Mayorista Farmacéutica

Sistema de Optimización de Rutas

Juan Niño 2240040. Daniver Hernandez 2240032. Luis Rueda 2240021. Juan Rivera 2240046

Problema

Una mayorista farmacéutica con alta demanda necesita distribuir medicamentos y otros insumos críticos de forma rápida y eficiente a hospitales, clínicas y farmacias. Ante la necesidad de minimizar tiempos de entrega, reducir el consumo energético y responder a imprevistos (como cambios de demanda o condiciones climáticas adversas), se opta por utilizar drones que realice entregas automáticas.

La mayorista tiene una base de datos con todas las zonas de entrega, y de acuerdo con los criterios seleccionados la ruta se crea. Los criterios son las características del paquete y se definen más adelante.

Los drones están programados para realizar una sola ruta semanal, en la que recorre la ciudad, yendo a cada punto de entrega en su base de datos que cumpla las condiciones, y entrega el paquete de farmacéuticos correspondiente. El dron recorre su ruta, ordenando las entregas para que sean geográficamente cercanas entre sí, de tal forma que se optimice el camino. La cantidad de entregas depende completamente de los criterios de selección y necesidades del cliente.

El problema a resolver consiste en planificar y optimizar las rutas de vuelo de losdrones, garantizando que cada entrega se realice en el menor tiempo y con el menor recorrido posible. Esto no solo permite mejorar la eficiencia logística, sino que también asegura la entrega oportuna de productos esenciales para la salud de la población.

Puesta en Contexto de Grafos y Nodos

La estructura de datos Grafos es usada para representar todos los caminos que puede tomar cada dron en su ruta de entrega. Así pues, el Nodo del Grafo representa cada punto al que el dron debe llevar un paquete, y todos los nodos están interconectados entre sí, incluyendo la base de donde sale el dron. Como todos los nodos se conectan el uno con el otro, hay que

establecer un orden para que se recorra la ruta, iniciando desde la base (origen) del dron y terminando ahí también. En un plano de coordenadas xy, tomando los cuatro cuadrantes, existe un nodo origen en x=0,y=0 que es de donde parte el dron,, es decir, es la base de operaciones, y tiene dos vértices y dos vecinos, por uno de ellos sale a recorrer la ruta, y por el otro llega luego de completar la última entrega.

El tipo de Grafo es: No dirigido y Etiquetado. Es etiquetado ya que se define con múltiples características, aunque no se muestren en las graficas. De acuerdo al contexto, este tipo ofrece varios beneficios y ventajas estratégicas, como:

- **Bidireccionalidad:**
  El grafo no dirigido permite que se recorra la arista en cualquier dirección.. Esto facilita el calculo de las rutas posibles del dron, ya que no discrimina si puede o no tomar un camino hacia otro nodo por su dirección, al poder recorrerlo en ambas direcciones simplemente se tomara la más adecuada,.

- **Optimización Natural:**

  Al ser etiquetados, cada nodo posee las caracteristicas generales, y esto permite el calculo de distancias entre nodos, pesos de carga en las rutas, se pueden aplicar algoritmos para encontrar la ruta más corta, la más ligera, o la de mayor valor, optimizando así los viajes del dron.

- **Reducción de Complejidad:**

  Ser etiquetado y no dirigido permite agregar o quitar nodos (puntos de entrega) fácilmente, sin alterar la estructura general del modelo. Además, ss útil para adaptarse rápidamente a cambios en la demanda, nuevas zonas de entrega o condiciones adversas, lo que mejora la resiliencia logística

Librerías

- **Math:** Esta librería proporciona funciones matemáticas estándar (como las que encuentras en una calculadora científica). Más que todo se usa math.sqrt() para sacar el valor de la raiz cuadrado, es decir, la distancia
- **Random:** Permite trabajar con generación de números aleatorios y selección aleatoria de elementos. En la demo del repositorio del proyecto, se utiliza para generar las zonas de entrega, nodos.

- **matplotlib.pyplot as plt:** Es parte de la librería Matplotlib, utilizada para graficar datos en Python. Se importa generalmente como plt por convención. Se utiliza para crear la gráfica (plot) del grafo en el plano, es decir, visualizar las rutas.
- **matplotlib.cm as cm:** Es parte de la librería Matplotlib, utilizada para manejar mapas de color en Python. Se importa generalmente como cm por convención. Se utiliza para dar colores l la a gráfica (plot) del grafo en el plano, y poder dar un color diferente a cada uno de los subgrafos y rutas diferentes.

