

Título:

Documentación Proyecto II del curso Estructuras De Datos.

Autores:

Bryan Jesús Sánchez Brenes.

bryan.jsb.1801@gmail.com

304760577.

Priscilla Castro Arroyo

mariacastro9712@gmail.com

116760798

Institución:

Universidad de Nacional. Escuela de Informática y Computación

Lugar:

Lagunilla de Heredia.

Fecha:

Sábado, 16 de noviembre del 2019.

Grupo :

02.

Contenido

INTRODUCCIÓN.

	1
Enunciado	2
Antecedentes	2
Problema y justificación	2
Objetivos	3
Alcances	3
2. MARCO DE REFERENCIA (MARCO TEÓRICO).	4
Descripción de la solución	8
Análisis de las estructuras de datos usadas en el programa.	
Métodos importantes del programa:	9
Diagramas	13
MARCO METODOLÓGICO.	15
4.CONCLUSIONES.	155. REFERENCIAS Y BIBLIOGRAFÍA.
	15

INTRODUCCIÓN.

Enunciado

Se requiere crear una solución de software de repartos en que se usen mapas con sus respectivas rutas y señalizaciones viales; se necesita de un modelo que muestre las rutas más cortas para el tránsito de los autos de reparto y así se optimicen las operaciones de la empresa adquisidora del producto.

Antecedentes

Los grafos en su teoría tienen un problema llamado: “el problema del camino más corto”, en el cuál el objetivo por cumplir para resolverlo es encontrar un camino entre dos vértices para obtener la suma de los pesos de ambos y obtener así el camino más conveniente. Por lo tanto, para su aplicación existen algoritmos que facilitan la resolución de este problema.

Problema y justificación

El objetivo de la teoría de grafos para la búsqueda de las rutas mínimas tiene como principal objetivo automatizar los mecanismos de búsqueda de las rutas con la trayectoria que genere la mínima distancia total. Algunos de sus usos más importantes son: inteligencia artificial, telecomunicaciones (empaquetamiento de los paquetes por routers), aplicaciones de sistemas de información geográfica, entre otros.

Objetivos

- Poner en práctica los conceptos técnicos explicados en clase por el profesor, aplicar una correcta encapsulación y estructura de datos no elementales para la eficiencia de los algoritmos implementados.
- Aplicar los conocimientos adquiridos de teoría de grafos así como investigar los algoritmos de búsqueda más óptima para así aplicar alguno de ellos.
- Buscar el correcto manejo de estructuras de datos no elementales y su encapsulación, además de la presentar de un análisis sobre la eficiencia de los algoritmos programados.
- Obtener un análisis teórico sobre la estructura de datos a implementar en el programa, así como la presentación de ideas, información y demás material que apoye el código creado.

Alcances

Se ha logrado la creación de una estructura de datos apoyada en la el algoritmo llamado Floyd-Warshall para recorrer los grafos contruidos y sumar los ponderados correspondientes según el análisis

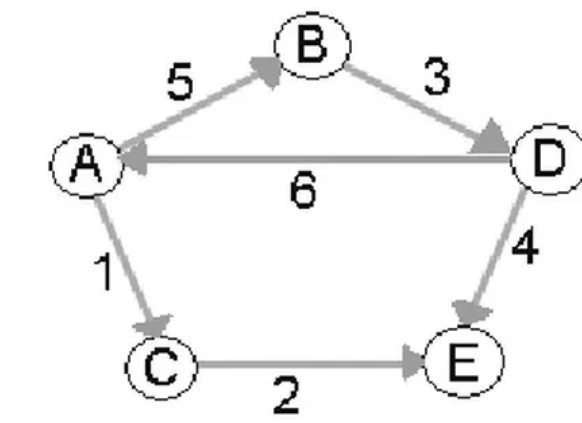
programado. Además, se ha alcanzado un análisis teórico antes y después de la creación del código, de modo que los conocimientos de los integrantes del equipo de trabajo han sido aumentados gracias a las exigencias del enunciado del presente proyecto.

2. MARCO DE REFERENCIA (MARCO TEÓRICO).

Descripción de la solución.

Primeramente se investigó sobre los algoritmos de utilidad para nuestro objetivo y se eligió el algoritmo llamado Floyd-Warshall el cual sirve para la búsqueda del camino mínimo en grafos dirigidos ponderados en una sola ejecución.

Teoría del método Floyd-Warshall:



_____ Consiste en crear dos matrices: una llamada ponderada y otra llamada de recorrido.

Matriz ponderada: Se procede a escribir en ella las ponderaciones de todos los recorridos teniendo en cuenta que: La diagonal no se usa y que en las casillas en que la dirección sea contraria se escribe el símbolo de infinito.

