

Facultad de Ingeniería y Ciencias Agropecuarias
Ingeniería en Sistemas de Computación e Informática
ACI320 Programación Orientada a Objetos
Período 2018-1

A. Identificación

Número de sesiones:	64
Número total de hora de aprendizaje:	64 h presenciales + 96 h de aplicación del aprendizaje y estudio autónomo = 160 h totales.
Créditos – malla actual:	6
Profesor:	Bernarda Sandoval
Correo electrónico del docente	bernarda.sandoval@udla.edu.ec
Director:	Marco Galarza
Campus:	Queri
Pre-requisito:	ACI220/Programación Estructurada
Co-requisito:	
Paralelo:	1, 2 y 70

B. 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.

C. Resultados de aprendizaje (RdA) del curso

1. Explica los principios y características propias de un paradigma de programación.
2. Aplica los principios y características propias de un paradigma y lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales.

D. Sistema y mecanismos de evaluación

De acuerdo con el Modelo Educativo de la UDLA la evaluación busca evidenciar el logro de los resultados de aprendizaje institucionales, de cada carrera y de cada 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 con el calendario académico:

Reporte de progreso 1: (5 semanas) 25%

- a) *Tareas autónomas* (Portafolio de ejercicios (5%)): Por cada tema se enviará una serie de ejercicios simples y complejos que serán realizados por el estudiante como parte de su trabajo autónomo.
- b) *Participación en clase* (Portafolio de laboratorios (10%)): Diferentes actividades realizadas en clase orientadas a identificar, corregir y reforzar los temas vistos en clase.
- c) *Evaluación continua* (Cuestionarios y ejercicios prácticos): Por cada tema analizado en clase se tendrá una evaluación a través de un cuestionario en el aula virtual y el desarrollo de un ejercicio práctico. (10%)

Reporte de progreso 2: (5 semanas) 35%

- a) *Tareas autónomas* (Portafolio de ejercicios (10%)): Por cada tema se enviará una serie de ejercicios simples y complejos que serán realizados por el estudiante como parte de su trabajo autónomo.
- b) *Participación en clase* ((Portafolio de laboratorios (12.5%)): Diferentes actividades realizadas en clase orientadas a identificar, corregir y reforzar los temas vistos en clase.
- c) *Evaluación continua* (Cuestionarios y ejercicios prácticos): Por cada tema analizado en clase se tendrá una evaluación a través de un cuestionario en el aula virtual y el desarrollo de un ejercicio práctico. (12.5%)

Progreso 3: (6 semanas) 40%

- a) *Tareas autónomas* (Portafolio de ejercicios (10%)): Por cada tema se enviará una serie de ejercicios simples y complejos que serán realizados por el estudiante como parte de su trabajo autónomo.
- b) *Participación en clase* (Portafolio de laboratorios (15%)): Diferentes actividades realizadas en clase orientadas a identificar, corregir y reforzar los temas vistos en clase.
- c) *Evaluación continua* (Cuestionarios y ejercicios prácticos): Por cada tema analizado en clase se tendrá una evaluación a través de un cuestionario en el aula virtual y el desarrollo de un ejercicio práctico. (5%)
- d) *Evaluación continua: (Proyecto final, desarrollo de una aplicación informática (10%))*: Para la evaluación final se considera la realización de un proyecto en el que los estudiantes forman grupos de 3 personas y desarrollan una aplicación informática para resolver un problema propuesto por ellos, en el cual demuestren el dominio de los temas revisados durante el semestre. La nota comprende el proyecto funcionando correctamente, la documentación y la defensa (individual). El proyecto final será evaluado por medio de una rúbrica.

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 La metodología debe contemplar el aprendizaje presencial, virtual y autónomo.

En el proceso de enseñanza – aprendizaje de la materia hay dos aristas; la primera es el estudiante, cuya participación en todas las actividades planificadas es parte integral de su formación académica, la segunda arista es la planificación sistemática del semestre. En las clases se usarán materiales didácticos que motiven a los estudiantes al aprendizaje como: Proyectos, foros participativos, trabajos colaborativos, juegos (gamificación), todo esto conlleva a los estudiantes se motiven y les guste la carrera de Ingeniería de Software.

Las clases se desarrollarán con una participación de los estudiantes quienes realizarán prácticas de laboratorio, talleres, 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.

F.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 las evaluaciones teóricas y/o prácticas de cada uno de los temas que permitirán evaluar el aprendizaje a través de preguntas de asociación, selección múltiple y el desarrollo de ejercicios prácticos.

F.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 de ejercicios y del trabajo en grupo

F.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 con 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 la información subida al aula virtual.

Proyecto final:

Para la evaluación final se considera la realización de un proyecto en el que los estudiantes se organizan en grupos de 3 personas y desarrollan una aplicación informática para resolver un problema propuesto por ellos, en el cual demuestren el

dominio de los temas revisados durante el semestre. La nota comprende el proyecto funcionando correctamente, la documentación y la defensa individual. El proyecto final será evaluado por medio de una rúbrica.

