



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
PROYECTO CURRICULAR DE INGENIERÍA DE SISTEMAS



I. IDENTIFICACIÓN DE LA ASIGNATURA

MODELOS DE PROGRAMACIÓN	
CÓDIGO DE LA ASIGNATURA	200403
ÁREA	PROGRAMACIÓN
SEMESTRE	SEGUNDO
PLAN DE ESTUDIOS	1996 - Ajuste 2002
HORAS TOTALES POR SEMESTRE	96
HORAS TEÓRICAS	32
HORAS PRÁCTICAS	64
SEMANAS POR SEMESTRE	16
PRE REQUISITO	NINGUNO
CO REQUISITO	NINGUNO
CRÉDITOS	3

II. CARACTERÍSTICAS DE LA ASIGNATURA

1. CONCEPTOS PREVIOS REQUERIDOS:

- programación orientada a objetos
- principios de diseño orientado a objetos
- modelamiento con diagrama de clases

2. PUNTOS DE APOYO PARA OTRAS ASIGNATURAS:

- Estructura lógica conceptual basada en paradigmas de programación
- Herramienta fundamental para ingeniería de software.
- Herramienta fundamental para bases de datos
- Herramienta fundamental para Redes

3. ASIGNATURAS RELACIONADAS

- Ingeniería de Software
- Bases de Datos
- Lógica
- Matemáticas Discretas

4. CONTENIDO FUNDAMENTAL:

Unidad 1. Programación Con Patrones De Diseño.,
Unidad 2. Patrones Creacionales.
Unidad 3. Patrones Estructurales.
Unidad 4. Patrones De Comportamiento.
Unidad 5. Otros Patrones.
Unidad 6. Antipatrones.
Unidad 7. De Las Objetos A Los Componentes.
Unidad 8. Construcción De Componentes.
Unidad 9. Representación De Contenido Mediante El Modelo Jerárquico.
Unidad 10. Representación Del Conocimiento Como Reglas.
Unidad 11. Representación De Información Persistente.
Unidad 12. Persistencia En Los Lenguajes Y Bases De Datos.



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
PROYECTO CURRICULAR DE INGENIERÍA DE SISTEMAS



5. JUSTIFICACIÓN:

Los modelos siempre han sido fundamentales para el establecimiento de un sistema terminado, con un mínimo de errores, gracias a una previa experimentación realizada sobre prototipos producidos cuando se modela. Un sistema cuyo eje es el software o la informática depende en gran medida del modelado que se haga y se plasme finalmente en un programa de computador, y es incluso en la misma programación donde se puede continuar explotando los beneficios que tiene el modelar. El diseño de un programa como mínimo de manera tácita tiene un modelo, por tanto se hace imprescindible profundizar sobre el modelamiento propiamente establecido en la programación. Es tarea entonces establecer como modelar programas que estén basados propuestas con principios y reglas que garanticen establecer parámetros de modelación junto con sus métricas, con los cuales se pueda tener el máximo control posible sobre cada instrucción que se codifique.

Modelos de programación es una materia que plantea fundamentalmente las formas, técnicas y mecanismos que pueden ser útiles al momento de plasmar la solución computacional de un problema, se pretende mostrar aquí que si bien es cierto la orientación a objetos es el paradigma mas empleado, muchas tecnologías alrededor de él, son desconocidas y por tanto poco utilizadas como los son por ejemplo los componentes o los patrones; incluso existen otras propuestas interesantes que vale la pena revisar como son por ejemplo la programación basada en jerarquías o de estructuración de contenidos, la programación funcional, la programación lógica, entre otras; que establecen sus propios principios y que pueden ser bastante útiles a la hora de enfrentar un problema.

6. OBJETIVO GENERAL:

Identificar los modelos de programación pertinentes para el planteamiento de una solución computacional.

