



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERIA

SYLLABUS

PROYECTO CURRICULAR: Ingeniería de Sistemas

NOMBRE DEL DOCENTE:

ESPACIO ACADÉMICO (Asignatura): Modelos de Programación I

Obligatorio (X)

Básico (X)

Complementario ()

CÓDIGO: 416

Electivo ()

Intrínsecas ()

Extrínsecas ()

NÚMERO DE ESTUDIANTES:

GRUPO:

NÚMERO DE CRÉDITOS: Tres(3)

TIPO DE CURSO: TEÓRICO ()

PRACTICO ()

TEO-PRAC (X)

Alternativas metodológicas:

Clase Magistral (X), Seminario (), Seminario – Taller (), Taller (X), Prácticas (X), Proyectos tutoriados(), Otro: _____

HORARIO:

DÍA

HORAS

SALÓN

I. JUSTIFICACIÓN DEL ESPACIO ACADÉMICO

Competencias del perfil a las que contribuye la asignatura:

Esta asignatura contribuye al desarrollo de la competencia “Resuelve problemas computacionales algorítmicamente” que se encuentra en el dominio de “programación” del área “básicas de ingeniería” del proyecto curricular de ingeniería de sistemas.

Contribución a la formación:

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 más 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.

Puntos de apoyo para otras asignaturas:	<ul style="list-style-type: none"> • Estructura lógica conceptual basada en paradigmas de programación • Herramienta fundamental para ingeniería de software. • Herramienta fundamental para bases de datos • Herramienta fundamental para Redes
Requisitos previos:	<ul style="list-style-type: none"> • programación orientada a objetos • principios de diseño orientado a objetos • modelamiento con diagrama de clases
II. PROGRAMACIÓN DEL CONTENIDO	
OBJETIVO GENERAL	
Presentar al estudiante la conceptualización y aplicación de los modelos de programación pertinentes para el planteamiento de una solución computacional basados en el paradigma orientado a objetos.	
OBJETIVOS ESPECÍFICOS	
<ol style="list-style-type: none"> 1. Realizar programas computacionales basados en modelos de programación robustos, empleando los principios propios del modelo que se emplee. 2. 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. 3. Explorar nuevos esquemas de solución planteando el modelo pertinente 4. 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. 	
COMPETENCIAS DE FORMACIÓN:	
Competencias que compromete la asignatura:	El estudiante está en capacidad de seleccionar y aplicar formas, técnicas y mecanismos útiles al momento de plasmar la solución computacional de un problema.
Competencias específicas de la asignatura:	<ul style="list-style-type: none"> • El estudiante entiende el concepto de patrón de diseño y sus implicaciones en el modo de resolver problemas. • Conoce y entiende las diferencias entre los diferentes tipos de patrones. • Entiende el tipo de problemas de desarrollo software que solucionan un uso correcto de los patrones creacionales. • Entiende el tipo de problemas de desarrollo software que solucionan un uso correcto de los patrones estructurales. • Entiende el tipo de problemas de desarrollo software que solucionan un uso correcto de los patrones de comportamiento • Comprende las implicaciones de los antipatrones de diseño.

	<ul style="list-style-type: none"> • Entiende el concepto y aplicabilidad del desarrollo por componentes. • El estudiante es capaz de reflexionar acerca del conocimiento y el uso de los patrones de diseño.
<p>Competencias Transversales a las que contribuye la asignatura:</p>	<ul style="list-style-type: none"> • El alumno tiene la capacidad de discernir que tecnología debe utilizar para la resolución de problemas particulares. • Comunica ideas de manera clara de forma oral o escrita. • Actúa estratégicamente dentro de un grupo de trabajo para el desarrollo de proyectos.

PROGRAMA SINTÉTICO:

1. Patrones de diseño

- 1.1. El concepto de patrón de diseño
- 1.2. Importancia de los patrones de diseño
- 1.3. Tipos de patrones

2. Patrones creacionales

- 2.1. Abstract factory
- 2.2. Factory method
- 2.3. Builder
- 2.4. Prototype
- 2.5. Singleton

3. Patrones Estructurales

- 3.1. Adapter
- 3.2. Bridge
- 3.3. Composite
- 3.4. Decorator
- 3.5. Facade
- 3.6. Flyweight
- 3.7. Proxy

4. Patrones de Comportamiento

- 4.1. Chain of responsibility
- 4.2. Command
- 4.3. Interpreter
- 4.4. Iterator
- 4.5. Mediator
- 4.6. Memento
- 4.7. Observer
- 4.8. State
- 4.9. Strategy
- 4.10. Template method
- 4.11. Visitor

