

	UNIVERSIDAD DISTRITAL "Francisco José de Caldas" Facultad de Ingeniería Ingeniería Eléctrica		
	Elaboró	Adriana Marcela Vega	Fecha de Elaboración
Revisó	[Escriba aquí el nombre]	Fecha de Revisión	[Escriba la Fecha de Revision]

1 DATOS DE IDENTIFICACIÓN

Nombre del espacio académico:	Fundamentos de Programación		
Pensum al que pertenece	2		
Código	701007		
Créditos Académicos	4		
Número de Horas Semanales	HTD	HTC	HTA
	4	2	6
Modalidad	Asignatura		
Área	Básicas de Ingeniería		

2 PREGUNTAS QUE BUSCA RESOLVER

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

3 JUSTIFICACIÓN

La Sociedad de la Información funciona en un nuevo espacio denominado "Ciberespacio". Este nuevo espacio facilita la comunicación entre personas, las cuales debido a su naturaleza de integrarse, ven en este "lugar" una gran posibilidad para actualizarse, compartir, transferir conocimientos, realizar comercio electrónico, comunicarse y aprender. Se considera que el estado evolutivo de esta sociedad concluirá en una sociedad denominada sociedad del conocimiento, en donde el conocimiento será la materia prima para producir riqueza y poder.

En esta nueva sociedad ya no son válidos ciertos modelos ni ciertos conceptos tradicionales, pues el mundo se rige actualmente según un nuevo orden que es la globalización. Los cambios que ha introducido la tecnología en nuestra forma de vida hacen necesario replantear los modelos tradicionales en casi todos los campos siendo uno de ellos la Informática. La informática ha venido ha revolucionar la vida del ser humano, pues esta presente en todos los campos del conocimiento tales como Artes, Literatura, Derecho, filosofía, medicina, arquitectura, administración y por supuesto en la rama de la Ingeniería.

La informática se soporta en la integración de metodologías para construcción de software, lenguajes de modelado, lenguajes de programación, sistemas operativos y en la integración de estos con computadores. El fundamento de la Informática han sido los lenguajes de programación, en un principio se utilizaron lenguajes de programación estructurada, pero con el pasar del tiempo evolucionaron hacia los lenguajes de programación orientada a objetos (C++, Java) y ahora último

con la aparición de nuevos lenguajes de programación orientada a objetos (C#, J#, C++), para operar en plataformas múltiples tenemos una herramienta poderosa para realizar desarrollos de una forma sencilla y rápida.

Los lenguajes de programación vienen entonces a ser en este nuevo entorno una herramienta fundamental para las diferentes ramas del conocimiento y la ingeniería y su aprendizaje es una necesidad imperiosa para los estudiantes, pues esta será la herramienta fundamental para poder avanzar en su proceso de formación profesional. Es así como se propone un sistema innovador donde se trabajen lenguajes de programación orientados a objetos C#, pues su sintaxis es muy similar. El buen ingeniero no se debe limitar al manejo de programas o paquetes existentes, sino a construir software de alta calidad mediante el uso de metodologías, lenguajes de modelado, lenguajes de programación orientada a objetos y herramientas de desarrollo.

4 OBJETIVOS

4.1 Objetivo General

Este curso tiene por objeto dar a conocer a los estudiantes de Proyecto Curricular de Ingeniería Eléctrica de primer semestre los fundamentos básicos sobre los siguientes temas: Metodologías para Diseño de Software, Lenguajes de Modelado de software (UML), Fundamentos de programación Orientada a Objetos (C#) y Manejo de Herramientas CASE para Diseño y Desarrollo.

4.2 Objetivos Específicos

Con el fin de alcanzar el objetivo general se proponen los siguientes objetivos específicos:

- Dar a conocer la arquitectura básica del computador, sus dispositivos de entrada, salida, almacenamiento y principalmente el funcionamiento de la memoria.
- Suministrar conceptos básicos sobre las características de los sistemas operativos.
- Dar a conocer la forma como interactúa, la memoria, el sistema operativo y los objetos.
- Realizar una introducción básica al Proceso Unificado de Desarrollo (Metodología RUP), para construcción de software de alta calidad, teniendo en cuenta los requerimientos de los usuarios, análisis, diseño, desarrollo, pruebas y evaluación.
- Realizar una introducción básica al Lenguaje de Modelado Unificado (UML), con el fin de indicar al alumno la importancia del modelado sobre los lenguajes de programación.
- Resaltar los objetivos del lenguaje de visualizar, especificar, construir y documentar software.
- Impartir instrucción sobre herramientas para modelado de software (CASE) con UML.
- Impartir instrucción sobre Lenguajes de Programación Orientados a Objetos (C#) desde los conceptos básicos hasta los avanzados, incluyendo: tipos de datos, estructuras de datos, variables, operadores, compiladores, instrucciones básicas, objetos y clases, librerías de clases, arreglos y matrices, clases básicas (*string*, numéricas), elementos importantes.
- Dar a conocer las características básicas de la programación orientada a objetos entre ellas: el principio de alta cohesión, débil acoplamiento, herencia, polimorfismo, encapsulación y dar ejemplos de su implementación.
- Impartir instrucción para construir clases ajustadas a estándares, teniendo en cuenta visibilidad de atributos, métodos, sobrecarga de métodos.
- Aprender el manejo de estructuras de control condicional básicas y bucles, aprender a

gestionar la memoria.

- Aprender a construir componentes y módulos con algoritmos, paso de parámetros, recursividad, acoplamiento y cohesión.
- Aprender a manejar herramientas de desarrollo como Visual Studio

5 COMPETENCIAS

La metodología utilizada en el desarrollo del curso apuntará a:

- Que el alumnos este en condiciones de realizar un proceso de análisis del problema antes de intentar emprender cualquier procedimiento de desarrollo, sin seguir una metodología.
- Que el alumno aprenda a manejar un lenguaje de modelado visual como UML, con el fin de que cree los modelos, los visualice, especificar su estructura y el comportamiento, generar el código mediante la ayuda de una herramienta CASE y adquiera la disciplina de documentar el proceso.
- Que aprenda a utilizar los principios básicos de la programación orientada a objetos "Alta cohesión, Débil acoplamiento", para que las clases que construya estén ajustadas a los estándares.
- Diseñar algoritmos a partir del planteamiento de un problema utilizando diagramas ajustados a UML.
- Implementar programas en C#, compilarlos y ejecutarlos.
- Realizar un programa final que resuelva un determinado problema de tipo práctico, siguiendo todo el proceso establecido por la metodología y el lenguaje de modelado, su implementación con un lenguaje de programación orientado a objetos y que se obtenga como resultado un ejecutable.
- Conocer los diferentes entornos y procesos de compilación y ejecución para los diferentes lenguajes de programación y la forma como se administra la memoria.
- Aprender a evaluar el software resultado del proceso de la Metodología.
- Aprender a manejar los depuradores y las Herramientas de desarrollo, que le dará posibilidades de continuar de forma autónoma y de conseguir trabajo.

6 CONTENIDOS

- Sistemas numéricos y unidades básicas de almacenamiento
- Arquitectura del Computador
- Descripción de los Sistemas Operativos.
- Metodología RUP, Requerimientos, Diseño, Desarrollo, Pruebas, Evaluación
- Lenguaje de Modelado UML
- Herramientas CASE
- Conceptos básicos sobre Programación
- Introducción a Clases y Objetos (notación UML)
- Lenguajes de Programación Orientados a Objetos.
- Conceptos básicos sobre Programación e Introducción a Clases y Objetos
- Herramienta de desarrollo Visual Studio .NET, Manejo de interfaz gráfica y lógica.
- Lenguaje de Programación C#
- Implementación del Software.

7 METODOLOGÍA

El programa de estudios no solo se enmarca en lo referente al proceso de modelado de software y su construcción mediante la implementación a través de lenguajes de programación orientados a objetos, sino que pretende dar a los estudiantes los conceptos básicos sobre la estructura que deben tener una aplicación que solucione un problema de la vida real, conociendo previamente de forma rápida modelo de tres capas (Presentación, lógica de aplicación y persistencia).

Incluye la parte conceptual sobre el ciclo de vida del software (especificación, requerimientos, diseño, construcción, implantación, pruebas y evaluación, entre otros) y la explicación de los métodos de desarrollo de software, enmarcado dentro del paradigma de orientación a objetos. También invita al estudiante a la utilización de técnicas y herramientas CASE, de manera que tenga la posibilidad de aplicar los conocimientos aprendidos en un caso práctico, el cual se refiere al modelado y desarrollo de una aplicación completa.

Para efectos del modelado de software el lenguaje utilizado será UML (Unified Modeling Language), para lo cual se pueden utilizar herramientas como Rational Rose, Eclipse, Microsoft Visio, UML Studio o la herramienta de preferencia del alumno. La metodología que se utilizará como guía de diseño y desarrollo será RUP (Rational Unified Process),

Para la parte de desarrollo se utilizarán principalmente lenguajes modernos de programación ajustados al estandar ECMA-335 (CLI), entre ellos C#, y los correspondientes entornos de ejecución (CLR). Se sugiere la herramienta de desarrollo Visual Studio NET.

