

# Estructuras de Datos

## Taller Grafos: Recorridos y Componentes Conexas

### Introducción

En esta práctica, usted implementará algunos algoritmos fundamentales del TDA Grafo. Para esto, utilizará el código adjunto, que provee una implementación en Java de los TDAs vértice (Vertex.java), arco (Edge.java), y grafo. Este último TDA está implementado tanto mediante una matriz de adyacencia (GraphMA.java) como mediante listas de adyacencias (GraphLA.java). Antes de empezar con la implementación de los métodos solicitados a continuación, asegúrese de familiarizarse con el código provisto. En particular, revise el programa principal de la clase Main, que ilustra la forma de crear un grafo con el TDA adjunto.

### Ejercicios

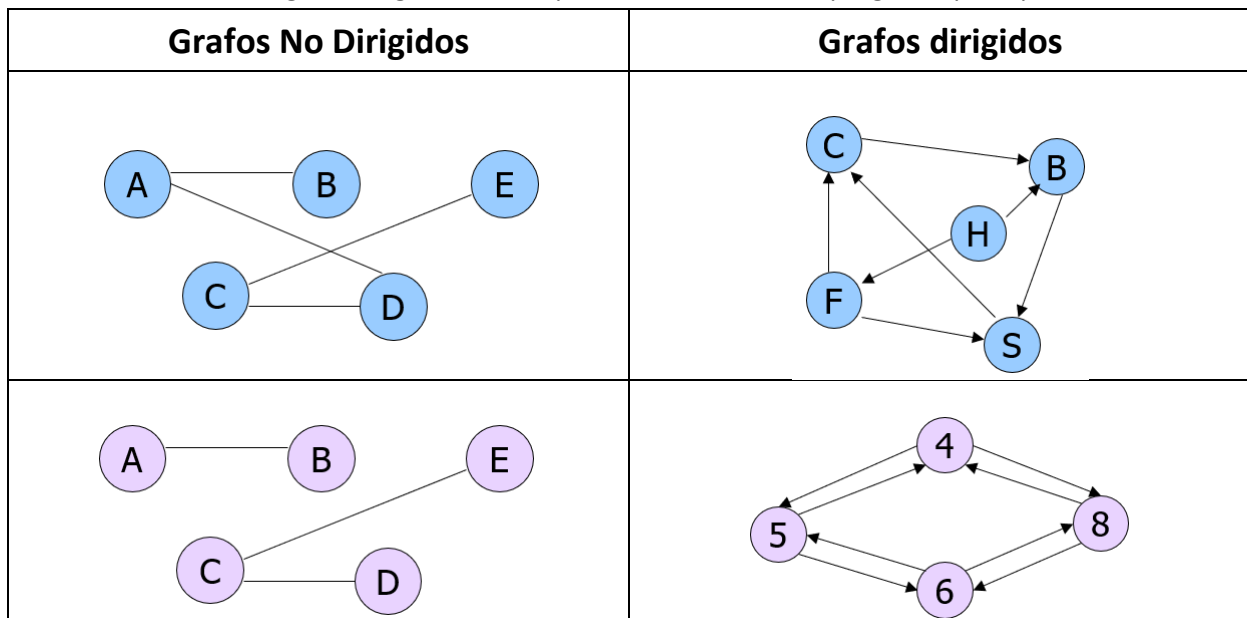
Los siguientes métodos deben ser incluidos **dentro de la clase GraphAL**, que utiliza listas de adyacencias para abstraer el concepto de grafos:

1. Implemente un método que recorra el grafo por anchura.
2. Implemente un método que recorra el grafo por profundidad.
3. Implemente un método que retorne las componentes conexas de un grafo no dirigido.
4. Implemente un método que retorne las componentes fuertemente conexas de un grafo dirigido.

Usted deberá, además, implementar todos los métodos (públicos o privados) que considere pertinente en la resolución de este taller.

La estructura GraphAM es incluida en el código de este practica únicamente con fines didácticos. Queda a criterio de cada estudiante explorar este código en mayor detalle.

5. Instancie los siguientes grafos e invoque sus métodos en un programa principal:



Note que ninguno de los grafos mostrados arriba incluye metadata en sus arcos. En ese caso, usted puede utilizar la clase Object para el parámetro de tipo correspondiente a los arcos y setear la metadata de los mismos a null.

Usted es responsable de decidir el tipo y número de argumentos que deben recibir y/o retornar los métodos que implementará.