

**Facultad de Ingeniería y Ciencias Agropecuarias**  
**Ingeniería en Sistemas de Computación e Informática**  
**ACI320/Programación Orientada a Objetos**  
Período 2016-2

**1. Identificación**

Número de sesiones: 64.

Número total de horas de aprendizaje: 160

Créditos – malla actual: 4

Profesor: Bernarda Sandoval

Correo electrónico del docente (Udlanet): b.sandoval@udlanet.ec

Director: Marco Antonio Galarza Castillo

Campus: Queri

Pre-requisito: ACI220/Programación Estructurada

Co-requisito:

Paralelo:

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**

Este curso permite al estudiante la solución de problemas por medio del diseño y la programación orientada a objetos. El énfasis está en el modelamiento de clases mediante UML, para luego realizar el diseño de la solución, la documentación y la implementación. Los alumnos utilizan bibliotecas para crear proyectos de software, utilizando java como lenguaje de programación para realizar los ejercicios propuestos en clase.

**3. Objetivo del curso**

Desarrollar aplicaciones informáticas utilizando las características de la programación orientada a objetos a través del diseño de clases para la solución de problemas.

#### 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. Explica los principios y características propias de un paradigma de programación.</p> <p>2. Aplica los principios y características propias de un paradigma y lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales.</p>	<p>1. Aplica metodologías de investigación, pensamiento lógico, fundamentos matemáticos, principios algorítmicos y teorías de Ciencias de la Computación en la fundamentación, modelación y diseño de soluciones informáticas.</p>	<p><b>Inicial</b> ( )</p> <p><b>Medio</b> ( X )</p> <p><b>Final</b> ( )</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:

<b>Reporte de progreso 1</b>	<b>35%</b>
Portafolio de ejercicios	10%
Portafolio de laboratorios	10%
Examen de 1er. Progreso	15%
<b>Reporte de progreso 2</b>	<b>35%</b>
Portafolio de ejercicios	10%
Portafolio de laboratorios	10%
Examen de 2do. Progreso	15%
<b>Evaluación final</b>	<b>30%</b>
Proyecto final	15%
Examen final	15%

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 clases se desarrollarán con una participación activa de los estudiantes quienes realizarán prácticas de laboratorio, investigaciones sobre temas específicos, ejercicios propuestos que facilitarán el desarrollo de su razonamiento y pensamiento lógico. Las estrategias para alcanzar un aprendizaje significativo incluyen:

- Investigaciones en bibliotecas virtuales, Internet.
- Clases cooperativas que promueven el aprendizaje integrador y el desarrollo del liderazgo personal y profesional.
- Desarrollo de trabajos de investigación y prácticas en laboratorio.
- Uso del aula Programación Orientada a Objetos II ACI320 en el apoyo virtual para compartir el material utilizado en clases para subir los deberes e investigaciones.

### 6.1. Escenario de aprendizaje presencial.

En el escenario de aprendizaje presencial se da énfasis a la enseñanza enfocada en el alumno mediante el uso de diferentes estrategias:

Instrucción directa:

En el escenario de aprendizaje presencial, se utilizará la instrucción directa para transmitir a los estudiantes información de forma organizada y sistemática sobre los principales conceptos de cada unidad temática.

Prácticas de Laboratorio:

En el transcurso de la materia se realizarán ejercicios prácticos usando herramientas a fin de que los estudiantes apliquen y refuercen los conocimientos teóricos adquiridos en los diferentes tópicos.

Los mecanismos de evaluación utilizados serán todas las actividades realizadas por los estudiantes y que estarán subidos a la plataforma virtual. Otra forma de evaluación son los exámenes teóricos y/o prácticos que permitirán evaluar el aprendizaje a través de preguntas de asociación, selección múltiple y el desarrollo de ejercicios prácticos.

### 6.2. Escenario de aprendizaje virtual.

Indagación en bases de datos:

El estudiante utilizará bases de datos a fin de investigar temas y/o ejercicios planteados por el docente. Esta actividad puede ser parte de las prácticas de laboratorio y trabajos en grupo.

### 6.3. Escenario de aprendizaje autónomo.

En el escenario de aprendizaje autónomo los estudiantes deberán realizar actividades que les permitan complementar y profundizar los conocimientos adquiridos en el escenario de aprendizaje presencial.