El conocimiento de la metodología, lenguajes de modelado y desarrollo, de las herramientas de desarrollo mencionadas anteriormente, junto con la formación obtenida durante el curso, permitirán que el estudiante este en condiciones de diseñar, desarrollar e implantar una aplicación sencilla, que solucione un problema real.

El programa se desarrollará mediante la exposición del profesor en el tablero o con ayudas visuales como proyector de acetatos o videobeam, de igual forma se desarrollarán talleres en clase con el objeto de afianzar los conocimientos adquiridos. Se efectuará la parte práctica en los laboratorios de forma que se puedan correr los códigos de los ejemplos efectuados teóricamente en el salón de clases. El estudiante debe canalizar sus esfuerzos en la parte de investigación y análisis efectuando desarrollos en su hogar para que tome destreza en la parte de programación.

8 REQUISITOS

Para un buen desarrollo del curso el estudiante necesita tener un buen manejo de:

- Manejo de Computadores
- Habilidades básicas de comunicación oral y escrita

9 RECURSOS

Espacio físico (Aula), Recurso docente, Recursos informáticos (C++, C#, Microsoft Visio, UML Studio)
Recursos bibliográficos

10 EVALUACIÓN

Se sugiere la realización de las siguientes actividades.

1 Parcial	15%
2 Parcial	15%
Trabajo Práctico	15%
Laboratorios	20%
Tareas, Quiz, etc	5%
Examen Final	30%

11 FUENTES DE INFORMACIÓN

11.1 Impresos

- Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design, and the Unified Process Craig Larman / Prentice Hall / 2001 / 0130925691 / 2nd
- Object Oriented Analysis and Design with applications - 2nd Ed.
- Grady Booch / Benjamin Cummings Publishing Co./ 1994
- Object-Oriented Software Engineering: A Use Case Driven Approach
- Ivar Jacobson / Addison-Wesley / 1992 / 0201544350
- Instant UML
- Pierre-Alain Muller / Wrox Press / 1997 / 1861000871
- Mastering Uml With Rational Rose
- Wendy Boggs, et al / Sybex / 1999 / 0782124534 / Bk&Cd Rom
- 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
- Microsoft Corporation.
- Rational Software. Manuales y ayudas de Rational Rose 2002.
- El lenguaje Unificado de Modelado. Grady Booch, James Rambaugh, Ivar Jacobson. Pearson Educación.2002.
- Manuales de las herramientas de desarrollo Visual Basic, Visual C++, C++, Java y Visual Studio NET.2004.
- Manuales y ayudas de Rational Rose y Microsoft Visio 2003.
- Manual del Programador Visual Basic NET. Microsoft Press. 2003.

11.2 Bibliografía Electrónica

- [Especifique aquí la bibliografía electronica]

12 RESUMEN ANALÍTICO DEL MICROCURRÍCULO

Semana	Tema	Actividades
	Introducción Fundamentos de Programación, presentación programa	
	Sistemas numéricos, Clases de Sistemas numéricos: Decimal, explicación detallada y sus ejemplos. Hexadecimal, explicación detallada y sus ejemplos. Binario: explicación detallada y sus ejemplos. Ejercicios para realizar en clase. Unidades básicas de almacenamiento, explicación detallada y ejemplos. Recordar las operaciones lógicas como and, or, not.	
	Arquitectura del computador: Explicación detallada de cada una de sus partes o hardware, dispositivos de entrada y salida. Proceso de arranque y chequeado, SETAP, configuración desde allí. Unidad Central de proceso (CPU), su funcionalidad, evolución, velocidad. Memorias: RAM, CACHE, BIOS, Virtual, ROM. Discos Duros, Tipos de buses	
	Los Sistemas Operativos: descripción, características, tipos de sistemas operativos, para que sirven cada uno de ellos. Clases de	
	Sistemas operativos y su evolución, Sistema operativo Windows, y su manejo desde el panel de control.	
	Metodología RUP, Requerimientos: Fundamentos de la metodología, Objetivos de la metodología, Las seis (6) mejores prácticas (Administración de requerimientos, Desarrollo Iterativo, Modelamiento visual, Verificación de la calidad, Arquitectura con componentes, Control de cambios), Etapas de la metodología (Análisis, Diseño y Desarrollo, Pruebas, Evaluación),Ciclo de vida del Desarrollo de Software (Inicio, Elaboración, Construcción, Transición, Flujo de trabajo de procesos, Flujo de trabajo de soporte). Requerimientos para el RUP, técnicas, formatos empleados, ejercicios prácticos	
	Lenguaje de Modelado UML : Objetivos de UML (Visualizar, Especificar, Construir, Documentar), Modelo conceptual de UML: Bloques de construcción UML: Elementos UML: estructurales, de comportamiento, de agrupación, de anotación. Relaciones UML: Dependencia, asociación, agregación, Generalización, Realización. Diagramas: Casos de uso, Clases, Componentes, Secuencia, Diagramas de Colaboración, Actividades. (Haciendo énfasis en los diagramas de clases, casos de uso y secuencia). Reglas UML: Nombres, visibilidad, Integridad, Ejecución. Mecanismos Comunes: Especificaciones, adornos, divisiones comunes, Mecanismos de extensibilidad	
	Herramientas CASE: instrucción teórica de algunas herramientas libres y manejo de las mismas para la elaboración de diagramas en UML. Para lo cual se pueden utilizar herramientas como Rational Rose, Microsoft Visio,UML Studio o la herramienta de	