La clase Dron

El dron es la clase que define las limitaciones preestablecidas para las entregas. Es el que recorre la ruta, y se crea desde el menu de opciones de usuario, donde puede definirse nuevos objetos Dron sus características limitantes.

El dron almacena los siguientes datos

- Nombre del dron
- El peso máximo que puede soportar. Es decir, la cantidad de paquetes que podrá llevar de acuerdo al peso de cada uno de ellos
- La ruta, osea, el nodo a recorrer.
- Un identificador de la ruta actual por la que irá el dron

La clase Nodo

El nodo es fundamental porque encapsula toda la información necesaria para que cada dron pueda realizar la entrega de manera precisa y segura. Al representar cada parada como un nodo, se facilita el cálculo de distancias, la optimización de la secuencia de entregas y la capacidad de adaptación ante cambios en la operación (por ejemplo, reordenar paradas según la prioridad o condiciones del entorno).

Aparte de los parámetros, el nodo posee un método toString para retornar toda la información en una cadena de texto., un metodo __hash__ para retornar su id, y un metodo __eq__ para compararse con otros nodos y definir si son iguales-

Cada nodo almacena los siguientes datos

- Nombre de la farmaceútica

- Identificador (ID) de la entrega (Código de tres números)
- La cantidad de paquetes que va a recibir el sitio
- Valor económico total de los paquetes a entregar
- Posición x del punto de entrega (Coordenadas del punto de entrega)
- Posición y (Coordenadas del punto de entrega)
- Referencias a todos los otros puntos de entrega introducidos en el programa

`La clase Graph`

El Grafo representa el mapa de vuelo completo de un dron, incluyendo todos los puntos de entrega (nodos) y las conexiones entre ellos (aristas). El dron parte del origen, que es también el origen en un plano cartesiano, con sus cuatro cuadrantes. A partir del (0,0), se reparten todos los nodos(zonas de entrega), y siguiendo la teoría de grafos, todos los nodos se conectan entre sí y con el origen. Esta red de conexiones se coloca con una opacidad menor para reconocer su existencia pero no obstruir la visión.

El dron siempre parte del origen y termina su ruta volviendo al origen desde un nodo distinto del cual partió.

De acuerdo a las restricciones de peso, distancia o la ruta en sí misma, el dron puede no pasar por ciertos nodos.

El constructor del Grafo Graph lo define con los siguientes parámetros:

- Vértices: Son conjunto de nodos. Si se proporciona una lista de nodos , se crea un conjunto de puntos de entrega.
- Edges: Son el conjunto de aristas o conexiones entre nodos.
- Weight: conjunto de pesos (distancias entre pares de nodos) total del grafo

Si hay vertices se construye automáticamente el grafo completo con conexiones entre todos los nodos posibles mediante el método `_build_full_graph`.

El grafo tiene los siguientes métodos:

- `_build_full_graph`: Este método conecta todos los nodos entre sí (grafo completo). Calcula la distancia entre cada par usando el teorema de pitágoras entre las coordenadas de nodo, y se agregan las aristas y sus respectivos pesos (distancia) al grafo.

- `_add_node`: Se añade un nuevo punto de entrega (nodo) al grafo. Se conecta con todos los nodos existentes, añadiendo las aristas y calculando sus distancias, y por último se incorpora al conjunto de vertices..