PONDERACIONES (PESO)					
	A	B	C	D	E
A	-	5	1	∞	∞
B	∞	-	∞	3	∞
C	∞	∞	-	∞	2
D	6	∞	∞	-	4
E	∞	∞	∞	∞	-

Por ejemplo: en la casilla [A,D] se pide el ponderado de A->D pero no existe, solamente existe D->A por lo tanto se pone infinito.

Matriz de recorrido:

RECORRIDO					
	A	B	C	D	E
A	A	B	C	D	E
B	A	B	C	D	E
C	A	B	C	D	E
D	A	B	C	D	E
E	A	B	C	D	E

Inicialmente la matriz de recorrido estará así y en base al análisis y cambios que se le haga a la matriz de ponderado ésta segunda matriz se irá editando. Se tomará cada letra y se analizará la fila-columna (fila A, columna A; fila B, columna B...etc) para sumar cada par de posiciones y bajo el criterio de $x+y < \text{pos}(\text{usando un nodo intermedio})$.

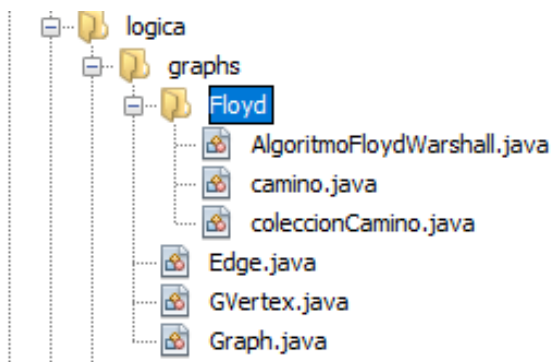
PONDERACIONES (PESO)					
	A	B	C	D	E
A	-	5	1	∞	∞
B	∞	-	∞	3	∞
C	∞	∞	-	∞	2
D	6	∞	∞	-	4
E	∞	∞	∞	∞	∞

Y finalmente cada cambio en la tabla de ponderaciones se hará en la matriz de recorrido poniendo una letra con la que se está trabajando, se harán n iteraciones hasta encontrar el camino óptimo.

RECORRIDO					
	A	B	C	D	E
A	A	B	C	B	C
B	D	B	D	D	D
C	A	B	C	D	E
D	A	A	A	D	E
E	A	B	C	D	E

Análisis de las estructuras de datos usadas en el programa.

