

	UNIVERSIDAD DISTRITAL "Francisco José de Caldas" Facultad de Ingeniería Ingeniería Eléctrica		
	Elaboró	Jhon Felipe Galvis	Fecha de Elaboración
Revisó	[Escriba aquí el nombre]	Fecha de Revisión	junio de 2010

1 DATOS DE IDENTIFICACIÓN

Nombre del espacio académico:	Programación Orientada a Objetos		
Pensum al que pertenece	2		
Código	702007		
Créditos Académicos	4		
Número de Horas Semanales	HTD 4	HTC 2	HTA 6
Modalidad	Asignatura		
Área	Ciencias Básicas		

2 PREGUNTAS QUE BUSCA RESOLVER

[Identificar aquellos problemas específicos que el curso de formación espera dar respuesta]

3 JUSTIFICACIÓN

La ingeniería de software ha diseñado una forma diferente para desarrollar aplicaciones informáticas, que consiste en visualizar el mundo tal y como es, es decir, colecciones de objetos que se pueden representar de forma general incluyendo sus características (atributos) y sus respectivas operaciones (funciones miembro). Aunque este paradigma Orientado a Objetos puede ser soportado en cierto grado en los lenguajes tradicionales; los lenguajes orientados a objetos proporcionan el esqueleto en la funcionalidad de los objetos, clases, métodos y mensajes, para hacer posible la creación de programas que sean más exactos y menos complejos. La orientación de objetos es un nivel de abstracción de computadora más allá del de los procedimientos y datos, anima al desarrollador de software a concentrarse en los temas importantes e ignorar el resto. Por tal motivo el ingeniero eléctrico debe estar en capacidad de generar soluciones acordes con la tecnología que va a encontrar en el mercado laboral con el ánimo de satisfacer sus necesidades informáticas.

La programación orientada a objetos permite, entre otros aspectos, escribir programas robustos debido a la posibilidad de modularizar y reutilizar código. Es por tanto necesario que todos los que se dediquen a realizar software en cualquier área, además de los conceptos de diseño y modelado, conozcan los principios de la orientación a objetos, el código, la sintaxis y la semántica de un lenguaje

de programación orientado a objetos con los cuales desarrollen habilidades para la construcción de software con múltiples propósitos.

La computación aplicada y avanzada redundante en grandes beneficios, en tanto como campo de investigación fundado en diversas áreas de las matemáticas, la técnica y los nuevos desarrollos, brinda a los futuros profesionales las bases teóricas y prácticas necesarias para su posterior desenvolvimiento en el desarrollo de sus propuestas investigativas y laborales; propende por materializar los conceptos teóricos del ciclo básico profesional en la comprensión, contextualización y conceptualización de los elementos constitutivos de la computación aplicada a la industria y en generar un nuevo valor en nuestro proceso de formación integral como profesionales, que dé cuenta de comprensión de los alumnos frente a la materia.

Se trata de brindar al estudiante las herramientas teórico - prácticas que le permitan asumir nuevos derroteros para la generación de propuestas en el área de la computación, uso y aplicación de las bases conceptuales y prácticas en la industria como un marco de referencia en su proceso de formación específica en el área científica en el cual nos movemos.

4 OBJETIVOS

4.1 Objetivo General

Dar a conocer a los estudiantes los conceptos básicos de la tecnología de programación orientada a objetos utilizando un lenguaje de programación sencillo, para que diseñen y desarrollen diferentes soluciones que involucren los temas aprendidos en la asignatura y soportarlos en la herramienta de programación C# de la suite Visual.Net de Microsoft

4.2 Objetivos Específicos

- Entender y aplicar el proceso de generación y declaración de clases (atributos y métodos) que definan las representaciones y comportamiento de los objetos.
- Asimilar las ideas orientadas a objetos manipulando a través de aplicaciones sencillas las diferentes etapas de la POO.
- Explorar la estructura sintáctica de las funciones del lenguaje Visual C#, para la construcción de programas orientados a objetos.
- Realizar comunicación entre objetos a través del envío de mensajes y conocer las propiedades de los objetos y los tipos abstractos de datos.
- Interactuar con Requerimientos, Modelado y codificación para la resolución de problemas con software.
- Recordar las sentencias de control, la sintaxis y la semántica básicas en la programación C#.
- Generar aplicaciones con soporte de almacenamiento secundario tanto en archivos planos como en bases de datos

