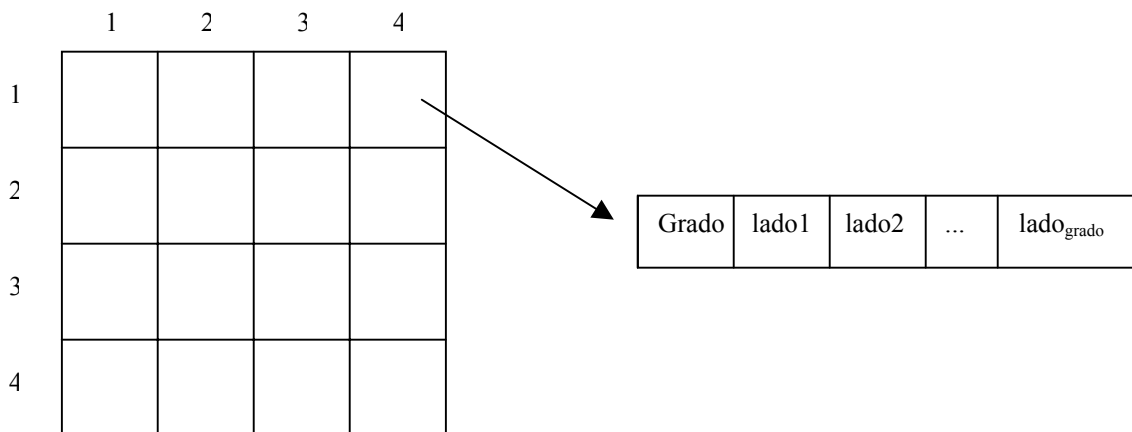

Proyecto #1

El objetivo de este proyecto es lograr que se familiarice con las operaciones básicas de los TADs *GRAFO NO DIRIGIDO* y *GRAFO DIRIGIDO*. Para ello se desea que implemente los TADs usando el lenguaje JAVA y luego desarrolle una pequeña aplicación que permita probar los TADs.

Para la implementación se creará una clase abstracta llamada *GRAFO* que contendrá las operaciones y estructura de datos asociados a un grafo, sea este dirigido o no. Los grafos podrán tener lados múltiples y bucles. La clase *GRAFO* tendrá dos clases concretas derivadas que llamaremos *NDGRAFO* (por grafo no dirigido) y *DIGRAFO* (por grafo dirigido). Se desea que implemente el TAD *GRAFO* utilizando Matriz de Adyacencias. Estos TADs serán necesarios para el desarrollo de otros proyectos que serán asignados durante el curso.

La estructura de datos para almacenar un grafo con arcos múltiples sería como sigue:



La Matriz debe implementarse utilizando la clase *Vector* de JAVA, lo cual permite utilizar todas las facilidades que brinda esta clase de JAVA. De esta manera, la matriz sería un vector de vectores. Cada elemento M_{ij} contiene el grado o grado de salida del nodo según que el grafo sea no dirigido o dirigido, así como una lista formada por los lados (aristas o arcos) que inciden en el nodo (arcos que salen en el caso dirigido).

Como tipos base se asumen los tipos *NODO* y *LADO*

El tipo *NODO* será una clase que poseerá al menos dos campos: uno tipo *string* que identifica unívocamente (en un grafo no debe haber dos nodos con el mismo identificador) a cada nodo y otro campo tipo entero que corresponde a la fila asociada al nodo en la matriz.

El tipo *LADO* es una clase que puede asumir como valores tuplas de la forma $\langle id, p \rangle$ donde *id* debe ser de tipo *string* e identifica unívocamente al lado (en un grafo no debe haber dos lados con un mismo identificador), y *p* es un campo tipo real que corresponde a un peso asociado al lado.

El TAD *GRAFO* debe incorporar al menos los siguientes operadores:

CrearGrafo: \rightarrow *GRAFO*

CrearGrafo:	<i>ARCHIVO</i>	→	<i>GRAFO</i>
AgregarNodo:	<i>GRAFO</i> x <i>datosNODO</i>	→	<i>GRAFO</i>
Nodos:	<i>GRAFO</i>	→	<i>Conjunto de identificadores de nodos</i>
Lados:	<i>GRAFO</i>	→	<i>Conjunto de identificadores de lados</i>
Grado:	<i>GRAFO</i> x <i>idNODO</i>	→	<i>ENTERO</i>
Ady:	<i>GRAFO</i> x <i>idNODO</i>	→	<i>Conjunto de identificadores de nodos</i>
Incidentes:	<i>GRAFO</i> x <i>idNODO</i>	→	<i>Conjunto de identificadores de lados</i>

- *datosNODO* significa “todos los datos de un nodo”.
- *idNODO* significa “el identificador de un nodo”.

