

Facultad de Ingeniería y Ciencias Agropecuarias
Ingeniería en Sistemas de Computación e Informática
Código del curso ACI520 y Asignatura Lenguaje de Programación I
Período 2016-1

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 proyecto 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	
Prueba Teórica	3.5%
Portafolio de Ejercicios	5.25%
Práctica de Laboratorio	5.25%
Ejercicios en Clase	7%
Evaluación Práctica	14%
Reporte de progreso 2	35%
Sub componentes	
Portafolio de Ejercicios	3.5%
Práctica de Laboratorio	5.25%
Ejercicios en Clase	3.5%
Avance Parcial Proyecto	8.75%
Evaluación Práctica	14%
Evaluación final	30%
Sub componentes	
Documentación Proyecto Final y Portafolio de Ejercicios	6%

Proyecto Final	9%
Evaluación Práctica	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.

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 5.25%: 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 3.5-5.25%: 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 3.5-5.25%: El estudiante deberá resolver los ejercicios e investigaciones indicadas por el docente, ejercicios prácticos rendidos en clases y subirlos a la plataforma virtual.

Avance Parcial del Proyecto 8.75%: El estudiante deberá presentar un avance de su proyecto final.

Documentación Proyecto Final 6%: El estudiante deberá presentar la documentación del proyecto final 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 y Modelo en capas	1.1. Arquitectura de .NET: CLR, Lenguaje Intermedio Y CLS, Biblioteca de Clases de .NET, Acceso a datos con ADO.NET, LINQ
		1.2. Modelado por capas
	2. Conceptos de Programación Visual C#	2.1 Sintaxis de C#
		2.2 Sistemas de tipos complejos: Enumeraciones, array
		2.3. Manejo de Clases y Objetos.
Emplea las características propias de un lenguaje de programación en la solución de problemas que necesitan aplicaciones computacionales orientados a eventos		2.4 Manejo de excepciones
	3. Desarrollo de aplicaciones en Visual C# y Librerías de clases .NET	3.1 Eventos y delegados
		3.2 Uso del diseñador de Visual Studio
		3.3. Diseño de Interfaz de usuario
		3.4 Manejo de controles comunes de Windows Forms
		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 Septiembre al 19 de Octubre

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	1.1. Arquitectura de .NET: CLR, Lenguaje Intermedio Y CLS, Biblioteca de Clases de .NET, Acceso a datos con ADO.NET, LINQ 1.2. Modelado por capas	(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Prácticas de Laboratorio (1,2) Trabajo en Grupo	Lectura Cursos On Line: Desarrollo en Microsoft Visual Studio http://www.microsoftvirtualacademy.com/training-courses/desarrollo-en-microsoft-visual-studio Y Desarrollando en punto net con Visual Studio http://www.microsoftvirtualacademy.com/training-courses/desarrollo-net	Módulos I, II, VIII, IX, XIX Capítulos II, V, VI, VII, VIII Prueba teórica de los temas revisados 21/09/2015
				(1,2) Tareas: p. 22 "Libro C# para Estudiantes"	Desarrollo de 5 primeros Ejercicios PARES (Rúbrica para Ejercicios de Programación) 23/09/2015
	2. Conceptos de Programación Visual C#	2.1 Sintaxis de C# 2.2 Sistemas de tipos complejos: Enumeraciones, array 2.3. Manejo de Clases y Objetos.		(1,2) Tareas: p.256 "Libro C# para Estudiantes"	Desarrollo de 5 primeros Ejercicios PARES (Rúbrica para Ejercicios de Programación) 23/09/2015

