

Práctica 8

Objetivo: Practicar con el ADT *Grafo*

Descripción: Se desea usar un grafo dirigido para representar la conectividad entre computadores de una red.

- Cada vértice contiene un String que identifica el nombre de un computador.
- Cada arista de un computador a otro indica que hay una conexión directa del primero al segundo.
- Las aristas no tienen pesos.

Práctica 8 (cont.)

Se desea hacer una clase Java con el diagrama de clases de la figura, y con estos métodos:

- **constructor:** crea el grafo vacío
- **insertaConexiones:** añade al grafo todas las conexiones posibles entre los computadores de un conjunto si no existen previamente
 - Lo que hace es añadir al grafo para cada variación (sin repetición) de dos computadores de ese conjunto, una arista del primer elemento al segundo, siempre que no exista ya en el grafo.

RedComputadores
Grafo<String> g
RedComputadores() void insertaConexiones(Set<String> nuevas) Set<String> alcanzables(String c) boolean esAlcanzable(String c1, String c2) throws NoExiste void muestra()

Práctica 8 (cont.)

- **alcanzables**: retorna el conjunto de los vértices del grafo que son alcanzables desde **c**; si **c** no está en el grafo retorna **null**
 - un vértice **b** es alcanzable desde otro, **a**, si existe un camino de **a** a **b**
- **esAlcanzable**: retorna un booleano que indica si el vértice **c2** es alcanzable desde el vértice **c1**; lanza **NoExiste** si **c1** o **c2** no son vértices del grafo
- **muestra**: muestra todos los vértices del grafo y sus aristas

Hacer también un programa de prueba

Práctica 8 (cont.)

Entregar:

- diseño de los métodos (pseudocódigo)
- eficiencia de los métodos
- código de la clase
- código del programa de prueba