

- Proyecto #1 Grafos // Hecho por: Luis Asenjo, Santiago Zabala y Alejandro Zamora
  - Descripción del problema
  - Descripción de los Diagramas de Clases
    - Diagrama del Json
    - Diagrama Estructuras Básicas
    - Diagrama Grafo y Redes
    - Diagrama Completo
  - Uso/Ejecución del proyecto
    - 1. Menú Principal
      - a) Cargar Archivo
      - B) Agregar Línea
        - i) Nombre Línea
        - ii) Paradas
        - iii) Información de la línea a agregar
      - c) Mostrar Grafo
        - i) Vista del Grafo
        - ii) Lista de Paradas en la Línea
        - iii) Cobertura de la Parada
        - iv) Lista de Sucursales
        - v) Lista de Sucursales Recomendadas
        - vi) Comportamiento del Grafo según las acciones del manejador
          - \*) Selección de una parada
          - \*\*) Agregar una línea
    - Uso de un archivo .bat para la ejecución del programa

# Proyecto #1 Grafos // Hecho por: Luis Asenjo, Santiago Zabala y Alejandro Zamora

---

## Descripción del problema

---

Problema:

Una cadena de supermercados quiere abrir nuevas tiendas en Sudamérica y necesita una herramienta digital para decidir dónde ubicarlas de manera estratégica.

### Criterios de ubicación:

- **Cercanía al transporte público:** Las tiendas deben estar a menos de 100 metros de una parada de autobús, metro o tren.
- **Cobertura de zonas comerciales:** Cada parada de transporte público es el centro de una zona. La tienda debe cubrir varias zonas para llegar a más clientes.
- **Radio de cobertura:** El número de paradas que una tienda puede "alcanzar" (t) varía según la ciudad. Por ejemplo, en Caracas, una tienda puede cubrir hasta 3 paradas, mientras que en Bogotá, puede cubrir hasta 10.

### Funcionalidades del programa:

- **Cargar datos:** El programa debe poder leer archivos con información sobre las paradas de transporte público de diferentes ciudades.
- **Seleccionar ubicaciones:** El usuario debe poder elegir las paradas donde se construirán las tiendas.
- **Calcular cobertura:** El programa debe mostrar qué zonas de la ciudad están cubiertas por las tiendas, según el radio de cobertura y la ubicación de las paradas.
- **Evaluar cobertura:** El objetivo es que las tiendas cubran toda la ciudad, especialmente las zonas cercanas a las rutas de transporte público.

*Todo parte de la elección de un archivo JSON donde se almacenan toda la información acerca de la red de metro a usar para el problema.*

## Descripción de los Diagramas de Clases

---

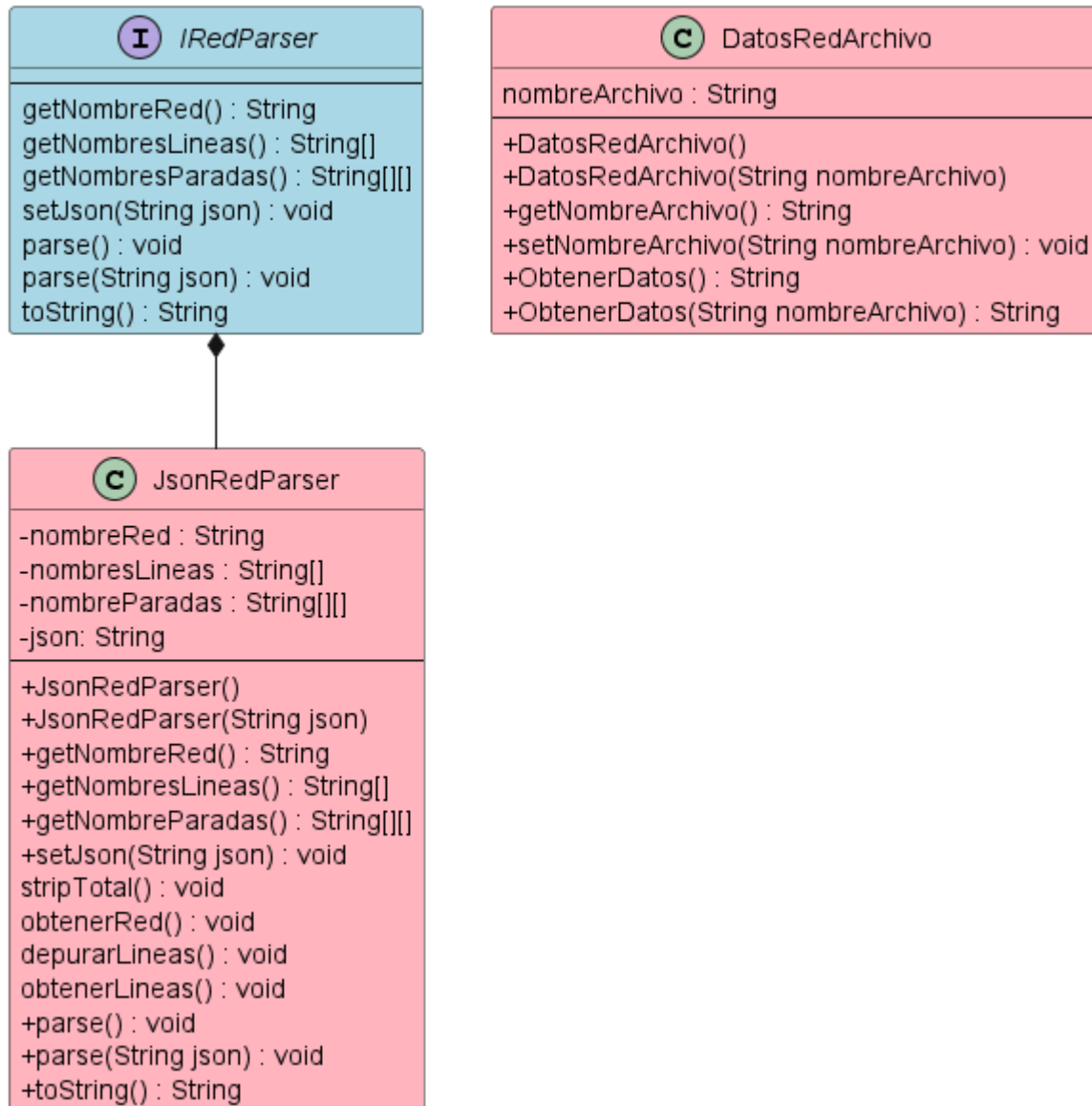
### Diagrama del Json

Existen 2 clases y una interfaz que se encargan de la extracción de los datos del archivo Json que se proporcione por el usuario:

- **IRedParser:** Interfaz de la clase JsonRedParser.
- **JsonRedParser:** Clase que obtiene la información de texto de una red de transporte de un string de texto con formato JSON.

- **DatosRedArchivo:** Clase que obtiene el string del archivo.

