

Facultad de Ingeniería y Ciencias Agropecuarias
Ingeniería en Sistemas de Computación e Informática
ACI520 - Lenguaje de Programación I
 Período 2017-2

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 Sistemas	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 () Medio (X) 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 ()

Resultados de aprendizaje (RdA)	RdA perfil de egreso de carrera Electrónica	Nivel de desarrollo (carrera)
1. Explica los conceptos de la programación orientada a eventos, a través de un lenguaje de programación.	Identifica oportunidades para mejorar el desempeño de las comunicaciones en las organizaciones a través de la incorporación y uso eficiente de plataformas de servicios de redes	Inicial () Medio (X) Final ()
2. Emplea las características propias de un lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales orientados a eventos	Identifica oportunidades para mejorar el desempeño de las comunicaciones en las organizaciones a través de la incorporación y uso eficiente de plataformas de servicios de redes	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	10%
Práctica de Laboratorio	20%
Avance Sistema Informático	30%
Evaluación Práctica	40%
Evaluación final	30%
Sub componentes	
Sistema Informático Final	40%
Evaluación Práctica	60%

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 ldocumentación 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
	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 6 de Marzo al 22 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#” (2) Informe de Lectura	Presentación Informe IEEE 13/03/2017
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) 16/03/2017
				(1,2) Tareas: p.165-168 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos (Rúbrica para Ejercicios de Programación) 20/03/2017

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) 23/03/2017
				(1,2) Tareas: p.274-290 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos y Resueltos (Rúbrica para Ejercicios de Programación) 27/03/2017
				(2)Informe de Práctica de Laboratorio	Práctica de Laboratorio 30/03/2017
				(1,2) Tareas: p.328-342 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos y Resueltos (Rúbrica para Ejercicios de Programación) 03/04/2017
				(1,2) Tareas: p.366-372 “Enciclopedia Microsoft Visual C#”	Desarrollo de Ejercicios Propuestos y Resueltos (Rúbrica para Ejercicios de Programación) 06/04/2017
					Evaluación Práctica Progreso I (Rúbrica Evaluación Práctica) 10/04/2017

Semanas 24 de Abril al 3 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>3.4 Eventos y delegados 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</p>	<p>(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Prácticas de Laboratorio (1,2) Trabajo e Grupo</p>		<p>Ejercicios planteados por Docente, Capas y DATASET (Rúbrica para Ejercicios de Programación) 27/04/2017</p>
				<p>(1,2) Lectura Capítulo 20 y Tarea: p. 740 "Cómo Programar en C#"</p>	<p>Desarrollar el ejercicio completo "20.6.1 Mostrar una tabla de base de datos en un control DataGridView" (Rúbrica para Ejercicios de Programación) 04/05/2017</p>
				<p>(1,2) Tarea: p. 751 "Cómo Programar en C#"</p>	<p>Desarrollar el ejercicio completo "20.7 Consulta de la base de datos Libros" (Rúbrica para Ejercicios de Programación) 08/05/2017</p>
				<p>(1,2) Tarea: p. 751 "Cómo Programar en C#"</p>	<p>Desarrollar el ejercicio completo " 20.8 Programación con ADO.NET: caso de estudio de libreta de direcciones" (Rúbrica para Ejercicios de Programación) 11/05/2017</p>

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) 15/05/2017
					Ejercicios planteados por Docente, CRUD BD/login (Rúbrica para Ejercicios de Programación) 18/05/2017
				(2) Informe de Práctica de Laboratorio	Práctica de Laboratorio 18/05/2017
				(1,2) Desarrollo Sistema Informático	Presentación Avance Sistema Informático (Rúbrica para Proyectos) 22/05/2017
					Evaluación Práctica Progreso II (Rúbrica Evaluación Práctica) 29/05/2017

Semanas 5 al 24 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 5. Introducción Windows Presentation Foundation</p>	<p>4.6 Transaccionalidad 4.7 Introducción a LINQ 5.1 Características y propiedades WPF</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) 19/06/2017</p> <p>Ejercicios planteados por Docente, LINQ (Rúbrica para Ejercicios de Programación) Y Evaluación Práctica Progreso III (Rúbrica Evaluación Práctica)26/06/2017</p> <p>Presentación Final del Sistema Informático y Documentación (Rúbrica para Proyectos) 03/07/2017</p>
---	--	--	--	--	--

9. 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.

10. Referencias bibliográficas

a. Principales.

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

b. Referencias complementarias.

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

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

11. 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

12. Proyecto Final y Rúbrica

Mecanismo de Evaluación

Proyecto Final: Desarrollar un Sistema Informático en Windows Forms en 3 capas, con interacción de base de datos, usando una metodología orientada a eventos, en lenguaje de programación C# sobre los requerimientos del Negocio asignado a cada grupo en clase.