1. Lógica (Paquete Graphs)

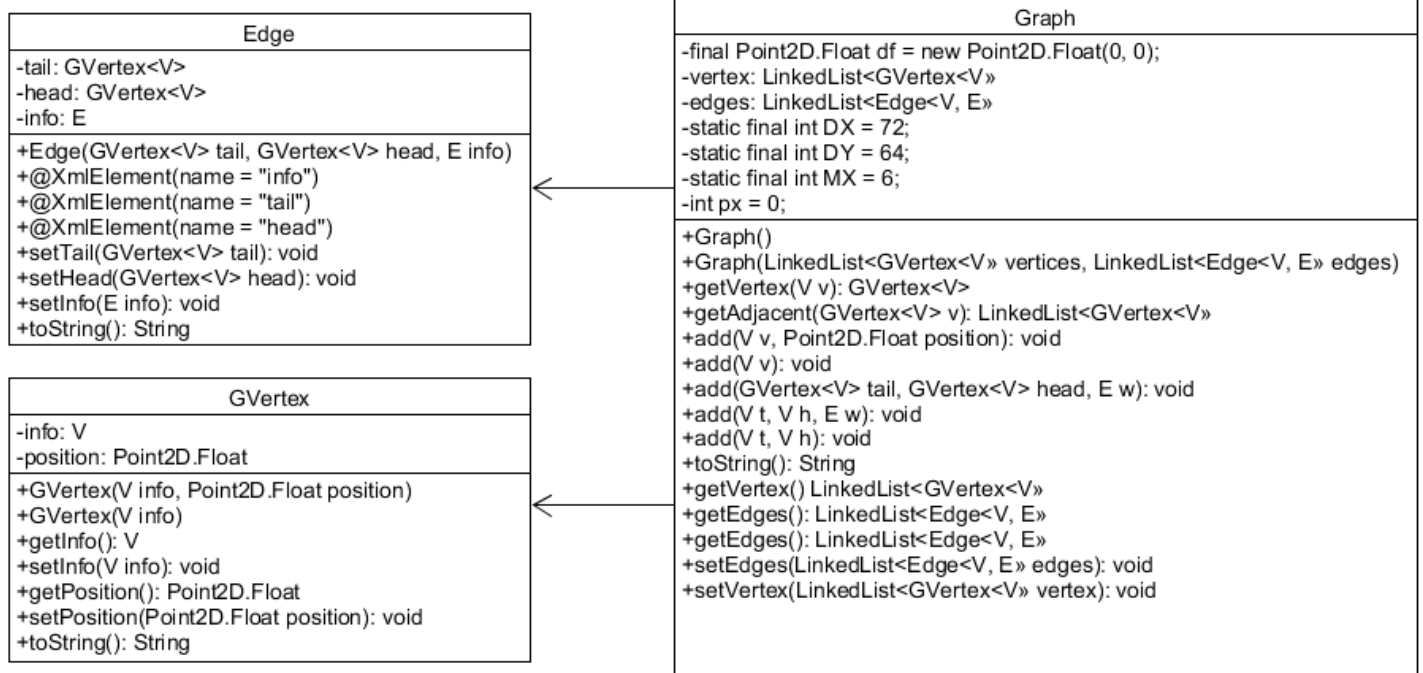


- Graphs contiene también Edge, GVertex y Graph ya que son necesitados para el algoritmo principal.

GVertex: Usa como atributos de tipo V y la clase Point2D para obtener las coordenadas y eventualmente poder dibujarla.

Edge: Contiene dos atributos de tipo Gvertex que representan cabeza y cola del vértice, también tiene la información del mismo.

Graph: Contiene una lista de GVertex y otra lista de Edge.



Encapsulación del método **Floyd-Warshall**: Se ha empaquetado en “Floyd”, junto con camino y “coleccioncamino”. “Floyd” contiene el algoritmo del camino más corto en la clase llamada “AlgoritmoFloydWarsahll” que necesita de Edge, GVertex y Graph.

camino: Tiene un vértice de inicio y otro de fin de tipo GVertex, un peso E y una lista de GVertex.

coleccionCamino: Es una lista de caminos en que tiene un método importante llamado buscarRuta el cual retorna el camino óptimo.

AlgoritmoFloyWarshall: La clase tiene 3 métodos en que cada uno devuelve un objeto de tipo coleccionCamino porque se necesita crear la matriz de pesos, la matriz de camino y la matriz resultante que representa el camino óptimo.

