llegar a la fase de diseño de un proyecto de software.

3. Objetivo del curso

Desarrollar las fases de análisis y diseño de un proyecto de desarrollo de software bajo el paradigma orientado a objetos usando la metodología tradicional y herramientas CASE para que el estudiante comprenda las actividades de las fases de análisis y diseño de software.

4. Resultados de aprendizaje deseados al finalizar el curso

Resultados de aprendizaje (RdA)	RdA perfil de egreso de carrera	Nivel de desarrollo (carrera)
1. Explica los conceptos de una metodología de desarrollo de software, en las fases de análisis y diseño.	(3) 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.	Inicial () Medio (X) Final ()
2. Aplica las características de la metodología de desarrollo, los diagramas de modelamiento y los patrones de diseño en las fases de análisis y diseño del proceso de desarrollo de software.		Inicial () Medio (X) Final ()

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%
Lecciones Escritas	15%
Proyecto	10%
Portafolio	10%
Reporte de progreso 2	35%
Lecciones Escritas	10%
Presentaciones	5%
Proyecto	10%
Portafolio	10%
Evaluación final	30%
Proyecto Final	20%
Examen 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.

- **Lecciones Escritas 25%:** Son preguntas de elección múltiple que implican el estudio de cada tema de la materia. Un elemento de este mecanismo de evaluación podrá ser recuperada con el examen de recuperación
- **Portafolio - 10%:** Ejercicios y tareas prácticas semanales conjuntamente con el profesor: Conforman el portafolio que se desarrollará a lo largo del curso y deben ser enviados al apoyo virtual al finalizar cada semana. Las participaciones serán evaluadas de acuerdo a la rúbrica de ejercicios.
- **Examen final - 10%:** Son preguntas de elección múltiple que implican el estudio de toda la asignatura. Esta evaluación podrá ser recuperada con el examen de recuperación

6.2. Escenario de aprendizaje virtual.

- **Portafolio de Ejercicios - 5%:** El estudiante deberá resolver los ejercicios e investigaciones indicadas por el docente, ejercicios prácticos rendidos en clases y subirlos a la plataforma virtual.

6.3. Escenario de aprendizaje autónomo.

- **Portafolio - 5%:** Ejercicios y tareas prácticas semanales de trabajo autónomo: Conforman el portafolio que se desarrollará a lo largo del curso y deben ser enviados al apoyo virtual al finalizar cada semana. Las participaciones serán evaluadas de acuerdo a la rúbrica de ejercicios.
- **Proyecto Final - 40%:** Análisis y diseño de una aplicación definida en un caso de negocio presentado al inicio del semestre. Las actividades relacionadas serán desarrolladas utilizando una herramienta CASE seleccionada por cada grupo de manera tal que como resultado final se obtenga código generado automáticamente que permita tener un punto de partida para el desarrollo posterior con el que se pueda atender las necesidades planteadas en el caso de negocio.
El proyecto será realizado en grupos de 2 estudiantes. Se calificará de acuerdo a la rúbrica de cada componente del proyecto integrador.
- **Presentaciones - 5%:** El estudiante debe realizar presentaciones de un tema asignado, se debe realizar en grupos de 2 estudiantes. Se calificará de acuerdo a la rúbrica de presentaciones.

7. Temas y subtemas del curso

RdA	Temas	Subtemas
1. Explica los conceptos de una metodología de desarrollo de software, en las fases de análisis y diseño.	1. Análisis de sistemas de software	1.1 Modelos de proceso de software. 1.2 Metodología Rational Unified Process (RUP)
2. Aplica las características de la metodología de desarrollo, los diagramas de modelamiento y los patrones de diseño en las fases de análisis y diseño del proceso de desarrollo de software.	2. Modelamiento en la fase de análisis de sistemas de software	2.1 Modelamiento del dominio del problema. 2.2 Modelamiento de requerimientos y creación del modelo de requerimientos del sistema. 2.3 Modelamientos de casos de uso y de las especificaciones casos de uso de un sistema. 2.4 Modelamiento de las reglas de negocio de un sistema
	3. Modelamiento en la fase de diseño de sistemas de software	3.1 Modelamiento de actividades y estado 3.2 Modelamiento del diagrama de clases del sistema. 3.3 Estrategias de abstracción del diagrama de clases en función del modelo de entidad relación de un sistema. 3.4 Modelamiento interacción (diagramas de secuencia y colaborativo) 3.5 Modelamiento de componentes 3.6 Modelamiento de despliegue.

