



Universidad Simón Bolívar  
Depto. de Computación y T.I.  
Taller de Algoritmos y Estructuras III (CI-2693)  
Septiembre – Diciembre 2.010

## **Proyecto 1**

### **Tad Grafo**

El objetivo de este proyecto es realizar la implementación de la estructura Grafo Dirigido usando una representación: nodos y arcos se almacenan en una Tabla de Hash. Se provee un conjunto de archivos con las firmas de las clases cuya implementación debe ser completada por Uds.:

- DiGraph
- DiGraphHash
- Arc
- List
- MyHashTable
- Vertice
- Algoritmo
- Alcance

La clase abstracta DiGrafo define, entre otros, el método Alcance. Este método debe ser implementado por las especializaciones de Digrafo: DiGraphHash. Cada una de estas implementaciones debe hacer uso de la representación interna para lograr una mayor eficiencia.

La aplicación que Ud. deberá desarrollar consiste en un programa que recibe un nombre de archivo (que contiene un grafo, usando el formato descrito en el anexo 1) y se debe construir un grafo a partir de la información contenida en el archivo y aplicará un algoritmo para calcular la clausura transitiva de un grafo y almacenarla en un archivo desalida en el mismo formato establecido en el Anexo 1.

Sistanxis:

```
# java Main <inFileName> <outFileName>
```

**inFileName** y **outFileName** son los nombres de los archivos donde estará el archivo de entrada y el de salida.

### **Entrega**

Para el martes 19 de Octubre (semana 5) a la 9:30 am usted deberá entregar a su profesor (en el salón de clases), un sobre sellado y debidamente identificado con su nombre, carnet y profesor de laboratorio. Éste debe contener:

- Códigos fuente **debidamente documentados** de los tipos implementados.
- El 18 de Octubre hasta las 11:59pm, Ud deberá colocar el archivo con su proyecto en Aula Virtual, para lo cual deberá crear el directorio PROY1 dentro de la carpeta documentos de su grupo. Note que debe estar suscrito a algún grupo en aula Virtual para poder optar a esta opción. En este directorio colocará los archivos con los fuentes de su aplicación. El archivo debe tener como nombre **P1G<#grupo>.zip** que contenga un directorio **P1G<#grupo>** que a su vez contenga todos los archivos .java de su proyecto.

## Observaciones

En la evaluación del proyecto se tomará en cuenta el estilo de programación, desempeño, uso de herencia, correcto uso y manejo de excepciones, etc. Proyectos que no compilen serán calificados con cero.

## Referencias

1. [Ortega,Meza] Maruja Ortega y Oscar Meza. Grafos y Algoritmos. Colección Tesis Ciencias Aplicadas. Equinoccio. Caracas, 1993.
2. [Ambler] Writing Robust Java Code. Scott W. Ambler Software Process Mentor 2000.
3. **Guía de estilo publicada en Aula Virtual [javaCodingStandards.pdf](#)**

## Anexo 1

Cada implementación de Grafo contará con un constructor que permitirá leer un grafo de un archivo. Este archivo debe cumplir con el siguiente formato:

n	numNodos
n0 c0	Idnodo costo
n1 c1	Idnodo costo
n2 c2	Idnodo costo
n3 c3	Idnodo costo
m	numArcos
n0 n1 c	nodoFuente nodoDestino costo
n1 n2 c	nodoFuente nodoDestino costo
n2 n3 c	nodoFuente nodoDestino costo
n3 n0 c	nodoFuente nodoDestino costo

Donde  $n$  es el número de nodos que contiene el grafo y  $m$  el número de arcos.  $n_i$  es un String que representa el Id del vértice  $i$  del grafo. El siguiente es un ejemplo del contenido de un archivo válido con este formato.

En el archivo	El grafo que representa
4 0 1 1 2 2 2 3 1 4 0 1 11 1 2 12 2 3 11 3 0 29	