**PRÁCTICA 31 | GRAFOS**

# Introducción

La clase **GrafoMatrizA** ha crecido bastante para ocupar los métodos u operaciones que se pueden hacer sobre ella. A pesar de esto, aún faltan algunos conceptos de ser incluidos y en esta práctica se van a complementar algunas (o todas) esas faltantes. Al leer la descripción de las actividades me pude percatar que podré reutilizar métodos que había programado antes, lo que me indica que probablemente haya hecho un buen diseño en algunas partes de la clase. En general, y como suelo decirlo, espero poder realizar un buen análisis y diseño para generar métodos que aprovechen al máximo la Programación Orientada a Objetos y al mismo tiempo sean lo más óptimos posible.

# Desarrollo

Comencé la práctica leyendo los requerimientos e identificando los métodos que iba a necesitar en cada actividad para agregarlos a mi diagrama de clases que utilizo como guía y base de lo que será programado. Una vez terminado esto dibujé en mi cuaderno dos grafos: uno dirigido y otro no dirigido para realizar el análisis de los primeros métodos que iba a implementar.

Las primeras funcionalidades que codifiqué fueron las que identifican si un grafo es conexo o fuertemente conexo en el caso de que sea no dirigido o dirigido respectivamente, a la par que hacía el método me percaté que debía comprobar el tipo de grafo del que se trataba (dirigido o no dirigido) para que éstos métodos y algunos posteriores pudiesen entregar la funcionalidad correcta; así que tuve que detenerme y realizar dicha función.

Comprobar que un grafo no dirigido era conexo, así como obtener sus componentes de este tipo no fue muy complicado. Las cosas cambiaron cuando me puse a realizar el mismo procedimiento pero para un grafo dirigido; aquí las cosas cambiaban un poco y me atoré un tiempo tratando de asegurarme que el método funcionara correctamente. Muchas correcciones tuve que hacer a esta parte del código después de haberlo ya hecho una vez, en unos casos para modularizar más o para optimizar.

Otro problema que se presentó en el desarrollo de las actividades es que en la última clase de la materia se le incluyó al grafo la opción de agregar una prioridad en forma ascendente o descendente. El constructor que recibe este parámetro inicializa la matriz de aristas con un valor diferente al cero que se consideraba en muchos métodos como la ausencia de una arista, lo que conllevo a que tuviese que hacer otro método para tomar todos los casos posibles en cuenta, además de sustituir en las condiciones ya programadas con este nuevo método para que no funcionaran incorrectamente.

Finalmente llegué a la última actividad, donde en lugar de los pasos para hacer el código se proporcionaba directamente el pseudocódigo. De esta forma fue mucho más fácil y rápido hacer los métodos de distancias mínimas (estos procedimientos se hacen sólo cuando el criterio es descendente), y el algoritmo que almacena a los nodos intermedios que forman las trayectorias más cortas entre cada vértice del grafo.

