



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERIA

SYLLABUS

PROYECTO CURRICULAR: Ingeniería de Sistemas

NOMBRE DEL DOCENTE:

ESPACIO ACADÉMICO (Asignatura): Programación Avanzada

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

Electivo () : Intrínsecas () Extrínsecas ()

CÓDIGO: 410

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 para la formulación de programas de software bien formados basados en los principios fundamentales para la constitución de una buena solución computacional. Esta habilidad se reconoce como clave dentro del dominio del perfil de “Programación”.

Puntos de apoyo para otras asignaturas:

- Estructura lógica conceptual basada en paradigmas de programación.
- Herramienta fundamental para Modelos 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 Básica • Programación Orientada a Objetos
II. PROGRAMACION DEL CONTENIDO	
OBJETIVO GENERAL	
<p>Desarrollar los elementos conceptuales necesarios para hacer una buena elección tecnológica y filosófica, y con ella poder construir programas bien diseñados.</p>	
OBJETIVOS ESPECÍFICOS	
<ol style="list-style-type: none"> 1. Realizar programas computacionales basados en estándares, normas y principios de diseño y reutilización que permitan ofrecer soluciones eficaces. 2. Establecer los criterios de selección de un lenguaje de programación que garanticen el desarrollo adecuado de un proyecto informático. 3. Identificar propuestas de desarrollo importantes y robustas que garanticen una buena solución informática. 4. Identificar algunas de las tendencias del modelo OO importantes para la solución de problemas más complejos y puntuales. 5. Manejar elementos conceptuales de mayor profundidad sobre el modelo OO. 	
COMPETENCIAS DE FORMACIÓN:	
Competencias que compromete la asignatura:	<p>El estudiante está en capacidad de pensar ordenadamente para modelar una solución a un problema, en donde se debe analizar e implementar dicha solución aplicando los elementos conceptuales necesarios para poder construir de manera bien formada esta solución.</p>
Competencias específicas de la asignatura:	<ul style="list-style-type: none"> • Conoce y aplica los estándares y recomendaciones para la codificación de programas computacionales. • Conoce la evolución de los esquemas y mecanismos de abstracción. • Comprende el modelo orientado a objetos y sus divisiones. • Modela clases, interfaces, sistemas y sus relaciones mediante diagramas de clases. • Realiza un estudio de requerimientos para problemas sencillos. • Descubre las clases que permiten modelar una solución basado en los requerimientos. • Hace uso de las tarjetas CRC para plantear clases. • Implementa clases basado en los principios de diseño de clases. • Conoce y utiliza las técnicas de gestión de almacenamiento de objetos. • Conoce y plantea correctamente las relaciones entre clases, paquetes y subsistemas. • Realiza una correcta gestión de código.

	<ul style="list-style-type: none"> • Aplica los principios fundamentales para obtener calidad en los programas. • Conoce y aplica técnicas de programación concurrente. • Conoce y aplica técnicas de programación distribuida. • Conoce y aplica técnicas de persistencia con bases de datos.
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.

PROGRAMA SINTÉTICO:

1. El Paradigma Orientado a Objetos

- 1.1. Recomendaciones y estándares para la codificación
 - 1.1.1. Notaciones.
- 1.2. Evolución de los esquemas de abstracción.
- 1.3. El modelo orientado a objetos y sus divisiones.
- 1.4. Evolución de los mecanismos de abstracción.

2. Diagramas de clases

- 2.1. Estudio de requerimientos.
- 2.2. Heurística general para encontrar las clases.
- 2.3. Otras fuentes de clases.
- 2.4. Herramientas para el planteamiento y modelamiento de clases, tarjetas CRC.
- 2.5. Modelamiento de clases.
- 2.6. Modelamiento de interfaces.
- 2.7. Modelamiento de relaciones.
- 2.8. Modelamiento de sistemas.

