



Tarea Integradora #3
Facultad de Ingeniería
Algoritmos y Estructuras De Datos
Grupo: 01
Semestre 2021 - B

Logistics SA



Integrantes:

Juan David Ballesteros Valencia - A00306456

Camilo González Velasco - A00370263

Samuel Guerrero Viveros – A00365567

DOCENTE: Johnatan Garzón Montesdeoca

Contenido

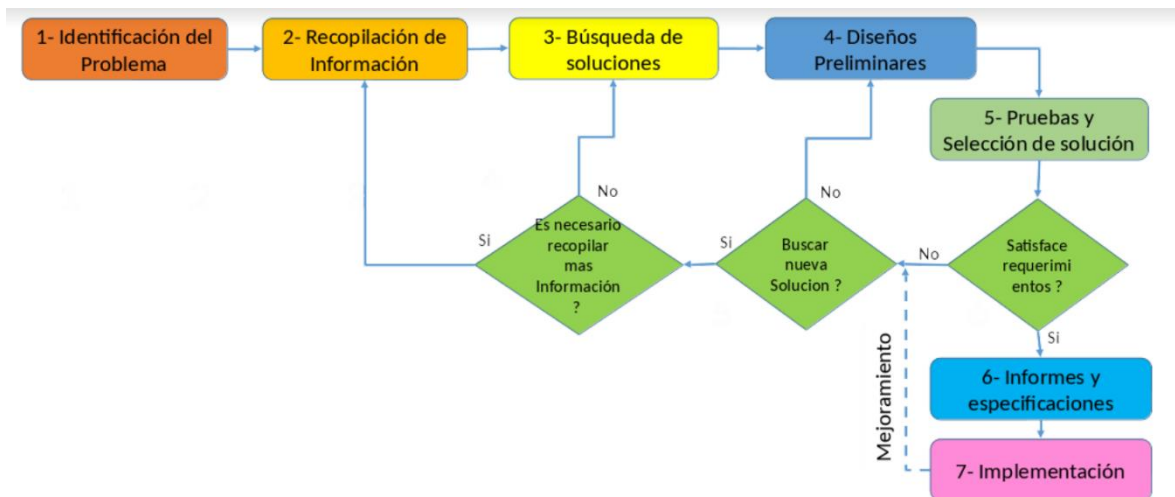
MÉTODO DE LA INGENIERÍA	3
Contexto	3
Desarrollo de la solución:	3
I. IDENTIFICACIÓN DEL PROBLEMA	4
<i>Definición del problema</i>	4
<i>Necesidades y Síntomas</i>	4
II. RECOPIACIÓN DE INFORMACIÓN	4
IV. TRANSICIÓN DE LAS IDEAS A LOS DISEÑOS PRELIMINARES	8
V. EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN	9
VI. PREPARACIÓN DE INFORMES Y ESPECIFICACIONES	11
VII. IMPLEMENTACIÓN DEL DISEÑO	11
Bibliografía:	12

MÉTODO DE LA INGENIERÍA

Contexto: Logis SA una startup de logística, la cual obtuvo un nuevo contrato con una empresa que suministra los productos de retail a las tiendas de barrio las cuales se encargan de vender los productos al consumidor final. La función de Logis SA es entregar los pedidos solicitados de la empresa intermediaria a las diferentes tiendas, para esto la empresa cuenta con una flota de vehículos, sin embargo, es la primera vez que se enfrentan al desafío de entregar tantos pedidos, además las estrategias que solían utilizar para hacer las entregas no son óptimas para un trabajo masivo como el que tienen sobre la mesa.

Desarrollo de la solución:

Para resolver la situación anterior se eligió el Método de la Ingeniería para desarrollar la solución siguiendo un enfoque sistemático y acorde con la situación problemática planteada. Con base en la descripción del Método de la Ingeniería del libro “Introduction to Engineering” de Paul Wright, se definió el siguiente diagrama de flujo, cuyos pasos seguiremos en el desarrollo de la solución.