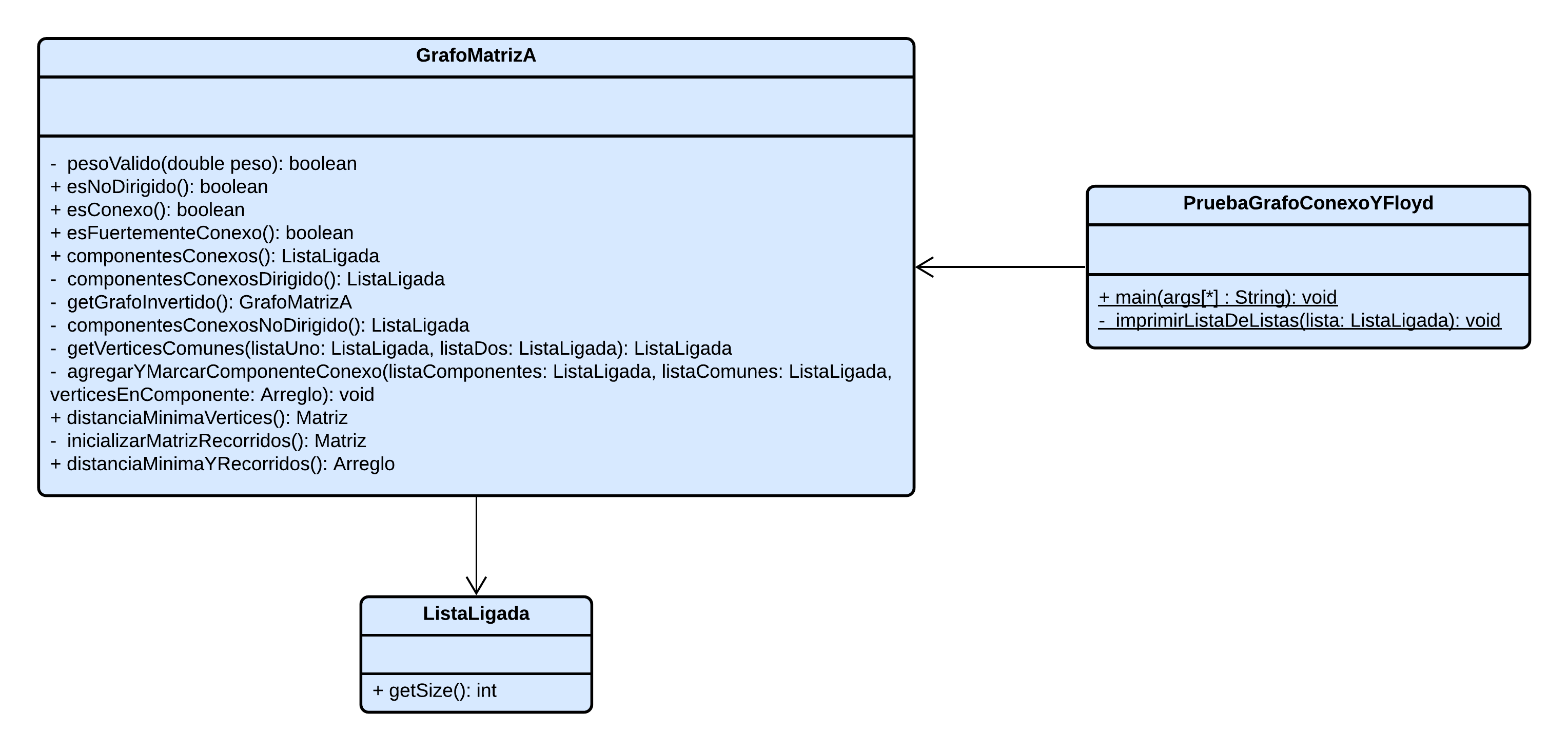
A continuación, se indican los archivos creados o modificados y su ubicación en el proyecto.

* ***GrafoMatrizA.java (paquete ednolineal).***
* ***ListaLigada.java (paquete edlineal).***
* ***PruebaGrafoConexoYFloyd.java (paquete interfaces).***

La clase anterior contiene las pruebas de esta práctica.

# Diagrama de clases

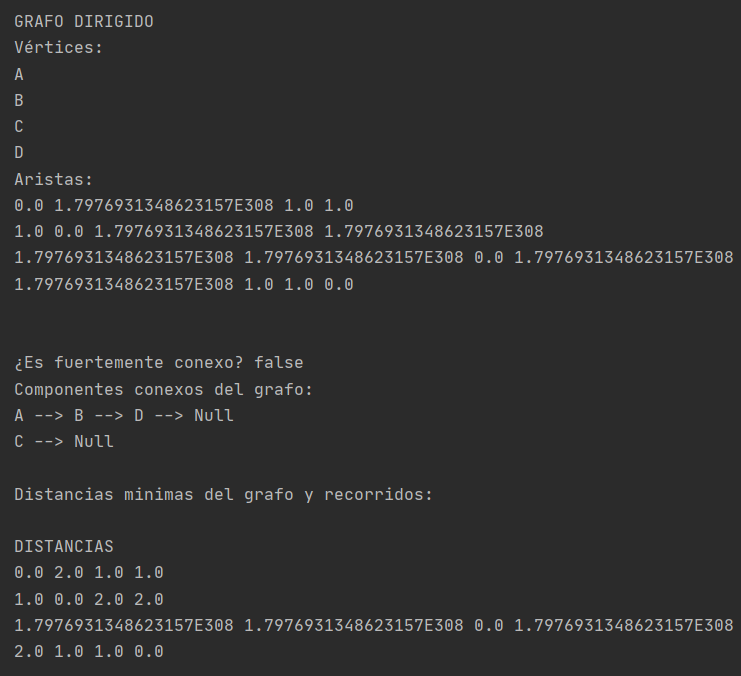
En el diagrama de clases se tiene una vista más cómoda de lo que se hizo en cada una de las clases mencionadas anteriormente. Cabe destacar que para las que ya existían en el proyecto solamente se ilustran los métodos o atributos que fueron agregados. En cambio, para las nuevas se muestra en su totalidad métodos y atributos que fueron incluidos.