G. Planificación alineada a los RdA

Planificación	Fechas	RdA 1	RdA 2
Unidad 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 Modelado de Clases en UML utilizando un editor	Semanas 1-4	X	
Actividades			
1. (P) Presentación por parte del docente del tema: "Introducción a la Programación Orientada a Objetos" 2. (P) Resolución de problemas: Identificación y creación de clases 3. (P) Presentación por parte del docente del tema: "Diagramas UML de Clases". 4. (A) Revisión de videos sobre relaciones entre clases presentados en el aula virtual 5. (P) Conversatorio en clase sobre los videos revisados sobre tipos de relaciones 6. (P) Presentación por parte del docente del tema: "Relaciones entre Clases". 7. (P) Resolución de problemas: diagramas UML (identificación de clases y relaciones) 8. (P) Ejercicios de modelo de clases utilizando un editor 9. (A) Portafolio de ejercicios aula virtual: Resolución de ejercicios presentados por el docente		X	
Evaluaciones			
1. Portafolio de ejercicios aula virtual: Resolución de ejercicios 2. Portafolio de laboratorios: Resolución de ejercicios propuestos en clases 3. Evaluación continua: Cuestionario y ejercicio de Características de POO, Clases y Objetos, Relaciones entre clases		X	
Unidad 2			
2. Aplicaciones de JAVA 2.1. Introducción a Java 2.2. Entorno de programación JAVA usando un IDE 2.2.1. Convenciones de escritura de código 2.2.2. Depuración de programas	Semanas 5-6		X
Lecturas			
Deitel, P., Deitel, H. (2012). Java cómo programar. (9na ed.). México, México: Pearson. <i>Capítulo 1(pág.18-26), Capítulo 2(pág. 38-56), Capítulo 3 (pág. 72-92).</i>			X
Actividades			

<ol style="list-style-type: none"> (A) Lectura del tema: Capítulo 1 "Introducción a las computadoras y a Java". (Deitel, 2012) (P) Presentación por parte del docente del tema: "Introducción a Java" (A) Lectura del tema: Capítulo 2 "Introducción a las aplicaciones en Java". (Deitel, 2012) (P) Presentación por parte del docente del tema: "Entorno de programación Java" (P) Resolución de ejercicios sencillos en Java. (A) Portafolio de ejercicios aula virtual: Resolución de ejercicios 			X
Evaluaciones			
<ol style="list-style-type: none"> Portafolio de ejercicios aula virtual: Resolución de ejercicios Portafolio de laboratorios: Resolución de ejercicios propuestos en clases Control de lectura. Resolución de cuestionario aula virtual. Capítulo 1 "Introducción a las computadoras y a Java". (Deitel, 2012) Capítulo 2 "Introducción a las aplicaciones en Java". (Deitel, 2012) Evaluación continua: resolución de ejercicios 			X
<p style="text-align: center;">Unidad 3</p> <ol style="list-style-type: none"> Implementación de clases <ol style="list-style-type: none"> Definiendo Clases Métodos Constructores Implementación de asociaciones entre clases Manejo de excepciones Herencia 	Semanas 7-11		X
Lecturas			
Deitel, P., Deitel, H. (2012). Java cómo programar. (9na ed.). México, México: Pearson. <i>Capítulo 3(pág.72-92), Capítulo 6(pág. 198-207), Capítulo 8 (pág. 312-326, Capítulo 11 (pág. 439-454).</i>			X
Actividades			
<ol style="list-style-type: none"> (A) Lectura del tema: Capítulo 3 "Introducción a las clases, objetos, métodos y cadenas". (Deitel, 2012) (P) Conversatorio en clase sobre la lectura del capítulo 3 (P) Resolución de ejercicios: definición de clases (A) Lectura del tema: Capítulo 6 "Métodos: un análisis más detallado". (Deitel, 2012) (A) Lectura del tema: Capítulo 8 "Clase y objetos: un análisis más detallado". (Deitel, 2012) (P) Presentación por parte del docente del tema: "Métodos y constructores". (P) Resolución de ejercicios utilizando diferentes métodos y constructores (P) Resolución de ejercicios pasando de diagramas UML a Java (A) Lectura del tema: Capítulo 11 "Manejo de excepciones un análisis más profundo". (Deitel, 2012) (P) Presentación por parte del docente del tema: "Manejo de excepciones". (P) Presentación por parte del docente del tema: "Herencia". (P) Resolución de ejercicios de herencia (A) Portafolio de ejercicios aula virtual: Resolución de ejercicios presentados por el docente 			X