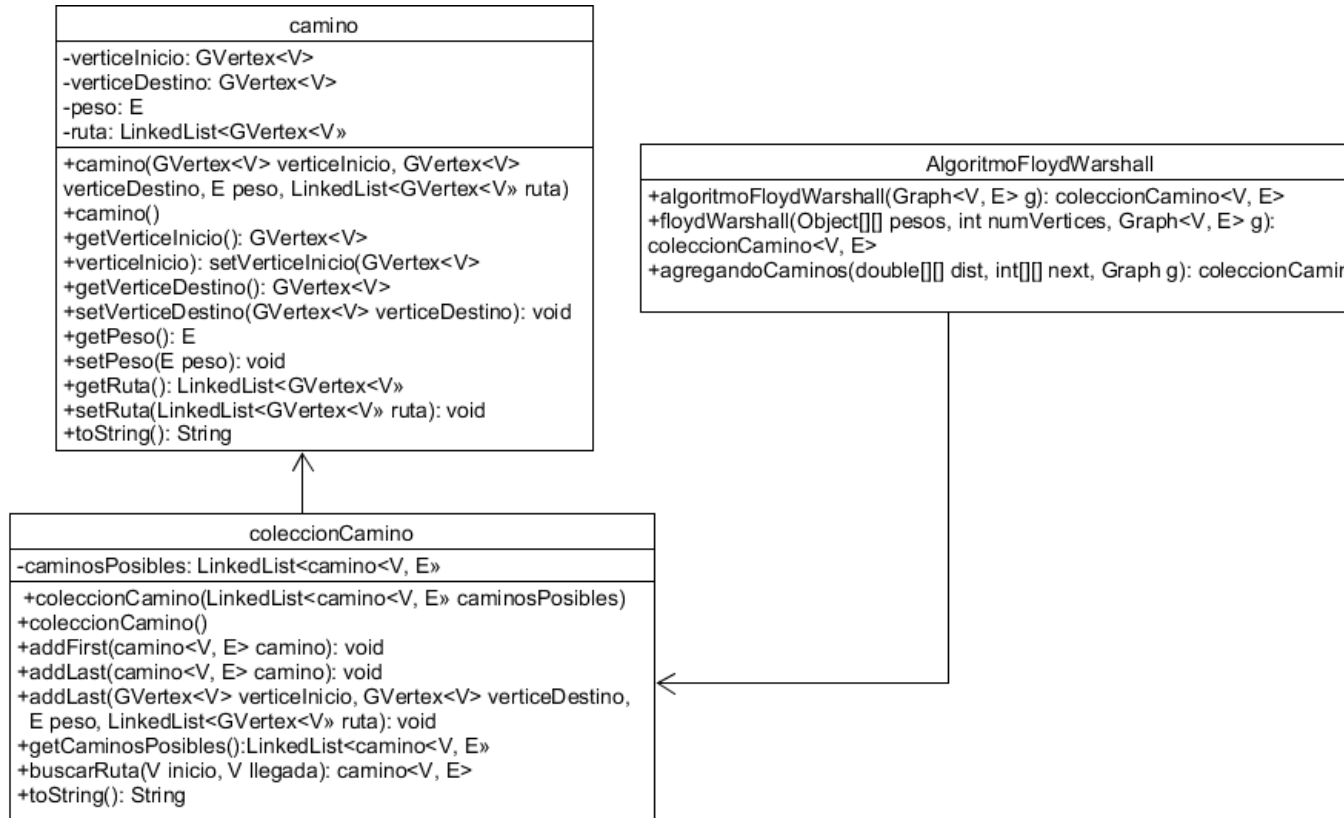


Diagrama por partes:

I: logica -> graphs -> floyd :

