

TAREA 1
ALGORITMOS Y ESTRUCTURAS DE DATOS 1^{DO} SEMESTRE 2020

Objetivo de la tarea: Diseñar e implementar un TDA.

Profesor: Karol Suchan

Ayudante: Yerko Ortiz

- Se aceptarán entregas de la tarea hasta el martes 7 de Abril a las 23:59 horas.
- Para la entrega se ha de subir un código con extensión .java vía classroom que de solución el enunciado.

El pergamino sagrado

Un maestro ninja pasó sus conocimientos del pergamino sagrado a sus discípulos, estos a su vez cuando se convirtieron en maestros, pasaron sus conocimientos a sus discípulos, continuando este ciclo por la posteridad hasta estos días; los ninjas que son discípulos del mismo maestro se les considera hermanos. Es sabido que todos los maestros enseñan de forma distinta y que a su vez todos los alumnos aprenden de distinta forma, es por esto que mientras las generaciones aumentan, el conocimiento del pergamino va cambiando hasta llegar al punto en que la enseñanza de un maestro, es un conocimiento absolutamente distinto al del pergamino original.

A usted como ninja de la programación, se le delega la tarea de crear un árbol genealógico que abarque a todos los ninjas que han obtenido parte del conocimiento del pergamino, donde para realizar esta tarea debe diseñar e implementar el TDA(tipo de dato abstracto) ArbolNinja tomando en cuenta las siguientes indicaciones:

Atributos de la clase Ninja(nodo):

- Ninja primerDiscipulo;
- Ninja nextHermano;
- int id;
- String[] nombre;

El ArbolNinja(TDA) debe tener los siguientes métodos:

- **void crearArbolNinja(String[] nombreMaestro):** inicializa el TDA con un ninja en su interior (el maestro inicial) con id = 0 y el nombre que usted estime conveniente.
- **void insertarDiscipulo(String[] nombreDiscipulo, int idMaestro):** busca el id del maestro en el TDA y luego inserta al ninja al final de la lista de discípulos (asignándole un id).
- **int contarGeneracion(int idNinja):** busca al ninja con idNinja en el TDA para luego retornar el número de generaciones de enseñanza que han pasado hasta ese ninja.
- **void imprimirArbol():** que imprime los nombres de todos los ninjas por generación, imprime el nombre del primer maestro, luego sus discípulos, luego los discípulos de esos discípulos y así sucesivamente.
- **int ancestroMinimoComun(int idNinja1, int idNinja2):** dado el id de dos ninjas, este método los busca y calcula el ancestro(maestro) mínimo que tienen en común.

Hint: para la asignación de id use un contador, así evita que existan dos ninjas con el mismo id en el TDA.