5 COMPETENCIAS

[Identificar aquellas competencias que se esperan obtener en el proceso]

6 CONTENIDOS

- Repaso de programación
- Repaso diagramas UML (casos de uso, Diagramas de clases, Diagramas de secuencia y Diseño de interfaces graficas de usuario)
- Características y herramientas de la programación orientada a objetos (POO)
- Constructores y destructores
- Almacenamiento temporal de objetos (vector tipo class)
- Búsqueda y ordenamiento de información
- Sobrecarga de operadores y sobrecarga de funciones
- Herencia simple y relaciones
- Almacenamiento de información sobre dispositivos secundarios archivos planos
- Aplicaciones Web y multimedia
- Introducción a las bases de datos

7 METODOLOGÍA

Las clases se desarrollaran en las aulas de sistemas y en un salón de la facultad, dado que es una asignatura de diseño y creación de programas basados en computadora. Para el buen desarrollo de la clase el alumno deberá realizar auto consultas previas a la misma y se hará uso de: material audiovisual (video beam, proyector de acetatos, etc.), esto con el fin de facilitar y agilizar la enseñanza por parte del docente y el aprendizaje por parte del alumno.

El profesor hará complementación del auto consulta de los alumnos de cada uno de los temas de la materia apoyados con el planteamiento y solución de ejercicios de acuerdo con cada tema, se desarrollaran programas de computadora en clase, guiado este desarrollo por el profesor. Además se propondrán talleres con el ánimo que el estudiante pueda despejar cualquier duda con respecto al tema; de igual forma esto permitirá hacer mas dinámica cada una de las clases.

También se implementara un proyecto el cual debe ser desarrollado por los estudiantes al final del semestre y en él se aplicaran los conceptos vistos en los talleres y clases.

EL DESARROLLO DE ESTE CURSO ES DE CARÁCTER TEÓRICO-PRACTICO Y POR LO TANTO NO TENDRÁ HABILITACIÓN.

8 REQUISITOS

[Escriba aquí los requisitos y competencias necesarios.]

9 RECURSOS

[Especifique aquí el equipo especial necesario, las direcciones URL, los asesores, etc.]

10 EVALUACIÓN

Primer corte 25% parcial teórico y programas
Segundo Corte 25% parcial teórico y programas
Laboratorio 20% Trabajo de investigación o proyecto
Examen Conjunto: 30%

11 FUENTES DE INFORMACIÓN

11.1 Impresos

- JAMsa, Kris. Superutilidades para visual basic.net. MC GRAW HILL.
- CHARTE OJEDA, Francisco. Programación con Visual Basic.net. ANAYA MULTIMEDIA.
- RUMBAUGH, James. Modelado y diseño orientado a objetos. ED. PRENTICE HALL.
- JOYANES, Luis. Programación orientada a objetos. MC GRAW HILL.
- BOUD, Timothy. Introducción a la programación orientada a objetos. ADDISON-WESLEY.
- El lenguaje unificado de modelado. Gaddy Booch, James Rumbaugh, Ivar Jacobson. Addison-Wesley / 2003
- El lenguaje unificado de modelado. Manual de Referencia Gaddy Booch, James Rumbaugh, Ivar Jacobson. Addison-Wesley / 2003
- MSDN: Microsoft Developers NetWork. Viene con los CD de Visual Studio .Net
- Manuales de las herramientas de desarrollo Visual Studio NET. 2000.
- Tutoriales de Visual C# y Access de Internet. 2005.
- Rational Software. Manuales y ayudas de Rational Rose 2002.
- El lenguaje Unificado de Modelado. Grady Booch, James Rambaugh, Ivar Jacobson. Pearson Educación.2002.
- Manuales y ayudas de Rational Rose y Microsoft Visio 2003.

