

Facultad de Ingeniería y Ciencias Agropecuarias
Ingeniería en Sistemas de Computación e Informática
ACI-120 - Introducción a los Sistemas de Información
Período 2016-2

1. Identificación (*Sílabo maestro*)

Número de sesiones: 64
 Número total de hora de aprendizaje: 160h
 Créditos – malla actual: 4
 Profesor: Anita Yáñez
 Correo electrónico del docente (Udlanet): a.yanez@udlanet.ec
 Coordinador: Marco Galarza
 Campus: Queri
 Pre-requisito: Co-requisito:
 Paralelo: 1

Tipo de asignatura:

Optativa	
Obligatoria	X
Práctica	

Organización curricular:

Unidad 1: Formación Básica	X
Unidad 2: Formación Profesional	
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

Es la primera materia dentro de la malla curricular en la línea de desarrollo de software que ofrece una visión introductoria para el desarrollo de un sistema de información con una connotación teórico – práctica de la funcionalidad cliente/servidor. Se revisará las nuevas herramientas de programación como la programación en bloques, para desarrollar la lógica de los estudiantes, los paradigmas de la programación orientación a objetos y estructurada, y además de una breve revisión de la programación web.

3. Objetivo del curso

Identificar las principales características de los sistemas de información, los lenguajes de programación, las técnicas de programación y la arquitectura cliente/servidor para el desarrollo de software, a través de nuevas herramientas de programación para que el estudiante desarrolle la lógica de la programación.

4. Resultados de aprendizaje deseados al finalizar el curso

Resultados de aprendizaje (RdA)	RdA perfil de egreso de carrera	Nivel de desarrollo (carrera)
1. Identifica las principales características de los sistemas de información, los lenguajes de programación, las técnicas de programación y la arquitectura cliente/servidor para el desarrollo de software.	1. Aplica el enfoque sistémico en el análisis y resolución de problemas relacionados con las TIC.	Inicial (X) Medio () 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%

- a) Portafolio de Ejercicios 30%, se adjuntará rúbrica.
- b) Pruebas online 30%.
- c) Examen 40%

Reporte de progreso 2: 35%

- a) Portafolio de Ejercicios 30% se adjuntará rúbrica.
- b) Pruebas online 30%.
- c) Examen 40%

Evaluación final: 30%

- a) Portafolio de Ejercicios 20%, se adjuntará rúbrica.
- b) Proyecto Final 40%.
- c) Examen Final 40%

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.

Para el aprendizaje presencial en el curso se realizará:

1. *Instrucción directa*: Los estudiantes recibirán explicación directa de los temas de la clase.
2. *Trabajo grupal*: Los estudiantes realizarán trabajos en grupo dentro del aula de clases. Todos los trabajos deben presentar su bibliografía académica que sustente su contenido, la evaluación será de manera individual de acuerdo a la rúbrica respectiva. Todos los trabajos formarán parte del portafolio de ejercicios.
3. *Trabajo individual*: Los estudiantes realizarán trabajos en el laboratorio de PC, los cuales van a ser dirigidos por el docente, su entrega va a ser al final de la clase y su evaluación de acuerdo a la rúbrica respectiva. Todos los trabajos formarán parte del portafolio de ejercicios.
4. *Evaluaciones*: Los estudiantes deberán rendir una evaluación por cada tema expuesto en clase, estos exámenes serán en el aula virtual y no tomarán más de 10 min. Todos los trabajos formarán parte del portafolio de evaluaciones. Además deberán rendir un examen al final de cada período (cátedras).

6.2. Escenario de aprendizaje virtual.

Para el aprendizaje virtual en el curso se realizará:

1. *Portafolio de ejercicios*. Recopilación de Trabajos individuales de los estudiantes subidos al apoyo virtual.
2. *Foros*. El estudiante debe aportar con ideas a foros virtuales en el apoyo virtual. Todos los foros formarán parte del portafolio de ejercicios.

6.3. Escenario de aprendizaje autónomo.

Para el aprendizaje autónomo en el curso se realizará:

1. *Prácticas individuales en laboratorio*: Explicado el tema el docente, propone a los alumnos una actividad de aplicación de dichos contenidos, las tareas deben ser subidos al apoyo virtual. Todos los trabajos formarán parte del portafolio de ejercicios.
2. *Portafolio de ejercicios*: El estudiante práctica los conocimientos y destrezas adquiridos. Todos los trabajos formarán parte del portafolio de ejercicios.
3. *Auto-evaluaciones y evaluaciones*: Se utilizará la herramienta de co-evaluación del aula virtual, la cual permite evidenciar el nivel de aprendizaje de los estudiantes, desarrollando en ellos responsabilidad y autonomía en las tareas enviadas.
4. *Proyecto final*: Para la evaluación final se considera la realización de un proyecto en el que los estudiantes desarrollen un tema asignado en el cual puedan demostrar el aprendizaje de los conceptos revisados durante la materia. El proyecto final será evaluado por medio de una rúbrica.