7. OBJETIVOS ESPECÍFICOS:

- Realizar programas computacionales basados en modelos de programación robustos, empleando los principios propios del modelo que se emplee.
- Establecer los criterios de selección de un lenguaje de programación sobre el cual se pueda efectuar el modelo de programación deseado ya sea este, un componente un patrón u otro dentro de la tecnología orientada a objetos.
- Explorar nuevos esquemas de solución planteando el modelo pertinente
- Manejar elementos conceptuales de mayor profundidad sobre el modelo OO y otros modelos de programación como el desarrollo por componentes o el desarrollo basado en patrones.
- Revisar otras propuestas de programación que permitan comparar los esquemas que se plantean en el momento de resolver problemas computacionales.

8. ESTRATEGIAS METODOLÓGICAS:

- La materia se desarrollará teórico práctica. Se hará primero conceptualización y luego prácticas de laboratorio sobre los conceptos previamente discutidos.

9. CRITERIOS DE EVALUACIÓN:

Proyecto Final	30%
Parcial	20%
Laboratorios y exposiciones	30%
Apreciativa, asistencia, trabajos	20%

10. RECURSOS FÍSICOS REQUERIDOS:



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
PROYECTO CURRICULAR DE INGENIERÍA DE SISTEMAS



Software

- Microsoft Visual studio . Net , Eclipse.
- JDK 1.4, BDK.

Hardware y logísticos

- Laboratorios de cómputo

11. PRÁCTICAS ESPECÍFICAS:

III. PARCELADO

No	TEMA A DESARROLLAR	SEMANAS ACADÉMICAS															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1.	Patrones de Diseño, Concepto e importancia de los patrones, Tipos de patrones: Abstract Factory, Builder,																
2.	Factory Meted, Prototype, Singleton.	X	X														
3.	patrones estructurales: Adapter, Bridge.			X													
4.	Composite, Decorador				X												
5.	Facade, Flyweight, Proxy.					X	X										
6.	patrones de comportamiento: Chain of Responsibility, Command.							X	X								
7.	Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor									X	X						
8.	Otros patrones: Patrones Arquitecturales, Organizacionales: Niveles, Tuberías y filtros, Pizarra. Sistemas Distribuidos: Broker.											X					
9.	Sistemas Interactivos: MVC, PAC, Sistemas Adaptables, Microkernel, Reflexion, Otros.												X				
10.	Antipatrones: gota, Fantasmas, Código Muerto, Código en espagueti, Islas, Otros													X	X		
11.	Construcción de componentes;_Concepto, Diseño de componentes, Tipos de componentes, Evolución de clases a componentes, Construcción de componentes.																X
12.	Otros modelos																X



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
PROYECTO CURRICULAR DE INGENIERÍA DE SISTEMAS



IV. BIBLIOGRAFÍA

AUTOR (ES)	TÍTULO	Editorial	Edición y/o año	Tipo*
Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides	Design Patterns: Elements of Reusable Object-Oriented Software	Addison Wesley		TG
Bertrand Meyer	Object-Oriented Software Constr	Prentice Hall		TR
Bjarne Stroustrup	El C++ lenguaje de programación segunda edición	Addison Wesley		TR
Ivor Horton	Beginning Java 2	Wrox		TC
Horstmann Cornell	Core Java 2 vol 1 y vol 2.	Prentice Hall.		TG
Ian F. Darw	Java Cookbook	O'Reilly		TR
Burton Harvey, Simon Robinson, Julian Templeman, Karli Watson	C# Programming	Wrox Press Ltda		TA
Paul Anderson and Gail Anderson	Navigating C++, and Object-Oriented Design.	Prentice Hall		TC
Kayshav Dattatri	C++ effective Object-Oriented Software Construction	Prentice Hall		TC
William H. Ford / William R Topp	Introduction to Computing Using C++ and Object Tecnology	Prentice hall		TC

TG: Texto Guía
TC: Texto Consulta
TR: Texto Referencia
TA: Texto Adicional