

Facultad de Ingeniería y Ciencias Agropecuarias
Ingeniería en Sistemas de Computación e Informática
Código del curso ACI520 y Asignatura Lenguaje de Programación I
Período 2016-2

1. Identificación

Número de sesiones: 48
 Número total de horas de aprendizaje: 120
 Créditos – malla actual: 3
 Profesor: Ing. Carlos Andrés Muñoz Cueva, MSc , MsF.
 Correo electrónico del docente (Udlanet): ca.munoz@udlanet.ec
 Director: Marco Galarza Castillo
 Campus: Sede Queri
 Pre-requisito: ACI421 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

Esta asignatura aporta al perfil del egresado de Ingeniería de Sistemas los conocimientos y habilidades necesarios para el diseño, desarrollo e implementación de aplicaciones visuales cliente-servidor. Usando un modelo de objetos, librerías y servicios que brinda la plataforma seleccionada.

Al final del curso el estudiante desarrollará un Sistema integrador (con la materia Base de datos I) aplicando la arquitectura en capas con manipulación de base de datos.

3. Objetivo del curso

Aplicar de forma óptima los conocimientos básicos e intermedios de la herramienta de programación visual, para el desarrollo tanto de aplicaciones tradicionales como aplicaciones cliente-servidor con interacción de base de datos, usando una metodología orientado al manejo de capas.

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 la programación orientada a eventos, a través de un lenguaje de programación.	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.	Inicial (X) Medio () Final ()
2. Emplea las características propias de un lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales orientados a eventos	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.	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%
Sub componentes	
Portafolio de Ejercicios	25%
Práctica de Laboratorio	35%
Evaluación Práctica	40%
Reporte de progreso 2	35%
Sub componentes	
Portafolio de Ejercicios	15%
Práctica de Laboratorio	20%
Avance Sistema Informático	25%
Evaluación Práctica	40%
Evaluación final	30%
Sub componentes	
Documentación Sistema Informático y Portafolio de Ejercicios	20%
Sistema Informático Final	30%
Evaluación Práctica	50%

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.

Una de las Evaluaciones Prácticas de los Progresos, podrá ser recuperada según decisión del alumno, si cubre la asistencia antes mencionada.

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.

Práctica de Laboratorio: El estudiante deberá resolver la práctica planteada por el profesor, tendrá la ayuda requerida y deberá elaborar un informe para la siguiente clase donde completará la práctica y contestará un grupo de preguntas planteadas.

Instrucción Directa: El docente presentará el tema a tratar de manera resumida, usando medios tecnológicos, se usará la técnica pregunta – respuesta.

Ejercicios individuales y en grupo: Que serán desarrollados dentro y fuera del aula para reforzar lo aprendido en clase.
Evaluaciones teóricas y prácticas.

6.2. Escenario de aprendizaje virtual

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

Avance Sistema Informático: El estudiante deberá presentar un avance de su aplicativo cliente-servidor.

Documentación Sistema Informático Final: El estudiante deberá presentar la documentación del Sistema Informático y resolver los deberes planteados.

7. Temas y subtemas del curso

RdAs	Temas	Subtemas
Explica los conceptos de la programación orientada a eventos, a través de un lenguaje de programación.	1. Plataforma .NET	1.1. Generalidades de .NET 1.2. CLR – Base Class Library - CLS
	2. Conceptos de Programación Visual C#	2.1 Sintaxis de C#
		2.2. Manejo de Clases y Objetos.
		2.3 Manejo de excepciones
Emplea las características propias de un lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales orientados a eventos	3. Desarrollo de aplicaciones en Visual C# y Librerías de clases .NET	3.1 Uso del diseñador de Visual Studio
		3.2. Diseño de Interfaz de usuario
		3.3 Manejo de controles comunes de Windows Forms
		3.4 Eventos y delegados
		3.5 Colecciones de datos
		3.6 Streams en .NET
		3.7 Acceso a datos de archivos
	4. Acceso a Datos y Arquitectura en capas	4.1 Introducción a ADO.NET. Acceso a datos
		4. 2. Arquitectura en capas Cliente-Servidor
		4.3. Acceso conectado a base de datos
		4.4. Acceso desconectado: Dataset y DataAdapters
		4.5.Manejo de controles con base de datos
		4.6 Transaccionalidad
		4.7 LINQ to SQL