**Diagrama de Clases Proyecto #1 (Json)**



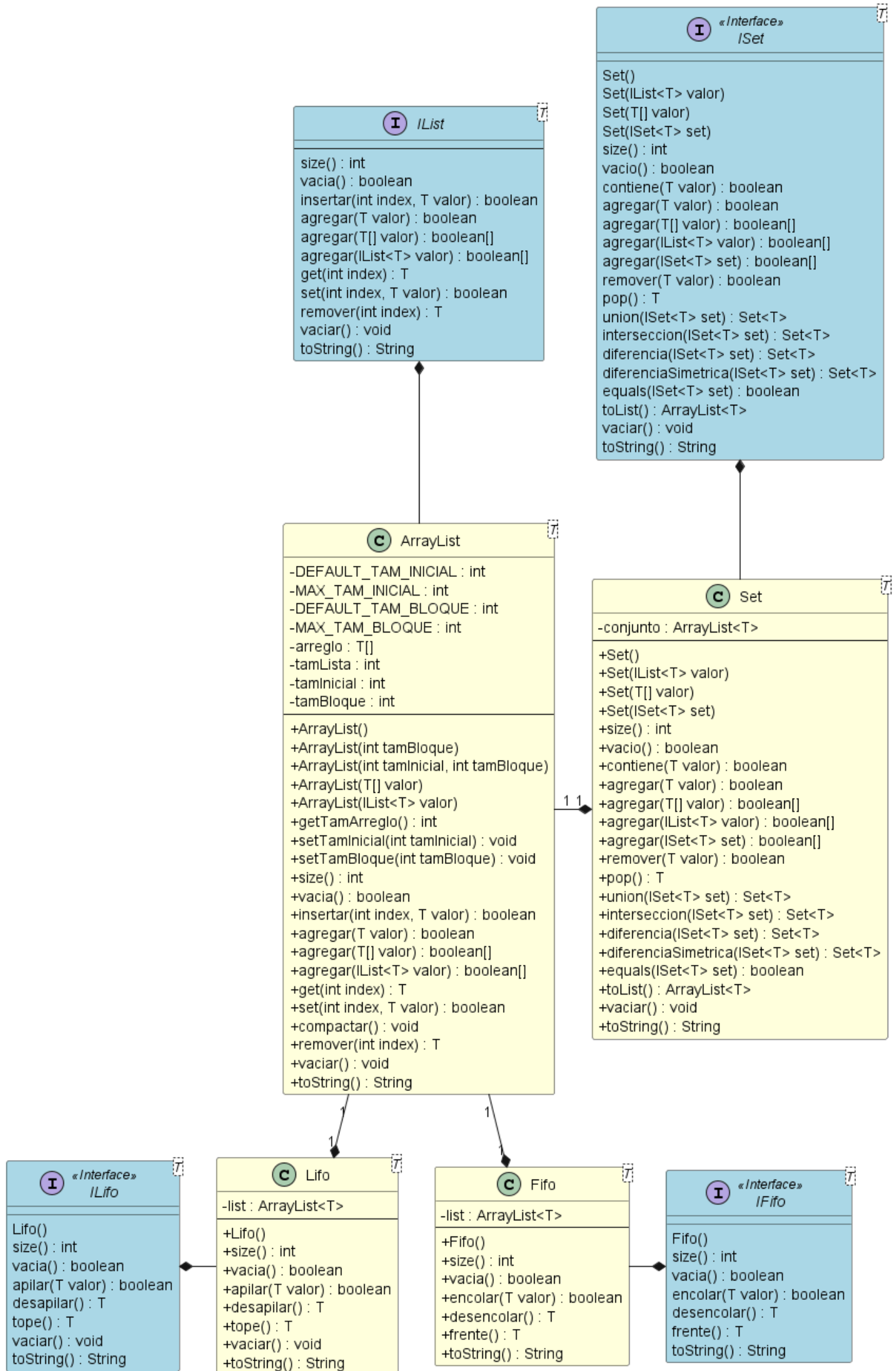
## Diagrama Estructuras Básicas

Existen diversas clases e interfaces de diversas estructuras de datos que se utilizan a lo largo del proyecto, estas son:

- **ICollection:** Interfaz de la clase ArrayList.
- **ISet:** Interfaz de la clase Set.
- **ILifo:** Interfaz de la clase Lifo.
- **IFifo:** Interfaz de la clase Fifo.
- **ArrayList:** Clase que implementa una lista de elementos de tipo genérico usando arreglos.
- **Set:** Clase que implementa un conjunto de elementos de tipo genérico usando un ArrayList.

- **Lifo:** Clase LifoLL (Linked List Lifo) para implementación de pilas (Lifo).
- **Fifo:** Clase FifoLL (Linked List Fifo) para implementación de colas (Fifo).

## Diagrama de Clases Proyecto #1 (Estructuras Básicas)

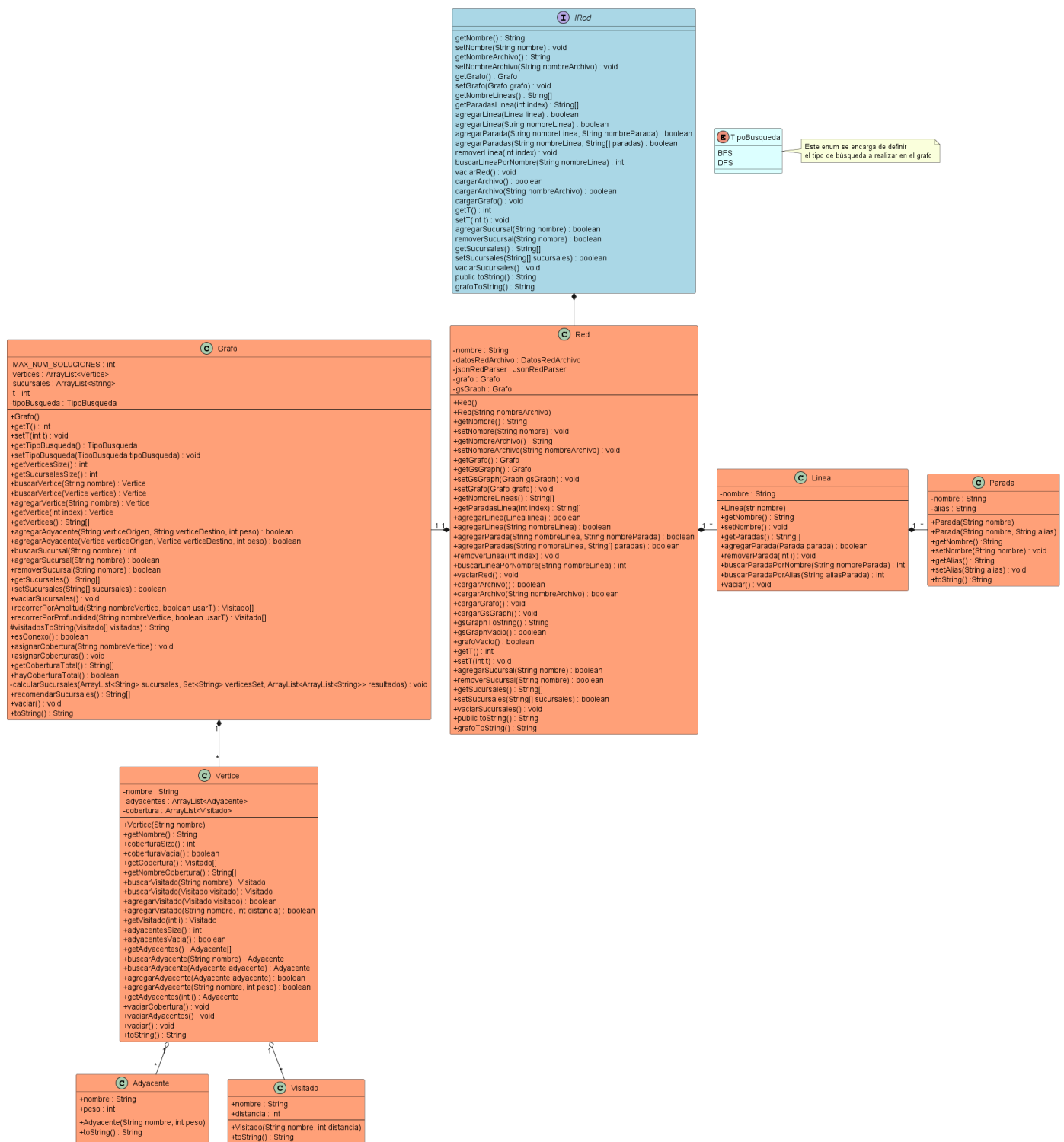


# Diagrama Grafo y Redes

Existen diversas clases e interfaces que se encargan de la creación de toda la estructura de la red y el grafo en general:

