

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

Facultad de Ciencias Básicas y Tecnologías
Física

Sílabo
Programación

- 1. Descripción.** La asignatura Programación se centrará en la aplicación de Python a problemas de la física y la ciencia de datos, con un enfoque en proyectos prácticos. A lo largo del curso, los estudiantes aprenderán los fundamentos de la programación orientada a objetos desde el inicio, junto con el uso de bibliotecas científicas como NumPy, SciPy, y Matplotlib, así como herramientas de manejo y análisis de datos con Pandas. Además, se introducirá a los estudiantes en el uso de SQL para la manipulación de bases de datos y se ofrecerá una introducción básica a las técnicas de machine learning. El curso está diseñado para desarrollar competencias prácticas y proporcionar a los estudiantes una sólida base para resolver problemas complejos en sus estudios y futuras investigaciones en física.
- 2. Justificación.** La asignatura Programación es fundamental en la formación de los estudiantes de física, ya que la programación se ha convertido en una herramienta indispensable en la comunidad científica. El uso de lenguajes de programación como Python permite realizar una amplia variedad de tareas de manera eficiente, como el procesamiento de datos, cálculos numéricos, visualización gráfica y modelado de fenómenos físicos. A través de este curso, los estudiantes adquirirán habilidades esenciales para el desarrollo de soluciones a problemas científicos, permitiéndoles simular y predecir comportamientos de la naturaleza, incluso en condiciones experimentales que no se pueden realizar debido a limitaciones de equipos o condiciones experimentales. Además, el curso servirá como base para asignaturas avanzadas como Métodos Numéricos y Física Computacional, integrando la programación en el conjunto de herramientas diarias del estudiante de física y preparando a los estudiantes para contribuir al desarrollo de nuevas tecnologías.
- 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

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

Enseñanzas	Unidad 1: Fundamentos de Programación en Python. Unidad 2: Fundamentos de Programación Orientada a Objetos (POO). Unidad 3: Bibliotecas Científicas en Python. Unidad 4: Análisis y Ciencia de Datos.
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.

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

- El estudiante es capaz de implementar algoritmos que le permiten solucionar problemas de la física y matemáticas
- 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 Programación en Python	<ul style="list-style-type: none"> - Introducción a Python: sintaxis básica
- print () y input () - Tipos de datos básicos: enteros, flotantes, cadenas, y booleanos
- Variables y operadores - Estructuras de control: condicionales (if, elif, else), ciclos (for, while) - Listas, tuplas, diccionarios y conjuntos - Operaciones básicas con colecciones 	I - II
Unidad 2: Fundamentos de Programación Orientada a Objetos (POO)	<ul style="list-style-type: none"> - Introducción a la programación orientada a objetos - Conceptos de clases y objetos - Atributos y métodos - Constructores y destructores - Encapsulamiento y visibilidad - Herencia y polimorfismo - Sobrescritura de métodos - Métodos mágicos y operadores sobrecargados 	III - V
Unidad 3: Bibliotecas Científicas en Python	<ul style="list-style-type: none"> - Introducción a bibliotecas científicas en Python - Instalación y configuración de bibliotecas: NumPy, SciPy, Matplotlib - NumPy: creación y manipulación de arreglos, funciones matemáticas y estadísticas - SciPy: uso para problemas numéricos - Matplotlib: creación de gráficos 2D y 3D, gráficos avanzados (gráficos de contorno) 	VI - VII

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

<p>Unidad 4: Manejo y Análisis de Datos, Ciencia de Datos y Machine Learning</p>	<ul style="list-style-type: none"> - Introducción a Pandas: Series y DataFrames - Operaciones básicas y avanzadas con Pandas - Importación y exportación de datos (CSV, Excel, SQL) - Limpieza y preparación de datos - Visualización de datos con Pandas y Matplotlib - Introducción a la ciencia de datos: conceptos básicos y flujo de trabajo - Técnicas de análisis de datos: estadísticas descriptivas y exploración de datos - Uso de SQL en Python: conexión a bases de datos, consultas básicas y avanzadas - Introducción a Machine Learning: conceptos básicos y aplicaciones, instalación y configuración de Scikit-learn 	<p>VIII - XVI</p>
---	--	-------------------

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 curso se orientará siguiendo una metodología de aprendizaje basado en proyectos, apoyado en clases magistrales y sesiones prácticas en el laboratorio de computación. Las clases magistrales proporcionarán la teoría necesaria y los ejemplos de código que servirán para orientar discusiones grupales y ejercicios prácticos. Los estudiantes trabajarán en proyectos individuales y en grupo, los cuales serán presentados y sustentados periódicamente. Además, se utilizarán herramientas como Git y GitHub para la gestión de versiones y la colaboración en proyectos. La evaluación continua y el feedback serán claves en el proceso de aprendizaje.

8. Evaluación.

La evaluación del curso se realizará mediante la presentación y sustentación de proyectos, la exposición de una librería especializada en Python y la entrega de informes escritos. La distribución de la evaluación será la siguiente:

- Proyecto 1: Implementación de un simulador físico utilizando POO y bibliotecas científicas (30%)**
 - Sustentación oral (15%):** Presentación y explicación del proyecto, demostración del funcionamiento y respuesta a preguntas.
 - Informe escrito (15%):** Documentación del proyecto, incluyendo descripción del problema, metodología, código fuente y resultados.

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

2. **Proyecto 2: Análisis de un conjunto de datos físicos utilizando Pandas y visualización con Matplotlib (30%)**
 - **Sustentación oral (15%):** Presentación y explicación del análisis de datos, visualización de resultados y discusión de hallazgos.
 - **Informe escrito (15%):** Documentación del proyecto, incluyendo descripción del conjunto de datos, análisis realizado, código fuente y conclusiones.
3. **Exposición de una librería especializada en Python (20%)**
 - **Presentación oral (10%):** Exposición detallada de una librería de Python especializada (e.g., Scikit-learn, Seaborn), incluyendo su instalación, funcionalidades principales y ejemplos de uso.
 - **Informe escrito (10%):** Documento explicativo de la librería presentada, con ejemplos de código y aplicaciones.
4. **Tareas y ejercicios prácticos (10%)**
 - **Tareas y ejercicios prácticos (10%):** Realización y entrega de ejercicios prácticos asignados a lo largo del curso, enfocados en la aplicación de los conceptos aprendidos.

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.

MODELO DE SÍLABO

MACROPROCESO DOCENCIA

- 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.
- 14-R. Wachenchauser, 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-Miktek. 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

Revisión y actualización: 2024-II

Vigencia del sílabo. Semestre II de 2024

Responsables:
Santiago Echeverri Arteaga, Ph.D.
Edgar Arturo Gómez, Ph.D.