8. Planificación secuencial del curso

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

Semanas 7 de Marzo al 16 de Abril					
RdAs	Temas	Subtemas	Actividad/ metodología/clase	Tarea/ trabajo autónomo	MdE/Producto/ fecha de entrega
Explica los conceptos de la programación orientada a eventos, a través de un lenguaje de programación.	1. Plataforma .NET y Modelo en capas 2. Conceptos de Programación Visual C#	1.1. Generalidades de .NET 1.2. CLR – Base Class Library – CLS 2.1 Sintaxis de C# 2.2 Manejo de Clases y Objetos. 2.3 Manejo de excepciones	(1) Instrucción Directa (1) Ejercicios en Clase	(1,2) Lectura Capítulo 1 y Tarea: p.3-14 “Enciclopedia Microsoft Visual C#”	Presentación Informe IEEE 15/03/2016
				(2) Informe de Lectura	
Emplea las características propias de un lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales orientados a eventos	3. Desarrollo de aplicaciones en Visual C# y Librerías de clases .NET	3.1 Uso del diseñador de Visual Studio 3.2. Diseño de Interfaz de usuario 3.3 Manejo de controles comunes de Windows Forms	(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Prácticas de Laboratorio (1,2) Trabajo e Grupo	(1,2) Tareas: p.106 -120 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos y Resueltos (Rúbrica para Ejercicios de Programación) 17/03/2016
				(1,2) Tareas: p.165-168 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos (Rúbrica para Ejercicios de Programación) 22/03/2016

Emplea las características propias de un lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales orientados a eventos	3. Desarrollo de aplicaciones en Visual C# y Librerías de clases .NET	3.1 Uso del diseñador de Visual Studio 3.2. Diseño de Interfaz de usuario 3.3 Manejo de controles comunes de Windows Forms	(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Prácticas de Laboratorio (1,2) Trabajo e Grupo	(1,2) Tareas: p.225-238 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos y Resueltos (Rúbrica para Ejercicios de Programación) 24/03/2016
				(1,2) Tareas: p.274-290 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos y Resueltos (Rúbrica para Ejercicios de Programación) 29/03/2016
				(2)Informe de Práctica de Laboratorio	Práctica de Laboratorio 31/03/2016
				(1,2) Tareas: p.328-342 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos y Resueltos (Rúbrica para Ejercicios de Programación) 05/04/2016
				(1,2) Tareas: p.366-372 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos y Resueltos (Rúbrica para Ejercicios de Programación) 07/04/2016
					Evaluación Práctica Progreso I (Rúbrica Evaluación Práctica) 12/04/2016

Semanas 18 de Abril al 4 de Junio					
Emplea las características propias de un lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales orientados a eventos	4. Acceso a Datos y Arquitectura en capas	3.4 Eventos y delegados 3.5 Colecciones de datos 3.6 Streams en .NET 3.7 Acceso a datos de archivos 4.1 Introducción a ADO.NET. Acceso a datos 4. 2. Arquitectura en capas Cliente-Servidor 4.3. Acceso conectado a base de datos 4.4. Acceso desconectado: Dataset y DataAdapters 4.5. Manejo de controles con base de datos	(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Prácticas de Laboratorio (1,2) Trabajo e Grupo		Ejercicios planteados por Docente, Capas y DATASET (Rúbrica para Ejercicios de Programación) 26/04/2016
				(1,2) Lectura Capítulo 20 y Tarea: p. 740 "Cómo Programar en C#"	Desarrollar el ejercicio completo "20.6.1 Mostrar una tabla de base de datos en un control DataGridView" (Rúbrica para Ejercicios de Programación) 28/04/2016
				(1,2) Tarea: p. 751 "Cómo Programar en C#"	Desarrollar el ejercicio completo "20.7 Consulta de la base de datos Libros" (Rúbrica para Ejercicios de Programación) 03/05/2016
				(1,2) Tarea: p. 751 "Cómo Programar en C#"	Desarrollar el ejercicio completo " 20.8 Programación con ADO.NET: caso de estudio de libreta de direcciones" (Rúbrica para Ejercicios de Programación) 05/05/2016

