



Ingeniería Eléctrica

División

Ingeniería en Computación

Departamento

Ingeniería en Computación

Carrera en que se imparte

Asignatura:

Obligatoria

Optativa

Horas:

Teóricas

Prácticas

Total (horas):

Semana

16 Semanas

Modalidad: Curso, laboratorio

Asignatura obligatoria antecedente: Computación para Ingenieros.

Asignatura obligatoria consecuente: Ninguna.

Objetivo(s) del curso:

El alumno empleará estructuras de almacenamiento de datos complejas para la resolución de problemas numéricos. Seleccionará y aplicará métodos numéricos para obtener soluciones aproximadas de modelos matemáticos que no se pueden resolver por métodos analíticos; y desarrollará programas tanto en lenguaje estructurado como orientado a objetos que implementen dichos métodos numéricos.

Temario

NÚM.	NOMBRE	HORAS
1.	Programación avanzada en lenguaje estructurado	6.0
2.	Aproximación numérica, errores y métodos numéricos iniciales	12.0
3.	Fundamentos de la programación orientada a objetos	6.0
4.	Programación orientada a objetos	6.0
5.	Métodos numéricos para solución de sistemas y ecuaciones avanzadas	12.0
6.	Programación orientada a objetos avanzada	6.0
		48.0
	Prácticas de laboratorio	32.0
	Total	80.0



1 Programación avanzada en lenguaje estructurado

Objetivo: El alumno elaborará programas en lenguaje C empleando estructuras de almacenamiento de datos complejas.

Contenido:

- 1.1 Arreglos de varias dimensiones y arreglos de apuntadores
- 1.2 Estructuras.
- 1.3 Archivos y bancos de datos
- 1.4 Desarrollo de programas simples con estructuras de almacenamiento complejas

2 Aproximación numérica, errores y métodos numéricos iniciales

Objetivo: El alumno aplicará los primeros métodos de solución numéricas considerando y minimizando los errores y la convergencia

Contenido:

- 2.1 Aproximación numérica y errores.
- 2.2 Solución numérica de ecuaciones algebraicas y trascendentes
- 2.3 Solución numérica de sistemas de ecuaciones lineales
- 2.4 Interpolación numérica
- 2.5 Desarrollo de programas en lenguaje estructurado para la implementación de los métodos de este tema

3 Fundamentos de la programación orientada a objetos

Objetivo: El alumno explicará los diferentes paradigmas de programación, así como los conceptos y diseño de la programación orientada a objetos en solución de problemas.

Contenido:

- 3.1 Paradigmas de programación: imperativa, funcional, lógica, declarativa y orientada a objetos
- 3.2 Conceptos manejados en la programación orientada a objetos: objetos, métodos, mensajes, clase, herencia, encapsulamiento, polimorfismo.
- 3.3 Diseño de programación orientada a objetos. Notación UML

4 Programación orientada a objetos

Objetivo: El alumno conocerá y aplicará las técnicas y herramientas de la programación orientada a objetos para la solución de problemas.

Contenido:

- 4.1 Teoría del Diseño de jerarquía de clases
- 4.2 Control de flujo.
 - 4.2.1 Sentencia if-else.
 - 4.2.2 Sentencia switch.
 - 4.2.3 Ciclo for.

PROGRAMACIÓN AVANZADA Y MÉTODOS NUMÉRICOS

(3 / 5)



- 4.2.4 Ciclo while y do-while.
- 4.3 Tipos de Clase.
 - 4.3.1 abstract, final, public, private.
 - 4.3.2 Métodos constructores.
 - 4.3.3 Interfaces.
- 4.4 Resolución de problemas matemáticos, físicos y químicos sencillos

5 Métodos numéricos para solución de sistemas y ecuaciones avanzadas

Objetivo: El alumno aplicará métodos numéricos para la solución numérica de sistemas y ecuaciones avanzadas.

Contenido:

- 5.1 Derivación e integración numérica
- 5.2 Solución numérica de ecuaciones y sistemas de ecuaciones diferenciales.
- 5.3 Solución de ecuaciones en derivadas parciales
- 5.4 Desarrollo de programas en lenguaje orientado a objetos para la implementación de los métodos de este tema

6 Programación orientada a objetos avanzada

Objetivo: El alumno aplicará los conceptos avanzados de la programación orientada a objetos para la resolución de problemas complejos.

Contenido:

- 6.1 Multihilos.
- 6.2 Flujos de Datos.
- 6.3 Resolución de problemas complejos

Bibliografía básica:

BURDEN, L. R. , FAIRES, J.D.
Análisis Numérico
7ª. Edición
México
Thomson International, 2003

CHAPRA, Steven C, CANALE, Raymond P.
Métodos Numéricos para Ingenieros
3ª. Edición
México
Mc. Graw-Hill, 1999

Temas de la asignatura para los que se recomienda

2 y 5

2 y 5

PROGRAMACIÓN AVANZADA Y MÉTODOS NUMÉRICOS

(4 / 5)



DEITEL, Harvey M.; DEITEL, Paul J
C++ How to Program
4th. Edition
USA
Prentice Hall., 2002

3, 4,5 y 6

DEITEL, HARVEY M.; DEITEL, PAUL J
Java How to Program
5th. Edition
Prentice Hall., 2002

3, 4,5 y 6

FELLEISEN, M.; FINDLET, R.B.; FLATT, M.
How to Design Class Hierarchies: an introduction to object-oriented programming.
Cambridge, USA
MIT Press, 2004

3 y 4

GERALD, Curtis F.
Análisis Numérico
6ª. Edición
México
Prentice Hall, 2001

2 y 5

KERNIGHAN,B.W.; RITCHIE, D.; RITCHIE, D.M.
C Programming Language
2th. Edition
USA
Prentice Hall, 1988

1

PENDER, TOM
UML Bible
Indianapolis, IN. USA
Wiley. 2003

3 y 4

Bibliografía complementaria:

ROQUES, PASCAL
UML in Practice
England, Great Britain
Jhon Wiley & Sons. 2004

3 y 4

**Sugerencias didácticas:**

Exposición oral	<input checked="" type="checkbox"/>	Lecturas obligatorias	<input checked="" type="checkbox"/>
Exposición audiovisual	<input checked="" type="checkbox"/>	Trabajos de investigación	<input type="checkbox"/>
Ejercicios dentro de clase	<input checked="" type="checkbox"/>	Prácticas de taller o laboratorio	<input checked="" type="checkbox"/>
Ejercicios fuera del aula	<input checked="" type="checkbox"/>	Prácticas de campo	<input type="checkbox"/>
Seminarios	<input type="checkbox"/>	Otras	<input type="checkbox"/>

Forma de evaluar:

Exámenes parciales	<input checked="" type="checkbox"/>	Participación en clase	<input checked="" type="checkbox"/>
Exámenes finales	<input checked="" type="checkbox"/>	Asistencias a prácticas	<input checked="" type="checkbox"/>
Trabajos y tareas fuera del aula	<input checked="" type="checkbox"/>	Otras	<input type="checkbox"/>

Perfil profesiográfico de quienes pueden impartir la asignatura

Egresado de la carrera de Ingeniero en Computación o afín. Conocimientos y experiencia en el diseño de algoritmos y programas en los paradigmas estructurado y orientado a objetos, así como en el desarrollo de proyectos de software.