I. IDENTIFICACIÓN DEL PROBLEMA

Definición del problema

El startup Logis SA, una empresa de logística desea consolidar un software que permita mejorar su gestión en las entregas que realizan. Software que le permita seleccionar las tiendas que se deben de visitar y de esta manera crear automáticamente la ruta más eficiente (corta) para el repartidor.

Necesidades y Síntomas

Logis SA ha solicitado la implementación de una herramienta para el manejo de información que permita ingresar tiendas, eliminar o consultar.

- La solución debe permitir agregar información a través de archivos .csv.
- La solución debe permitir agregar información de manera manual a través de una interfaz gráfica.
- La solución debe brindar un recorrido mínimo para hacer los envíos dispuestos durante la jornada laboral de la empresa.
- La solución debe almacenar y procesar la información en memoria secundaria, dado que el tiempo es crucial y no se puede perder un segundo de producción.

II. RECOPIACIÓN DE INFORMACIÓN

Hay algunas definiciones que son necesarias abordar para poder entender en su totalidad el problema, algunos son conceptos técnicos en informática que se deben de tener en cuenta para poder llevar a cabo una solución eficiente del problema. Otros conceptos necesarios son sobre el startup de logística y sus funciones.

Logística: Una empresa logística es aquella que se ocupa de gestionar las labores de transporte y distribución de productos y mercancías que producen y fabrican otras empresas. A grandes rasgos, su labor permite que los productos lleguen desde los centros de producción hasta el lugar en el que se van a consumir, ya sea otra empresa o un cliente particular.

Complejidad algorítmica: Concepto matemático que describe la eficiencia en recursos informáticos a la hora de realizar una secuencia de operaciones.

Data set: Conjunto o colección de datos específica, recopilada de varias fuentes orientada a características o aspectos específicos.

Vértice: Un vértice o nodo es la unidad fundamental de la que están formados los grafos. Llamaremos grado de un vértice al número de aristas de las que es extremo. Además, se le dice vértice “par” o “impar” según sea su grado.

Arista: Una arista o línea¹ corresponde a una relación entre dos vértices de un grafo.

Grafo: En matemáticas y ciencias de la computación, un grafo es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.

Caminos: En Teoría de Grafos, se llama camino a una secuencia de vértices dentro de un grafo tal que exista una arista entre cada vértice y el siguiente. Se dice que dos vértices están conectados si existe un camino que vaya de uno a otro, de lo contrario estarán desconectados. Dos vértices pueden estar conectados por varios caminos. El número de aristas dentro de un camino es su longitud.

Deep first search: Una búsqueda en profundidad (DFS) es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo. Su funcionamiento consiste en ir expandiendo cada uno de los nodos que va localizando, de forma recurrente (desde el nodo padre hacia el nodo hijo). Cuando ya no quedan más nodos que visitar en dicho camino, regresa al nodo predecesor, de modo que repite el mismo proceso con cada uno de los vecinos del nodo. Cabe resaltar que si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

Breadth first search: Una búsqueda en anchura (BFS) es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo, comenzando en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo), para luego explorar todos los vecinos de este nodo. A continuación, para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el grafo. Cabe resaltar que si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

Dijkstra: Algoritmo de Dijkstra. También llamado algoritmo de caminos mínimos es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de vértices en un grafo con pesos en cada arista.

Algoritmo de Floyd-Warshall: En informática, el algoritmo de Floyd-Warshall, descrito en 1959 por Bernard Roy, es un algoritmo de análisis sobre grafos para encontrar el camino mínimo en grafos dirigidos ponderados. El algoritmo encuentra el camino entre todos los pares de vértices en una única ejecución.

Árbol de recubrimiento mínimo: Dado un grafo conexo y no dirigido, un árbol recubierto, árbol de cobertura o árbol de expansión de ese grafo es un subgrafo que tiene que ser un árbol y contener todos los vértices del grafo inicial. Cada arista tiene asignado un peso proporcional entre ellos, que es un número representativo de algún objeto, distancia, etc.; y se usa para asignar un peso total al árbol recubierto mínimo computando la suma de todos los pesos de las aristas del árbol en cuestión. Un árbol recubierto mínimo o un árbol de expansión mínimo es un árbol recubierto que pesa menos o igual que todos los otros árboles recubridores. Todo grafo tiene un bosque recubridor mínimo.