- `remove_node`: elimina un nodo de la red de entregas, también se eliminando todas las conexiones (aristas) y distancias relacionadas con ese nodo.
- `get_weight`: fecibe dos vértices y devuelve el valor de la arista entre ellos, es decir, su distancia. Si no encuentra esa arista, lanza un error.
- `nearest_neighbor:` devuelve una lista de nodos que describe una ruta inicial eficiente para el dron utilizando el algoritmo del vecino más cercano (Nearest Neighbor). Es una solución aproximada al problema del viajero (TSP), en donde el dron intenta visitar todos los nodos con pedidos (has_order=True) partiendo de un nodo inicial (start), minimizando la distancia total de vuelo. En cada nodo, el dron busca el nodo con menor distancia en ese precisa iteración y se coloca en él, para iniciar de nuevo su busqueda, hasta visitar todos los nodos, entonces vuelve a la base.
- `tour_length:`Este método calcula la distancia total de una ruta de entrega (una lista de nodos en orden). Siguiendo la ruta, suma todas las distancias usando get_weight(), y retorna un número
- `adjust_route:`Este método recalcula la ruta entrega del dron en caso de que se elimine un nodo (se cancele un pedido. Devuelve una nueva lista de nodos (`new_tour`) que representa la ruta ajustada del dron, finalizando nuevamente en el nodo de inicio (`start`).
- `plot_tour:`Este método permite visualizar gráficamente una ruta completa del dron (una lista de nodos), destacando qué zonas tienen pedidos activos y cuáles no. Utiliza matplotlib.pyplot.
- `plan_routes:`Este método divide la entrega total en múltiples rutas optimizadas según restricciones físicas del dron como peso máximo, distancia máxima o volumen de entregas (número de paquetes). Retorna una lista de rutas, segun la restriccion elegida,`max_weight`, `max_volume`, `max_distance.`
- `plot_full_graph:`Este método permite visualizar el grafo completo que representa todas las posibles rutas de conexión entre zonas de entrega.
- `plot_route:`Este método dibuja en un gráfico una sola ruta (una de las rutas del dron), permitiendo visualizar el trayecto de un viaje específico..
- `plot_all_routes:`Este método permite ver todas las rutas generadas por el algoritmo de planificación en un solo gráfico, cada una en un color distinto.

Ya al final, se encuentra un menú interactivo , (User Interface)que permite utilizar el programa. Cuenta con las opciones de Agregar Nodo, paragenerar un nuevo grafo (crear un objeto clase Graph si hay al menos dos nodos), generar subruta por criterios como distancia maxima a recorrer o peso maximo, Visualizar rutas usa matplotlib.pyplot para graficar las rutas generadas, Agregar un dron agrega un dron, Asignar ruta al dron para indicar que camina va a tomar, Mostrar menu, y salir del programa. Hay una ultima opcion que es una Demo

La Demo, Interfaz gráfica de la ruta y Prueba con datos reales

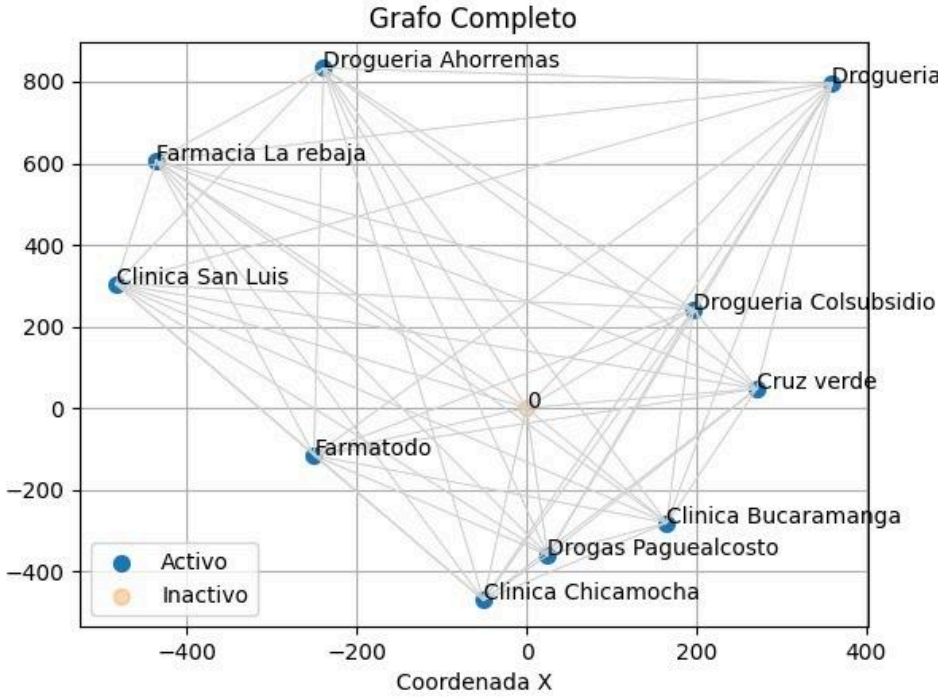
La opcion 9 del menu es para ejecutar un escenario de prueba, y se puede realizar con datos aleatorios o con un set de datos reales de Bucaramanga. Si se eligen reales, usa 10 droguerías reales con coordenadas específicas. Si se elige aleatorio: genera 15 nodos con posiciones y pedidos aleatorios. Luego, hace una demostraión del funcionamiento de los métodos para crear el grafo, hacer la ruta ineficiente de vecinos más cercanos, cancela un pedido, reajusta la ruta, generasubrutas con un peso máximo, y muestra y grafica las subrutas

