

Contenido

Aunque en la [descripción](#) de la materia las temáticas están en un orden establecido, en la materia se tendrá el siguiente orden de temáticas:

1. Análisis de algoritmos.
2. Verificación de algoritmos.
3. Diseño de algoritmos.
4. Técnicas avanzadas de programación.
5. Intratabilidad

Bibliografía recomendada

Para análisis de algoritmos y técnicas de diseño de algoritmos:

- Técnicas de diseño de algoritmos ([ver](#)): Está disponible para descarga libre
- Introducción a los algoritmos ([ver](#))

Distribución de notas y evaluación

[Ver](#) fechas generales del profesor para información exacta sobre los días de parciales, examen, talleres y entregas.

- 30%
 - 10% 1 Taller en clase en la semana previa al parcial.
 - 20% Parcial
 - ** 5% Entrega de proyecto de Aula
- 30%
 - 10% 1 Taller en clase en la semana previa al parcial
 - 20% Parcial
 - ** 5% Entrega de proyecto de Aula
- 40%
 - 20% Proyecto de clase
 - Se califica la última entrega del 10% se suma a las dos entregas de 5% cada una hechas en los cortes anteriores
 - 20% Examen final

**** Nota importante:** Aunque la nota de proyecto final se subirá al sistema en el último 20%, este se calificará en tres entregas, una después de cada parcial, la primera y la segunda tienen una calificación correspondiente al 5% y la última una calificación correspondiente al 10%

Proyecto de clase de investigación formativa

Se desea que el estudiante realice un proyecto sobre análisis experimental de algoritmos. El proyecto consiste en realizar el análisis experimental de algoritmos de ordenamiento aplicados a datos de personas.

Entregas ([ver](#) fechas)

Modo de entrega para las 3 entregas: Por correo, **dos archivos** adjuntos, un **pdf** con el documento escrito y un archivo comprimido (formato zip) con los **códigos fuentes** del proyecto a profesor.niquefa@gmail.com al menos 24 horas antes de la clase de sustentación en donde deben defender el avance ante el profesor ([ver](#) fechas exactas). Asunto del mensaje:

- AVA **X** Entrega **N** (Nombre Completo Integrante 1, Nombre Completo Integrante 2)
- Donde X es B o C dependiendo del grupo, N es 1, 2 o 3 dependiendo de la entrega y los nombres de los integrantes van completos y entre paréntesis.

- Primera (5%)
 - Tareas:
 - Hacer un conjunto de clases y métodos en Java (o el lenguaje de su preferencia) que sirva para generar conjuntos de datos de personas, estas personas deben tener las siguientes características como mínimo:
 - Nombres
 - Primer Apellido
 - Segundo Apellido
 - Tipo de documento (CC, TI, Pasaporte, CC Extranjería)
 - Número de documento
 - Tipo de Sangre
 - Género (Masculino o Femenino)
 - Fecha de nacimiento (Usar GregorianCalendar o similar)
 - Correo electrónico
 - Número de teléfono
 - Número de teléfono celular
 - Para el punto anterior, se recomienda hacer métodos sencillos como los siguientes, y a partir de ellos generar métodos más complejos como el de obtener una persona generada aleatoriamente.
 - `int getRandomNumber(int max)`
 - `String getRandomString(ArrayList<String> list)`
 - Diseñar un esquema de experimentación para los algoritmos de ordenamiento propuestos.

- Traer una lista de algoritmos de ordenamiento basados en comparaciones que se puedan aplicar al problema de “Ordenar” personas. Se pide sólo la lista, ningún algoritmo implementado. La lista debe tener al menos 7 algoritmos.
 - El programa implementado debe producir archivos de texto plano con datos generados que tengan sentido en el universo que se está modelando, es decir, las personas. [Ver](#) ejemplo de cómo se podría ver un archivo generado con el programa hecho. (deben entregar el fuente documentado usando javadoc o el análogo en el lenguaje que deseen).
- Segunda (5%)
 - Traer implementados y probados al menos 6 algoritmos de los mostrados en la primera entrega. Estos algoritmos deben funcionar con **Personas**, es decir, deben ordenar conjuntos de datos como los que produce el programa hecho en la primera entrega.
 - Entregar también un documento escrito (un pdf) en donde se muestran las complejidades teóricas de cada algoritmo:
 - Caso mejor
 - Caso medio
 - Caso peor
- Tercera (10%)
 - Aplicar el esquema de experimentación para analizar los algoritmos implementados y redactar un documento con el respectivo análisis. Este documento debe tener al menos:
 - Comparación del análisis teórico con el experimental con sus respectivas conclusiones.
 - Gráficas que muestran los análisis teóricos y experimentales para cada algoritmo
 - Justificación de las conclusiones.
 - Dificultades encontradas durante el proyecto
 - Lecciones aprendidas en la realización del proyecto
 - El esquema de experimentación debe estar plasmado en un programa que lo corre todo, es decir, se deben poder correr todos los experimentos con la ejecución de un programa, y este programa es el que se debe enviar en esta última entrega.

Material adicional de estudio:

- Libro de teoría y práctica de demostraciones matemáticas (en inglés) ([Ver](#))
- Libro guía: Building Blocks of Theoretical Computer Science (en inglés) ([Ver](#))
- Polinomios online
 - Multiplicación : Página 1 ([ver](#))
 - Multiplicación : Página 2 ([ver](#))

- Búsqueda de raíces : Página 1 ([ver](#))
 - No es exacto, pero ayuda, ejemplo: Colocar $x^4-4x^3-x^2+16x+12 = 0$
- [algorithm.net](#) ([ver](#))
- Algorithms and data structures sit ([ver](#))