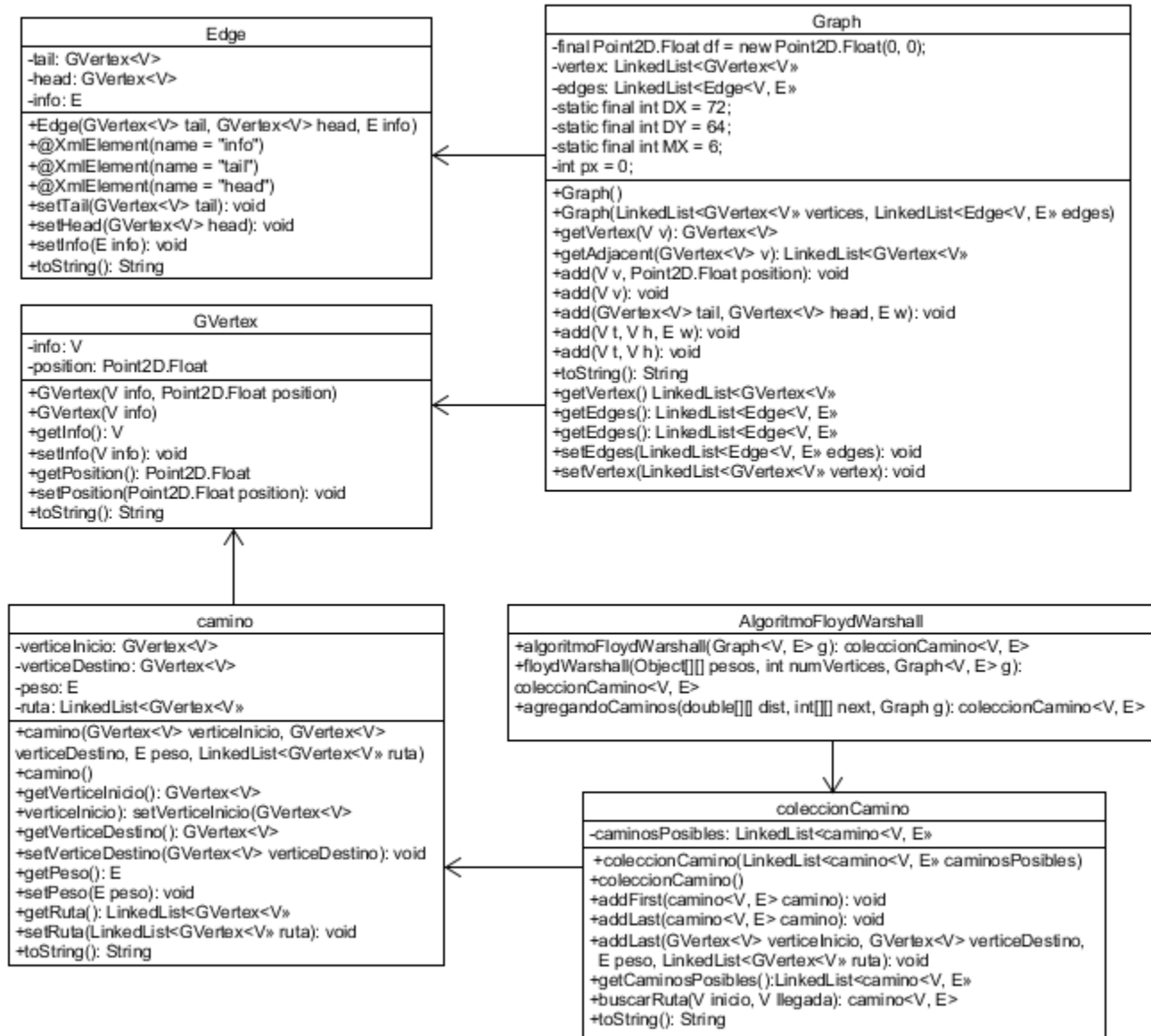
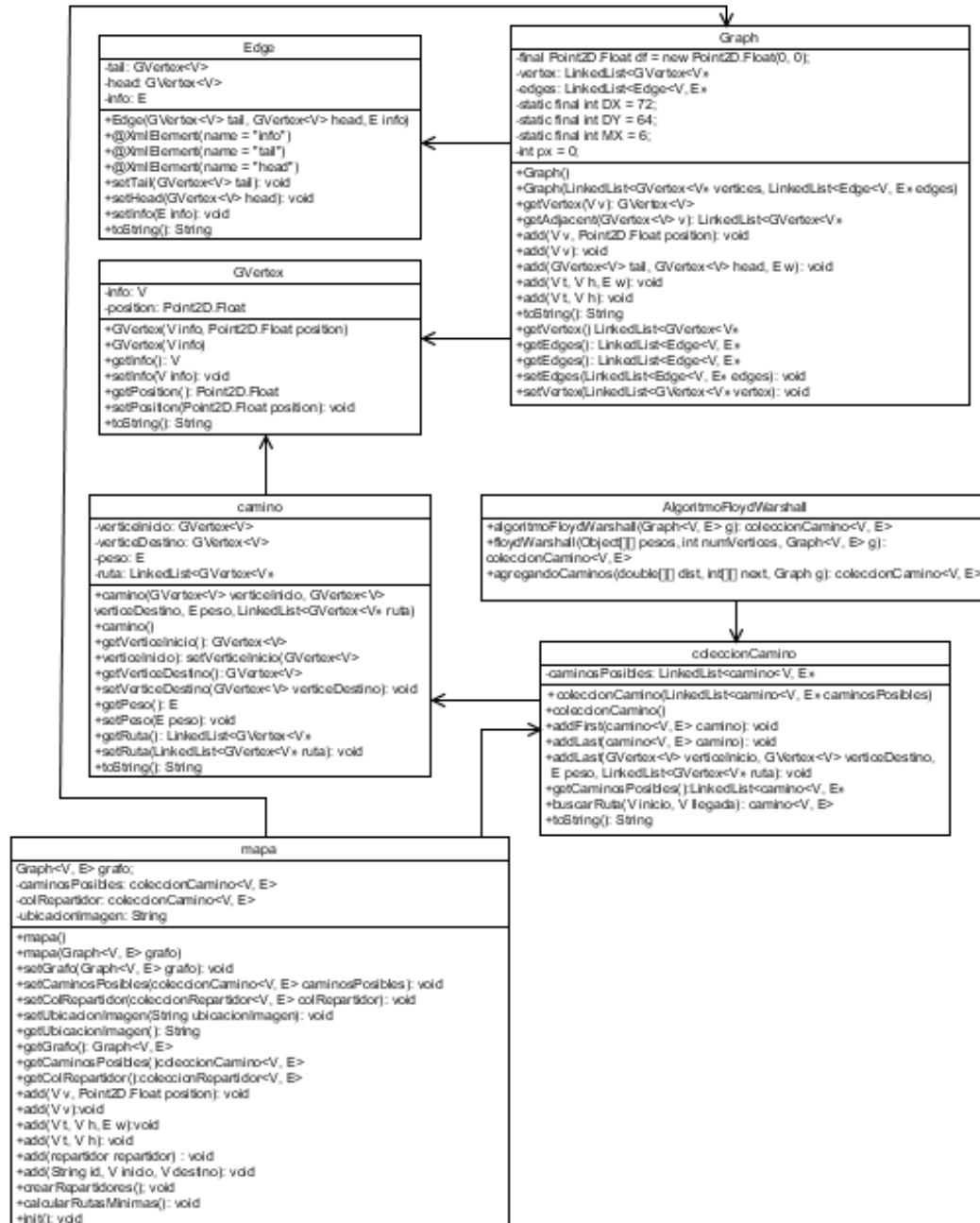