RdA	Temas	Subtemas
	4. Patrones de Diseño	4.1 Patrones de diseño 4.2 Patrones de diseño estructurales. 4.3 Patrones de diseño creacionales 4.4 Patrones de diseño de comportamiento
	5. Herramientas para desarrollo de sistemas de software	5.1 Herramientas CASE 5.2 Herramientas de diseño de aplicaciones

8. Planificación secuencial del curso

Semana 1- 4 (del 6 de marzo del 2017 al 31 de marzo del 2017)					
RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#1	1. Análisis de sistemas de software orientado a objetos	1.1 Modelos de proceso de software 1.2 Metodología Rational Unified Process (RUP)	(1) Instrucción directa (1) Trabajo en grupo (1) Evaluación	(3) Lectura del tema Pressman, R. (2015). Capítulo 7 (3) Realiza ejercicios propuestos de modelo de procesos de desarrollo de software Pressman, R. (2015). Capítulo 7 (3) Lectura de RUP Capítulo 2.3: Sommerville, I. (2011). (3) Resolución del caso: Uso de plantillas RUP Capítulo 2.3: Sommerville, I. (2011).	Cuestionario RUP Consulta de opciones de Herramientas CASE Open Source Fecha entrega: 17/03/2017 Portafolio de Ejercicios: documento en base a las plantillas RUP (Rubrica de Portafolio de ejercicios) Fecha entrega: 24/03/2017 Portafolio de Ejercicios: modelamiento del dominio del problema proyecto integrador (Rubrica de Portafolio de ejercicios) Fecha entrega: 31/03/2017 Proyecto: Propuesta de una aplicación – Elaboración del caso de negocio (Rubrica de primera entrega del proyecto) Fecha entrega: 31/03/2017

Semana 5 – 6 (del 3 al 14 de abril del 2017) (Considerar recuperación de feriado de Viernes Santo)

RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#2	2. Modelamiento en la fase de análisis de sistemas de software	<p>2.1 Modelamiento del dominio del problema.</p> <p>2.1 Modelamiento de requerimientos y creación del modelo de requerimientos del sistema.</p> <p>2.2 Modelamientos de casos de uso y de las especificaciones casos de uso de un sistema.</p> <p>2.3 Modelamiento de las reglas de negocio de un sistema</p>	<p>(1) Instrucción directa</p> <p>(1) Aprendizaje basado en problemas</p>	<p>(3) Realiza ejercicios propuestos de diagrama casos de uso. Revisión Pressman, R. (2015). Capítulo 9</p> <p>(3) Realiza ejercicios propuestos de las especificaciones de los casos de uso. Pressman, R. (2015). Capítulo 9</p> <p>(3) Realiza ejercicios propuestos de diseño individual de diagrama de negocios para un sistema de software orientada a objetos Pressman, R. (2015). Capítulo 9</p>	<p>Portafolio ejercicios: documento de modelo Requerimientos (Rubrica de Portafolio de ejercicios) Fecha entrega: 07/04/2017</p> <p>Portafolio ejercicios: diagramas de casos de uso (Rubrica de Portafolio de ejercicios) Fecha entrega: 14/04/2017</p> <p>Portafolio ejercicios: especificaciones de casos de uso basado en un problema (Rubrica de Portafolio de ejercicios) Fecha entrega: 14/04/2017</p> <p>Lección escrita Análisis de sistemas Fecha entrega: 14/04/2017</p>

Semana 7 – 13 (del 17 de abril del 2017 al 2 de junio del 2017)

RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#2	3. Modelamiento en la fase de diseño de sistemas de software	3.1 Modelamiento de actividades y estado	(1) Instrucción directa Pressman, R. (2015). Capítulo 11	(3) Realiza el Diseño de los diagramas de actividades y estados de la aplicación orientada a objetos	Portafolio ejercicios: diagramas de actividades y de estado (Rubrica de Portafolio de ejercicios) Fecha entrega: 21/04/2017
		3.2 Modelamiento del diagrama de clases del sistema.	(1) Aprendizaje basado en problemas	(3) Realiza el Diseño de los diagramas de clases de la aplicación orientada a objetos tomando en cuenta los patrones de diseño	Portafolio ejercicios: diagramas de clases basado en el problema propuesto (Rubrica de Portafolio de ejercicios) Fecha entrega: 05/05/2017
		3.3 Estrategias de abstracción del diagrama de clases en función del modelo de entidad relación de un sistema.	(1) Trabajo en grupo	Lectura previa Parte IV (Larman C. 2010).	Portafolio ejercicios: diagramas de componentes y despliegue de un sistema (Rubrica de Portafolio de ejercicios) Fecha entrega: 05/05/2017
		3.4 Modelamiento interacción (diagramas de secuencia y colaborativo)	(1) Lección Escrita.	(3) Preparan una presentación sobre los patrones de diseño de creación	Portafolio ejercicios: diagramas de secuencia y colaboración de un sistema (Rubrica de Portafolio de ejercicios) Fecha entrega: 19/05/2017
		3.5 Modelamiento de componentes			Presentación: Diapositivas y diseño de diagramas de ejemplo de los patrones de diseño de creación. (Rubrica de Presentación) Fecha entrega: 19/05/2017
		3.6 Modelamiento de despliegue.			
		4.1 Patrones de diseño			
	4. Patrones de diseño	4.2 Patrones de diseño estructurales.			

