

## MÉTODOS COMPUTACIONALES 1

NOMBRE DEL CURSO: Métodos Computacionales 1

CÓDIGO DEL CURSO: FISI-2526

UNIDAD ACADÉMICA: Departamento de Física

PRERREQUISITOS: Introducción a la Programación (ISIS-1221). Álgebra Lineal I (MATE-1105).

CRÉDITOS: 3

PERIODO ACADÉMICO: 202020

HORARIO: Mi y Vi, 17:00 a 18:15

---

NOMBRE PROFESOR MAGISTRAL: Jaime Ernesto Forero Romero

CORREO ELECTRÓNICO: [je.forero@uniandes.edu.co](mailto:je.forero@uniandes.edu.co)

HORARIO Y LUGAR DE ATENCIÓN: con cita previa

NOMBRE PROFESOR COMPLEMENTARIA: John Fredy Suárez Pérez

CORREO ELECTRÓNICO: [jf.suarez@uniandes.edu.co](mailto:jf.suarez@uniandes.edu.co)

HORARIO DE ATENCIÓN: con cita previa.

---

### I Introducción

Los métodos computacionales son un aspecto inseparable de cualquier área de trabajo en ciencia e ingeniería. Esto se debe a la facilidad de acceso a computadoras programables y su aumento exponencial en capacidad de procesamiento. Estos recursos para el cómputo sólo se puede aprovechar si las personas interesadas son capaces de utilizarlos tecnología de manera eficiente. De manera complementaria, la obtención y comprensión de los resultados obtenidos con estos métodos computacionales requieren una comprensión básica de probabilidad y estadística.

### II Objetivos

En el curso se presentan algoritmos y técnicas computacionales básicas para:

- analizar datos y modelos con conceptos probabilísticos y métodos estadísticos
- resolver numéricamente problemas que involucren derivadas, integrales y sistemas de ecuaciones algebraicas

### III Competencias a desarrollar

Al finalizar el curso, se espera que el estudiante adquiera las siguientes habilidades.

- Analizar datos usando apropiadamente conceptos básicos de probabilidad y estadística en la solución de problemas en ciencias y otras áreas: variables aleatorias, valores esperados, métodos de Monte-Carlo, pruebas de hipótesis:
  - Diferenciar distribuciones de probabilidad comunmente encontradas en ciencias: normal, poisson, log-normal, etc.
  - Hacer histogramas para identificar la distribución de los datos como función de una variable específica.
  - Estimar la moda, media, mediana, desviación estandar, curtosis y otros momentos de series de datos.
  - Realizar pruebas cuantitativas para ver si dos series de datos corresponden a la misma distribución de probabilidad.

- Utilizar datos para proponer modelos paramétricos y acotar los terminos libres de este.
  - Solucionar sistemas de ecuaciones algebraicas lineales.
  - Resolver problemas de mínimos cuadrados, planteados desde el sistema de ecuaciones lineales correspondiente.
  - Realizar reducción de dimensionalidad de datos a través del método de componentes principales.
- Usar herramientas de Python para análisis estadístico de datos, para la solución de problemas en ciencias y otras áreas.
    - Implementar computacionalment modelos físico–matemáticos para el análisis de datos utilizando **clases** y **funciones**.
    - Utilizar de la biblioteca **pandas** para el manejo eficiente de los datos teniendo en cuenta los siguientes procedimientos: preprocesamiento, re-formato de columnas y normalización.
    - Utilizar la biblioteca **numpy** en la implementación de los algoritmos matemáticos que representen los diferentes sistemas en estudio.
    - Utilizar la biblioteca **matplotlib** para visualizar datos.
  - Solucionar numéricamente problemas sencillos de cálculo diferencial, cálculo integral y álgebra lineal: cálculo de integrales, cálculo de derivadas, sistemas de ecuaciones lineales, integración de funciones, problemas de valores propios.
    - Calcula numéricamente la primera y segunda derivada de funciones de una variable.
    - Calcula numéricamente integrales unidimensionales definidas e indefinidas con los métodos siguientes: trapecio, y cuadratura gaussiana.
    - Encuentra numéricamente la solución a sistemas de ecuaciones algebraicos lineales.
    - Utiliza bibliotecas de álgebra lineal para calcular valores y vectores propios.
    - Interpreta los vectores y valores propios de una matriz de covarianza como herramientas para la reducción de dimensionalidad de un conjunto de datos.

## IV Contenido por semanas

### Semana 1

- Temas: Presentación del curso. Uso de Python en la nube. Comandos básicos Unix. Repaso de Python: variables, listas, iteración diccionarios, lectura de archivos, condicionales.

### Semana 2

- Temas: Repaso de Python: funciones, clases, objetos, numpy, matplotlib.

### Semana 3

- Temas: Introducción a la estadística. Estadística descriptiva.

### Semana 4

- Temas: Elementos básicos de probabilidad.

### Semana 5

- Temas: Variables aleatorias y valores esperados.

### Semana 6

- Temas: Variables aleatorias especiales.

### Semana 7

- Temas: Distribuciones de estadísticas de sampleo.

### Semana 8

- Temas: Estimación de parámetros.

### Semana 9

- Temas: Prueba de hipótesis.

#### Semana 10

- Temas: Regresión.

#### Semana 11

- Temas: Integración.

#### Semana 12

- Temas: Derivadas. Raíces de ecuaciones.