3. Relaciones entre clases, paquetes y espacios de nombres como subsistemas.

- 3.1. Relación de asociación.
- 3.2. Relación de generalización o herencia.
- 3.3. Relación de dependencia.
- 3.4. Relación de realización.
- 3.5. Delegación con herencia.
- 3.6. Delegación con asociación.
- 3.7. Delegación con dependencias.
- 3.8. Delegación con realización.
- 3.9. Principio de la Burocracia Perfecta.
- 3.10. Establecimiento de sistemas y subsistemas vía espacio de nombres o paquetes.
- 3.11. Anidación de subsistemas.
- 3.12. Acceso en subsistemas.

4. Implementación de clases con diseño

- 4.1. Principio del Egoísmo.
- 4.2. Clase-instancia-objeto.
- 4.3. Principio de seguridad en una clase mediante el this.
- 4.4. Control de acceso e interfaces de una clase.
- 4.5. Acceso en clases e instancias.
- 4.6. Atributos de una clase.
- 4.7. Seguridad de los atributos.
- 4.8. Ley de Demeter.

5. Programación Con Diseño Orientado A Objetos.

- 5.1. El principio de abierto/cerrado.
- 5.2. El principio de sustitución de Liskov.
- 5.3. El principio de inversión de dependencia.
- 5.4. El principio de separación de la interfaz.
- 5.5. El principio de equivalencia reutilización/revisión.
- 5.6. El principio de cierre común.
- 5.7. El principio de reutilización común.
- 5.8. El principio de dependencia acíclica.
- 5.9. El principio de las dependencias estables.
- 5.10. El principio de las abstracciones estables.

6. Gestión del almacenamiento de objetos

- 6.1. Concepto e importancia de la gestión de objetos.
- 6.2. Tipos de gestión.
- 6.3. Recuperación del espacio de los objetos.
- 6.4. Objetos alcanzables e inalcanzables.
- 6.5. Problemática y necesidad de evitar pérdida de memoria.
- 6.6. Recuperación de los objetos estáticos.
- 6.7. Recuperación de los objetos en pila.
 - 6.7.1. El papel de las pilas en la recuperación de objetos en pila.
- 6.8. Recuperación de los objetos libres o en montículo.
- 6.9. Programación O.O con recolección.
 - 6.9.1. Liberación o recuperación manual y automática.

7. Gestión de código y pautas para obtener calidad en los programas.

- 7.1. Principios fundamentales para la gestión de código.
- 7.2. Principios fundamentales para establecer pautas de calidad en los programas.

8. Otros Tópicos En Programación Orientada A Objetos.

- 8.1. Programación concurrente.
- 8.2. Programación distribuida.
- 8.3. Persistencia y el papel de la programación en las bases de datos.
- 8.4. Generalidades sobre programación en bases de datos y conectividad.

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	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 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.

VI. EVALUACIÓN			
	TIPO DE EVALUACIÓN	FECHA	PORCENTAJE
PRIMER CORTE			
SEGUNDO CORTE			
PROYECTO FINAL			30,00%

ASPECTOS A EVALUAR DEL CURSO

- Claridad y entendimiento de los conceptos.
- Que se haya identificado correctamente el problema y que el modelo lo represente adecuadamente.
- Que la solución diseñada resuelva el problema.
- Apego a la formalidad y estándares requeridos.
- Que el análisis de corrección sea exhaustivo.
- Que el prototipo corresponda al modelo diseñado y no presente errores de sintaxis.
- La asistencia a las clases magistrales y a los laboratorios.
- El esfuerzo y dedicación en la resolución de problemas.
- Que la documentación permita reconocer la forma en que se ha abordado el problema y la estructura del programa implementado.
- En las pruebas escritas se consideran en forma parcial los aspectos considerados en proyectos de programación bajo problemas que requieren un menor tiempo de desarrollo y en una modalidad que no requiere uso del computador, así como la comprensión conceptual.

DATOS DEL DOCENTE

NOMBRE :

PREGRADO :

POSTGRADO :

ASESORIAS: FIRMA DE ESTUDIANTES

NOMBRE	FIRMA	CÓDIGO	FECHA
1.			
2.			
3.			

FIRMA DEL DOCENTE

FECHA DE ENTREGA: _____