Diagrama por partes:

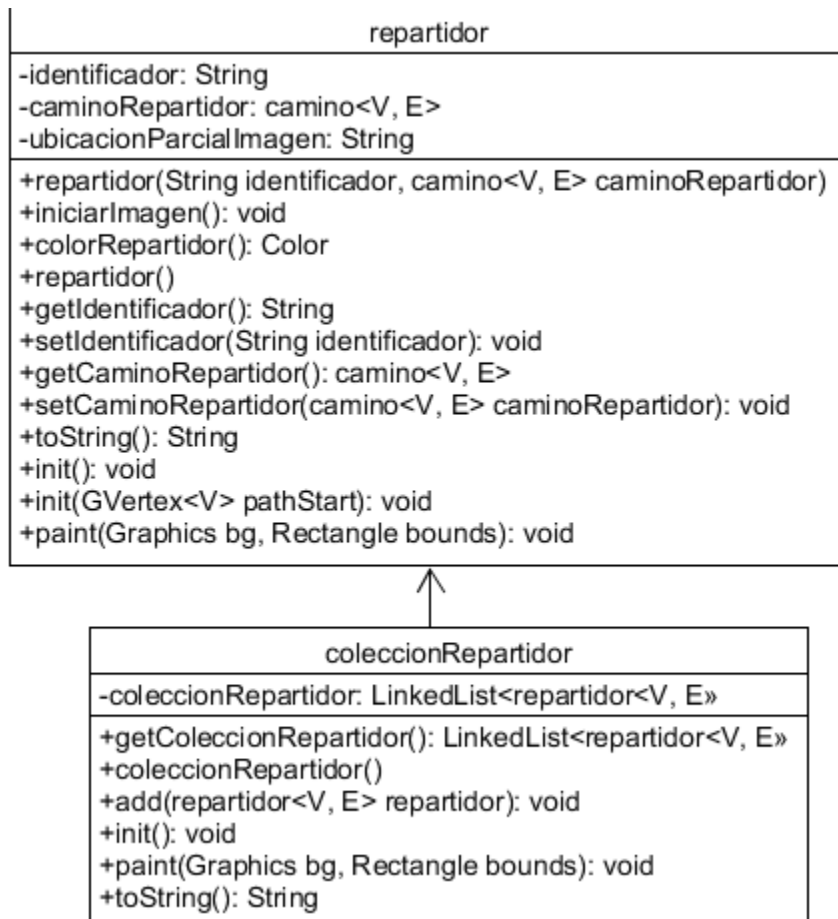
II logica -> mapa:

Mapa necesita de Graph y coleccionCamino:

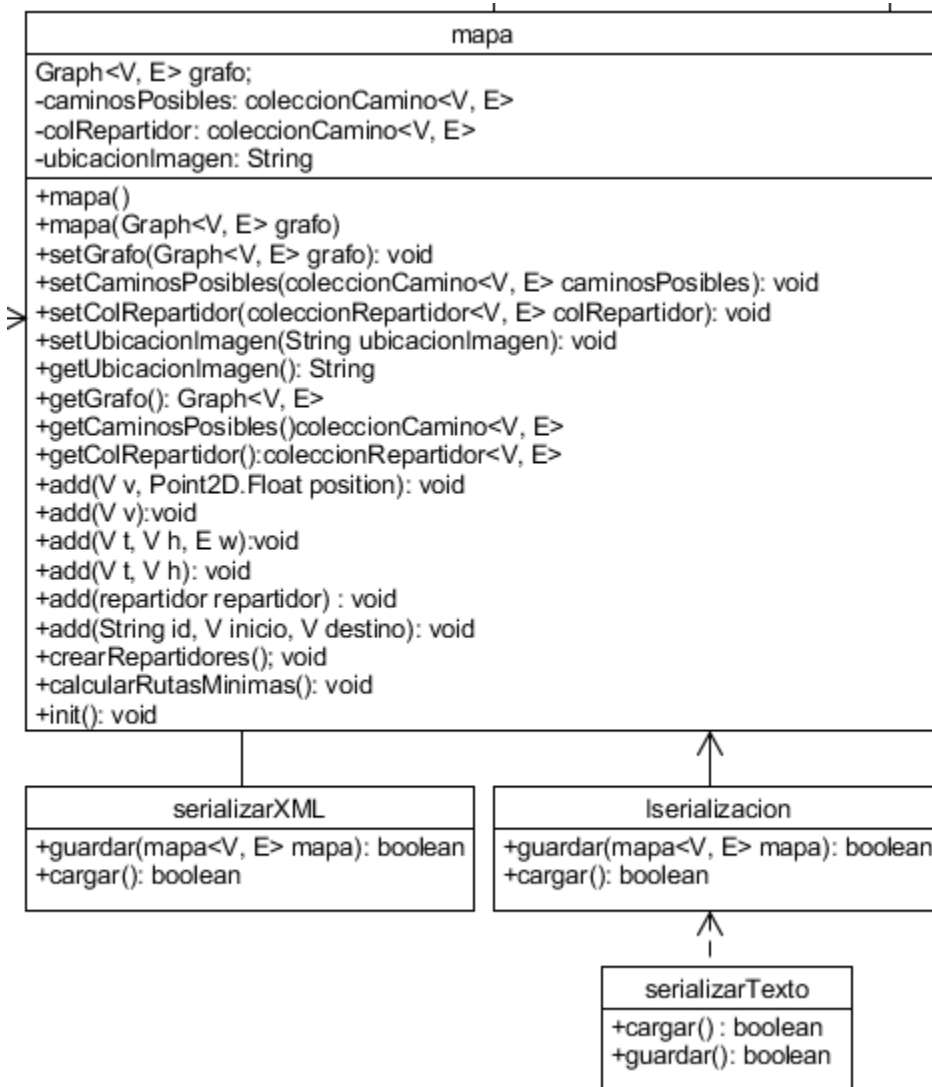


III repartidor: lógica -> repartidor

Repartidor necesita de camino y de este se crea la coleccionRepartidor:



IV xml para cargar los mapas:



Métodos importantes del programa:

Buscar ruta: En la clase coleccionCamino se necesita de este método para la ruta óptima recibiendo por parámetro el vértice de inicio y el de llegada.

```
public camino<V, E> buscarRuta(V inicio, V llegada) {  
    camino ptr = null;  
  
    for(camino<V,E> i:caminosPosibles){  
        ptr = i;  
        if (ptr.verticeDestino.getInfo().toString().equals(llegada)  
            && ptr.verticeInicio.getInfo().toString().equals(inicio)) {  
            break;  
        }  
    }  
}
```