Explica los conceptos de la programación orientada a eventos, a través de un lenguaje de programación.	2. Conceptos de Programación Visual C#	2.4 Manejo de excepciones	(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Prácticas de Laboratorio (1,2) Trabajo en Grupo	(1,2) Tareas: p.271 "Libro C# para Estudiantes"	Desarrollo de 5 primeros Ejercicios IMPARES (Rúbrica para Ejercicios de Programación) 28/09/2015
		2.1 Sintaxis de C# 2.2 Sistemas de tipos complejos: Enumeraciones, array 2.3. Manejo de Clases y Objetos. 2.4 Manejo de excepciones		(1,2) Tareas: p.310 "Libro C# para Estudiantes"	Desarrollo de 5 primeros Ejercicios PARES (Rúbrica para Ejercicios de Programación) 28/09/2015
				(1,2) Tareas: p.310 "Libro C# para Estudiantes"	Desarrollo de 5 primeros Ejercicios IMPARES (Rúbrica para Ejercicios de Programación) 30/09/2015
				(1,2) Tareas: p.239 "Libro C# para Estudiantes"	Desarrollo de TODOS los Ejercicios(Rúbrica para Ejercicios de Programación) 30/09/2015
				(1,2) Lectura Capítulo 13 y Tarea: p.420-448 "Cómo Programar en C#"	Se deben crear ejemplos similares (NO LOS MISMOS) a los desplegados en el libro en programación C# windows forms. (Rúbrica para Ejercicios de Programación) 05/10/2015
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 Eventos y delegados 3.2 Uso del diseñador de Visual Studio 3.3. Diseño de Interfaz de usuario 3.4 Manejo de controles comunes de Windows Forms			Ejercicios planteados por Docente, Transferencia de Variables (Rúbrica para Ejercicios de Programación) 07/10/2015

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 Eventos y delegados 3.2 Uso del diseñador de Visual Studio 3.3. Diseño de Interfaz de usuario 3.4 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) Lectura Capítulo 14 y Tarea: p.452-506“Cómo Programar en C#”	Se deben crear ejemplos similares (NO LOS MISMOS) a los desplegados en el libro en programación C# windows forms. (Rúbrica para Ejercicios de Programación) 07/10/2015
					Consulta Eventos y Delegados y Consulta Abrir una solo instancia de forma (Rúbrica para Ejercicios de Programación) 12/10/2015
				(2)Informe de Práctica de Laboratorio	Práctica de Laboratorio 14/10/2015
					Evaluación Práctica Progreso I (Rúbrica Evaluación Práctica) 19/10/2015
Semanas 4 de noviembre al 14 de diciembre					
Emplea las características propias de un lenguaje de programación en la solución de problemas que	4. Acceso a Datos y Arquitectura en capas	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	(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Prácticas de Laboratorio (1,2) Trabajo e Grupo	(1,2) Tareas: p.333 “Libro C# para Estudiantes”	Desarrollo de 10 ejercicios (Rúbrica para Ejercicios de Programación) 09/11/2015

necesitan aplicaciones computacionales orientados a eventos	4. Acceso a Datos y Arquitectura en capas	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) 11/11/2015
				(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) 16/11/2015
				(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) 18/11/2015
				(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) 23/11/2015
					Presentación del 50% del Proyecto (Rúbrica para Ejercicios de Programación) 23/11/2015

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		Ejercicios planteados por Docente, CRUD BD/login (Rúbrica para Ejercicios de Programación) 30/11/2015
				(2)Informe de Práctica de Laboratorio	Práctica de Laboratorio 02/12/2015
					Evaluación Práctica Progreso II (Rúbrica Evaluación Práctica) 07/12/2015
				(1,2)Desarrollo Proyecto Parcial	Presentación Avance Parcial del Proyecto (Rúbrica para Proyectos) 09/12/2015
Semanas 4 al 11 de Enero					
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	4.6 Transaccionalidad 4.7 Introducción a LINQ	(1) Instrucción Directa (1) Ejercicios en Clase (1,2) Prácticas de Laboratorio (1,2) Trabajo e Grupo	(1,2)Desarrollo Proyecto Parcial	Ejercicios planteados por Docente,Transaccionalidad (Rúbrica para Ejercicios de Programación) Y Presentación Final del Proyecto (Rúbrica para Proyectos) 4/01/2016 Ejercicios planteados por Docente, LINQ (Rúbrica para Ejercicios de Programación) Y Evaluación Práctica Progreso III (Rúbrica Evaluación Práctica) 11/01/2016

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.

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

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

b. Referencias complementarias.

