



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



ESCUELA  
NACIONAL  
de CIENCIAS  
de la TIERRA

Escuela Nacional de Ciencias de la Tierra  
Plan de Estudios de la Licenciatura en Ciencias de la Tierra

Programa			
Herramientas Computacionales			
Clave	Semestre	Créditos	Campo de conocimiento:
			1
			Etapa de formación:
			Básica
Modalidad	Curso(X) Taller() Lab() Seminario() Otras		Tipo
			T ( ) P ( ) T/P (X)
Carácter	Obligatorio (X) Optativo ( )		Horas
	Obligatorio E ( ) Optativo E ( )		
Duración	16 semanas		Semana
			Semestre
			Teóricas: 2
			Teóricas: 32
		Prácticas: 2	Prácticas: 32
		Total:4	Total:64
Seriación			
Ninguna ( )			
Obligatoria ( )			
Asignatura antecedente			
Asignatura subsecuente			
Indicativa ( X )			
Asignatura antecedente		Ninguna	
Asignatura subsecuente		Herramientas Computacionales Avanzadas	
Objetivo general:			
Comprender el uso informado y aplicado de las funciones matemáticas en sistemas de ambientes de cómputo y software libre.			
Objetivos particulares:			
-Identificar las estructuras básicas de programación en distintos ambientes.			
-Manejar lenguajes de programación orientado a problemas relacionados con el Sistema Tierra.			
-Elaborar bases de datos robustas.			
Índice temático			
	Tema	Horas	
		Semestre	
		Teóricas	Prácticas
1	Construcción y estructura de pseudocódigo y programas de cómputo	4	6
2	Estructuras básicas de programación	10	8
3	Programación y visualización de datos	6	6
4	Diferentes lenguajes de programación y aplicaciones en Ciencias de la Tierra	8	8



5	Manejo de archivos	4	4
<b>Subtotal</b>		<b>32</b>	<b>32</b>
<b>Total</b>		<b>64</b>	

<b>Contenidos temáticos</b>	
Temas	Subtemas
1	Construcción y estructura de pseudocódigo y programas de cómputo 1.1 Algoritmos y diagramas de flujo de un programa 1.2 Estructura de la programación modular 1.3 Sistemas numéricos de punto flotante y cálculo del error numérico
2	Estructuras básicas de programación 2.1 Estructura lógica. (Ciclos más utilizados FOR, WHILE, IF, CASE) 2.2 Estructura de la programación modular en GNU (ejemplo Python) <sup>[1]</sup> <sub>[2]</sub> 2.3 Estructura de la programación modular en metalenguajes (ejemplo MATLAB) <sup>[1]</sup> <sub>[2]</sub> 2.4 Herramientas de comunicación con dispositivos externos (ejemplos Python, MATLAB, LabView). 2.6 Solución de ecuaciones diferenciales ordinarias, problemas matriciales y valores propios, métodos de derivación e integración numérica. 2.7 Programación de captura de datos y lectura-escritura de archivos.
3	Programación y visualización de datos 3.1 Uso de las bibliotecas Matplotlib y ndimage, y programas de visualización gráfica 3.2 Visualización de sistemas de ecuaciones diferenciales 3.3 Manipulación de imágenes digitales e interfaces gráficas
4	Diferentes lenguajes de programación y aplicaciones en Ciencias de la Tierra 4.1 Presentación básica de otros lenguajes formales como C/C++, Java, R. 4.2 Aplicaciones para modelos de poblaciones. 4.3 Diferencias finitas para EDP relacionadas con el flujo de agua. 4.4 Modelos sobre interacciones en ecosistemas. Sistemas dinámicos. 4.5 Análisis de datos experimentales.
5	Manejo de archivos 5.1 Creación, apertura y manipulación de archivos con lenguajes de programación. 5.2 Modificación de datos dentro de un archivo. 5.3 Manipulación de bases de datos y grandes cantidades de archivos. 5.4 Manipulación de imágenes y edición automatizada de imágenes.

<b>Estrategias didácticas</b>
Ejercicios fuera del aula
Ejercicios en clase
Exposición oral
Presentación de proyectos en grupo
Generar una página tipo blog con las actividades que se realicen en clase

<b>Evaluación del aprendizaje</b>
-----------------------------------

Programas y algoritmos
Trabajos y tareas
Exámenes parciales
Examen final
Proyecto final

<b>Perfil profesiográfico del docente</b>	
Título o grado	Matemático(a), Físico(a), Actuario(a), Científico(a) de la Computación, Ing. en Sistemas Computacionales, u otro afín. Nivel mínimo de Maestría.
Experiencia docente	Con experiencia docente de al menos dos años en: - El nivel de licenciatura de las carreras de Ciencias de la Computación, Matemáticas, Física, Ingeniería, Ciencias de la Tierra, u otras afines. - Posgrados afines.
Otras características	Haber tomado el curso de Formación Docente impartido en la Escuela Nacional de Ciencias de la Tierra. Con experiencia en el manejo de los distintos lenguajes de programación y software libre.

<b>Bibliografía básica</b>
Cormen TH, Leiserson CE, Rivest RL, Stein C. (2009) Introduction to algorithms. 3rd ed. USA: The MIT Press;.
Downey AB, Elkner J, Meyers C. (2002) How to think like a computer scientist: learning with Python. USA: Green Tea Press;.
Angela B. Shiflet y George W. Shiflet. (2014) Introduction to Computational Science: Modeling and Simulation for the Sciences - Second Edition. Princeton University Press; Edición: 2.
John M. Stewart. (2014) Python for Scientists. Cambridge University Press; Edición: 1
Johnny Wei-Bing Lin. (2012) A Hands-On Introduction to Using Python in the Atmospheric and Oceanic Sciences. First Edition () <a href="https://www.johnny-lin.com/pyintro/">https://www.johnny-lin.com/pyintro/</a>
<b>Mesografía</b>
<a href="http://jupyter.org">http://jupyter.org</a>
<a href="http://matplotlib.org">http://matplotlib.org</a>
<a href="http://www.scipy.org">http://www.scipy.org</a>

<b>Bibliografía complementaria</b>
Newman M. (2012) Computational Physics. CreateSpace Independent Publishing Platform;.
Press WH, Teukolsky SA, Vetterling WT, Flannery BP. (2007) Numerical recipes, the art of scientific computing. 3rd ed. Cambridge (UK): Cambridge University Press
<b>Mesografía</b>
<a href="https://github.com/">https://github.com/</a>
<a href="http://www.python.org">http://www.python.org</a>
<a href="http://www.sagemath.org/">http://www.sagemath.org/</a>
<a href="https://julialang.org/">https://julialang.org/</a>