Algoritmo de Prim: El algoritmo de Prim es un algoritmo perteneciente a la teoría de los grafos para encontrar un árbol recubridor mínimo en un grafo conexo, no dirigido y cuyas aristas están etiquetadas. En otras palabras, el algoritmo encuentra un subconjunto de aristas que forman un árbol con todos los vértices, donde el peso total de todas las aristas en el árbol es el mínimo posible. Si el grafo no es conexo, entonces el algoritmo encontrará el árbol recubridor mínimo para uno de los componentes conexos que forman dicho grafo no conexo.

Algoritmo de Kruskal: El algoritmo de Kruskal es un algoritmo de la teoría de grafos para encontrar un árbol recubridor mínimo en un grafo conexo y ponderado. Es decir, busca un subconjunto de aristas que, formando un árbol, incluyen todos los vértices y donde el valor de la suma de todas las aristas del árbol es el mínimo. Si el grafo no es conexo, entonces busca un bosque expandido mínimo (un árbol expandido mínimo para cada componente conexa).

III. BÚSQUEDA DE SOLUCIONES CREATIVAS

Para desarrollar las alternativas expuestas a continuación se realizó la técnica de lluvia de ideas: Generación espontánea de ideas diseñadas para resolver un problema, en combinación con una revisión secuencial a conceptos previamente adquiridos relacionados con las necesidades del problema expuesto.

Es preciso destacar que la solución tiene que ser una aplicación (software) con interfaz gráfica. Por ende, es razonable no considerar una solución que implique un software mediante línea de comandos (CLI) o en un programa en 3D, ya que la solución a este problema dado por Logis SA demanda un programa con interfaz gráfica (GUI). En este orden de ideas, el producto final será un software con interfaz gráfica y las posibles soluciones exhibidas a continuación estarán dirigidas al funcionamiento y/o comportamiento interno de dicho programa.

Implementación estructura de datos:

- a. LinkedList
- b. ArrayList.
- c. Grafo ponderado.
- d. Árbol n-ario.
- e. Grafo no ponderado.

Forma de implementación de estructura de datos:

- a. Lista de adyacencia
- b. Matriz de adyacencia
- c. Ninguna de las anteriores (en caso de que se elijan las opciones que implican grafos, es decir, a, b y d)

Tipo de estructura de datos:

- a. Grafo dirigido
- b. Grafo no dirigido
- c. Multigrafo dirigido
- d. Multígrafo no dirigido

- e. Ninguna de las anteriores (en caso de que se elijan las opciones que implican grafos, es decir, a, b y d)

Forma en la cual se harán los recorridos mínimos para las entregas:

- a. Deep first search
- b. Breadth first search
- c. Dijkstra
- d. Algoritmo de Floyd-Warshall
- e. Árbol de recubrimiento mínimo
- f. Algoritmo de Prim
- g. Ninguna de las anteriores (en caso de que se elijan las opciones que implican grafos, es decir, a, b y d)

IV. TRANSICIÓN DE LAS IDEAS A LOS DISEÑOS PRELIMINARES

En esta fase vamos a descartar las peores alternativas que no brindan una solución adecuada a las necesidades del problema.

Implementación estructura de datos

Analizando a profundidad lo requerido, las estructuras tales como LinkedList, ArrayList y árbol n-ario no permiten resolver los requerimientos que se piden en el enunciado. Las listas no tienen una relación entre ellas y esto no permite que existan camino y los árboles tienen una conexión, pero para llegar a una tienda puede existir más de un camino y en este problema es mejor no tener este tipo de limitaciones.

Forma de implementación estructura de datos

En este caso, no se eligió la opción ninguno de las anteriores dado que la estructura LinkedList, ArrayList y árbol n-ario se descartaron. Por otro lado, las opciones a y b siguen siendo viables por tanto no se descarta ninguna de las dos.