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

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

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

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

Horario de atención al estudiante: Solo en Universidad

ANEXOS

Guía de Práctica de Laboratorio 1 – ACI520



FACULTAD DE INGENIERÍA
Y CIENCIAS AGROPECUARIAS

ACI520 - LENGUAJE DE PROGRAMACIÓN I

GUÍA PARA PRÁCTICAS DE LABORATORIO

INGENIERIA EN SISTEMAS DE
COMPUTACION E INFORMATICA

Presentación

La presente Guía para Prácticas de Laboratorio ha sido desarrollada para que los estudiantes de la asignatura de LENGUAJE DE PROGRAMACION I dispongan de la información necesaria para la realización de las prácticas correspondientes de acuerdo a los temas, objetivos y resultados de aprendizaje definidos. En este documento se incluye el proceso en el laboratorio de experimentación e investigación, con los respectivos recursos y resultados esperados, para que el estudiante pueda desarrollar su práctica-taller y la elaboración de sus respectivos informes o cualquier otra evidencia de aprendizaje, que serán evaluadas con las rúbricas que constan en los anexos.

La Guía presenta una secuencia donde se especifica cada sesión de prácticas en laboratorio con su respectivo proceso didáctico y formativo.

SIGLA	ASIGNATURA	PARALELO	PERIODO
ACI520	LENGUAJE DE PROGRAMACIÓN I	1	2015-1
FECHA	TEMA	PRACTICA No	SESIONES
16/10/2014	Controles Visuales	1	1

1.- OBJETIVO

- Realizar un proyecto Windows Forms donde aplicarán los conceptos sobre la Programación Visual con clases, menús y listas.

2.- RESULTADOS DE APRENDIZAJE ESPERADOS

Con la ejecución de la práctica el estudiante fortalecerá las siguientes habilidades:

- Utilizar controles comunes para representar procesos.
- Interactuar con clases y objetos para representar una funcionalidad.
- Utilizar listas para transferir información entre formas dentro de un MDI.

Los resultados de aprendizaje listados apoyan a la consecución del primer resultado de aprendizaje general de la materia:

- Desarrolla aplicaciones visuales orientados eventos, con colecciones de datos y manejo de archivos.

3.- MATERIALES/RECURSOS Y EQUIPOS

Para la ejecución de la práctica verifique que disponga de los siguientes recursos:

1. Computador de laboratorio con acceso a máquina virtual con el siguiente software:
 - a. Microsoft Visual Studio 2010.
 - b. Internet para Aula Virtual
2. Guía de laboratorio.

4.- ACTIVIDAD FORMATIVA

Prerrequisitos:

Para la elaboración de la práctica, es indispensable que el estudiante:

- Revise los ejercicios referentes a Controles Comunes y MDI
- Revisar ejercicios de clases y objetos
- Realice una lectura previa de la guía completa de la práctica.

Descripción de la actividad:

Escenario:

Se requiere crear un programa en Windows forms que maneje una tienda de video, donde se renten películas.

Proceso de la práctica:

Para cumplir con los requerimientos de la práctica se deben realizar las siguientes actividades:

1. Debe haber un MDI que contenga las demás formas.
2. Debe existir una enumeración para los estados de las películas: Disponible, Rentada.
3. Debe crearse la clase Película.
4. La clase película debe tener los siguientes atributos: nombrePeli, cantidad, estado
5. En el MDI deben existir menús con: Ingreso de películas, Renta Película, Devolución Películas, Reporte y Salir.
6. En la primera pantalla de ingreso de películas se ingresa en controles necesarios: nombre película y cantidad, estado es asignado como Disponible con enumeración y mediante un botón debe ingresarse cada película en una lista de películas List<clsPelicula>
7. En la segunda pantalla de Renta Película se recupera las películas ingresadas en la lista (como el ejemplo del apoyo virtual Ejemplo Listas con MDI) en un combobox/listbox el nombre, al escoger la película