11.2 Electrónicos

<http://www.monohispano.org/>
<http://www.josanguapo.com/>
<http://mono.es.gnome.org/>
<http://www.mono-project.com/downloads/>
<http://programacionfacil.com/>

Adicionalmente el material entregado por el docente y lo dejado en aulas virtuales.

12 RESUMEN ANALÍTICO DEL MICROCURRÍCULO

Semana	Tema	Actividades
	Repaso de programación 0.1 Constantes, tipos de variables, operadores y tipos 0.2 Estructuras de decision: if-else; if-else if- else; switch-case 0.2 Estructuras de repetición: for; while; do-while 0.3 Funciones, tipos de función, retorno y parámetros 0.4 Ejercicio: programa en C#	
	1. Repaso de diagramas UML 1.1 Diagrama de casos de uso 1.2 Diagrama de clases y relaciones 1.3 Diagrama de secuencias 1.4 Diseño de interfaces graficas de usuario 1.5 Ejercicios	
	2. Introducción a la programación orientada a objetos 2.1 Reseña histórica 2.2 Características de la programación orientada a objetos POO 2.2.1 Abstracción 2.2.2 Encapsulamiento 2.2.3 Herencia 2.2.4 Polimorfismo 2.3 Herramientas de la POO 2.3.1 Clases 2.3.2 Objetos o instancias de una clase 2.3.3 Métodos o funciones miembro 2.3.4 Mensajes o eventos 2.4 Implementación de la POO en Visual C# 2.4.1 Declaración y ámbitos de una clase 2.4.2 Partes de una clase: atributos y métodos 2.4.3 Creación de objetos	
	3. Constructores y destructores 3.1 Definición, características y uso de constructores 3.2 Tipos de constructores: por defecto, por argumento y de copia 3.2 Definición e implementación de un destructor	
	4. Colecciones de objetos a través de un vector tipo class 4.1 Generación del vector 4.2 Control del vector y llamado a los métodos por cada uno de los objetos 4.3 Envió del vector a otros formularios como parámetro	
	5. Operaciones básicas con una lista dinámica de objetos	

	<p>5.1 Inserción, eliminación, consulta e impresión de objetos</p> <p>5.2 Tipos de ordenamiento y búsqueda de objetos</p>	
	<p>6. Almacenamiento de objetos sobre dispositivo secundario (disco duro, disketts, cd_rw o memorias USB).</p> <p>6.1 Tipos de acceso a archivos (binarios)</p> <p>6.2 Operaciones para el manejo de archivo (apertura e y/o creación, lectura. Escritura, búsqueda y cierre).</p>	
	<p>7. Sobrecarga (Polimorfismo)</p> <p>7.1 De operadores</p> <p>7.2 De funciones</p>	
	<p>8. Herencia</p> <p>8.1 Herencia simple</p> <p>8.2 Herencia múltiple</p> <p>8.3 Constructores y destructores para el manejo de herencia</p> <p>8.4 Clases abstractas y concretas</p>	
	<p>9. Desarrollo de aplicaciones WEB y uso de la multimedia</p> <p>9.1 Introducción a la WEB</p> <p>9.2 Que es un servicio WEB?</p> <p>9.3 Por que servicios WEB?</p> <p>9.4 Componentes de software</p> <p>9.5 Características de los servicios WEB</p> <p>9.6 Ejemplos de servicios WEB</p> <p>9.7 Creación de una Web con C#</p> <p>9.7 Introducción e importancia de la multimedia</p> <p>9.8 Concepto y componentes de la multimedia</p> <p>9.8 Creación de un ejemplo utilizando multimedia en C#</p>	
	<p>10. Introducción a las Bases de datos</p> <p>10.1 Concepto y tipos de Bases de Datos</p> <p>10.2 Sistemas para diseñar y estructurar bases de datos</p> <p>10.3 Creación, diseño y relaciones de Bases de datos en Access</p> <p>10.4 Conexión de bases de datos desde C#</p> <p>10.5 Operaciones básicas de bases de datos desde C#</p> <p>10.5.1 Consulta o visualización de registros SQL select</p> <p>10.5.2 Inserción o adición de registros SQL insert</p> <p>10.5.3 Borrado o eliminación de registros SQL delete</p>	