5. Anti-patrones

- 5.1. Gota
- 5.2. Fantasmas
- 5.3. Código muerto
- 5.4. Código en espagueti
- 5.5. Islas
- 5.6. Los peligros de los patrones de diseño

6. Componentes

- 6.1. Concepto de Componente
- 6.2. Diseño de componentes
- 6.3. Tipos de componentes
- 6.4. Evolución de clases a componentes
- 6.5. Construcción de componentes

III. ESTRATEGIAS

Metodología Pedagógica y Didáctica:

- Asistencia a clases expositivas y de discusión
- Elaboración y lectura de paper (documentación).
- Se debe procurar incentivar el trabajo de grupo más que el trabajo individual. (se recomienda trabajar en grupos de dos o tres estudiantes)
- Implementación y prueba de prototipos (programas) en laboratorio de computación

Tipo de Curso	Horas			Horas profesor/semana	Horas Estudiante/semana	Total Horas Estudiante/semestre	Créditos
	TD	TC	TA	(TD + TC)	(TD + TC +TA)	X 16 semanas	
	2	2	8	4	12	192	3

Trabajo Presencial Directo (TD): trabajo de aula con plenaria de todos los estudiantes.

Trabajo Mediado_Cooperativo (TC): Trabajo de tutoría del docente a pequeños grupos o de forma individual a los estudiantes.

Trabajo Autónomo (TA): Trabajo del estudiante sin presencia del docente, que se puede realizar en distintas instancias: en grupos de trabajo o en forma individual, en casa o en biblioteca, laboratorio, etc.

IV. RECURSOS

Medios y Ayudas:

- Aula normal con tablero para sesiones de cátedra y para sesiones de discusión.
- Disponibilidad para acceder a proyector multimedia.
- Laboratorio de computación, para las sesiones de laboratorio.
- IDE's para desarrollar en java (Eclipse, Netbeans, ...)
- Página web para publicar material didáctico, guías de ejercicios, soluciones, tareas, etc.
- Acceso al material bibliográfico recomendado.
- Asignación de una persona que tenga las plenas competencias del curso (monitor) para asesorar a los estudiantes en dudas durante las sesiones del laboratorio de computación.

BIBLIOGRAFÍA

TEXTOS GUÍA

- Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley
- Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra. Head First Design Patterns. O'Reilly Media
- James W. Cooper. Design Patterns. Addison Wesley

TEXTOS COMPLEMENTARIOS

- Bertrand Meyer. Construcción de Software Orientado a Objetos. Prentice Hall.
- Horstmann Cornell, Core Java 2 vol 1 y vol 2. Prentice Hall.
- Bruce Eckel. Thinking Java. Prentice Hall
- Deitel & Deitel. Java2 How To Program. Prentice Hall.
- Agustín Froufe Quintas. Java 2 Manual de usuario y tutorial. Alfaomega.
- Guía de certificación de java. Sun Microsystem.

REVISTAS

DIRECCIONES DE INTERNET

- www.objectmentor.com
- www.sourcemaking.com
- <http://download.oracle.com/javase/6/docs/api/>
- <http://download.oracle.com/javase/6/docs/>

V. ORGANIZACIÓN / TIEMPOS

Espacios, Tiempos, Agrupamientos:

Se recomienda trabajar una unidad cada cuatro semanas, trabajar en pequeños grupos de estudiantes, utilizar Internet para comunicarse con los estudiantes para revisiones de avances y solución de preguntas (esto considerarlo entre las horas de trabajo cooperativo).

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Patrones de diseño																
	El concepto de patrón de diseño																
	Importancia de los patrones de diseño																
	Tipos de patrones																
2	Patrones creacionales																
	Abstract factory																
	Factory method																
	Builder																
	Prototype																
	Singleton																
3	Patrones Estructurales																
	Adapter																
	Bridge																
	Composite																
	Decorator																
	Facade																
	Flyweight																
	Proxy																
4	Patrones de Comportamiento																
	Chain of responsibility																
	Command																
	Interpreter																
	Iterator																
	Mediator																
	Memento																
	Observer																
	State																
	Strategy																
	Template method																
	Visitor																

DATOS DEL DOCENTE

NOMBRE :

PREGRADO :

POSTGRADO :

ASESORIAS: FIRMA DE ESTUDIANTES

NOMBRE	FIRMA	CÓDIGO	FECHA
1.			
2.			
3.			

FIRMA DEL DOCENTE

FECHA DE ENTREGA: _____