- **IRed:** Interfaz de la clase Red.
- **Red:** Clase que representa una red de transporte la cual es implementada con ArrayList.
- **Linea:** Representa una linea de una red de transporte la cual es implementada con ArrayList.
- **Parada:** Representa una parada de una red de transporte.
- **Grafo:** Clase para representar un grafo. En el caso del proyecto, los vértices son paradas o estaciones de una linea de transporte.
- **Vertice:** Clase para representar un vertice en un grafo. En el caso del proyecto, los vértices son paradas o estaciones de una linea de transporte
- **Adyacente:** Clase que contiene el nombre del vertice adyacente y el peso (es parte de un grafo no dirigido).
- **Visitado:** Clase para representar un vertice visitado en un grafo. Sirve tanto para recorrer el grafo, como para determinar cuales son los vertices (paradas) que están dentro de un area de cobertura de una sucursal.
- **TipoBusqueda:** Clase enumerada que define el tipo de búsqueda a realizar, por DFS (por profundidad) o por BFS (por amplitud).

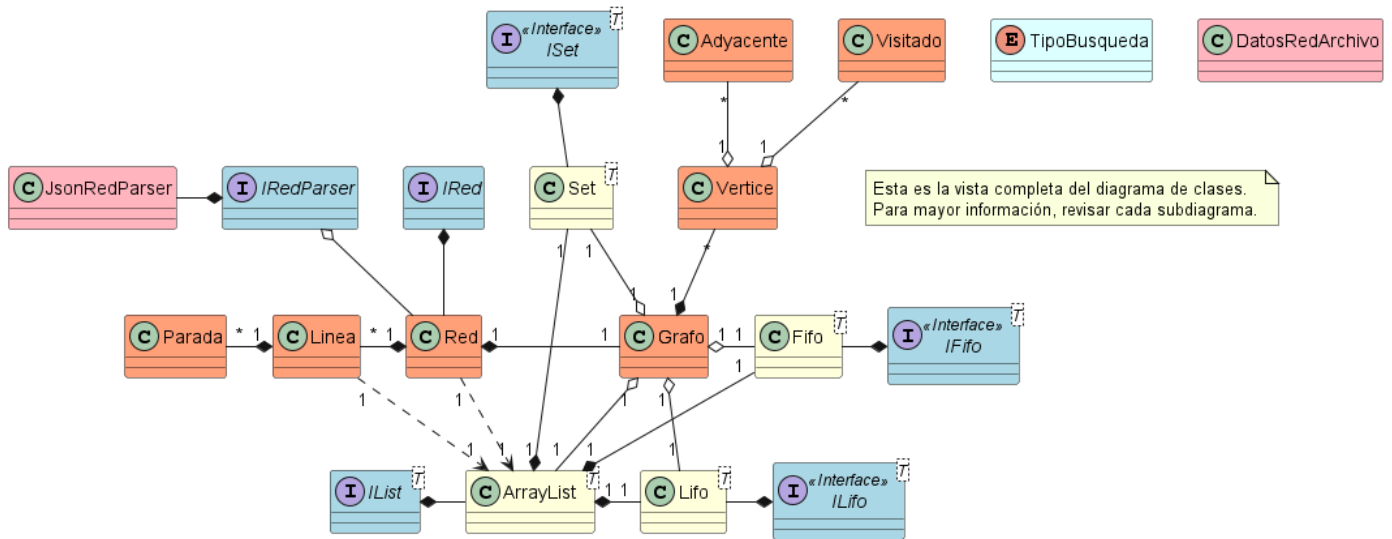
Diagrama de Clases Proyecto #1 (Grafos y Redes)



# Diagrama Completo

En este diagrama se muestran las relaciones entre todas las clases e interfaces del proyecto las cuales están simplificadas con tan solo el nombre de la clase o interfaz para visualizar de mejor manera todo el proyecto:

Diagrama de Clases Proyecto #1 (Completo)



# Uso/Ejecución del proyecto

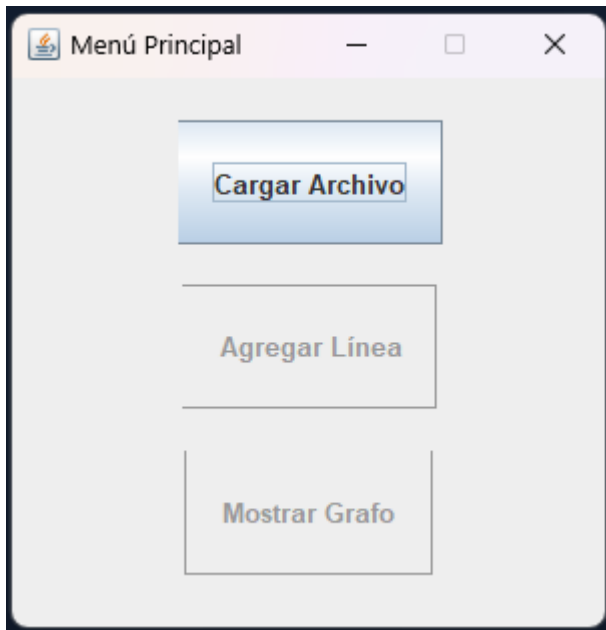
Se explicarán las partes más importantes para entender cómo funciona la ejecución del proyecto mediante las diferentes interfaces gráficas que lo conforman:

## 1. Menú Principal

Al correr el programa, la primera ventana que se nos muestra es la ventana del menú principal el cual posee 3 opciones:

- a) Cargar Archivo
- b) Agregar Línea
- c) Mostrar Grafo

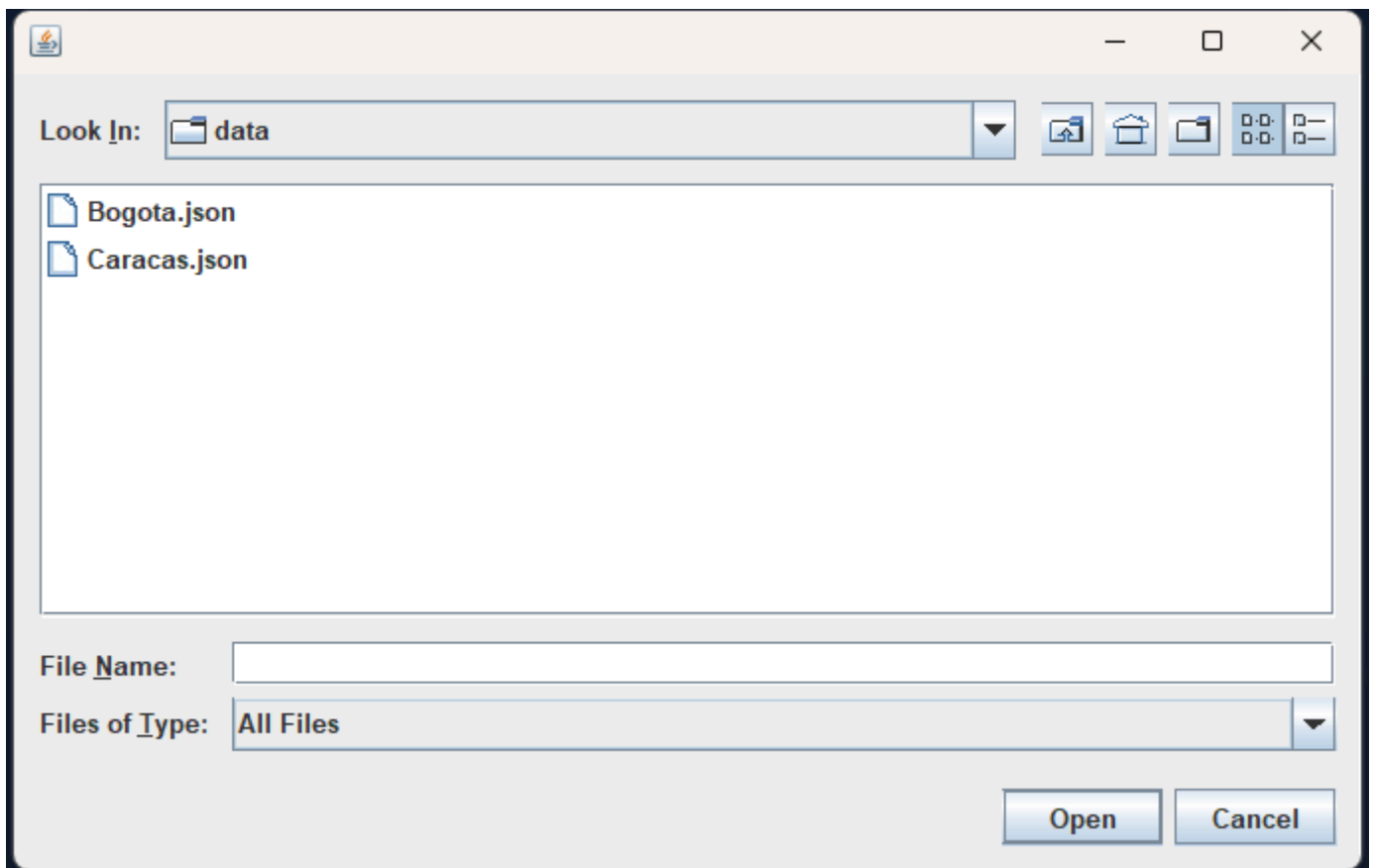




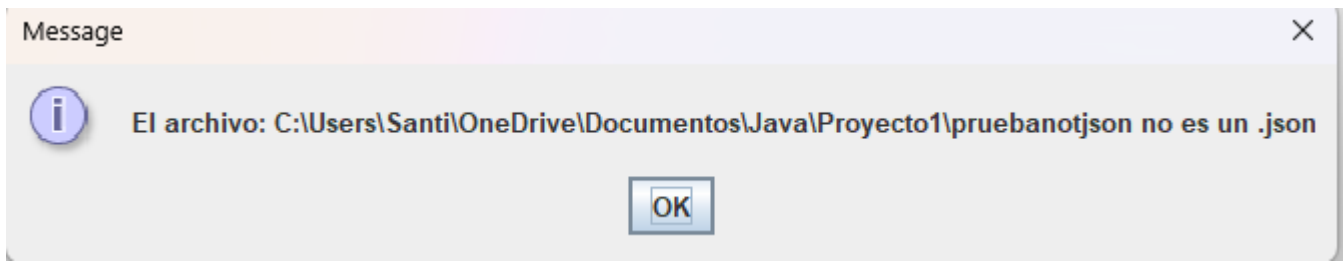
Al iniciar el programa, solamente aparece disponible la opción de Cargar Archivo porque, hasta que el usuario no seleccione un archivo JSON no se pueden usar las opciones de Agregar Línea ni la de Mostrar Grafo

#### a) Cargar Archivo

Al abrir esta ventana, se nos mostrará una ventana de selección de archivo como esta:



*El selector de archivo solamente acepta archivos tipo JSON, cualquier otro tipo de archivo que elijas hará que se te muestre un mensaje como el siguiente:*



## B) Agregar Línea

Al haber ya seleccionado un archivo tipo JSON, podemos seleccionar la opción Agregar Línea y se nos mostrará la siguiente ventana:

Nombre Línea:

Paradas:

*Nombres de Paradas: separadas por coma o por una nueva línea.*  
*Transferencias en formato Parada1:Parada2*  
*Los nombres son sencibles a Mayúsculas y Minúsculas.*

Agregar

Información de la línea a agregar:

Aceptar Cancelar

Vamos a ir parte por parte:

i) Nombre Línea

Es un espacio de texto que te permite poner el nombre de la línea que se quiere agregar o crear.

**Nombre Línea:**

***Se puede poner cualquier tipo de nombre, siempre y cuando no sea igual a ningún nombre de otra línea ya existente en la red. Cabe resaltar la presencia del CaseSensitive.***

## ii) Paradas

Es un espacio de texto que le permite al usuario introducir o crear las paradas que van a existir dentro de la nueva línea. Existen diversas maneras de agregar nuevas paradas:

- Escribiendo los nombres de las paradas separados por una línea de texto
- Escribiendo los nombres de las paradas separados por una coma ","
- Si se quiere crear una transición, es decir 2 vértices adyacentes con peso 0, se escribe "nombreParada1:nombreParada2"

**Paradas:**

***Nombres de Paradas: separadas por coma o por una nueva línea.***  
***Transferencias en formato Parada1:Parada2***  
***Los nombres son sencibles a Mayúsculas y Minúsculas.***

Agregar

***Cabe resaltar la existencia de CaseSensitive.***

Ejemplo:

**Nombre Línea:**

Línea Prueba

**Paradas:**

Parada1  
Parada2, Parada3  
Parada4:Parada5

*Nombres de Paradas: separadas por coma o por una nueva línea.*  
*Transferencias en formato Parada1:Parada2*  
*Los nombres son sencibles a Mayúsculas y Minúsculas.*

**Agregar**

iii) Información de la línea a agregar

Al haber terminado de añadir las paradas de la nueva línea en la parte (ii), se muestra un panel de texto que muestra el resultado e información de la línea a agregar con su respectivo nombre y el nombre de cada una de las paradas que contiene dicha línea.