- automáticamente en un primer label aparecerán las unidades disponibles y en un segundo label el estado disponible si existen unidades o Rentada si las disponibles son cero.
8. En la misma pantalla junto a los labels antes mencionados debe haber un botón llamado rentar el cual va disminuyendo las unidades disponibles de cada película(se actualiza atributo cantidad de la película seleccionada en la lista), al cambiar de selección en el combobox/listbox se actualizarán los labels, si el estado es rentada se deshabilita el botón rentar.
 9. En pantalla Devolución películas, de igual manera se recupera la lista modificada de películas en un combobox/listbox, con los mismos labels de cantidad de disponibles y estado, pero con botón devolver, el cual aumenta una unidad en disponibles de cada película (se actualiza lista), botón se deshabilita si estado cambia a Disponible
 10. En pantalla reporte se despliega en un mismo listbox las películas, su cantidad y estado.
 11. El programa no termina hasta escoger el menú SALIR.
 12. Debe ponerse al menos 2 Excepciones.

Preguntas de la Práctica:

1. Entre las opciones para habilitar/deshabilitar un botón, cuáles encontró?
2. Investigue otra forma de controlar ingreso solo numérico de datos que no sea maskedtextbox y ponga un ejemplo.
3. Sí al borrar elementos de un listbox solo se borran visualmente, como haría para que también se borren de la lista de objetos List<>();
4. Implemente el método Tryparse en alguna de las conversiones, despléguelo aquí.
5. En qué parte del proceso tuvo mayor dificultad?
6. Cuál cree que son las propiedades más usadas de los controles tipo lista?
7. Cuál de los controles tipo lista le parece más útil para su proyecto parcial?
8. Investigue elementos de enmascaramiento dentro de la propiedad Mask del Maskedtextbox, cuando es 9 , cuando es 0, la coma, el punto, etc.?
9. Resuelva el inconveniente de las máscaras y decimales, ponga un ejemplo.
10. Dentro del combobox como resolvió la asignación de valores e índices?
11. Le parece que el uso de listas es una buena alternativa de estructura de datos antes las bases de datos?

6.- REFERENCIAS BIBLIOGRÁFICAS

- Cevallos, Javier (2013). Enciclopedia de Microsoft Visual C#, México, Editorial Alfaomega, ISBN 978-607-707-024-5 (Libro Físico)

7.- MECANISMO DE EVALUACIÓN Y ANEXOS

El puntaje asignado al estudiante por la elaboración de la práctica, será el obtenido por la presentación del informe de la práctica realizada, el cual será calificado con base a la rúbrica correspondiente.

No se recibirán informes a estudiantes que no hayan asistido puntualmente a la práctica de laboratorio.

Guía de Práctica de Laboratorio 2 – ACI520



FACULTAD DE INGENIERÍA
Y CIENCIAS AGROPECUARIAS

ACI520 - LENGUAJES DE PROGRAMACION I

GUÍA PARA PRÁCTICAS DE LABORATORIO

INGENIERIA EN SISTEMAS

ASIGNATURA: Lenguajes de Programación		SIGLA: ACI520	PARALELO: 1	PERIODO: 2015-2
Fecha: XXXXX	PRÁCTICA No: 2	SESIONES: 1	TEMA: Aplicativo en capas con manejo de Base de datos	

Presentación

La presente Guía para Prácticas de Laboratorio ha sido desarrollada para que los estudiantes de la asignatura de ACI520 dispongan de la información necesaria para la realización de las prácticas correspondientes de acuerdo a los temas, objetivos y resultados de aprendizaje definidos. En este documento se incluye el proceso en el laboratorio de experimentación e investigación, con los respectivos recursos y resultados esperados, para que el estudiante pueda desarrollar su práctica-taller y la elaboración de sus respectivos informes o cualquier otra evidencia de aprendizaje, que serán evaluadas con las rúbricas que constan en los anexos.

La Guía presenta una secuencia donde se especifica cada sesión de prácticas en laboratorio con su respectivo proceso didáctico y formativo.

1.- OBJETIVO

Crear aplicaciones Windows Forms con conexión a Base de datos SqlServer, aplicando arquitectura en capas en C# .NET.

2.- RESULTADOS DE APRENDIZAJE ESPERADOS

El estudiante podrá:

- Desarrollar aplicaciones Windows Forms en C# con manipulación de base de datos SQL Server 2008 o superior, usando el componente ADO.NET y arquitectura 3 capas, Capa de base de datos, capa de negocios, y capa del cliente.