		4.3 Patrones de diseño creacionales 4.4 Patrones de diseño de comportamiento			<p>Proyecto: Documento de Especificación de Requerimientos de software de la aplicación seleccionada en la primera entrega. (Rubrica de 2da entrega Proyecto) Fecha entrega: 26/05/2017</p> <p>Lección escrita Diseño e implementación de un sistema de software orientado a objetos Fecha entrega: 02/06/2016</p>
--	--	---	--	--	--

Semana 14-16 (del 5 de junio de 2017 al 23 de junio del 2017)

RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#2	5. Herramientas para proyectos	5.1 Herramientas CASE 5.2 Herramientas de diseño de aplicaciones	(1) Trabajo en grupo: (1) Lección Escrita	(3) Entrega del proyecto final y documentos con los diagramas hechos en las herramientas	<p>Proyecto: Documento de Arquitectura del Software (diseño de la aplicación) y código generado por la herramienta CASE seleccionada (Rubrica de 3ra entrega Proyecto) Fecha entrega: 16/06/2017</p> <p>Lección escrita de análisis y diseño para un sistema de software. Fecha entrega: 23/06/2017</p>

Toda fecha de entrega de productos podrá ser modificada por necesidades de la asignatura, y previo acuerdo entre docente y estudiantes.

9. Normas y procedimientos para el aula

- Se permitirá entregar una tarea hasta con 48 horas de retraso con una penalidad del 50% de la nota asignada
- Se tomará lista dentro de los primeros 5 minutos luego de iniciado cada módulo, si el estudiante llega después, podrá ingresar de forma silenciosa, pero no se registrará la asistencia
- Los estudiantes deberán practicar la “honestidad académica” para todas las actividades de esta asignatura (ejercicios, exámenes, proyectos, y todas las actividades de aprendizaje solicitadas por el docente), caso contrario se calificará con la mínima calificación (cero).
- El uso de celulares, redes sociales, audífonos, Etc. (objetos distractores) y el consumo de alimentos deben ser siempre fuera del aula
- Se recibirán trabajos solo en aula virtual
- El estudiante puede acceder a tutoría personal en los horarios establecidos por el docente
- En el caso de inasistencia es responsabilidad del estudiante igualarse en los contenidos de la materia dictada en dicha clase
- Las calificaciones obtenidas en sesiones destinadas a pruebas o prácticas de laboratorio son exclusivas para los estudiantes que asistan a las mismas

10. Referencias bibliográficas

10.1. Principales.

Pressman, R. (2015). *Software Engineering – A Practitioner’s Approach*. (8th edition). USA: Mc.Graw Hill.

10.2. Referencias complementarias.

Kimmel, P. (2010). *Manual de UML*. (1era edición). Madrid, España: MCGRAW-HILL INTERAMERICANA EDITORES.

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

11. Perfil del docente

Nombre de docente:

Hugo Fernando Chimbo Acosta

Perfil:

Ingeniero en Sistemas (Escuela Politécnica Nacional), Magister en Gerencia de Sistemas (Universidad de las Fuerzas Armadas) y Profesional en la Gestión de Proyectos (PMP - Project Management Institute), con varias certificaciones en productos y marcos de trabajo referentes a TI, posee una experiencia profesional de más de 15 años en el ámbito de las Tecnologías de la Información, Infraestructura de Redes – Comunicaciones y

la Gestión de Proyectos de TI en empresas Públicas y Privadas de todo tamaño, gestionando equipos de trabajo altamente productivos para extraer el máximo beneficio del HW, SW y procesos existentes generando rentabilidad a los negocios. Docente de pre grado por 12 años en la Escuela Politécnica Nacional dictando materias como Sistemas Distribuidos, Redes, Interredes, Intranets y Extranets, Administración de Sistemas Operativos y Redes, Administración de Servidores Linux, Aplicaciones distribuidas, Comercio Electrónico. Instructor certificado en Infotraining en la línea de marcos de trabajo en la gestión de proyectos y sus herramientas de administración.

Contacto:

hugo.chimbo@udlanet.ec