Tipo de estructura de datos

En este caso, no se eligió la opción ninguno de las anteriores dado que la estructura LinkedList, ArrayList y árbol n-ario se descartaron. Se descarta además el grafo no dirigido y multígrafo no dirigido dado que el tipo de grafo que mejor se acomoda a nuestros requerimientos son los grafos dirigidos, por el simple hecho de que se tienen entregas obligatorias (dirigidas) a alguna tienda en específico.

Forma en la cual se harán los recorridos mínimos para las entregas:

En este caso, no se eligió la opción ninguno de las anteriores dado que la estructura LinkedList, ArrayList y árbol n-ario se descartaron. Por tanto, la solución ira ligada a los grafos y los algoritmos expuestos en este componente son precisamente sobre grafos. Las opciones a y b pueden ser descartadas ya que son algoritmos de búsqueda los cuales no solucionarían este apartado, sin embargo, hay que tener en cuenta que son opciones muy factibles para el apartado de consulta que exige los requerimientos del problema.

Las opciones e y f no solucionan este apartado, además exigen grafos de carácter no dirigido y como se expuso en el anterior apartado se descartaron este tipo de grafos, entonces se descartan también estas alternativas.

V. EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN

Para evaluar las soluciones expuestas anteriormente, se plantearán diversos criterios para cada problema propuesto hallando así la mejor solución en cada contexto.

Criterios: (Cada alternativa tiene una puntuación de 5)

Implementación estructura de datos

- a. Permita asignar los tipos de relación (calles) entre las tiendas.
- b. Permita encontrar el camino más conveniente a el repartidor para hacer la entrega en la respectiva tienda.

Forma de implementación estructura de datos

No tiene criterios, dado que ninguno tiene una diferencia significativa o característica diferencial entre ellos, entonces se puede hacer uso de uno o de ambos a total libertad.

Tipo de estructura de datos

- a. Existe varios caminos que conectan a las tiendas.

Forma en la cual se harán los recorridos mínimos para las entregas:

- a. Encuentra el recorrido mínimo de manera eficiente.
- b. Posee una complejidad que permita un tiempo optimo.

Implementación estructura de datos

	Criterio A	Criterio B	Total
Alternativa C	5	5	10
Alternativa E	5	0	5

Se escoge la alternativa C la cual corresponde a un grafo ponderado, dado que el valor de las aristas permitirá encontrar el camino más corto.

Tipo de estructura de datos

	Criterio A	Total
Alternativa A	0	0
Alternativa C	5	5

Se escoge la alternativa C la cual corresponde a un multígrafo dirigido.

Forma en la cual se harán los recorridos mínimos para las entregas:

	Criterio A	Criterio B	Total
Alternativa C	5	0	5
Alternativa D	5	0	5

Se eligen ambas alternativas dado que cumplen con las condiciones mínimas y no tienen una diferencia significativa a menos de que se utilicen colas de prioridad. Sin embargo, para fines prácticos ambos algoritmos cumplen las condiciones.

VI. PREPARACIÓN DE INFORMES Y ESPECIFICACIONES

Todos los informes y especificaciones que se llevaron a cabo durante el desarrollo del software para solucionar el problema exhibido se encuentra ubicado en el repositorio del proyecto, más específicamente en la carpeta docs:

<https://github.com/JuanDavidBallesteros/logisticsSA/tree/master/docs>

VII. IMPLEMENTACIÓN DEL DISEÑO

La implementación del diseño fue realizada en el lenguaje de programación Java, junto con Scene Builder el cual se usó para construir la interfaz gráfica GUI. El proyecto se encuentra almacenada en el siguiente repositorio: <https://github.com/JuanDavidBallesteros/logisticsSA>

Bibliografía:

Algoritmo de Kruskal. (s. f.). Kruskal. Recuperado 14 de noviembre de 2021, de

http://i3campus.co/CONTENIDOS/wikipedia/content/a/algoritmo_de_kruskal.html

colaboradores de Wikipedia. (2020a, octubre 20). *Árbol recubridor mínimo*. Wikipedia, la enciclopedia libre.