14. (A) Proyecto grupal: Desarrollo de una aplicación informática (entrega parcial: diagrama UML, requerimientos funcionales)			
Evaluaciones			
4. Portafolio de ejercicios aula virtual: Resolución de ejercicios 5. Portafolio de laboratorios: Resolución de ejercicios propuestos en clases 6. Control de lectura. Resolución de cuestionario aula virtual. Capítulo 3 "Introducción a las clases, objetos, métodos y cadenas" Capítulo 6 "Métodos: un análisis más detallado" Capítulo 8 "Clase y objetos: un análisis más detallado" 7. Evaluación continua: Cuestionario de herencia y manejo de excepciones 8. Evaluación continua: resolución de ejercicios 9. Proyecto grupal: Desarrollo de una aplicación informática (entrega parcial: diagrama UML, requerimientos funcionales)			
Unidad 4	Semanas 12-13		
1. Colecciones 4.1 Arreglos estáticos 4.2 Arreglos dinámicos genéricos(ArrayList)			X
Lecturas			
Deitel, P., Deitel, H. (2012). Java cómo programar. (9na ed.). México, México: Pearson. <i>Capítulo 7</i> (pág.241-254, 259-262, 284-286).			X
Actividades			
1. (A) Lectura del tema: Capítulo 7 "Arreglos y objetos ArrayList". (Deitel, 2012) (pág. 241-254 y 259-262). 2. (P) Presentación por parte del docente del tema: "Arreglos estáticos". 3. (A) Lectura del tema: Capítulo 7 "Arreglos y objetos ArrayList". (Deitel, 2012) (pág. 284-286). 4. (P) Presentación por parte del docente del tema: "Arreglos dinámicos". 5. (P) Resolución de ejercicios: definición de clases 6. (A) Portafolio de ejercicios aula virtual: Resolución de ejercicios presentados por el docente			X
Evaluaciones			
1. Portafolio de ejercicios aula virtual: Resolución de ejercicios 2. Portafolio de laboratorios: Resolución de ejercicios propuestos en clases 3. Control de lectura. Resolución de cuestionario aula virtual. Capítulo 7 "Arreglos y objetos ArrayList" 4. Evaluación continua: resolución de ejercicios			

Unidad 5	Semanas 14-16		X
5. Diseño avanzado de clases 5.1 Enumeraciones, miembros static y final 5.2 Clases Abstractas 5.3 Polimorfismo 5.4 Interfaces			
Lecturas			
Deitel, P., Deitel, H. (2012). Java cómo programar. (9na ed.). México, México: Pearson. <i>Capítulo 8</i> (pág. 331-339), <i>Capítulo 10</i> (pág. 395-402, 419-430).			X
Actividades			
1. (A) Lectura del tema: Capítulo 8 “Clases y objetos: un análisis más detallado”. (Deitel, 2012) 2. (P) Presentación por parte del docente del tema: “Enumeraciones, miembros static y final”. 3. (A) Lectura del tema: Capítulo 10 “Programación orientada a objetos: polimorfismo”. (Deitel, 2012) (pág. 395-402). 4. (P) Presentación por parte del docente del tema: “Clases abstractas y polimorfismo”. 5. (P) Resolución de ejercicios 6. (A) Lectura del tema: Capítulo 10 “Programación orientada a objetos: polimorfismo”. (Deitel, 2012) (pág. 319-430). 7. (P) Presentación por parte del docente del tema: “Interfaces”. 8. (P) Resolución de ejercicios 9. (A) Portafolio de ejercicios aula virtual: Resolución de ejercicios presentados por el docente 10. (A) Proyecto grupal: Desarrollo de una aplicación informática			X
Evaluaciones			
1. Portafolio de ejercicios aula virtual: Resolución de ejercicios 2. Portafolio de laboratorios: Resolución de ejercicios propuestos en clases 3. Control de lectura. Resolución de cuestionario aula virtual. Capítulo 8 “Clases y objetos: un análisis más detallado” Capítulo 10 “Programación orientada a objetos: polimorfismo” 4. Evaluación continua: resolución de ejercicios 5. Proyecto grupal: Desarrollo de una aplicación informática (entrega final)			X

H. Normas y procedimientos para el aula

1. Solo se recibirán trabajos en el aula virtual y dentro del plazo establecido.
2. 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.
3. 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.

4. 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, se calificará con la mínima calificación (1.0).
5. Se acepta el uso de cualquier dispositivo electrónico (iPads, tablets, celulares, audífonos) únicamente con fines académicos. El uso para fines no académicos equivaldrá a una inasistencia.
6. No se podrán ingresar alimentos al aula.
7. El estudiante tiene derechos a recibir tutoría 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. Las fechas de las evaluaciones serán publicadas en el apoyo virtual de la materia.

I. Referencias

1. Principales.

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

2. Complementarias.

1. Joyanes Aguilar, Luis. (2011). *Programación en Java: algoritmos, programación orientada a objetos e interfaz gráfica de usuario*. México, México: McGraw Hill.
2. Cadenhead, R. (2012). *Programación Java 7*. Madrid: Anaya

J. 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 a educación virtual y web semántica.