

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA



PROGRAMA DE ESTUDIO

PROGRAMACIÓN AVANZADA Y MÉTODOS NUMÉRICOS

1312

3°

8

Asignatura

Clave

Semestre

Créditos

Ingeniería Eléctrica

Ingeniería en Computación

Ingeniería en Telecomunicaciones

División

Departamento

Carrera en que se imparte

Asignatura:

Obligatoria

Optativa

Horas:

Teóricas

Prácticas

Total (horas):

Semana

16 Semanas

Aprobado:
Consejo Técnico de la Facultad

Consejo Académico del Área de las Ciencias
Físico Matemáticas y de las Ingenierías

Fecha:
25 de febrero, 17 de marzo y 16 de junio de 2005

11 de agosto de 2005

Modalidad: Curso, laboratorio.

Asignatura obligatoria antecedente: Computación para Ingenieros.

Asignatura obligatoria consecuente: Ninguna.

Objetivo(s) del curso:

El alumno elaborará programas en lenguaje C que implementen algoritmos de métodos numéricos utilizando tipos de datos de mayor complejidad. Además conocerá y aplicará las técnicas y herramientas de la programación orientada a objetos para la solución de problemas de métodos numéricos utilizando un lenguaje orientado a objetos para su implementación (Java, C++ ó C#).

Temario

NÚM.	NOMBRE	HORAS
1.	Programación de métodos numéricos en lenguaje C	12.0
2.	Fundamentos de la programación orientada a objetos	6.0
3.	Programación orientada a objetos	9.0
4.	Programación orientada a objetos de métodos numéricos	12.0
5.	Programación orientada a objetos avanzada	9.0
		48.0
	Prácticas de laboratorio	32.0
	Total	80.0



1 Programación de métodos numéricos en lenguaje C

Objetivo: El alumno elaborará programas en lenguaje C implementando algunos métodos numéricos.

Contenido:

- 1.1 Apuntadores y arreglos de apuntadores
- 1.2 Estructuras y Archivos
- 1.3 Solución numérica de ecuaciones algebraicas y trascendentes
- 1.4 Solución numérica de sistemas de ecuaciones lineales
- 1.5 Interpolación, derivación e integración numéricas

2 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:

- 2.1 Paradigmas de Programación: Imperativa, funcional, lógica, declarativa y orientada a objetos
- 2.2 Conceptos de Programación Orientados a Objetos.
 - 2.2.1 Objetos (Estado, Comportamiento, Identidad).
 - 2.2.2 Métodos.
 - 2.2.3 Clase.
 - 2.2.4 Mensajes.
 - 2.2.5 Herencia.
 - 2.2.6 Encapsulamiento.
 - 2.2.7 Polimorfismo.
- 2.3 Diseño de Programación Orientada a Objetos.
 - 2.3.1 Planteamiento de Clases y Objetos.
 - 2.3.2 Diagrama de clases (UML).
 - 2.3.3 Diagrama de estado.
 - 2.3.4 Casos de uso.
 - 2.3.5 Relación entre clases y objetos.

3 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:

- 3.1 Teoría del Diseño de jerarquía de clases
- 3.2 Control de flujo.
 - 3.2.1 Sentencia if-else.
 - 3.2.2 Sentencia switch.
 - 3.2.3 Ciclo for.
 - 3.2.4 Ciclo while y do-while.
- 3.3 Tipos de Clase.



3.3.1 abstract, final, public, private.

3.3.2 Métodos constructores.

3.3.3 Interfaces.

3.4 Resolución de problemas matemáticos, físicos y químicos sencillos

4 Programación orientada a objetos de métodos numéricos.

Objetivo: El alumno aplicará las técnicas y herramientas de la programación orientada a objetos para la implementación de métodos numéricos.

Contenido:

4.1 Implementación de interpolación, derivación e integración numérica

4.2 Implementación de solución numérica de ecuaciones y sistemas de ecuaciones diferenciales.

4.3 Implementación de solución numérica de ecuaciones en derivadas parciales

5 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:

5.1 Multihilos.

5.1.1 Creación de un hilo.

5.1.2 Arranque de un hilo

5.1.3 Manipulación de un hilo

5.1.4 Suspensión de un hilo

5.1.5 Parada de un hilo

5.2 Flujos de Datos.

5.2.1 La clase system(stdin, stdout, stderr).

5.2.2 Archivos

5.3 Resolución de problemas complejos

Bibliografía básica:

BURDEN, L. R. , FAIRES, J.D.

Análisis Numérico

7ª. Edición

México

Thomson International, 2003

Temas para los que se recomienda:

1 y 4

CHAPRA, Steven C, CANALE, Raymond P.

Métodos Numéricos para Ingenieros

3ª. Edición

México

Mc. Graw-Hill, 1999

1 y 4

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 y 5

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

3, 4 y 5

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

2 y 3

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

1 y 4

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

2

Bibliografía complementaria:

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

2

**Sugerencias didácticas:**

Exposición oral	<input checked="" type="checkbox"/>
Exposición audiovisual	<input checked="" type="checkbox"/>
Ejercicios dentro de clase	<input checked="" type="checkbox"/>
Ejercicios fuera del aula	<input checked="" type="checkbox"/>
Seminarios	<input type="checkbox"/>

Lecturas obligatorias	<input checked="" type="checkbox"/>
Trabajos de investigación	<input type="checkbox"/>
Prácticas de taller o laboratorio	<input checked="" type="checkbox"/>
Prácticas de campo	<input type="checkbox"/>
Otras	<input type="checkbox"/>

Forma de evaluar:

Exámenes parciales	<input checked="" type="checkbox"/>
Exámenes finales	<input checked="" type="checkbox"/>
Trabajos y tareas fuera del aula	<input checked="" type="checkbox"/>

Participación en clase	<input checked="" type="checkbox"/>
Asistencias a prácticas	<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 estructurada y orientado a objetos, así como en el desarrollo de proyectos de software.