	preferencia del alumno	
	Conceptos básicos sobre Programación: Revisión a las Técnicas de Programación: Programación Estructurada, Programación no Estructurada, Programación Procedimental, Programación Modular, Problemas de la Programación Modular, Programación Orientada a Objetos	
	Introducción a Clases y Objetos (notación UML): Definición de Clases y Objetos, atributos y métodos. Comportamiento, Conceptos relacionados con el paradigma de objetos, Clases, Instancias, referencias.	
	Lenguajes de Programación Orientados a Objetos: Descripción general de ella, principios básicos: "Alta cohesión, Débil acoplamiento". Conocer en que consiste la Herencia, el Polimorfismo y la Encapsulación. Principales lenguajes de Programación Orientada a Objetos.	
	Herramienta de desarrollo Visual Studio .NET, Manejo de interfaz gráfica y lógica. Compiladores del lenguaje, Código Intermedio, Compilador en tiempo de ejecución, Cargador de clases, CLR (Common Language Runtime), Como crear soluciones de "n" capas, Como crear proyectos, Conceptos de espacios de nombres, Compilación a código intermedio, Compilación en tiempo de ejecución (JIT), Creación de una solución de dos (2) capas (presentación y lógica de aplicación) uso de herramientas y sus propiedades, uso de los exploradores, Uso del depurador.	
	Fundamentos de Programación con C#: Creación de clases y componentes (Código Intermedio), Creación de una solución, Creación de clases, Implementación de Interfaces, Implementación de métodos Implementación de clases con Herencia, Implementación de "Herencia múltiple", implementación de Polimorfismo, Sobrecarga de métodos, Tipos de Datos, Conversión de tipos (Int, string, boolean, byte, char, etc), Constantes, Nombres de variables, Operadores aritméticos, operadores relacionales, Op. Lógicos, Forma de evaluar expresiones lógicas, precedencia de operadores, Sentencias Básicas de asignación, de llamada a un método, de retorno de un método: Instrucción IF, ELSE; IF Anidados; Instrucción DO, WHILE; Instrucción FOR, Instrucción SWITCH, Instrucción FOREACH; Introducción a los Algoritmos; Conceptos sobre algoritmos; Atributos, Visibilidad de atributos, checked y unchecked, instrucción try-cath, Manejo de arrays, propiedades, Manejo de funciones (sin(), cos(), exp(), tan(), asin(), acos(), max(), float(), long(), parseLong(), etc, Variables Static, El objeto this, super. Manejo de Eventos (mouseDown, MouseUp, mouseDrag, etc), La Clase Component (Eventos sobre componentes TextField, Button, CheckBox, Choice, List, etc), La	

	Clase Color y sus métodos (brighter, darker, getBlue(), getRed(), etc). La Clase Graphics y sus métodos (drawRect, drawLine, drawArc, etc). La Clase Fonts y FontMetrics (getName, getSize, isBold, isItalic, etc), La Clase Applets, Los Applets y los documentos HTML.	
	Implementación del Software: Diseño y Desarrollo de una aplicación completas ajustada al modelo de dos (2) capas- Metodología RUP. Requerimientos, Diseño y Modelado (Casos de uso, Clases, Diagrama de Clases), Construcción, Pruebas, Evaluación. (Durante todo el curso)	
	Desarrollo de una guía práctica en cada una de las semanas que se dure el programa	