Emplea las características propias de un lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales orientados a eventos	4. Acceso a Datos y Arquitectura en capas	4.1. Introducción a ADO.NET. Acceso a datos 4. 2. Arquitectura en capas Cliente-Servidor 4.3. Acceso conectado a base de datos 4.4. Acceso desconectado: Dataset y DataAdapters 4.5. Manejo de controles con base de datos	(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Prácticas de Laboratorio (1,2) Trabajo e Grupo		Presentación del 50% del Sistema Informático (Rúbrica para Ejercicios de Programación) 10/05/2016 Visita Técnica
					Ejercicios planteados por Docente, CRUD BD/login (Rúbrica para Ejercicios de Programación) 12/05/2016
				(2) Informe de Práctica de Laboratorio	Práctica de Laboratorio 17/05/2016
				(1,2) Desarrollo Sistema Informático	Presentación Avance Sistema Informático (Rúbrica para Proyectos) 26/05/2016
					Evaluación Práctica Progreso II (Rúbrica Evaluación Práctica) 31/05/2016

Semanas 6 al 25 de Junio					
<p>Emplea las características propias de un lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales orientados a eventos</p>	<p>4. Acceso a Datos y Arquitectura en capas</p>	<p>4.6 Transaccionalidad 4.7 Introducción a LINQ</p>	<p>(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Trabajo e Grupo</p>	<p>(1,2)Desarrollo Sistema Informático</p>	<p>Ejercicios planteados por Docente, Transaccionalidad (Rúbrica para Ejercicios de Programación) Y Presentación Final del Sistema Informático y Documentación (Rúbrica para Proyectos) 14/06/2016</p> <p>Ejercicios planteados por Docente, LINQ (Rúbrica para Ejercicios de Programación) Y Evaluación Práctica Progreso III (Rúbrica Evaluación Práctica) 05/07/2016</p>

Normas y procedimientos para el aula

- a. 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.
- b. Si un estudiante llega dentro de los 10 primeros minutos, pero el docente se encuentra en clase, es responsabilidad total del alumno indicar y verificar que el docente le ponga asistencia.
- c. Los estudiantes deberán practicar la honestidad académica, no se admitirá por ningún motivo la copia parcial o total de ejercicios, exámenes, proyectos y todas las actividades de aprendizaje solicitadas por el docente, y se calificará con la mínima calificación (cero).
- d. 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.
- e. Solo se recibirán trabajos dentro del aula virtual, trabajo atrasado solo será recibido por causa de fuerza mayor comprobable en Secretaría Académica.
- f. Solo se puede comer fuera del aula.
- g. El estudiante puede acceder a tutoría personal en los horarios establecidos por el docente.
- h. En el caso de inasistencia es responsabilidad del estudiante igualarse en los contenidos de la materia dictada en dicha clase.
- i. 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, si no se justifica.
- j. Si el estudiante no asiste a la práctica de laboratorio, no podrá realizar el informe del mismo.
- k. El informe relacionado con la práctica de laboratorio, debe contener pantallas como evidencias de haber completado la práctica.
- l. Es responsabilidad total del alumno subir tareas, exámenes, prácticas y demás asignaciones del docente, subir correcta y de forma completa al apoyo virtual.
- m. Se tomarán muy en cuenta las faltas ortográficas, las cuales significarán disminución en la calificación final.

9. Referencias bibliográficas

a. Principales.

Ceballos, J. (2013). *Enciclopedia de Microsoft Visual C#*. (3era ed). México: Editorial Alfaomega. ISBN 978-607-707-024-5 (Físico)

Bell, D. y Parr, M. (2010). *C# Para Estudiantes*. (2da ed). México: Pearson Educación, ISBN 9786073203289.(Digital).

b. Referencias complementarias.