1. Temas y subtemas del curso

RdA	Temas	Subtemas
Identifica las principales características de los	1. Introducción a los Sistemas de	1.1 Conceptos básicos
		1.2 Componentes

sistemas de información, los lenguajes de programación, las técnicas de programación y la arquitectura cliente/servidor para el desarrollo de software.	información (1 semana)	1.3 Tipos de Sistemas de información
	2. Álgebra booleana (1 semana)	2.1. Propositiones
		2.2 Conocimientos básicos de álgebra booleana.
		2.3 Tablas de verdad.
	3. Introducción a la programación (2 semana)	3.1 Historia y evolución de la programación.
		3.2. Algoritmia, pseudocódigo básico,
		3.3. Proceso de programación, programa fuente y programa compilado.
		3.4 Paradigmas de programación
		3.5. Lenguajes de programación
	4. Programación en bloques (2 semana)	4.1. Scratch
		4.2. App Inventor
		4.3. Blockly
	5. Introducción a la programación estructurada (3 semana)	5.1 Visión general del paradigma de la programación estructurada.
		5.2. Diagramas de flujo.
		5.3. Pseudocódigo.
		5.4. Revisión de algoritmos. Métodos formales para verificación de programas
	6. Introducción a la programación orientada a objetos. (2 semana)	6.1 Visión general del paradigma de la programación orientada a objetos.
		6.2 Conocimiento básico de las estructuras de control del lenguaje de programación java para implementar programas orientados a objetos.
	7. Introducción a los nuevos lenguajes de programación (1 Semana)	7.1. Ruby
		7.2. Python
	8. Introducción a aplicaciones Cliente-Servidor. (1 semana)	8.1 Programación HTML5
		8.2 Arquitectura cliente/ servidor
		8.2 Conocimiento de las herramientas para trabajar con formularios y servlets.

programación (2 semana)	3.2. Algoritmia, pseudocódigo básico	(1) Evaluación	Lectura del libro: Introducción a la Programación Lógica y Diseño (Farrell 2013, Capítulo 1)	Fecha: 29-03-2016
	3.3. Proceso de programación, programa fuente y programa compilado.			Portafolio de ejercicios: Trabajo grupal: Exposición e informe sobre “Paradigmas de programación” (rúbrica) Fecha: 30-03-2016
	3.4 Paradigmas de programación			Evaluación: Resolución de cuestionario en el aula virtual Fecha: 05-04-2016
	3.5. Lenguajes de programación			

SEMANA: 06-04-2016 - 13-04-2016

1	4. Introducción Programación en bloques (2 semana)	4.1. Scratch	(1) Instrucción Directa (1) Portafolio del Estudiante	Lectura del material publicado en el apoyo. Curso programación en bloques: https://studio.code.org/s/20-hour	Portafolio de ejercicios: Proyecto aplicativo sobre programación en bloques usando bloques en Scratch. Fecha: 19-04-2016 (rúbrica) Evaluación: Curso programación en bloques: https://studio.code.org/s/20-hour Fecha: 19-04-2016
		4.2. App Inventor			
		4.3. Blockly			

EVALUACIÓN CATEDRA 1 19-04-2016

REALIMENTACIÓN CATEDRA I Y REGISTRO DE NOTAS 20-04-2016

SEMANA: 25- 04-2016 al 13-05-2016

1	5. Introducción a la programación estructurada (3 semana)	5.1 Visión general del paradigma de la programación estructurada.	(1) Instrucción Directa	Lectura del libro: Programación Estructurada y Orientada a Objetos(López, 2011, Capítulo 1)	Portafolio de ejercicios: Resolución del ejercicios sobre Programación estructurada (López, 2011, Capítulo 3,4,5) Fecha: 11-05-2016
		5.2. Diagramas de flujo	(1) Evaluación		
		5.3. Pseudocódigo			
		5.4. Revisión de algoritmos. Métodos formales para verificación de programas	(1) Portafolio del Estudiante	Lectura del libro: Fundamentos de programación Piensa e C(Cairo, 2006, Capítulo 1)	Evaluación: Resolución de cuestionario: Introducción a la programación estructurada Fecha: 11-05-2016

SEMANA: 16-05-2016 al 27-05-2016)