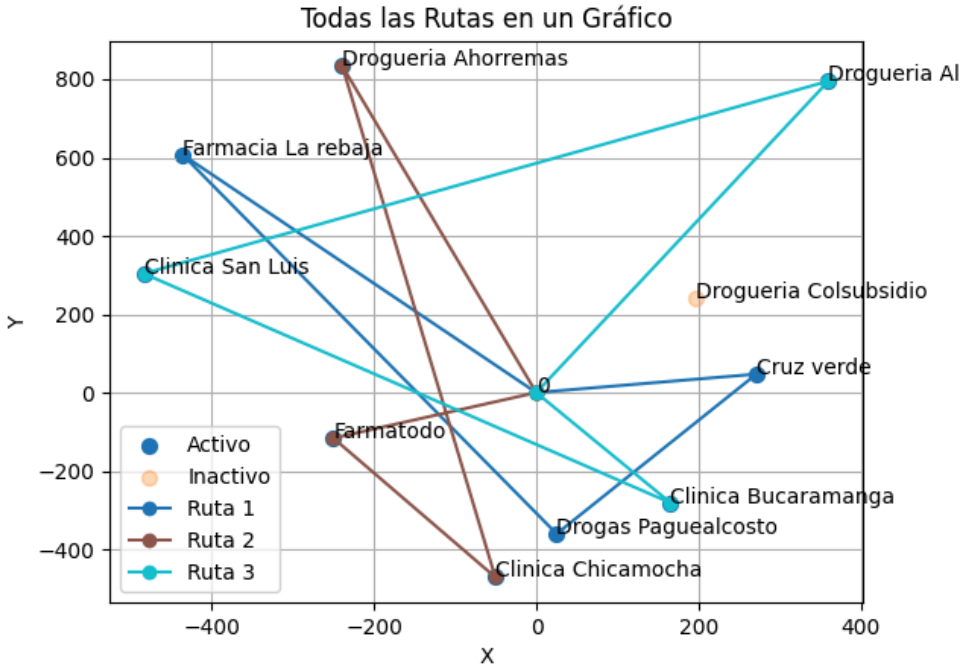
Las clínicas utilizadas para la ruta son:

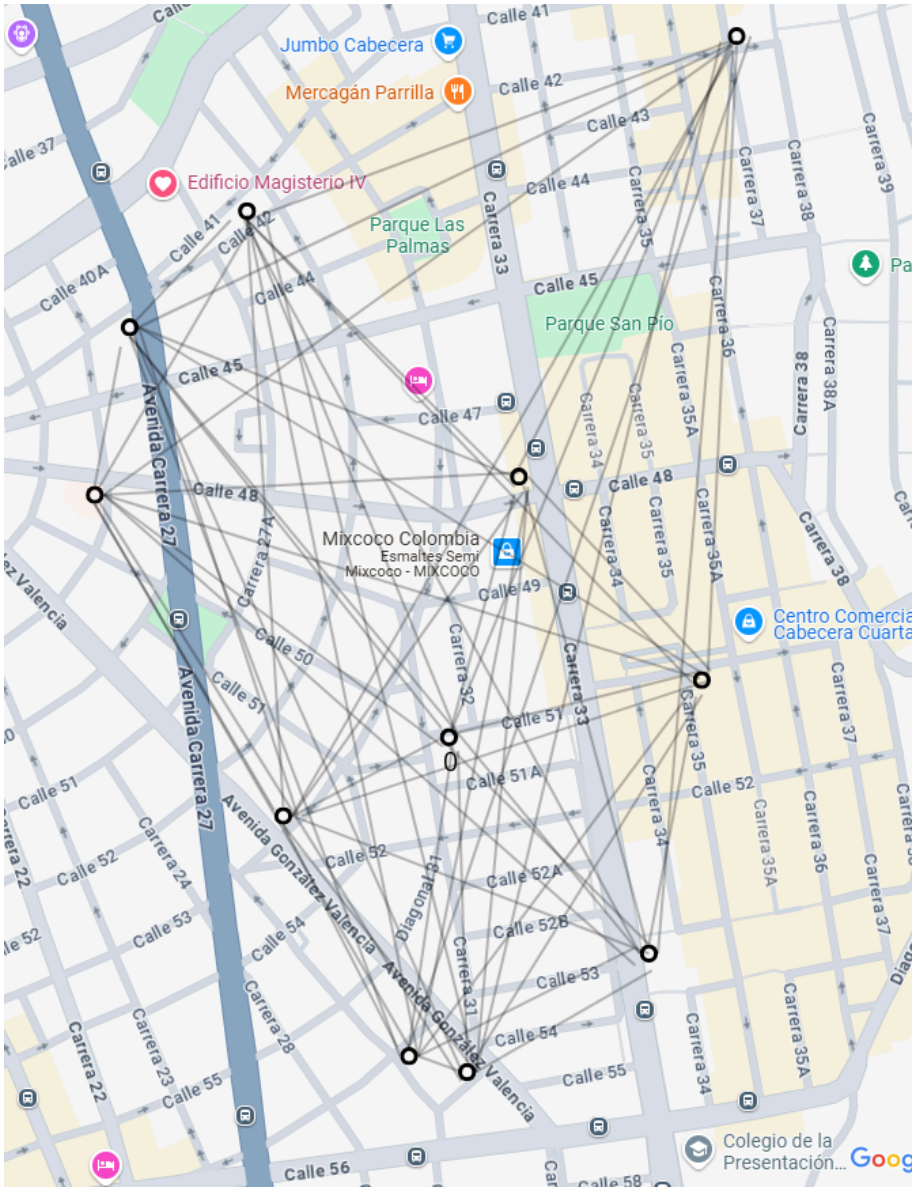
- Clínica Bucaramanga
- Clínica Chicamocha
- Clínica San Luis

Las Farmacias utilizadas en la ruta son:

- Farmatodo
- Drogas La Rebaja
- Droguería Colsubsidio
- Cruz Verde
- Drogas paguealcosto
- Drogas Ahorramás
- Droguería Alemana


Grafo Completo
Drogueria Ahorremas
Drogueria
Farmacia La rebaja
Clinica San Luis
Drogueria Colsubsidio
Cruz verde
0
Farmatodo
Clinica Bucaramanga
Drogas Paguealcosto
Clinica Chicamocha
Activo
Inactivo
800
600
400
200
0
−200
−400
−400
−200
0
200
400
Coordenada X


Todas las Rutas en un Gráfico
Drogueria Ahorremas
Drogueria Al
Farmacia La rebaja
Clinica San Luis
Drogueria Colsubsidio
Cruz verde
0
Farmatodo
Clinica Bucaramanga
Drogas Paguealcosto
Clinica Chicamocha
Activo
Inactivo
Ruta 1
Ruta 2
Ruta 3
800
600
400
200
0
−200
−400
Y
−400
−200
0
200
400
X


Jumbo Cabecera
Mercagán Parrilla
Edificio Magisterio IV
Parque Las Palmas
Parque San Pío
Mixcoco Colombia
Esmaltes Semi Mixcoco - MIXCOCO
Centro Comercial Cabecera Cuarta
Colegio de la Presentación...
Avenida Carrera 27
Avenida Gonzalez Valencia
Diagonal 51
Calle 41
Calle 42
Calle 43
Calle 44
Calle 45
Calle 47
Calle 48
Calle 49
Calle 50
Calle 51
Calle 51A
Calle 52
Calle 52A
Calle 52B
Calle 53
Calle 54
Calle 55
Calle 56
Calle 58
Calle 40A
Carrera 22
Carrera 22A
Carrera 23
Carrera 24
Carrera 28
Carrera 31
Carrera 32
Carrera 33
Carrera 34
Carrera 35
Carrera 35A
Carrera 36
Carrera 37
Carrera 38
Carrera 38A
Carrera 39