Deitel, H. y Deitel, P.,(2012) . *Cómo programar en C#*. (5 ed). Inglaterra: Pearson Education, ISBN 97802737933004 (Digital).

Microsoft Corp. (2013), Academia Virtual Microsoft , Recuperado 06/03/15 de <http://www.microsoftvirtualacademy.com/training-courses/desarrollo-en-microsoft-visual-studio>.

10. Perfil del docente

Nombre de docente: Carlos Muñoz Cueva

Maestría en Gerencia de Sistemas, Maestría en Finanzas Empresariales

Más de 10 años de trabajo en el ámbito informático en Instituciones Financieras

Contacto: ca.munoz@udlanet.ec

11. Rúbrica del Sistema Informático Final

Criterio	Peso	Excelente	Bueno	Regular	Insuficiente	Nada
		100%	75%	50%	25%	0%
Funcionalidad (Casos de Uso)	2	Si cumplen con todas las funcionalidades anunciadas en el documento de especificación	Cumple con al menos 50% de la funcionalidad indicada en el documento de especificaciones	Cumple con la cuarta parte de la funcionalidad indicada en el documento de especificaciones	No cumple con la funcionalidad indicada en el documento de especificaciones	No presenta nada
Interfaz de Software	2	La interface gráfica es completamente amigable con el usuario. Utiliza orden en de tabulación de los controles, fácilmente se puede usar teclado y ratón. Utiliza diseño especial, decoraciones para resaltar el programa. La interacción con usuario es adecuada: Mensajes, tooltips, ayudas, menús, tipos de ventanas, manejo de perfiles	La interface gráfica es amigable con el usuario pero hay errores de en la interfaz, errores de tabulación, faltas de ortografía en las pantalla. La interacción con usuario: Mensajes, tooltips, ayudas, menús, tipos de ventanas, manejo de perfiles, no es adecuada.	La interface es muy poco amigable con el usuario y la interacción con usuario: Mensajes, tooltips, ayudas, menús, tipos de ventanas, manejo de perfiles, no es adecuada.	La interface gráfica no es amigable con el usuario. Funciones incorrectas, errores de interfaz, errores de inicialización.	No presenta nada
Programación	3	Identifico adecuadamente las capas del proyecto: Datos, Negocios y Cliente, y las clases están creadas correctamente. Todos Los eventos son adecuados, código fuente es legible, y están documentados, comentarios adecuados al programa. Usa conectividad en capas parametrizada Siempre usa los controles de usuario en la interfaz de usuario. Maneja los errores, y los mensajes son adecuados al usuario del aplicativo Valida adecuadamente todos los datos de entrada, especificando los mensajes de validación.	Identifico adecuadamente las capas del proyecto: Datos, Negocios y Cliente, pero la programación no es la adecuada. Los eventos son adecuados, pero su código fuente no es legible Usa de manera esporádica los controles de usuario en la interfaz. Usa conectividad en capas pero esta no es parametrizada Maneja errores en algunas ocasiones y los mensajes no son los adecuados Valida algunos datos de entrada, pero no especifica los mensajes de validación	No tiene las 3 capas necesarias del modelo. No todas las fuentes están codificadas correctamente. Muy poco manejo de errores. Valida muy poco la entrada de Datos.	Los eventos no son adecuados No se conecta a la base de datos No usa controles de usuario No aplica modelo en capas No maneja errores , ni valida los datos de entrada	No presenta nada
Dominio del Proyecto	2	El dominio de la programación es clara, contundente, y bien fundamentada	Aunque la programación es clara no demuestra un dominio claro de la programación	El dominio de la programación no es suficiente	No Conoce el proyecto	No presenta nada
Originalidad, diseño e innovación	1	El programa demuestra innovación y originalidad.	Su diseño es vistoso pero no lo suficiente	El programa demuestra originalidad aunque no es innovador	Su diseño es poco vistoso	No presenta nada