



Curso de propedéutico -- Programación
Maestría en Ciencias en Ciencia de Datos
Maestría en Ciencia de Datos e Información

Formato para el desarrollo de las unidades de estudio

[Llenar un formato por cada unidad de estudio que se proponga]

Nombre de la Unidad de estudio	Programación y uso de estructuras de datos
Tiempo estimado de estudio	40 hrs

Conocimientos requeridos para abordar la unidad

- Conocimientos básicos de matemáticas discretas
- Uso de un lenguaje de programación imperativo u orientado a objetos
- Conocimiento básico de la arquitectura de una computadora

Objetivo(s) de aprendizaje

Al finalizar la unidad de estudio será capaz de:

- Usar estructuras de datos básicas como arreglos, conjuntos, cadenas, secuencias, listas.
- Usar arreglos de números en punto flotante como estructuras vectoriales y combinarlos mediante operaciones básicas de álgebra lineal
- Implementar y aplicar funciones de distancia entre vectores
- Aplicar algoritmos elementales de ordenamiento y búsqueda
- Usar notación asintótica para describir el costo de los algoritmos

Contenido (Temario):

Señalar los contenidos concretos que son los objetos de aprendizaje de la unidad

1. Conceptos básicos de programación en Python
 - a. Instalación de Anaconda y uso de Jupyter
 - b. Variables, constantes, literales
 - c. Condicionales
 - d. Ciclos
 - e. Funciones
 - f. Entrada, salida y graficación simple de datos
 - g. Librerías para análisis de datos
 - i. Pandas
 - ii. Scikit-learn
2. Algoritmos
 - a. Costo computacional de los algoritmos
 - b. Notación asintótica
3. Uso de estructuras de datos elementales
 - a. Arreglos
 - i. Mínimo, máximo, elementos únicos, ordenamientos simples, búsqueda en arreglos ordenados
 - ii. Media, mediana, media armónica, media geométrica, varianza y desviación estándar

- b. Representación de vectores
 - i. Suma, resta, producto punto, norma de vectores
 - ii. Funciones de distancia:
 - Minkowski: L_p , L_1 , L_2 , L -infinito
 - Coseno y el ángulo entre vectores
 - Basadas en conjuntos: distancia de Jaccard y Dice
- c. Tablas hash
 - i. Acceso eficiente a pares asociados
 - ii. Contadores
- d. Sets
 - i. Representación en listas ordenadas
 - ii. Representación basada en tablas hash sin par asociado
 - iii. Multisets
 - iv. Consultas por membresía y conteo de elementos diferentes en una colección
- 4. Modelado de datos en espacio vectorial
 - a. Preprocesamiento de texto
 - i. Tokenizado de datos
 - ii. Pesado de términos
 - Binario
 - Frecuencia
- 5. Distribución de términos en una colección de libros
 - a. Ley Heaps para la colección
 - b. Ley Zipf para la colección

Bibliografía general y otros recursos recomendados:

- Thomas H., Cormen, Leiserson, C. E., Rivest, R. L., & Stein; C. Introduction to algorithms. Cambridge, MIT press. 2001.

- Sedgewick, R., & Flajolet, P; An introduction to the analysis of algorithms. Addison-Wesley. 2013.
- El curso completo de introducción en algoritmos MIT 6.046J / 18.410J Introduction to Algorithms (SMA 5503), Fall 2005
Primera lección: <https://www.youtube.com/watch?v=JPYuH4qXLZ0>

Ejercicios, problemas y actividades para el estudio de los temas

Indicar la bibliografía y/o los recursos a utilizar.

Indicar los aspectos a tomar en cuenta para la solución de las actividades propuestas.

Bibliografía

- Thomas H.. Cormen, Leiserson, C. E., Rivest, R. L., & Stein; C. Introduction to algorithms. Cambridge, MIT press. 2001. Parte I, II, y III, todos los capítulos.
- Sedgewick, R., & Flajolet, P; An introduction to the analysis of algorithms. Addison-Wesley. 2013. Cap. 1, 2, 3 y 5.

Orientaciones

- Las implementaciones deben ser concisas y correctas.
- Los análisis requeridos deben ser correctos y descritos explícitamente
- Las implementaciones de estructuras básicas deben ser creadas sin el uso de las versiones incluidas en bibliotecas del lenguaje de programación utilizado
-