**Lecturas:**

Todos los estudiantes deben realizar la lectura de capítulos específicos de la bibliografía que serán indicados por el profesor de acuerdo a cada tema. La lectura tiene como objetivo conocer, complementar o profundizar los contenidos del programa de la asignatura.

**Portafolio de ejercicios:**

El portafolio del estudiante recopilará evidencia del aprendizaje dentro del desarrollo del curso. A lo largo de toda la materia se realizarán varias actividades de aprendizaje autónomo:

**Trabajo en grupo (colaborativo):**

A lo largo de la materia, se realizarán trabajos grupales de búsqueda y análisis de información que serán aplicados al desarrollo de su proyecto final, estos serán evaluados en base a informes subidos al aula virtual.

La prueba y examen de cada progreso, a más de evaluar el aprendizaje presencial, incluirán los temas desarrollados en el portafolio del estudiante y las lecturas.

**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.

## 7. Temas y subtemas del curso

RdA	Temas	Subtemas
Explica los principios y características propias de un paradigma de programación.	1. Diseño de clases	1.1 Características de la POO 1.2 Clases y objetos 1.3 Diagramas UML de Clases 1.4 Relaciones entre Clases 1.5 Estructuras Comunes de Clases 1.6 Modelado de Clases en UML utilizando un editor
	2. Aplicaciones de JAVA	2.1 Entorno de programación JAVA usando un IDE 2.2 Convenciones de escritura de código 2.3 Manejo de excepciones 2.4 Depuración de programas

<p>Aplica los principios y características propias de un paradigma y lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales.</p>	3. Implementación de clases	3.1 Definiendo Clases 3.2 Implementación de asociaciones entre clases 3.3 Herencia
	4. Colecciones	4.1 Arreglos estáticos 4.2 Arreglos dinámicos genéricos(ArrayList)
	5. Diseño avanzado de clases	5.1 Clases Abstractas 5.2 Polimorfismo 5.3 Interfaces
	6. Diseño e Implementación de aplicaciones	6.1 Selección y diseño de una aplicación a implementar 6.2. Implementación de la solución informática

## 8. Planificación secuencial del curso

Semana 1 -4					
RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#1	1. Diseño de clases	1.1 Características de la POO 1.2 Clases y objetos 1.3 Diagramas UML de Clases 1.4 Relaciones entre Clases 1.5 Estructuras Comunes de Clases 1.6 Modelado de Clases en UML utilizando un editor	(1) Socialización del Sílabo e Indicaciones Generales  (1) Instrucción Directa: Diseño de clases  (1) Ejercicios de implementación usando Netbeans  (1) Prácticas de laboratorio: Modelamiento de clases	(3) Portafolio de Ejercicios 63-64 (Deitel)	Portafolio De Prácticas de laboratorio Fecha de entrega: Del 7 marzo al 2 abril  Portafolio de Ejercicios Fecha de entrega: Del 7 marzo al 2 abril
Semana 5 - 7					
RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#2	2. Aplicaciones de JAVA	2.1 Entorno de programación JAVA usando un IDE 2.2 Convenciones de escritura de código	(1) Instrucción Directa: Aplicaciones de Java  (1) Portafolio de Ejercicios	(3) Portafolio de Ejercicios p. 65-70 (Deitel)  (3) Portafolio de Ejercicios p.	Portafolio De Prácticas de laboratorio Fecha de entrega: Del 4 abril al 22 abril  Portafolio de Ejercicios

		2.3 Manejo de excepciones 2.4 Depuración de programas	(1) Prácticas de laboratorio.  (2) Investigación de Excepciones en java.	22,41 (Flores,H., Joyanes, L.)	Fecha de entrega: Del 4 abril al 22 abril  Examen, teórico práctico Progreso 1: Fecha de entrega: Del 11 al 16 Abril
--	--	--	--	--------------------------------	--

**Semana 8 - 10**

RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#2	3. Implementación de clases	3.1 Definiendo Clases 3.2 Implementación de asociaciones entre clases 3.3 Herencia	(1) Instrucción Directa: Implementación de clases  (1) Portafolio de Ejercicios  (1) Prácticas de laboratorio.	(3) Portafolio de Ejercicios p. 144-150 (Deitel)  (3) Portafolio de Ejercicios de p. 139 (Flores,H., Joyanes, L.)	Portafolio De Prácticas de laboratorio Fecha de entrega: Del 25 de abril al 14 de mayo.  Portafolio de Ejercicios Fecha de entrega: Del 25 de abril al 14 de mayo.