Recuperado 13 de noviembre de 2021, de

https://es.wikipedia.org/wiki/%C3%81rbol_recubridor_m%C3%ADnimo#:~:text=Un%20%C3%A1rbol%20recubridor%20m%C3%ADnimo%20o,cable%20a%20una%20nueva%20vecindad.

colaboradores de Wikipedia. (2020b, diciembre 19). *Vértice (teoría de grafos)*. Wikipedia, la enciclopedia

libre. Recuperado 12 de noviembre de 2021, de

[https://es.wikipedia.org/wiki/V%C3%A9rtice_\(teor%C3%ADa_de_grafos\)#:~:text=En%20teor%C3%ADa%20de%20grafos%2C%20un,que%20est%C3%A1n%20formados%20los%20grafos.&text=Un%20v%C3%A9rtice%20w%20es%20adyacente,los%20v%C3%A9rtices%20adyacentes%20a%20v.](https://es.wikipedia.org/wiki/V%C3%A9rtice_(teor%C3%ADa_de_grafos)#:~:text=En%20teor%C3%ADa%20de%20grafos%2C%20un,que%20est%C3%A1n%20formados%20los%20grafos.&text=Un%20v%C3%A9rtice%20w%20es%20adyacente,los%20v%C3%A9rtices%20adyacentes%20a%20v.)

colaboradores de Wikipedia. (2021a, abril 27). *Arista (teoría de grafos)*. Wikipedia, la enciclopedia libre.

Recuperado 12 de noviembre de 2021, de

[https://es.wikipedia.org/wiki/Arista_\(teor%C3%ADa_de_grafos\)#:~:text=Gr%C3%A1ficamente%20las%20aristas%20se%20representan,y%20apunta%20a%20nodo%20destino.](https://es.wikipedia.org/wiki/Arista_(teor%C3%ADa_de_grafos)#:~:text=Gr%C3%A1ficamente%20las%20aristas%20se%20representan,y%20apunta%20a%20nodo%20destino.)

colaboradores de Wikipedia. (2021b, junio 23). *Algoritmo de Floyd-Warshall*. Wikipedia, la enciclopedia

libre. Recuperado 12 de noviembre de 2021, de https://es.wikipedia.org/wiki/Algoritmo_de_Floyd-Warshall

colaboradores de Wikipedia. (2021c, agosto 4). *Grafo*. Wikipedia, la enciclopedia libre. Recuperado 12 de

noviembre de 2021, de

<https://es.wikipedia.org/wiki/Grafo#:~:text=En%20matem%C3%A1ticas%20y%20ciencias%20de,de%20la%20teor%C3%ADa%20de%20grafos.>

colaboradores de Wikipedia. (2021d, agosto 22). *Algoritmo de Dijkstra*. Wikipedia, la enciclopedia libre.

Recuperado 12 de noviembre de 2021, de https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra

colaboradores de Wikipedia. (2021e, octubre 13). *Camino (teoría de grafos)*. Wikipedia, la enciclopedia libre.

Recuperado 12 de noviembre de 2021, de

[https://es.wikipedia.org/wiki/Camino_\(teor%C3%ADa_de_grafos\)](https://es.wikipedia.org/wiki/Camino_(teor%C3%ADa_de_grafos))

colaboradores de Wikipedia. (2021f, noviembre 14). *Algoritmo de Prim*. Wikipedia, la enciclopedia libre.

Recuperado 14 de noviembre de 2021, de

https://es.wikipedia.org/wiki/Algoritmo_de_Prim#:~:text=El%20algoritmo%20de%20Prim%20es,y%20cuyas%20aristas%20est%C3%A1n%20etiquetadas.&text=Por%20esta%20raz%C3%B3n%20el%20algoritmo,DJP%20o%20algoritmo%20de%20Jarnik.

DFS vs BFS. (2020, 25 mayo). Encora. Recuperado 12 de noviembre de 2021, de

<https://www.encora.com/es/blog/dfs-vs-bfs>

¿Qué es y qué hace una empresa logística? (s. f.). Logística. Recuperado 12 de noviembre de 2021, de

<https://www.beetrack.com/es/blog/que-es-una-empresa-logistica>

Imagen tomada de:

<https://www.logimov.com/nwlib6/includes/phpthumb/phpThumb.php?src=/imagenes/3software.jpg&w=700&f=jpg>