1	6. Introducción a la programación orientada a	6.1 Visión general del paradigma de la programación orientada a objetos.	(1) Instrucción Directa (1) Evaluación	Lectura del libro Programación Estructurada y Orientada a Objetos(López,	Evaluación: Cuestionario teórico en el aula : Conceptos Básicos de la programación Orientada a objetos
---	---	--	---	---	--

objetos. (2 semana)	6.2 Conocimiento básico de las estructuras de control del lenguaje de programación java para implementar programas orientados a objetos.	(1) Portafolio del Estudiante	2011, Capítulo 10)	Fecha: 25-05-2016
			Resolución de ejercicios	Portafolio de ejercicios: Resolución del ejercicios sobre Programación Estructurada y Orientada a Objetos(López, 2011, Capítulo 10) Fecha: 25-05-2016

EVALUACIÓN CATEDRA 2 31-05-2016

REALIMENTACIÓN CATEDRA I Y REGISTRO DE NOTAS 01-06-2016

SEMANA: 06-06-2016 al 17-06-2016)

1	7. Introducción a los nuevos lenguajes de programación (2 semana)	7.1 Programación Ruby	(1) Instrucción Directa	Lectura del material publicado en el aula virtual.	Evaluación: Resolución de cuestionario en el aula virtual Fecha: 08-06-2016
		7.2 Programación Python	(1) Evaluación (1) Portafolio del Estudiante	Resolución de ejercicios (2) Foro Virtual, Discusión sobre los nuevos lenguajes de programación	Exposición grupal sobre los nuevos lenguajes de programación. Fecha: 08-06-2016 Foro virtual: 2 intervenciones fundamentadas (Rúbrica para foros) Fecha: 08-06-2016

SEMANA:30-06-2016 al 24-06-2016

1	8. Introducción a aplicaciones Cliente Servidor. (1 semana)	8.1 Programación HTML5	(1) Instrucción Directa	Lectura del material publicado en el aula virtual.	Evaluación: Resolución de cuestionario en el aula virtual Fecha: 22-06-2016
		8.2Arquitectura cliente/ servidor	(1) Evaluación		
		8.3 Conocimiento de las herramientas para trabajar con formularios y servlets.	(1) Portafolio del Estudiante	Resolución de ejercicios	Solución de ejercicios enviados por el docente Fecha: 22-06-2016

EVALUACIÓN FINAL : 05-07-2016

PRESENTACIÓN PROYECTO FINAL 12-07-2016

3. Normas y procedimientos para el aula (Docente)

1. Solo se recibirán trabajos en el aula virtual y dentro del plazo establecido.
2. No se admitirá la entrega de tareas fuera del aula virtual y fuera del plazo establecido.
3. 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.

4. 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.
5. 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 (cero).
6. Se acepta el uso de cualquier dispositivo electrónico (ipads, tablets, celulares, audifonos) únicamente con fines académicos. El uso para fines no académicos equivaldrá a una inasistencia.
7. No se podrán ingresar alimentos al aula
8. El estudiante tiene derechos a recibir tutoría en los horarios establecidos por el docente.
9. En el caso de inasistencia es responsabilidad del estudiante igualarse en los contenidos de la materia dictada en dicha clase.
10. 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.

4. Referencias bibliográficas (*Docente*)

a. Principales.

1. Stair, R., Reynolds, G. (2010). *Principios de sistemas de información*. (9a ed). Mexico, Mexico: Cengage Learning Editores. ISBN 978-0-324-66528-4 007
2. López, L. (2011). *Programación Estructurada y Orientada a Objetos*. (3a ed) México, México: Alfaomega Grupo Editor. ISBN 978-607-707-211-9.
3. Noguera, F., Riera, D. (2013). *Programación*. (1a ed.) Barcelona, España: Editorial UOC. ISBN: 978-84-9788-110-4

b. Referencias complementarias.

1. Villalpando, M. García, A., (2014). *Matemáticas discretas: aplicaciones y ejercicios*. México, Mexico. Larousse Grupo Editorial Patria. ISBN ebook: 9786074389258.
2. Márquez, G. Osorio, S., & Olvera, N. (2010). *Introducción a la Programación Estructurada en C. (1a ed)* México, México: Pearson Educación. ISBN 978-607-32-0600-6

a. Perfil del docente

Ing. Anita Yáñez

Magíster en Gerencia Empresarial mención en Gerencia de Proyectos, Ingeniera de Sistemas de Computación e Informática de la Escuela Politécnica Nacional, con 15 años de experiencia en el análisis, desarrollo e implementación de Software en el campo público y privado. En la área docente 10 Años de experiencia.

Correo: a.yanez@udlanet.ec

Horario de atención:

Lunes: 8:00 a 10:00

Viernes: 7:00 a 9:00