2.- MATERIALES/RECURSOS Y EQUIPOS

Para realizar la práctica se requiere:

Visual Studio .Net 2010 o superior: Necesario para la programación del sistema
SQLServer 2008 o superior para la base de datos.

3.- ACTIVIDAD FORMATIVA

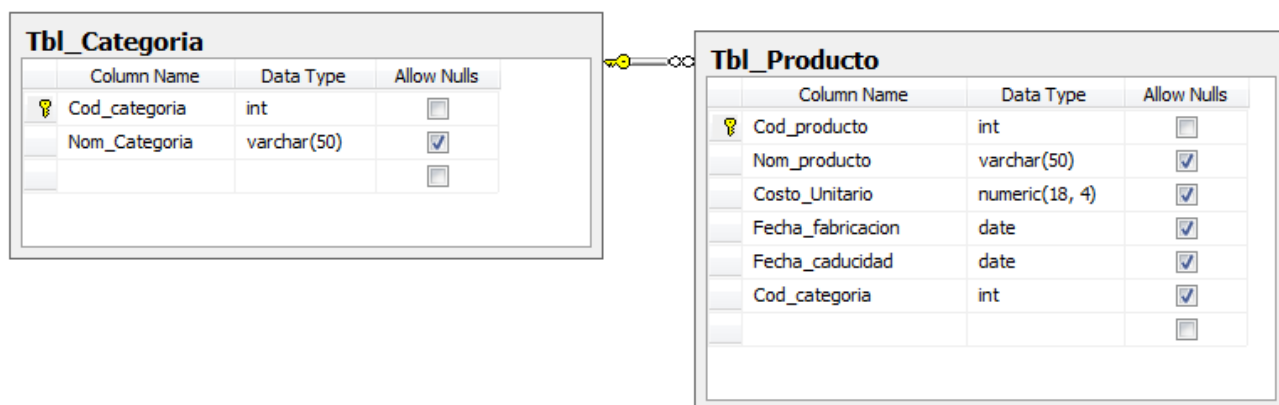
Prerrequisitos:

Para realizar esta práctica el estudiante debe dominar:

1. Arquitectura de .NET: CLR, Lenguaje Intermedio Y CLS, Biblioteca de Clases de .NET, Acceso a datos con ADO.NET, LINQ
2. Sintaxis de C#
3. Sistemas de tipos complejos: Enumeraciones, Array
4. Manejo de Clases y Objetos.
5. Manejo de excepciones
6. Eventos
7. Uso del diseñador de Visual Studio: IDE
8. Diseño de Interfaz de usuario
9. Manejo de controles comunes de Windows Forms.
10. Colecciones de datos
11. Streams en .NET
12. Acceso a datos de archivos
13. Arquitectura en capas Cliente-Servidor
14. Acceso conectado a base de datos
15. Acceso desconectado: Dataset y DataAdapters
16. Manejo de controles con base de datos

Descripción de la actividad:

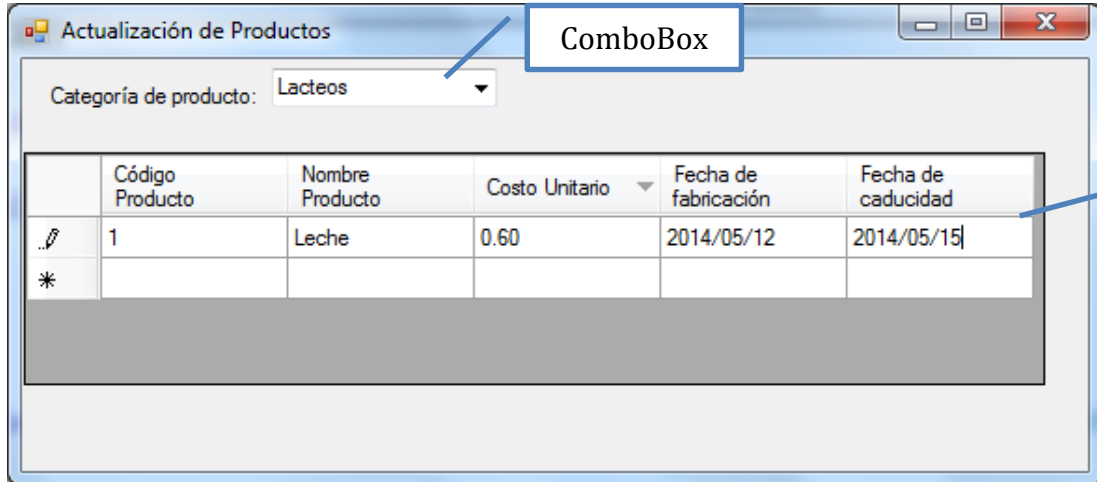
En SQLServer 2008 crear una base de datos LAB1, crear las tablas *Tbl_Categoria* y *Tbl_Producto* siguientes con su respectiva relación.