**Semana 11 - 12**

RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#2	4. Colecciones	4.1 Arreglos estáticos 4.2 Arreglos dinámicos genéricos(ArrayList)	(1) Instrucción Directa: Colecciones  (1) Portafolio de Ejercicios  (1) Prácticas de laboratorio.  (2) Investigación de Tipos de Colecciones en java.	(3) Portafolio de Ejercicios p. 295-310 (Deitel)  (3) Portafolio de Ejercicios p. 102 (Flores,H., Joyanes, L.)	Portafolio De Prácticas de laboratorio Fecha de entrega: Del 16 al 27 de mayo.  Portafolio de Ejercicios Fecha de entrega: Del 16 al 27 de mayo.

--	--	--	--	--	--

Semana 13 - 14					
RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#2	5. Diseño avanzado de clases	5.1 Clases Abstractas 5.2 Polimorfismo 5.3 Interfaces	(1) Instrucción Directa: Diseño avanzado de clases  (1) Portafolio de Ejercicios  (1) Prácticas de laboratorio.	(3) Portafolio de Ejercicios p. 436-437 (Deitel)  (3) Portafolio de Ejercicios p. 132 (Flores,H., Joyanes, L.)	Examen, teórico práctico Progreso 2: Fecha de entrega: Del 30 de mayo al 3 de junio  Portafolio De Prácticas de laboratorio Fecha de entrega: Del 30 de mayo al 10 de junio.  Portafolio de Ejercicios Fecha de entrega: Del 30 de mayo al 10 de junio.

Semana 15 - 16					
RdA	Tema	Sub tema	Actividad/ estrategia de clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
#2	6. Diseño e Implementación de aplicaciones	6.1 Selección y diseño de una aplicación a implementar 6.2. Implementación de la solución informática	(2) Selección de una aplicación  (2) Implementación de la solución informática	(3) Diseño e implementación de una aplicación informática ( Proyecto final)	Informe y defensa del proyecto final Fecha de entrega: Del 20 al 24 de junio 2016  Examen, teórico práctico Evaluación Final: Fecha de entrega: Del 4 al 8 de julio 2016

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

## 9. Normas y procedimientos para el aula

1. Se permitirá entregar una tarea hasta con 24 horas de retraso con una penalidad del 50% de la nota asignada.

2. 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.
3. Los estudiantes deberán practicar la honestidad académica como: ejercicios, exámenes, proyectos, y todas las actividades de aprendizaje solicitadas por el docente. La falta de honestidad se calificará con la mínima calificación (cero).
4. El uso de cualquier dispositivo electrónico se aceptará en clase solo para fines académicos. El uso para fines no académicos equivaldrá a una inasistencia.
5. Todas las actividades serán receptadas únicamente a través del aula virtual.
6. El estudiante puede acceder a tutoría personal en los horarios establecidos por el docente.
7. En el caso de inasistencia es responsabilidad del estudiante igualarse en los contenidos de la materia dictada en dicha clase.
8. En el caso de que un estudiante falte a una sesión en la que se realicen pruebas, ejercicios o prácticas de laboratorio, no se podrán recuperar las calificaciones.

## 10. Referencias bibliográficas

### 10.1. Principales.

#### Libros Físicos:

- Héctor, F. (2012). *Programación Orientada a Objetos usando JAVA*. (1era ed). Bogotá, Colombia: Ecoe Ediciones.
- Deitel, P., Deitel, H. (2012). *Java cómo programar*. (9na ed). México, México: Pearson

### 10.2. Referencias complementarias.

#### Libros Físicos:

- Joyanes Aguilar, Luis. (2011). *Programación en Java: algoritmos, programación orientada a objetos e interfaz gráfica de usuario*. México, México: McGraw Hill.

## Perfil del docente

Ing. Bernarda Sandoval, Msc.

Master en Ciencias de la Computación obtenido en la Universidad Estadual Paulista “Júlio de Mesquita Filho” UNESP-Brasil, Diploma superior en Docencia Universitaria obtenida en la Universidad de la Fuerzas Armadas ESPE, Ingeniera en Sistemas e Informática por la Universidad de la Fuerzas Armadas ESPE. Tiene experiencia en trabajos de investigación en el área de Inteligencia Artificial específicamente en Web semántica, agentes inteligentes y ontologías. Ha realizado publicaciones referentes al educación virtual y web semántica.