



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA

SYLLABUS

PROYECTOS CURRICULAR: Ingeniería de sistemas

NOMBRE DEL DOCENTE:

ESPACIO ACADÉMICO (Asignatura) : Programación Orientada a objetos

Obligatorio (X) : Básico(X) Complementario ()

CÓDIGO: 10

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	En este espacio académico se establecen las bases de la aplicación del paradigma orientado a objetos y se le brindan al estudiante las herramientas para la aplicación de los principios y características de este paradigma para fortalecer en el estudiante las habilidades en el desarrollo de programas computacionales. Estas habilidades se reconocen como claves dentro del dominio del perfil de "Programación".

<p>Puntos de apoyo para otras asignaturas</p>	<p>En Ingeniería de Sistemas herramienta fundamental para:</p> <ul style="list-style-type: none">• Estructura lógica conceptual basada en paradigmas de programación.• Programación orientada a objetos, Programación avanzada y modelos de programación. Ingeniería de software. Bases de datos, Redes, Ciencias de la computación. <p>En Ingeniería Industrial herramienta fundamental para:</p> <ul style="list-style-type: none">• Programación lineal, gestión tecnológica, teoría de colas y simulación, programación y control de producción, logística, gestión de operaciones. <p>En Ingeniería Catastral herramienta fundamental para:</p> <ul style="list-style-type: none">• SIG, bases de datos, interfaces SIG <p>En Ingeniería Eléctrica herramienta fundamental para:</p> <ul style="list-style-type: none">• Área de circuitos, área de electrónica, probabilidad y estadística, sistemas dinámicos, redes de comunicaciones, digitales, herramientas computacionales, campos, generación hidráulica y generación térmica. <p>En Ingeniería Electrónica herramienta fundamental para:</p> <ul style="list-style-type: none">• Sobre escritura, variables polimórficas y genericidad.• Entiende las relaciones entre los distintos tipos de polimorfismo.• Entiendo los mecanismos de gestión de errores que ofrecen algunos lenguajes de programación.• Entiende el concepto de concurrencia.• Entiende el concepto de persistencia.• Entiende y aplica los conceptos básicos sobre modelado de datos.
---	---

	<ul style="list-style-type: none"> • Entiende los conceptos básicos sobre manejo de puertos. • Entiende los conceptos básicos sobre modelado. • Entiende y aplica los conceptos básicos sobre aplicaciones en ambiente web.
Requisitos previos	<ul style="list-style-type: none"> • Programación básica
Competencias transversales a las que contribuye la asignatura:	<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.
OBJETIVO GENERAL	
Presentar al estudiante la conceptualización y aplicación del paradigma orientado a objetos, enfatizando en los elementos conceptuales propios de este que permitan plantear y aplicar modelos bien formados utilizando un lenguaje de programación orientado a objetos	
OBJETIVOS ESPECÍFICOS	
<ol style="list-style-type: none"> 1. Determinar los tipos de aplicación y las situaciones en las que se debe aplicar el paradigma orientado a objetos. 2. Comprender, interpretar y analizar el cambio de enfoque en el modo de resolver problemas que supone el uso del paradigma orientado a objetos respecto a otros paradigmas. 3. Aplicar los conceptos del paradigma de programación orientada a objetos tales como: polimorfismo, encapsulamiento, herencia, sobrecarga, funciones virtuales, etc. 4. Manejar adecuadamente conceptos tales como: recursividad, objetos transientes, residentes y persistentes; generalización y generacidad; clases plantillas; asociación, agregación y composición. 5. Identificar problemas de: portabilidad, efectos colaterales y transparencia referencial. 6. Comprender la enorme importancia de crear software fiable, reutilizable y mantenible. 7. Dominar estrategias básicas de reutilización como son el uso de librerías o paquetes de software. 8. Aplicar el modelo orientado a objetos en programación de dispositivos de cómputo. 	

