

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

Facultad de Ciencias Básicas y Tecnologías
Física

Sílabo
Programación

- 1. Descripción.** La asignatura Programación de la Facultad de Ciencias Básicas y Tecnologías se describe como un espacio académico que hace parte de las Actividades Académicas Básicas del Programa de Física. En esta materia se estudian los fundamentos de la programación secuencial. Se busca principalmente desarrollar en el estudiante un amplio conjunto de herramientas conceptuales inherentes a las ciencias de la computación y en particular en Física que le permitan desarrollar, implementar y proponer soluciones a problemas de carácter científico. Adicionalmente, la asignatura de programación es considerada como el punto de partida hacia asignaturas avanzadas de los semestres superiores tales como: Métodos Numéricos y Física computacional entre otras.
- 2. Justificación.** Hoy en día el uso de los computadores y lenguajes de programación son considerados por la comunidad científica como una herramienta indispensable y fundamental en la formación de todo profesional en Física. Puede decirse que los lenguajes de programación son una herramienta bastante versátil que permite realizar una gran cantidad tareas de manera eficiente, como por ejemplo: el procesamiento de datos, cálculos numéricos, visualización gráfica y cálculo simbólico, entre muchas otras. Adicionalmente, gracias al uso de lenguajes de programación es posible hoy en día simular-modelar- fenómenos físicos bajo condiciones de interés, predecir posibles comportamientos de la naturaleza. Aun cuando los experimentos no se pueden realizar debido a limitaciones de equipos o por las condiciones mismas del experimento.
- 3. Administración del espacio académico, núcleo o cátedra.**

Generalidades	Detalle
Modalidad	Presencial
Espacio académico, núcleo o cátedra	Programación
Código	120710206
Tipo de Actividad Académica	Profesional
Naturaleza	Teórico-practica
Enseñanzas	Unidad 1: Fundamentos de la programación Unidad 2: Python y algoritmia

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

	Unidad 3: Programación orientada a objetos y librerías Unidad 4: Programación para la Física
Evaluación	Cuantitativa y cualitativa
Créditos	3 (tres)
Horas semanales	6 (seis)
Total de horas por semestre	196 (ciento noventa y dos)
Horas de docencia directa (HDD) o de trabajo interactivo (HTIC)	6 horas por semana
HDD/HTIC en docencia	6 horas por semana
HDD/HTIC en investigación	1 horas por semana
HDD/HTIC en extensión	0 horas por semana
Horas de trabajo independiente	6 horas por semana
Habitable	Sí
Validable	Sí
Homologable	Sí
Requisitos	Ecuaciones diferenciales

4. Relación de los alcances de la formación.

4.1. Unidades de competencia propias del espacio académico, núcleo o cátedra.

Unidad de competencia – UC2

El estudiante identifica e interpreta fenómenos de la física y resuelve problemas en los ámbitos teórico-experimentales, expresándolos de manera coherente y utilizando competencias comunicativas en lengua materna o alguna lengua extranjera.

4.2. Resultados de aprendizaje.

Se espera que al finalizar el curso el estudiante alcance los siguientes resultados:

- El estudiante comprende a profundidad el funcionamiento de un computador y sabe de que manera puede aportar, como físico, al desarrollo de nuevas tecnologías.
- El estudiante integre la programación a su conjunto de herramientas del día a día para resolver problemas de la física.
- El estudiante es capaz de encontrar varias posibles soluciones (secuenciales y recursivos) para un problema y entre ellas seleccionar la más efectiva.
- El estudiante es capaz de interpretar códigos realizados por otros autores.
- El estudiante es capaz de implementar algoritmos que le permiten solucionar problemas de la física y matemáticas

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

- El estudiante manipula correctamente archivos de entrada o salida de diferente tipo y los representa gráficamente de forma estética, de acuerdo a lo que pueda ser solicitado por una revista científica internacional.
- Manipula y desarrolla rutinas y librerías de uso general en los programas de cómputo científico.

5. Enseñanzas.

UNIDAD	CONTENIDO	DURACIÓN semanas
UNIDAD 1- FUNDAMENTOS DE LA COMPUTACIÓN	Breve discusión sobre aspectos de los lenguajes de programación: compilados vs. Interpretados, funcionamiento interno de un computador, diseño de algoritmos y pseudo código, tipos de datos, operadores y expresiones, entrada y salida de datos por teclado, variables locales, globales, programa principal y parámetros, programación orientada a objetos, clases, instancias, constructor de una clase, métodos mágicos principales <code>__str__</code> , <code>__add__</code> , <code>__eq__</code> , <code>__sub__</code> , <code>__call__</code> , fundamentos que retornan valores con argumentos posicionales y paramétricos, problemas resueltos y propuestos.	I – IV
UNIDAD 2- PYTHON Y ALGORÍTMIA	Buenas prácticas de la programación, enteros, punto flotante, punto flotante de precisión, booleanos, listas, cadenas, tuplas, diccionarios, operaciones sobre objetos, métodos <code>__getitem__</code> <code>__setitem__</code> , entrada y salida de datos por teclado, cadenas con formato, flujo, anidamiento, construcción de algoritmos y	V - VII

