



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE CIENCIAS
PLAN DE ESTUDIOS DE LA LICENCIATURA EN
FÍSICA BIOMÉDICA
Programa de la asignatura



Algoritmos Computacionales

Clave: 1430	Semestre: 4°	Campo de conocimiento: Tecnologías de la Información	No. Créditos: 6
Carácter: Obligatorio		Horas	Horas por semana
Tipo: Teórico-Práctica		Teoría: 2	Práctica: 2
Modalidad: Curso		Duración del programa: 16 semanas	

Seriación: No (x) Si () Obligatoria () Indicativa ()

Asignatura antecedente: Ninguna

Asignatura subsecuente: Ninguna

Objetivo general: Describir la metodología y la estructura de programas de cómputo en pseudocódigo.

Objetivos específicos:

1. Identificar la lógica de construcción de pseudocódigo.
2. Desarrollar programas en pseudocódigo.
3. Escribir programas computacionales para resolver problemas específicos.

Índice Temático

Unidad	Tema	Horas	
		Teóricas	Prácticas
1	Estructura básica de la programación	3	3
2	Construcción de pseudocódigo para cómputo científico	9	9
3	Estructuración de código computacional	8	8
4	Programación y visualización de datos	9	9
5	Introducción a otros lenguajes de programación	3	3
Total de horas:		32	32
Suma total de horas:		64	

Contenido Temático

Unidad	Temas y subtemas
1	Estructura básica de la programación 1.1. Algoritmos y diagramas de flujo de un programa. 1.2. Estructura de la programación modular. 1.3. Estructura lógica. (Ciclos más utilizados FOR, WHILE, IF, CASE). 1.4. Sistemas numéricos de punto flotante y cálculo del error numérico.
2	Construcción de pseudocódigo para cómputo científico 2.1. Solución de ecuaciones diferenciales ordinarias.

	2.2. Método de Runge-Kutta. 2.3. Solución de problemas matriciales. 2.4. Problema de eigenvalores. 2.5. Transformada de Laplace y Fourier. 2.6. Caminante aleatorio. 2.7. Programa de comunicación y/o captura de datos con un dispositivo externo.
3	Estructuración de código computacional 3.1. Estructura de la programación modular en GNU (ejemplo Python). 3.2. Estructura de la programación modular en metalenguajes (ejemplo MATLAB). 3.3. Herramientas de comunicación con dispositivos externos (ejemplos Python, MATLAB, LabView).
4	Programación y visualización de datos 4.1. Uso de la librerías Matplotlib y ndimage, y programas gnuplot y VPython. 4.2. Visualización de sistemas de ecuaciones diferenciales. 4.3. Manipulación de imágenes digitales. 4.4. Integración de programas con interfaces gráficas. 4.5. Aplicaciones computacionales (proyecto).
5	Introducción a otros lenguajes de programación 5.1. Presentación básica de otros lenguajes formales como C/C++, Java, R. 5.2. Actualización básica de plataformas de computación.

Bibliografía básica:

Cairó O. Metodología de la programación, algoritmos, diagramas de flujo y programas. México: Alfaomega; 2003.

Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms. 3rd ed. USA: The MIT Press; 2009.

Downey AB, Elkner J, Meyers C. How to think like a computer scientist: learning with Python. USA: Green Tea Press; 2002.

Elden L, Linde WK. Numerical analysis, an introduction. Boston (USA): Boston Academic Press; 1990.

Felleisen M, Findler RB, Flatt M, Krishnamurthi S. How to design programs, an introduction to programming and computing. Massachusetts (USA): The MIT Press; 2001.

Greenbaum A, Chartier TP. Numerical methods, design, analysis, and computer implementation of algorithms. USA: Princeton University Press; 2012.

Hamming RW. Numerical methods for scientists and engineers. USA: Dover; 1987.

Kahaner D, Moler C, Nash S. Numerical methods and software. USA: Prentice Hall; 1989.

Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipes, the art of scientific computing. 3rd ed. Cambridge (UK): Cambridge University Press; 2007.

Bibliografía complementaria:

Mathews J, Fink K. Métodos numéricos con MATLAB. (España): Prentice Hall; 2007.

<http://www.python.org>

<http://www.scipy.org/>

<http://matplotlib.org/>

Sugerencias didácticas:

Exposición oral	(x)
Exposición audiovisual	(x)
Ejercicios dentro de clase	(x)
Ejercicios fuera del aula	(x)
Seminarios	()
Lecturas obligatorias	()
Trabajo de investigación	()
Prácticas de taller o laboratorio	(x)
Prácticas de campo	()
Otras:	(x)
Aprendizaje basado en problemas	

Mecanismos de evaluación del aprendizaje de los alumnos:

Exámenes parciales	()
Examen final escrito	()
Trabajos y tareas fuera del aula	(x)
Exposición de seminarios	(x)
Participación en clase	()
Asistencia	()
Seminario	()
Otras:	(x)
Entrega de programas de cómputo para la solución de problemas específicos.	

Perfil profesiográfico: Matemático, Físico o Ingeniero con experiencia en cómputo científico y en docencia.