1. Introducción al paradigma Orientado a Objetos

- 1.1. El progreso de la abstracción
- 1.2. El paradigma orientado a objetos
- 1.3. Lenguajes orientados a objetos
- 1.4. Metas del paradigma orientado a objetos

2. Fundamentos de la programación orientada a objetos

- 2.1. Clases
- 2.2. Atributos
- 2.3. Operaciones (métodos)
- 2.4. Encapsulación y ocultamiento de la información.
- 2.5. Modularidad (criterios, principios y reglas)
- 2.6. El concepto de interfaz
- 2.7. El concepto de objeto
- 2.8. Metaclases
- 2.9. El diseño de aplicaciones OO
- 2.10. Relaciones entre clases y relaciones entre objetos
- 2.11. Documentación del código

3. Herencia y polimorfismo

- 3.1. Introducción a la Herencia
- 3.2. Herencia Simple
- 3.3. Herencia Múltiple
- 3.4. Herencia de Interfaz
- 3.5. Herencia de Implementación
- 3.6. Beneficios y costes de la herencia
- 3.7. Elección de la técnica de reutilización
- 3.8. Polimorfismo y reutilización
- 3.9. Sobrecarga
- 3.10. Polimorfismo en jerarquías de herencia
- 3.11. Variables Polimórficas
- 3.12. Genericidad

4. Gestión de errores y otras características

- 4.1. Gestión de errores
- 4.2. Concurrencia
- 4.3. Persistencia
- 4.4. Persistencia con serialización
- 4.5. Persistencia con archivos
- 4.6. Recogiendo la basura

5. Temas complementarios

- 5.1. Conceptos básicos sobre modelado de datos y persistencia.
- 5.2. Conceptos básicos sobre manejo de puertos.
- 5.3. Conceptos básicos sobre modelado.
- 5.4. Conceptos básicos sobre aplicaciones web.

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 Lectivas/sem	Horas Estud.te/sem	Total Horas Estud.te/sem	Créditos
	TD	TC	TA	(TD + TC)	(TD + TC + TA)	X 16 semanas	
	2	4	6	6	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 pizarrón 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

- Bertrand Meyer. Construcción de Software Orientado a Objetos. Prentice Hall.
- Bruce Eckel. Thinking Java. Prentice Hall
- Deitel & Deitel. Java2 How To Program. Prentice Hall.

TEXTOS COMPLEMENTARIOS

- Agustín Froufe Quintas. Java 2 Manual de usuario y tutorial. Alfaomega.
- Horstmann Cornell, Core Java 2 vol 1 y vol 2. Prentice Hall.
- Horstmann Cornell, Core Java 1. Prentice Hall.

PAGINAS DE INTERNET

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

V. ORGANIZACIÓN /TIEMPOS

Espacios, tiempos, agrupamientos:

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Introducción al paradigma Orientado a Objetos																
	El progreso de la abstracción																
	El paradigma orientado a objetos																
	Lenguajes orientados a objetos																
	Metas del paradigma orientado a objetos																
2	Fundamentos de la programación orientada a objetos																
	Clases																
	Atributos																
	Operaciones (métodos)																
	Encapsulación y ocultamiento de la información.																
	Modularidad (criterios, principios y reglas)																
	El concepto de interfaz																
	El concepto de objeto																
	Metaclases																
	El diseño de aplicaciones OO																
	Relaciones entre clases y relaciones entre objetos																
	Documentación del código																
3	Herencia y polimorfismo																
	Introducción a la Herencia																
	Herencia Simple																
	Herencia Múltiple																
	Herencia de Interfaz																
	Herencia de Implementación																
	Beneficios y costes de la herencia																
	Elección de la técnica de reutilización																
	Polimorfismo y reutilización																
	Sobrecarga																
	Polimorfismo en jerarquías de herencia																
	Variables Polimórficas																
	Genericidad																
4	Gestión de errores y otras características																

ASESORIAS: FIRMA DE ESTUDIANTES

NOMBRE	FIRMA	CODIGO	FECHA
1.			
2.			
3.			

FIRMA DEL DOCENTE

FECHA DE ENTREGA: _____