**Información de la línea a agregar:**

**Aceptar** **Cancelar**

Ejemplo:

**Nombre Línea:**

Línea Prueba

**Paradas:**

Parada1  
Parada2, Parada3  
Parada4:Parada5

*Nombres de Paradas: separadas por coma o por una nueva línea.*  
*Transferencias en formato Parada1:Parada2*  
*Los nombres son sencibles a Mayúsculas y Minúsculas.*

**Agregar**

**Información de la línea a agregar:**

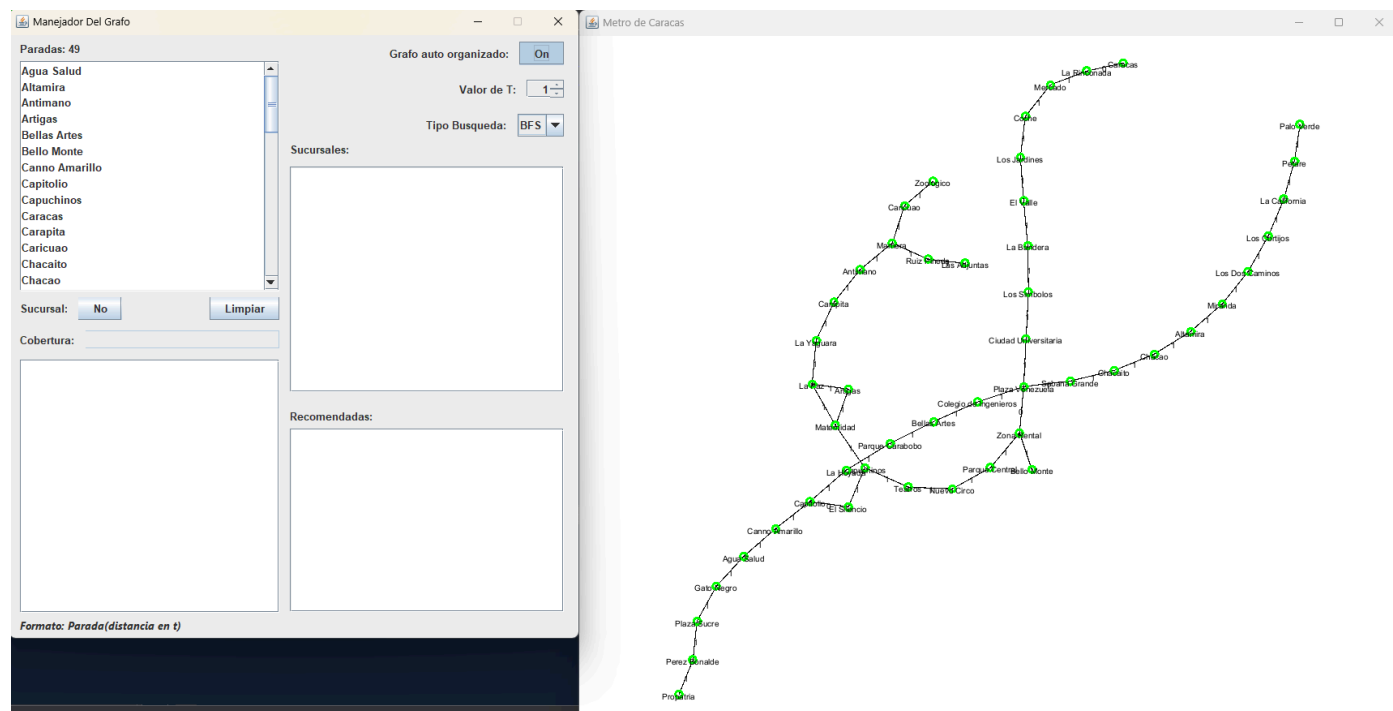
Línea: Línea Prueba  
Paradas:  
Parada1  
Parada2  
Parada3  
Parada4:Parada5

**Aceptar** **Cancelar**

Al terminar todo el proceso, al darle al botón de aceptar se agregará la línea al grafo de la red cargada.

## c) Mostrar Grafo

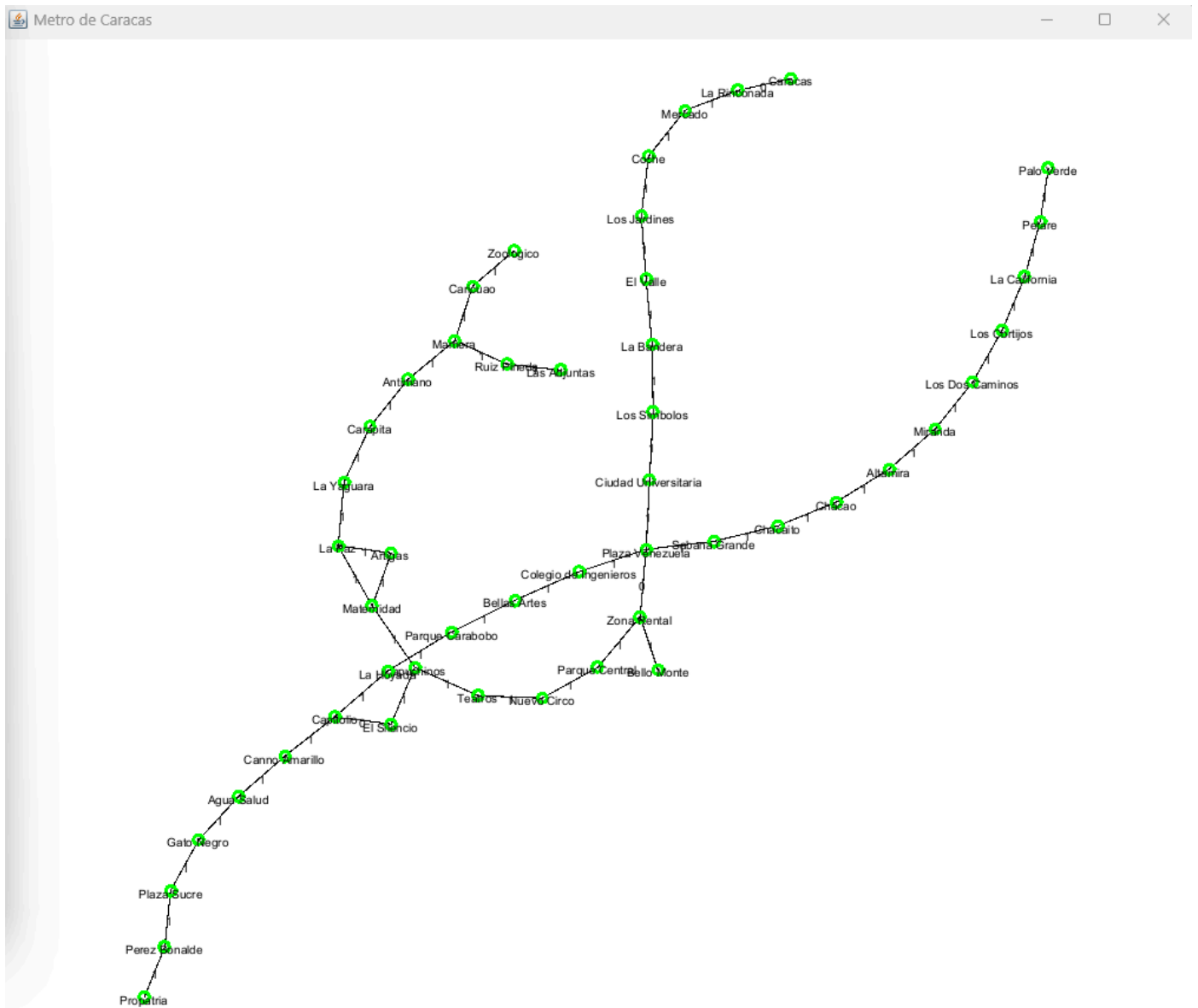
Al haber seleccionado un archivo JSON, si seleccionamos esta opción se nos mostrarán las siguientes ventanas:



La ventana de la izquierda la llamaremos "Manejador del Grafo" y la de la derecha la llamaremos "Vista del Grafo". Vayamos parte por parte:

### i) Vista del Grafo

Es una ventana en la cual se nos muestra la representación gráfica del grafo de la red en cuestión.



Aquí se nos muestran todas las paradas como puntos de distintos tamaños y colores con sus respectivos nombres y adyacencias con otras paradas, además del peso entre cada adyacencia. A lo largo de la ejecución, este grafo puede ir variando dependiendo de las acciones que hagamos en la otra ventana (Manejador del Grafo), luego hablaremos más a fondo de esas acciones.