Ingresar datos a las tablas para realizar las pruebas del aplicativo.

Crear un proyecto de Windows llamado *Laboratorio1* usando C# de Visual Studio 2010 o superior, crear los siguientes formularios.

Formulario 1:



	Código Producto	Nombre Producto	Costo Unitario	Fecha de fabricación	Fecha de caducidad
✎	1	Leche	0.60	2014/05/12	2014/05/15
*					

Para enlazar datagridview SIN smartag se lo puede hacer de la siguiente manera:

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    DataSet ds = new DataSet();

    ds=claseNegocio.consultaUsuariosBase(int.Parse(comboBox1.SelectedValue.ToString()));
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "NombreTabla";
}
```

Otra forma para realizar una consulta directamente con condición where:

```
public System.Data.DataSet consultaUsuariosBase(int codCatego)
{
    SqlDataAdapter da;
    DataSet ds = new DataSet();

    try
    {
        conexion.Open();
        string sql = "select * from Producto where cod_categoria= " +
        codCatego;

        da = new SqlDataAdapter(sql, conexion);
        da.Fill(ds, "Producto");

        return ds;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        conexion.Close();
    }
}
```

}

Formulario 2:

Actividades:

1. Crear un proyecto en 3 capas, *capa de Base de datos, capa de Negocios y la capa de Interfaz.*
2. Crear las clases que involucra los objetos del proyecto: *Producto y Categoría*

Formulario 1:

3. Cuando cargue el formulario debe llenar el ComboBox con, los datos de la tabla *Tbl_categoria*.
4. Cuando se seleccione un elemento del ComboBox en el DataGridView debe desplegar los producto correspondientes a esa categoría que se encuentran almacenados en la bases de datos.
5. Cuando se selecciona un registro del DataGridView y se presiona Enter o se realiza Doble Click con el ratón se desplegará el formulario 2 con los datos del producto seleccionado.

Formulario 2:

6. Se debe controlar que la *fecha de caducidad* no sea menor a la *fecha de fabricación*
7. El campo *precio unitario* solo debe admitir valores numéricos.
8. El campo código del Producto no es editable.
9. Al presionar el botón Grabar los datos serán grabados en la base de datos se desplegará el mensaje de confirmación de la acción, y se regresará al formulario 1, con la información del grid actualizada.

6.- REFERENCIAS BIBLIOGRÁFICAS

1. Bell D. y Parr M.L, (2010). C# Para Estudiantes. Mexico Mexico. Pearson. ISBN 978-607-32-0328-9. (Digital)
2. Deitel H. y Deitel P. (2007). Como Programar en C#. Mexico Mexico. Pearson. ISBN 970-26-1056-7.(Digital).
3. Microsft Developer Network, Recuperado el 06-Marzo del 2014 de <http://msdn.microsoft.com/es-es/library/ms123401.aspx>

7.- MECANISMO DE EVALUACIÓN Y ANEXOS

El puntaje asignado al estudiante por la elaboración de la práctica, será el obtenido por la presentación del informe de la práctica realizada, el cual será calificado con base a la rúbrica correspondiente.

No se recibirán informes a estudiantes que no hayan asistido puntualmente a la práctica de laboratorio.