#### Semana 13

- Temas: Interpolación.

#### Semana 14

- Temas: Solución de sistemas de ecuaciones lineales.

#### Semana 15

- Temas: Ajustes por mínimos cuadrados con métodos matriciales.

#### Semana 16

- Temas: Autovalores y autovectores. Análisis de Componentes Principales.

## V Metodología

Los 3 créditos del curso corresponden a 9 horas de dedicación semanal. Cada semana habrá 3,75 horas sincrónicas distribuidas en  $2 \times 1,25 = 2,5$  horas de magistral y 1,25 horas de complementaria. Las 5,25 horas restantes corresponden a trabajo individual para la preparación de los contenidos de la semana. De esto sugerimos dedicarle 4 horas a la magistral y 1,25 horas a la complementaria.

- Antes de cada clase magistral al estudiante se le propone una lectura, videos explicativos de 15 minutos de duración aproximada y un ejercicio para resolver. El tiempo de estudio y trabajo de este material no debe exceder el tiempo previsto para trabajo individual.
- Durante las clases sincrónicas de la magistral habrá una sesión de zoom de 30 minutos para presentar la parte teórica del tema del día, mostrar un ejemplo de implementación y resolver las dudas que se tengan sobre el tema del día. Las dudas serán recogidas en el **Padlet de la magistral**. Los siguientes 45 minutos se dedicarán a la resolución de un ejercicio que se debe responder a través de BrightSpace. La interacción se hará a través de slack.
- Durante las clases sincrónicas de la complementaria habrá una sesión de Clase Remota en SICUA de 30 minutos para resolver dudas de la clase magistral de la semana anterior. Las dudas serán recogidas en el **Padlet de la complementaria**. Los siguientes 45 minutos se dedicarán a la resolución de un ejercicio que se deberá responder a través de SICUA. La interacción se dará a través de slack.

Para el desarrollo de la clase se usará el ambiente Binder accesible desde el repositorio del curso.

## VI Criterios de evaluación

En cada clase de magistral y complementaria habrá un ejercicio para entregar. Solamente recibirán calificación 16 de esos ejercicios, 8 de la magistral y 8 de la complementaria.

El profesor de la clase magistral anunciará los ejercicios que reciben calificación cada dos semanas. Al publicar el enunciado de un ejercicio no se sabe si será calificado o no.

La nota numérica se calcula a partir de estas entregas de ejercicios de programación. Los porcentajes se dividen en

- Ocho (8) Ejercicios de programación propuestos en la Magistral: 70 %.
- Ocho (8) Ejercicios de programación propuestos en la Complementaria: 30 %.

Cada ejercicio recibe el siguiente porcentaje de calificación:

- Entrega de código fuente que incluye las funciones, algoritmos y procedimientos que se solicitan en el enunciado: 30 %.
- El código corre sin errores dentro del intérprete de python: 30 %.
- El código produce la respuesta correcta: 40 %.

El código se calificará dentro del intérprete de binder del repositorio del curso. La nota definitiva se reporta con dos cifras decimales.

Todas las entregas son **individuales**. Si en las entregas se detecta que el trabajo no fue individual (esto incluye colaboración con personas no inscritas en el curso, i.e. a través de “monitorías”) se llevará el caso a comité disciplinario y la nota del curso queda como Pendiente Disciplinario hasta que el comité tome alguna decisión.

Todas las entregas de talleres y ejercicios se harán a través de la plataforma habilitada (BrightSpace para la Magistral y SICUA para la Complementaria) **No se aceptará ninguna tarea por fuera de esas plataformas**, a menos que ocurra una falla en los servidores que afecte a **todos** los estudiantes del curso. No se toman en cuenta entregas marcadas como **tarde** por la plataforma.

Las entregas de ejercicios que se publican durante la sesión sincrónica de magistral y complementaria tendrán una ventana de tiempo de 4 horas para entregarse.

Las entregas de ejercicios que se publican 36 horas antes de la sesión sincrónica de la magistral tendrán una ventana de tiempo de 36 horas para entregarse.

## VII Bibliografía

Bibliografía principal:

- *Introduction to Probability and Statistics for Engineers and Scientists*. Sheldon M. Ross. Third Edition. Elsevier Academic Press. 2004.
- *A survey of Computational Physics - Enlarged Python Book*. R. H. Landau, M. J. Páez, C. C. Bordeianu. WILEY. 2012. <https://psrc.aapt.org/items/detail.cfm?ID=11578>

Bibliografía secundaria:

- *A student's guide to numerical methods*, I. H. Hutchinson. Cambridge University Press. 2015.
- *Introduction to statistics with python*. Thomas Haslwanter. Springer. 2016.
- *Pattern recognition and Machine Learning*. Christopher Bishop. Springer. 2011.
- *Data Analysis: A Bayesian Tutorial*. D. S. Sivia, J. Skilling. Second Edition, Oxford Science Publications. 2006.
- *Statistical Mechanics: Algorithms and Computations*. W. Krauth, Oxford Univ. Press.
- *An Introduction to Statistical Learning*. G. James, D. Witten, T. Hastie, R. Tibshirani, Springer. <http://www-bcf.usc.edu/~gareth/ISL/>
- *Elements of Scientific Computing* Tveito A., Langtangen H.P., Nielsen B.F., Cai X. Springer. 2010.
- *Introduction to Computation and Programming Using Python*, Guttag, J. V. The MIT Press. 2013.