Cabe aclarar que el grafo es interactivo y posee sus propias físicas y movimientos. Por ejemplo, si el usuario desea, puede mover cualquier parada y toda la red se va a mover junto a ella. Pero si no queremos esto y solamente queremos mover la parada seleccionada, podemos activar o desactivar el movimiento de la red con el siguiente botón el cual se encuentra en la otra ventana (Manejador del Grafo):

Grafo auto organizado:

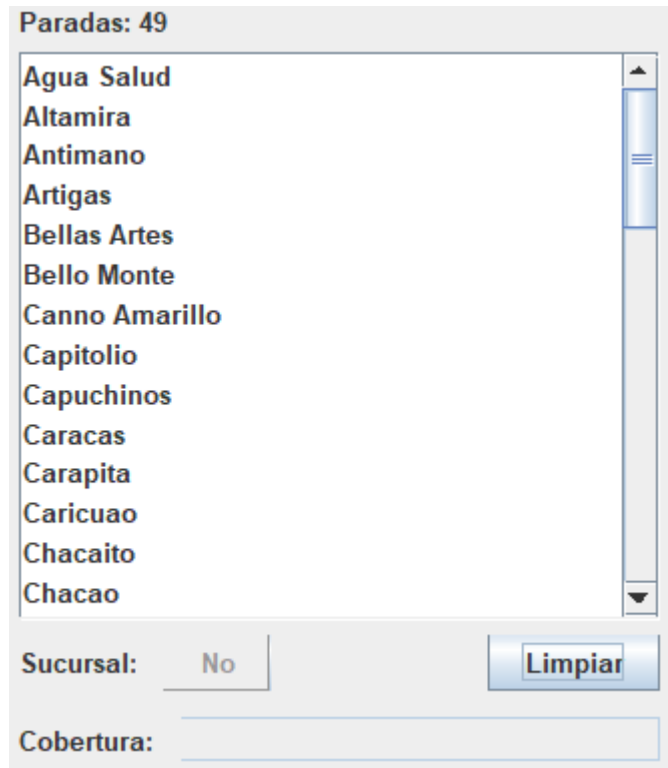
On

Es un botón que contiene 2 opciones entre "On" (que la red se mueve en conjunto) o "Off" (que la red se queda estática).



## ii) Lista de Paradas en la Línea

Es una lista scrolleable donde se muestran todas las paradas que contiene la red en cuestión en orden alfabético. Arriba del todo se muestra la cantidad de paradas totales que contiene la red, en este caso, al haber cargado el archivo JSON del Metro de Caracas se nos señala que existen 49 paradas alrededor de toda la red o en el metro.



Paradas: 49

- Agua Salud
- Altamira
- Antimano
- Artigas
- Bellas Artes
- Bello Monte
- Canno Amarillo
- Capitolio
- Capuchinos
- Caracas
- Carapita
- Caricuao
- Chacaito
- Chacao

Sucursal:

Cobertura:

Al seleccionar una de las paradas, ocurrirán varios eventos en ambas ventanas. Pronto explicaremos lo que sucede a mayor profundidad.

Una vez seleccionada la parada, existen 2 botones que podemos utilizar. El primero de ellos es el botón de Sucursal, el cual intercambia entre las opciones "Sí" o "No" y el cual decide si una parada es una sucursal o no. El botón de Limpiar simplemente deselecta la opción o parada previamente elegida y lo deja en blanco, como si nunca hubieras elegido una. Al haber elegido alguna parada, al lado del texto que dice Cobertura se nos mostrará la el nombre de la parada seleccionada.

Ejemplo:

Paradas: 49

Agua Salud
Altamira
Antimano
Artigas
Bellas Artes
Bello Monte
Canno Amarillo
Capitolio
Capuchinos
<b>Caracas</b>
Carapita
Caricuao
Chacaito
Chacao

Sucursal:

Cobertura:


### iii) Cobertura de la Parada

Al haber seleccionado alguna parada se nos mostrará algo como esto

Cobertura:


**Formato: Parada(distancia en t)**

Es una ventana que nos muestra la información de la cobertura que posee la parada seleccionada. El tema de la cobertura depende totalmente del valor "t" el cual, mientras mayor sea el número, mayor será la cobertura de la parada. Este valor se puede cambiar en la opción que se encuentra en esta misma ventana (Manejador del Grafo):

Valor de T:  

Es un botón que tiene opciones de subir o bajar el valor de la variable de 1 en 1.

Sumado a todo esto, en relación al tema de la cobertura está el tema del tipo de búsqueda el cual puede variar entre DFS (por profundidad) y BFS (por amplitud). Esta opción se puede cambiar desde el siguiente botón el cual se encuentra en esta misma ventana (Manejador del Grafo):

Tipo Búsqueda:  

Es un botón con 2 opciones desplegables entre DFS y BFS.

Ejemplo: Supongamos que elegimos la parada Caracas con  $t=3$  y BFS como tipo de búsqueda:



**Sucursales:**

Ejemplo: Supongamos que elegimos la parada de Caracas como sucursal:

**Paradas: 49**

Agua Salud  
Altamira  
Antimano  
Artigas  
Bellas Artes  
Bello Monte  
Canno Amarillo  
Capitolio  
Capuchinos  
**Caracas**  
Carapita  
Caricuao  
Chacaito  
Chacao

Sucursal: ☐ Si

Cobertura:

Limpiar

**Grafo auto organizado:**

**Valor de T:**

**Tipo Búsqueda:**

**Sucursales:**

Caracas

En la ventana se nos mostrará que efectivamente, la parada Caracas es una sucursal.

#### v) Lista de Sucursales Recomendadas

En esta ventana se mostrará un texto que posee una lista de sucursales que son recomendadas por el programa para poder cubrir todos los vértices o paradas de la red, es decir, tener una cobertura total.

### Recomendadas:

Ejemplo: Supongamos que elegimos la parada Caracas con  $t=3$

Paradas: 49

Agua Salud  
Altamira  
Antimano  
Artigas  
Bellas Artes  
Bello Monte  
Canno Amarillo  
Capitolio  
Capuchinos  
**Caracas**  
Carapita  
Caricuao  
Chacaito  
Chacao

Sucursal: ☒ Si

Cobertura: **Caracas**

Caracas(0)  
La Rinconada(0)  
Mercado(1)  
Coche(2)  
Los Jardines(3)

Grafo auto organizado:

Valor de T:

Tipo Búsqueda:  ▼

### Sucursales:

Caracas

### Recomendadas:

Antimano  
Artigas  
Canno Amarillo  
Ciudad Universitaria  
Los Dos Caminos  
Palo Verde  
Propatria

En la ventana se nos muestran varias paradas como Antimano, Artigas, Canno Amarillo, etc... Cada una de ellas son paradas que, si son seleccionadas como sucursales (con  $t = 3$ ), cubrirían toda la cobertura de la red. Mientras mayor sea  $t$ , menor será el número de recomendaciones que hará el programa o distintas sucursales serán recomendadas. Vea este ejemplo cuando se selecciona la parada Caracas con  $t = 5$ :