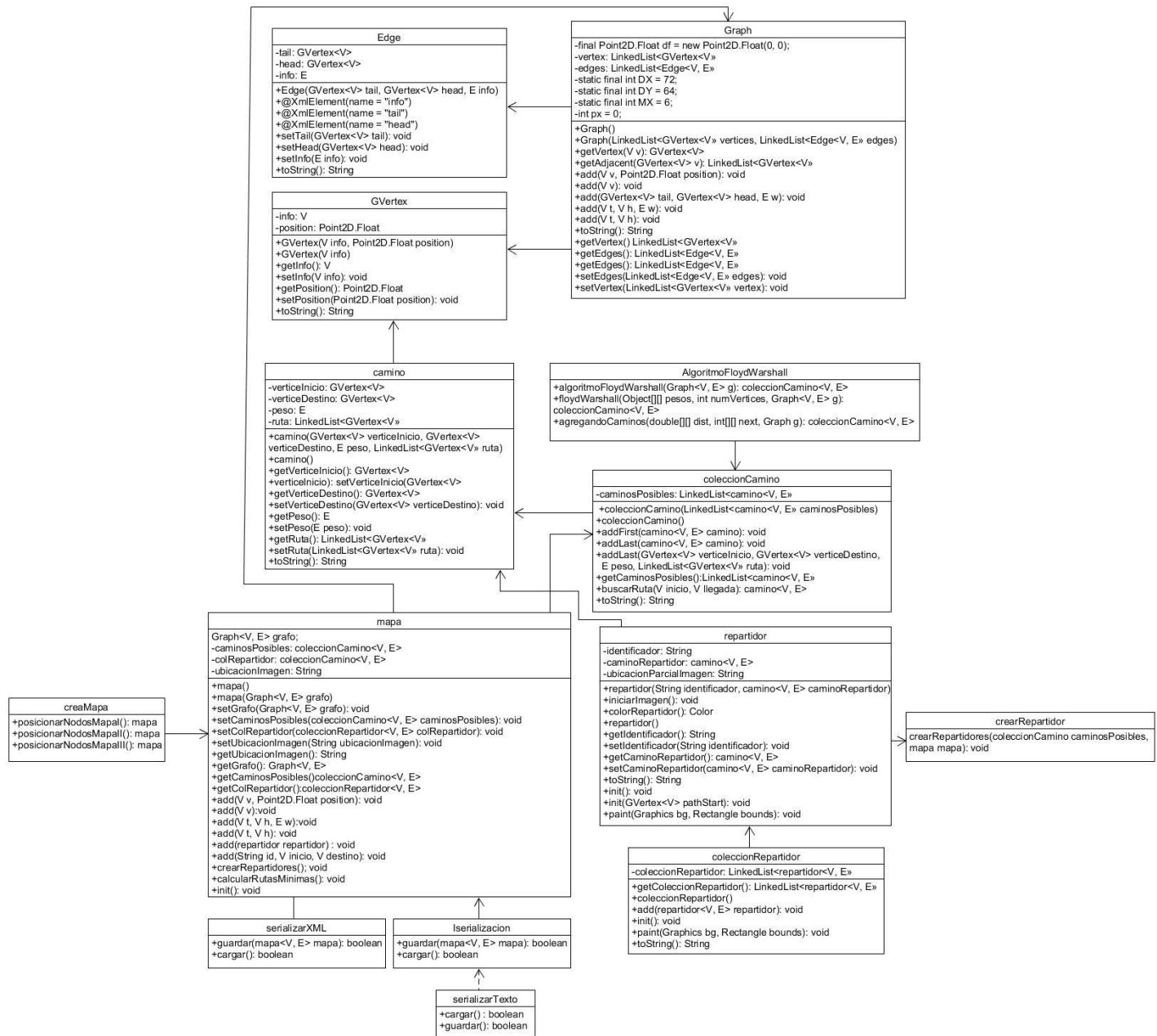
Desde la clase logica.mapa el siguiente método crea una instancia del algoritmo de Floyd Warshall:

```
private void calcularRutasMinimas() {  
    AlgoritmoFloydWarshall<V, E> floyd = new AlgoritmoFloydWarshall();  
    this.caminosPosibles = floyd.algoritmoFloydWarshall(grafo);  
}
```

Desde la clase mapa este método xml para guardar el mapa:

```
public boolean guardarMapa() {  
    serializarXML guardar = new serializarXML();  
    return guardar.guardar(this);  
}
```

Diagrama final del proyecto:



MARCO METODOLÓGICO.

Se ha usado el entorno de desarrollo netbeans para escribir el código Java; se la logrado encapsular los conceptos básicos para manejar un grafo básico conexo dirigido tal como vértices, nodos y la creación de conexión

entre vértices a modo de aristas. Además se unió el algoritmo Floyd Warshall para darle al código la utilidad de encontrar la ruta mínima según la suma de sus pesos.

4.CONCLUSIONES.

Hemos decidido usar un algoritmo no estudiado en clase porque después de investigar encontramos que sería de gran utilidad para fortalecer las habilidad de ser ingenieros más autodidactas, además pretendemos presentar un proyecto un caracter más innovador. Después de haberlo implementado podemos concluir que nuestra idea fue exitosa y cumplimos con los requerimientos del enunciado.

Con el presente trabajo el conocimiento adoptado en clase ha sido puesto en práctica, el código ha sido de gran utilidad para el análisis de grafos y aprendimos a implementar los recursos brindados por el profesor con nuestras ideas apoyadas por una investigación previa; hemos logrado usar logrado implementar conocimiento aprendido en el curso de programación III y nos hemos enfrentado a un análisis para la encapsulación del enunciado.

El grupo de trabajo ha logrado el análisis sobre teoría de grafos, se ha logrado crear un programa que guía camiones repartidores hacia las rutas óptimas según el mapa seleccionado a desplegarse como el enunciado lo pedía; para lo anterior se ha usado el método Floyd Warshall para lograr los cálculos en tiempo reducido y se logra una visualización del camino resultante. Por ende, consideramos nuestro proyecto II como un gran aprendizaje del curso de estructura de datos.

5. REFERENCIAS Y BIBLIOGRAFÍA.

<https://revistas.unal.edu.co/index.php/dyna/article/view/25773/39408>

https://www-m9.ma.tum.de/graph-algorithms/spp-floyd-warshall/index_en.html

<https://estructurasite.wordpress.com/algortimo-de-floyd-warshall/>

<http://dalila.sip.ucm.es/~manuel/Informatica/FloydWarshall.pdf>