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

	recurrencia, problemas resueltos y propuestos	
UNIDAD 3 – PROGRAMACIÓN ORIENTADA A OBJETOS Y LIBRERÍAS	Funciones, clases, herencia, polimorfismo, cohesión, abstracción, creación de librerías, importación de librerías, visualización de datos en Matplotlib (Tiro parabólico, movimiento planetario, graficas 2D y 3D, graficas de contorno, entre otros), importación y exportación de datos desde el disco duro, librerías para trabajar datos, álgebra lineal y estadística	VIII – XI
UNIDAD 4. – PROGRAMACIÓN PARA LA FÍSICA	Implementación de la recursión en problemas complejos (torre de Hanoi, explorar un laberinto, métodos numéricos), estructuras lineales, pila de datos (<i>stacks</i>), cola de datos (<i>queues</i>), remoción de elementos de la cola (<i>deques</i>), tipos de datos abstractos, implementación de listas desordenadas, listas ligadas, vectores, introducción al análisis de datos	XII - XVI

6. Procesos integrativos.

Los estudiantes tienen la opción de pertenecer a los semilleros de la Facultad y/o vincularse a los grupos de investigación del Programa mediante procesos de iniciación científica.

7. Metodología.

El espacio académico de la asignatura de programación se desarrollará mediante exposiciones del profesor acompañadas de explicaciones de código fuente que sirva para orientar discusiones grupales en clase, desarrollo de ejercicios de aplicación y trabajo por parte del estudiante relacionados con problemas propuestos en clase. En

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

principio, los temas de la asignatura están enfocados para orientar el aprendizaje del lenguaje de programación en Python y los fundamentos de la algoritmia.

8. Evaluación.

El proceso de evaluación de los estudiantes de la asignatura Programación se realizará teniendo en cuenta la normativa establecida en el estatuto estudiantil, la cual está consignada en el acuerdo No. 066 del 22 de diciembre del año 2000 del Consejo Superior de la Universidad del Quindío. Teniendo presente que la naturaleza del presente espacio académico es de carácter teórico-práctico se propone como metodología de evaluación lo siguiente:

- 4 parciales25% cada uno

9. Bibliografía.

Textos guía:

- 1- Brian Heinold. *Practical Introduction to Python Programming*. Department of Mathematics and Computer Science Mount St. Mary's University. 2012.
- 2- Jennifer Campbell, Paul Gries, Jason Montojo, Greg Wilson. *Practical Programming An Introduction to Computer Science Using Python 3*. Pragmatic Bookshelf. 2013.
- 3- John M. Zelle. *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates Inc. 2005.
- 4- Wes McKinney. *Python for data analysis*. O'Reilly. 2018.
- 5- Sandro Tosi. *Matplotlib for developers*. Packt Publishing. 2009.
- 6- Pablo M. García Corzo. *Proyecto Phythones Simulaciones Físicas en Visual Python*. Alqua. 2008.

Textos de lectura complementaria:

- 7- S. S. Skiena. *The Algorithm Design Manual*, Springer. 2008.
- 8- Z.A. Shaw. *Learn Python 3 the Hard Way*. Addison-Wesley Professional. 2013.
- 9- B. Miller, D. Ranum. *Problem Solving with Algorithms and Data Structures*, Franklin, Beedle & Associates, 2006.
- 10-A. Lancaster, G. Webster. *Python for the life sciences*. Apress. 2019.
- 11-J.M. Kinder, P. Nelson. *A student's guide to Python for physical modeling*. Princeton University Press. 2015.
- 12-Phuong Vo T H, Martin Czygan, Ashish Kumar, *Python: Data Analytics and Visualization: Understand, evaluate, and visualize data*, Packt Publishing. 2017
- 13-Zach Radtka, Donald Miner, *Hadoop with Python*, O'Reilly Media. 2015.

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

- 14-R. Wachenchauzer, M. Manterola, M. Curia, M. Medrano, N. Paez. *Algoritmos de Programación con Python*. Librosweb. Recuperado de <https://uniwebsidad.com/libros/algoritmos-python> Recuperado el 17/02/2021
- 15-SQLModel, *SQL databases in Python, designed for simplicity, compatibility, and robustness* Recuperado de <https://sqlmodel.tiangolo.com/#license> el 14/02/2022
- 16-FastAPI framework, *high performance, easy to learn, fast to code, ready for production*, Recuperado de <https://fastapi.tiangolo.com/> el 14/02/2022
- 17-Ron White. *How Computers Work The Evolution of Technology*. QUE. 2014.
- 18-Google. Google Python Style Guide. Recuperado de: <https://google.github.io/styleguide/pyguide.html> recuperado en 22/07/2020.
- 19-Miktex. Recuperado de <https://miktex.org/download/ctan/systems/win32/miktex/setup/windows-x64/basic-miktex-2.9.7442-x64.exe> recuperado el 17/06/2020
- 20-Anaconda. Recuperado de <https://www.anaconda.com/products/individual#Downloads> recuperado el 17/06/2020

10. Historial de revisión.

Primera versión: 2020-II

Revisión y actualización: 2021-I

Revisión y actualización: 2022-I

Vigencia del sílabo. Semestre I de 2023

Responsables:

Santiago Echeverri Arteaga, Ph.D.

Edgar Arturo Gómez, Ph.D.