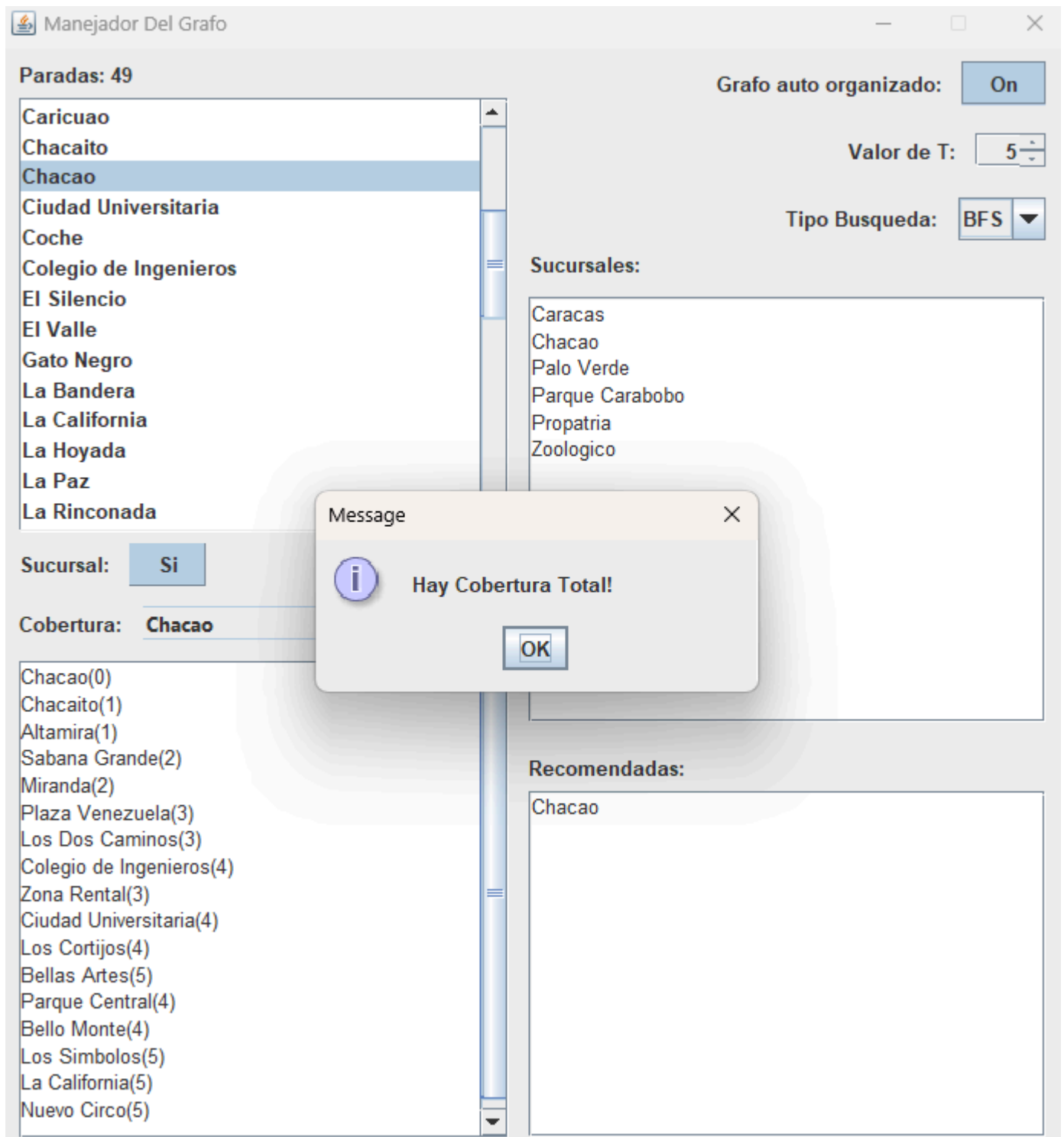
The screenshot shows the 'Manejador Del Grafo' application window. On the left, a list of 49 stops is shown, with 'Caracas' selected. Below this list, the 'Sucursal' is set to 'Si' and 'Cobertura' is 'Caracas'. The 'Cobertura' section lists the following stops with their respective counts: Caracas(0), La Rinconada(0), Mercado(1), Coche(2), Los Jardines(3), El Valle(4), and La Bandera(5). On the right, the 'Grafo auto organizado' is set to 'On', 'Valor de T' is 5, and 'Tipo Búsqueda' is 'BFS'. The 'Sucursales' section shows 'Caracas' as the selected node. The 'Recomendadas' section lists the following recommended stops: Chacao, Palo Verde, Parque Carabobo, Propatria, and Zoologico.

Paradas	Cobertura
Caracas	0
La Rinconada	0
Mercado	1
Coche	2
Los Jardines	3
El Valle	4
La Bandera	5

Recomendadas
Chacao
Palo Verde
Parque Carabobo
Propatria
Zoologico

Se observa que no solo las paradas cambiaron sino que también se recomendaron menos paradas ya que, debido a un mayor valor de  $t$  la cantidad de sucursales necesarias para cubrir toda la cobertura de la red es menor.

Ahora, supogamos que seleccionamos todas las sucursales que se nos recomiendan. Si hacemos eso, se nos mostrará el siguiente mensaje:



Al darle click a "ok", el usuario observará que la lista de sucursales recomendadas ahora posee un texto que dice que existe una cobertura total:



Manejador Del Grafo

Paradas: 49

- Caricuao
- Chacaito
- Chacao
- Ciudad Universitaria
- Coche
- Colegio de Ingenieros
- El Silencio
- El Valle
- Gato Negro
- La Bandera
- La California
- La Hoyada
- La Paz
- La Rinconada

Sucursal:

Cobertura:

Grafo auto organizado:

Valor de T:

Tipo Busqueda:

Sucursales:

- Caracas
- Chacao
- Palo Verde
- Parque Carabobo
- Propatria
- Zoologico

Recomendadas:

Hay Cobertura Total!

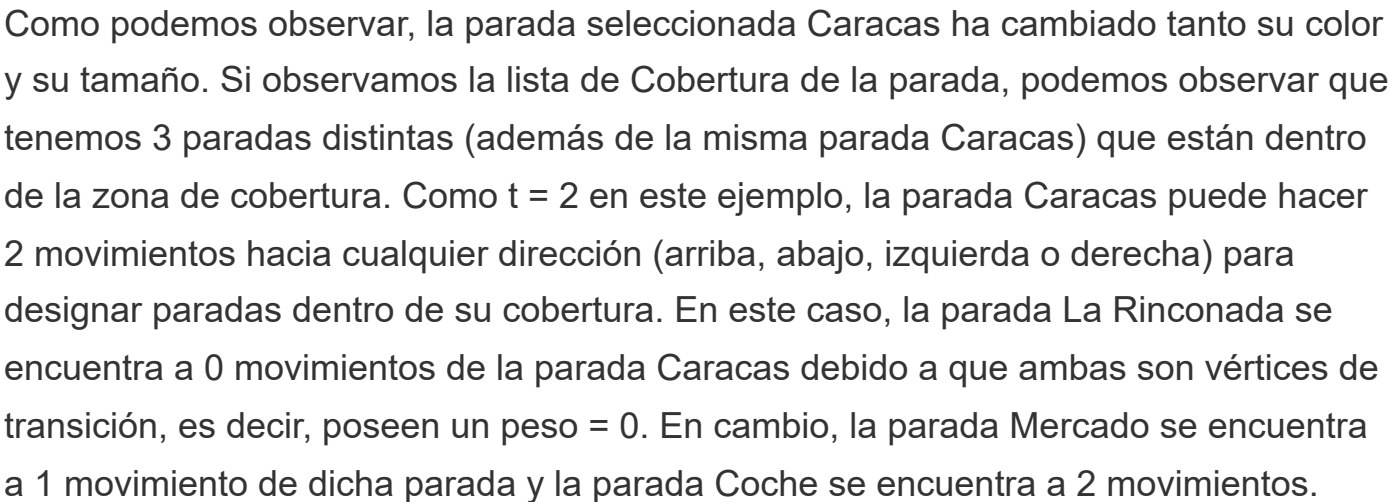
***Cabe aclarar que las sucursales que se seleccionen se pueden deseleccionar volviendo a seleccionarlas en la lista de paradas y dándole a "no" en Sucursal. Y además se puede ir cambiando el valor de  $t$  en medio de la ejecución del programa.***

