

HERRAMIENTAS COMPUTACIONALES

CONTENIDO PROGRAMÁTICO

NOMBRE DEL CURSO: Herramientas Computacionales

CÓDIGO DEL CURSO: FISI 2026

UNIDAD ACADÉMICA: Departamento de Física

PRERREQUISITOS: Algorítmica y Programación Orientada por Objetos 1 (ISIS 1204)

I Introducción

Los computadores nos ayudan a organizar, comunicar y procesar información. Hoy en día son esenciales en los mundos de la ciencia, la academia y la técnica. Este curso enseña algunas herramientas computacionales básicas para hacer de los computadores herramientas útiles, poderosas y versátiles para estudiantes de ciencias naturales. El curso desarrolla habilidades de programación en un lenguaje de alto nivel, por ejemplo Python o Matlab; enseña algunos métodos de análisis numérico; y exhibe algunas herramientas útiles en el análisis de datos.

II Objetivos

Los objetivos del curso son:

- Ofrecer herramientas computacionales básicas útiles en la investigación y la vida académica.
- Introducir rutinas sencillas de análisis numérico.
- Desarrollar habilidades de programación.

III Competencias

Al finalizar el curso, se espera que el estudiante esté en capacidad de:

- Utilizar computadores con sistema operativo tipo UNIX.
- Implementar en un lenguaje de programación de alto nivel la solución de problemas computacionales sencillos.
- Manipular, analizar y visualizar datos usando un lenguaje de programación de alto nivel.

IV Contenido

El siguiente enlace contiene los videos referenciados en el programa:

https://www.youtube.com/playlist?list=PLHQztzvthdVM_MGC9dPFKe4hPAwBd_7RJ3

Semana 1 Introducción. Presentación y examen diagnóstico. Uso de binder.

Semana 2 Linux. Introducción a UNIX: filosofía, comandos básicos.

Semana 3 Linux. Editores de texto, control de procesos, redirección.

Videos: 1 y 2 Introducción a UNIX (Ubuntu).

Entrega ejercicio 1.

Semana 4 Introducción a Python: filosofía, sintaxis básica, operaciones aritméticas, condicionales.

Videos: 4 y 5 Introducción a Python.

Entrega ejercicio 2.

Semana 5 Operaciones con cadenas de caracteres, listas y estructuras iterativas.

Video: 6 Python: Listas y strings.

Entrega ejercicio 3.

Semana 6 Definición de funciones, tipos de variables, recursividad.

Video: 7 Python: Funciones, tipos de variables y recursividad.

Entrega ejercicio 4.

Semana 7 Programación orientada a objetos en Python.

Videos: 8 Programación Orientada a Objetos en Python.

Entrega ejercicio 5.

Semana 8 Instalación (pip) e importación de módulos. Cuadernos de iPython.

Videos: 9 Python: Módulos, PIP y cuadernos de Jupyter.

Entrega ejercicio 6.

Semana 9 Semana de trabajo individual.

Semana 10 Introducción a NumPy. Arrays de numpy y operaciones entre arrays.

Video: 10 Numpy con IPython

Entrega ejercicio 7.

Semana 11 Importación de datos. Visualización de datos con matplotlib: plot, scatter, imshow y subplot.

Video: 11 import Matplotlib as plt.

Entrega ejercicio 8.

Semana 12 Análisis numérico: métodos de bisección, método de Newton-Raphson.

Video: 12 Métodos numéricos: ceros de funciones con IPython.

Entrega ejercicio 9.

Semana 13 Introducción a SciPy. Ajustes polinomiales y no polinomiales.

Video: 13 Ajuste a funciones lineales y no lineales.

Entrega ejercicio 10.

Semana 14 Herramientas estadísticas: funciones estadísticas, histogramas y ejemplos de distribuciones.

Video: 14 Herramientas estadísticas con Python.

Entrega ejercicio 11.

Semana 15 Métodos de Monte Carlo.

Video: 15 Métodos Monte Carlo

Entrega ejercicio 12.

Semana 16 Simulación de sistemas físicos.

V Metodología

Se hará énfasis en el trabajo individual por fuera del horario de clase. El profesor entregará una lista de recursos que cada estudiante debe preparar **antes de llegar a la clase presencial**. En este curso los temas a preparar se encuentran en videos.

Al inicio de cada clase el profesor responderá preguntas y hará aclaraciones sobre el material de la semana. Luego los estudiantes resolverán individualmente los ejercicios de la semana. Todo el trabajo es individual y las calificaciones se otorgarán por los resultados entregados exclusivamente en horario de clase por SICUA. Bajo ninguna circunstancia se recibirán ejercicios entregados por fuera del tiempo estipulado (hora oficial del final de la clase más 10 minutos).

Se recomienda traer a clase una memoria USB si le interesa guardar su trabajo. Se espera del estudiante que evite activamente que su trabajo sea plagiado: cualquier sospecha de copia hará que el caso sea llevado automáticamente a proceso disciplinario, por lo que se aconseja que borre sus archivos del computador que usó cuando no los necesite más. La memoria USB también podrá ser útil si por algún motivo su computador pierde la conexión a internet y necesita cambiar de ordenador para subir su trabajo a la plataforma de SICUA.

VI Calificación del curso

- Examen diagnóstico. Bono 20 %.

- 4 entregas en clase 80 % (20 % cada una).
- Examen final 20 %.

Durante el semestre se deben entregar las soluciones a 12 ejercicios tal como indica el programa. Estos ejercicios se deben resolver y subir a sicuaplus antes de la finalización de la clase. Entre esos doce ejercicios se elegirán cuatro al azar que serán calificados, cada uno con un peso del 15 % de la nota final.

VII Bibliografía

Bibliografía principal:

- H. P. Langtangen. *A Primer on Scientific Programming with Python*, 2009.
<http://link.springer.com.ezproxy.uniandes.edu.co:8080/book/10.1007%2F978-3-642-18366-9>

Bibliografía complementaria:

- J. V. Guttag. *Introduction to Computation and Programming Using Python*, 2013.
- K. D. Lee. *Python Programming Fundamentals*, 2011.
<http://link.springer.com.ezproxy.uniandes.edu.co:8080/book/10.1007%2F978-1-84996-537-8>
- S. van Vugt. *Beginning the Linux Command Line*, 2009.
<http://link.springer.com.ezproxy.uniandes.edu.co:8080/book/10.1007%2F978-1-4302-1890-6>
- G. Grätzer. *More Math Into L^AT_EX*, 2007.
<http://link.springer.com.ezproxy.uniandes.edu.co:8080/book/10.1007%2F978-0-387-68852-7>