Las clase concreta *NDGRAFO* tendrá los operadores adicionales:

AgregarArista:	<i>GRAFO</i> x <i>datosARISTA</i>	→	<i>GRAFO</i>
EliminarArista:	<i>GRAFO</i> x <i>idLADO</i>	→	<i>GRAFO</i>

La clase concreta *DIGRAFO* tendrá los operadores adicionales:

AgregarArco:	<i>GRAFO</i> x <i>datosARCO</i>	→	<i>GRAFO</i>
EliminarArco:	<i>GRAFO</i> x <i>idLADO</i>	→	<i>GRAFO</i>
GradoInt:	<i>GRAFO</i> x <i>idNODO</i>	→	<i>ENTERO</i>
GradoExt:	<i>GRAFO</i> x <i>idNODO</i>	→	<i>ENTERO</i>
Sucesores:	<i>GRAFO</i> x <i>idNODO</i>	→	<i>Conjunto de identificadores de nodos</i>
Predecesores:	<i>GRAFO</i> x <i>idNODO</i>	→	<i>Conjunto de identificadores de nodos</i>

- *datosARISTA* significa “todos los datos de la arista”.
- *datosARCO* significa “todos los datos del arco”.

Especificación semántica:

CrearGrafo (): Crea un grafo nulo.

CrearGrafo (a): Crea un grafo a partir del archivo a.

AgregarNodo (G, n): Agrega el nodo n al grafo G previamente creado. Notese que al agregar un nodo, se debe agregar una nueva fila y una nueva columna a la matriz de adyacencias que representa el grafo.

AgregarArco (G, l): Agrega al grafo ya creado G el arco l.

AgregarArista (G, l): Agrega al grafo ya creado G la arista l.

EliminarArco (G, l): Elimina el arco l del grafo ya creado G.

EliminarArista (G, l): Elimina la arista l del grafo ya creado G.

Nodos (G): Obtiene el conjunto de nodos del grafo G.

Lados (G): Obtiene el conjunto de lados del grafo G.

Grado (G, n): Calcula y devuelve el grado del nodo n del grafo G.

GradoInt (G, n): Calcula y devuelve el grado interior del nodo n del grafo G.

GradoExt (G, n): Calcula y devuelve el grado exterior del nodo n del grafo G.

Ady (G, n): Obtiene el conjunto de nodos adyacentes al nodo n en el grafo G.

Predecesores (G, n): Obtiene el conjunto de nodos predecesores del nodo n en el grafo G.

Sucesores (G, n): Obtiene el conjunto de nodos sucesores del nodo n en el grafo G.

Incidentes (G, n): Obtiene el conjunto de lados incidentes al nodo n en el grafo G.

Para verificar el funcionamiento de la implementación del TAD *GRAFO*, debe desarrollar una pequeña aplicación que permita, por medio de un menú, tener acceso a todas las funciones del TAD, además de proveer un medio que permita visualizar la matriz de un grafo en cualquier momento. Adicionalmente, debe escribir una rutina que permita leer un grafo de un archivo. El archivo tiene el siguiente formato:

```
o      /* Orientacion 0 = No Orientado 1 = Orientado */
n      /* Número de Nodos */
m      /* Número de Lados */
x1     /* Identificación del nodo */
:
.
xn
i1 t1 id1 p1 /*Descripción de los lados <inicial, terminal, id, peso>*/
i2 t2 id2 p2
:
.
im tm idm pm
```

Para la entrega final, **lunes de la semana 5 a las 9:30 am** usted deberá entregar en un sobre debidamente identificado:

- Listados (documentados) De los fuentes del TAD grafo y la aplicación de prueba. Recuerde incluir las rutinas de visualización de grafos y lectura de archivos.
- Un diskette debidamente identificado, **LIBRE DE VIRUS Y DEFECTOS FÍSICOS**, cuyos únicos directorios sean \bin (que contenga los archivos .CLASS), \sources (que contenga los archivos .JAVA de su proyecto) y \doc que contenga la documentación en javadoc de las clases. Recuerde incluir en el informe el nombre de la clase que debe ser utilizada para ejecutar el programa de prueba.

Adicionalmente, se harán 2 entregas parciales:

1ra. entrega (Semana 3) : Listado DOCUMENTADO que contenga la declaración de TODAS las estructuras de datos utilizadas.

2da. Entrega (Semana 4) : Deben estar operativas y serán evaluadas en clase, la pequeña aplicación que permita, por medio de un menú, la visualización de la matriz del grafo, así como el acceso a las funciones del TAD que se indican a continuación:

CrearGrafo:		→	<i>GRAFO</i>
CrearGrafo:	<i>ARCHIVO</i>	→	<i>GRAFO</i>
AgregarNodo:	<i>GRAFO x datosNODO</i>	→	<i>GRAFO</i>
Grado:	<i>GRAFO x idNODO</i>	→	<i>ENTERO</i>
AgregarArista:	<i>GRAFO x datosARISTA</i>	→	<i>GRAFO</i> (para grafos no dirigidos)
AgregarArco:	<i>GRAFO x datosARCO</i>	→	<i>GRAFO</i> (para grafos dirigidos)