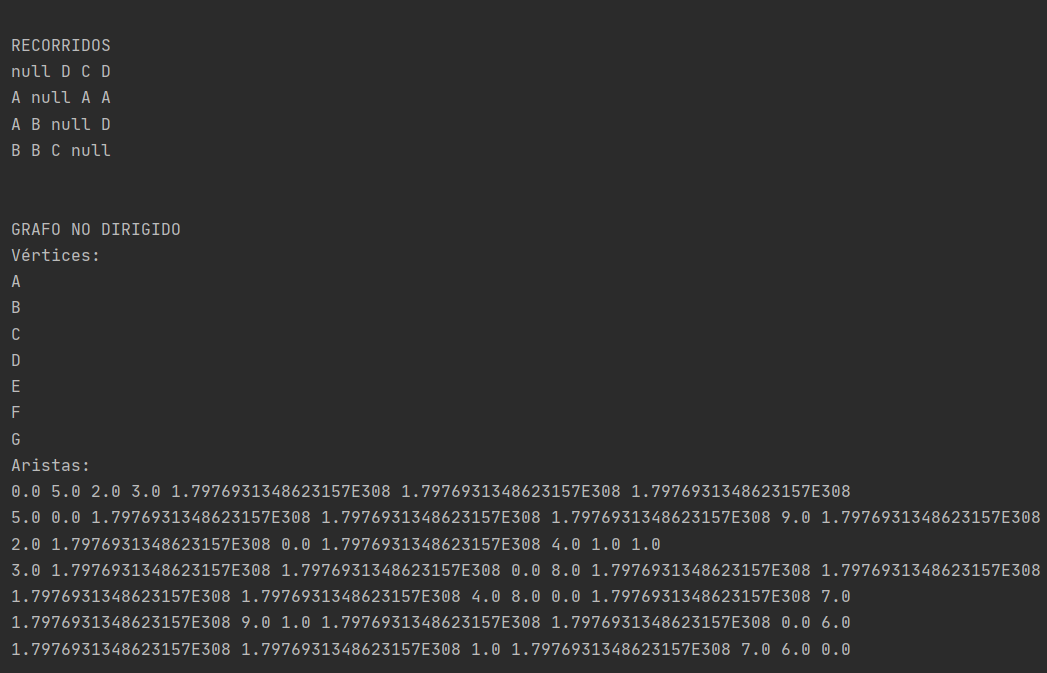


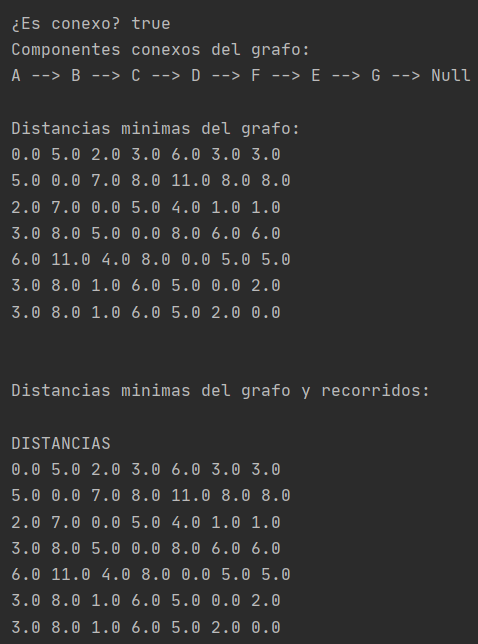
**Si desea revisar la documentación de lo que fue agregado o modificado en el proyecto vaya a las siguientes rutas:**

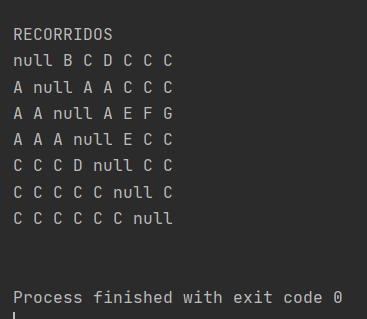
* **/docs/ednolineal/GrafoMatrizA.html**
* **/docs/edlineal/ListaLigada.html**
* **/docs/interfaces/PruebaGrafoConexoYFloyd.html**

# Capturas del programa funcionando









# Conclusiones

Con todo lo que se ha hecho en la clase **GrafoMatrizA** siento que ya está preparado para que se pueda hacer alguna práctica donde ahora los grafos se tomen como aplicación en la vida real.