Criterio	Satisfactorio	Bueno	Regular	Insatisfactorio	Malo
	4	3	2	1	0
Análisis del problema planteado(15%)	Identifica todos los requerimientos definidos en el documento de especificación de requerimientos del proyecto y es completamente funcional.	Identifica al menos el 75% de los requerimientos definidos en el documento de especificación de requerimientos del proyecto y es completamente funcional.	Identifica al menos el 50% de los requerimientos definidos en el documento de especificación de requerimientos del proyecto y es completamente funcional.	Identifica menos de la mitad de los requerimientos definidos en el documento de especificación de requerimientos del proyecto y es completamente funcional.	No presenta
Diseño de la solución del problema(20%)	La interface gráfica es amigable con el usuario en el 100% de la funcionalidad del proyecto. Utiliza los siguientes elementos: Tabulación de los controles, Se usa teclado y ratón de manera indistinta. Utiliza diseño especial, decoraciones para resaltar el programa. Usa Mensajes, tooltips, ayudas, menús, tipos de	La interface gráfica es amigable con el usuario en el 75% de la funcionalidad del proyecto. Utiliza los siguientes elementos: Tabulación de los controles, Se usa teclado y ratón de manera indistinta. Utiliza diseño especial, decoraciones para resaltar el programa. Usa Mensajes, tooltips, ayudas, menús, tipos de	La interface gráfica es amigable con el usuario en el 50% de la funcionalidad del proyecto. Utiliza los siguientes elementos: Tabulación de los controles, Se usa teclado y ratón de manera indistinta. Utiliza diseño especial, decoraciones para resaltar el programa. Usa Mensajes, tooltips, ayudas, menús, tipos de	La interface gráfica es amigable con el usuario menos de la mitad de la funcionalidad del proyecto. Utiliza los siguientes elementos: Tabulación de los controles, Se usa teclado y ratón de manera indistinta. Utiliza diseño especial, decoraciones para resaltar el programa. Usa Mensajes, tooltips, ayudas, menús, tipos de	No presenta

	ventanas, manejo de perfiles.	ventanas, manejo de perfiles.	ventanas, manejo de perfiles.	ventanas, manejo de perfiles.	
Implementación de la solución al tema planteado(30%)	Identifica las capas del proyecto: Datos, Negocios y Cliente, y las clases están creadas correctamente en el 100% de la funcionalidad. Todos Los eventos son adecuados, código fuente es legible, y están documentados, comentarios adecuados al programa. Usa conectividad en capas parametrizada Usa los controles de usuario en la interfaz. Maneja los errores, y los mensajes son adecuados al usuario del aplicativo Valida adecuadamente todos los datos de entrada, especificando los mensajes de validación.	Identifica las capas del proyecto: Datos, Negocios y Cliente, y las clases están creadas correctamente en el 75% de la funcionalidad. Todos Los eventos son adecuados, código fuente es legible, y están documentados, comentarios adecuados al programa. Usa conectividad en capas parametrizada Usa los controles de usuario en la interfaz. Maneja los errores, y los mensajes son adecuados al usuario del aplicativo Valida adecuadamente todos los datos de entrada, especificando los mensajes de validación.	Identifica las capas del proyecto: Datos, Negocios y Cliente, y las clases están creadas correctamente en la mitad de la funcionalidad. Todos Los eventos son adecuados, código fuente es legible, y están documentados, comentarios adecuados al programa. Usa conectividad en capas parametrizada Usa los controles de usuario en la interfaz. Maneja los errores, y los mensajes son adecuados al usuario del aplicativo Valida adecuadamente todos los datos de entrada, especificando los mensajes de validación.	Identifica las capas del proyecto: Datos, Negocios y Cliente, y las clases están creadas correctamente en menos de la mitad de la funcionalidad. Todos Los eventos son adecuados, código fuente es legible, y están documentados, comentarios adecuados al programa. Usa conectividad en capas parametrizada Usa los controles de usuario en la interfaz. Maneja los errores, y los mensajes son adecuados al usuario del aplicativo Valida adecuadamente todos los datos de entrada, especificando los mensajes de validación.	No presenta
Dominio del Proyecto(20%)	El Dominio de la programación visual y orientada a eventos es integral y pertinente.	El Dominio de la programación visual y orientada a eventos es adecuada.	El Dominio de la programación visual y orientada a eventos es breve.	El Dominio de la programación visual y orientada a eventos es superficial.	No presenta

Documentación(15%)	Presenta la documentación técnica del proyecto de manera completa: Manual de usuario Manual de Instalación Manual de diseño	Presenta la documentación técnica del proyecto de manera adecuada: Manual de usuario Manual de Instalación Manual de diseño	Presenta la documentación técnica del proyecto de manera incompleta, solo 2 de los 3 documentos.	Presenta la documentación técnica del proyecto de manera incompleta, solo 1 de los 3 documentos.	No presenta
---------------------------	--	--	--	--	-------------