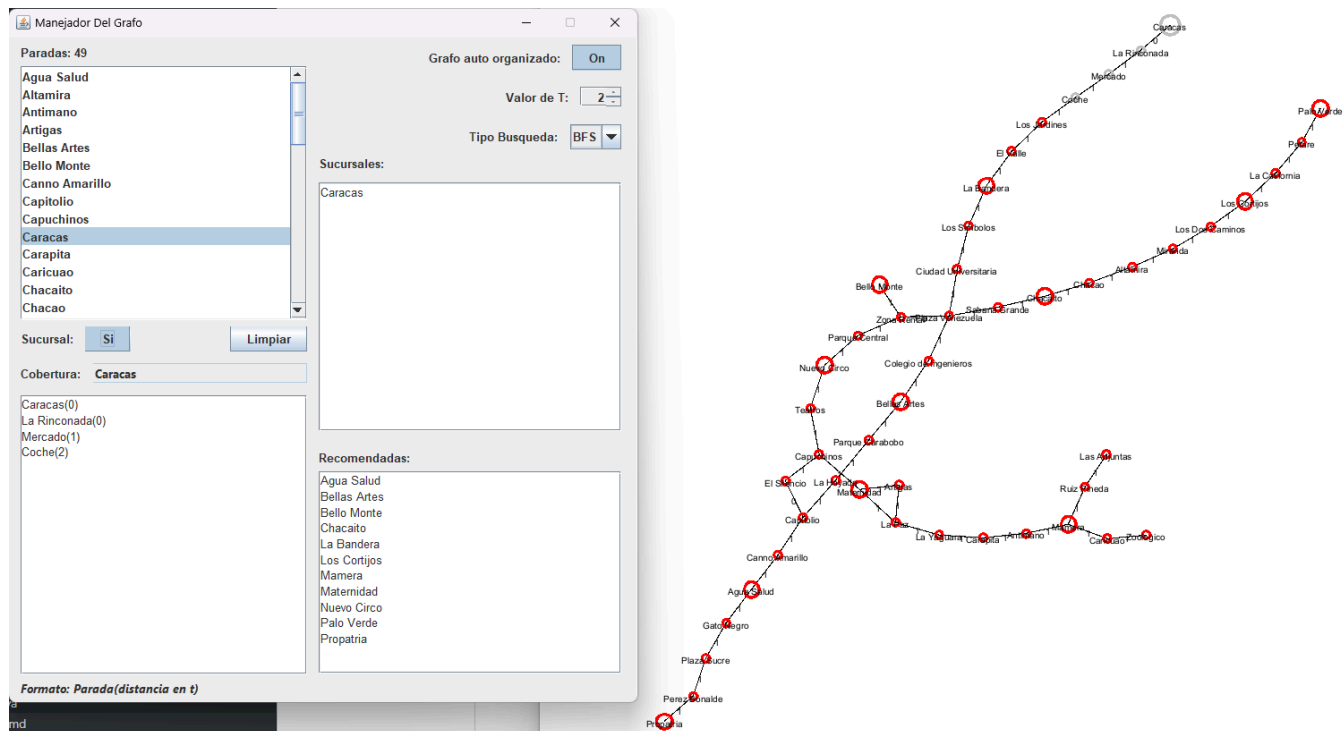
#### vi) Comportamiento del Grafo según las acciones del manejador

Ya que cubrimos todas las áreas de la ventana Manejador del Grafo, ahora veremos las acciones que realiza la ventana de Visualización del Grafo cuando hacemos cambios en el manejador.

\*) Selección de una parada

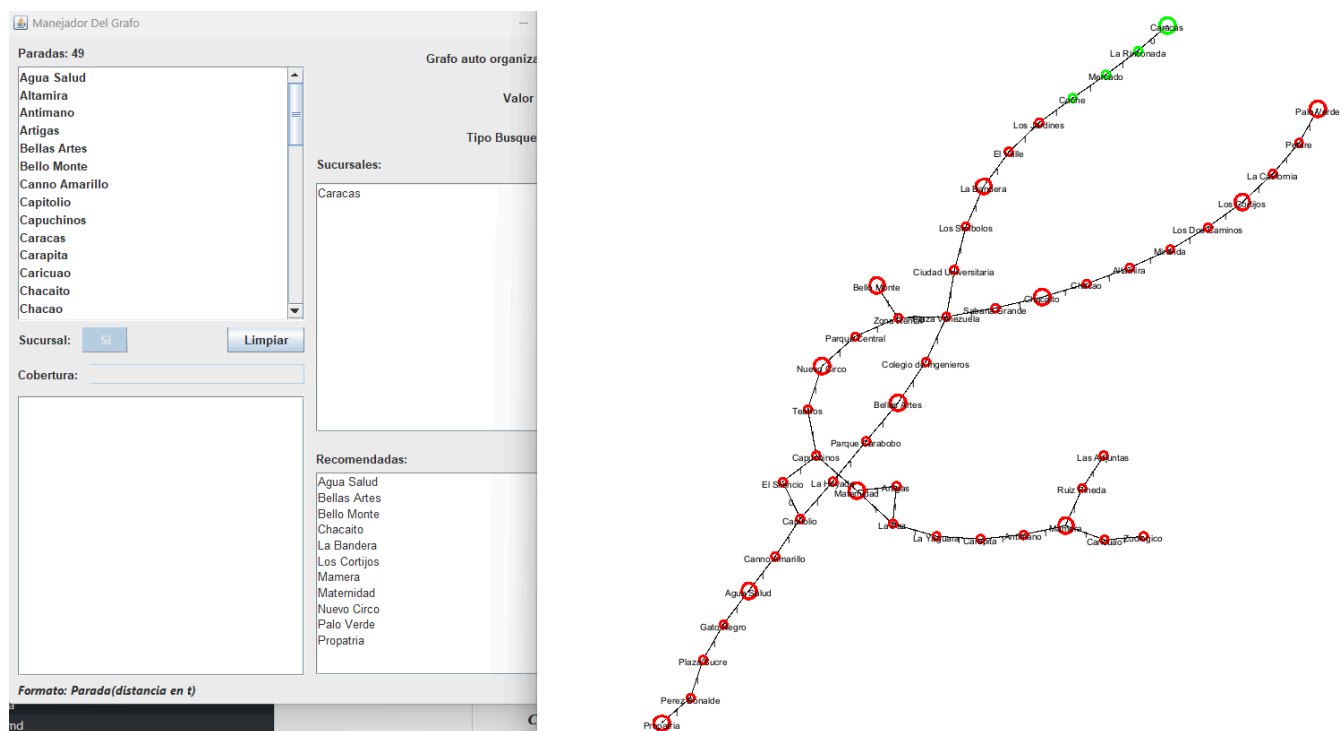
Ejemplo: Supongamos que elegimos la parada Caracas con  $t=2$



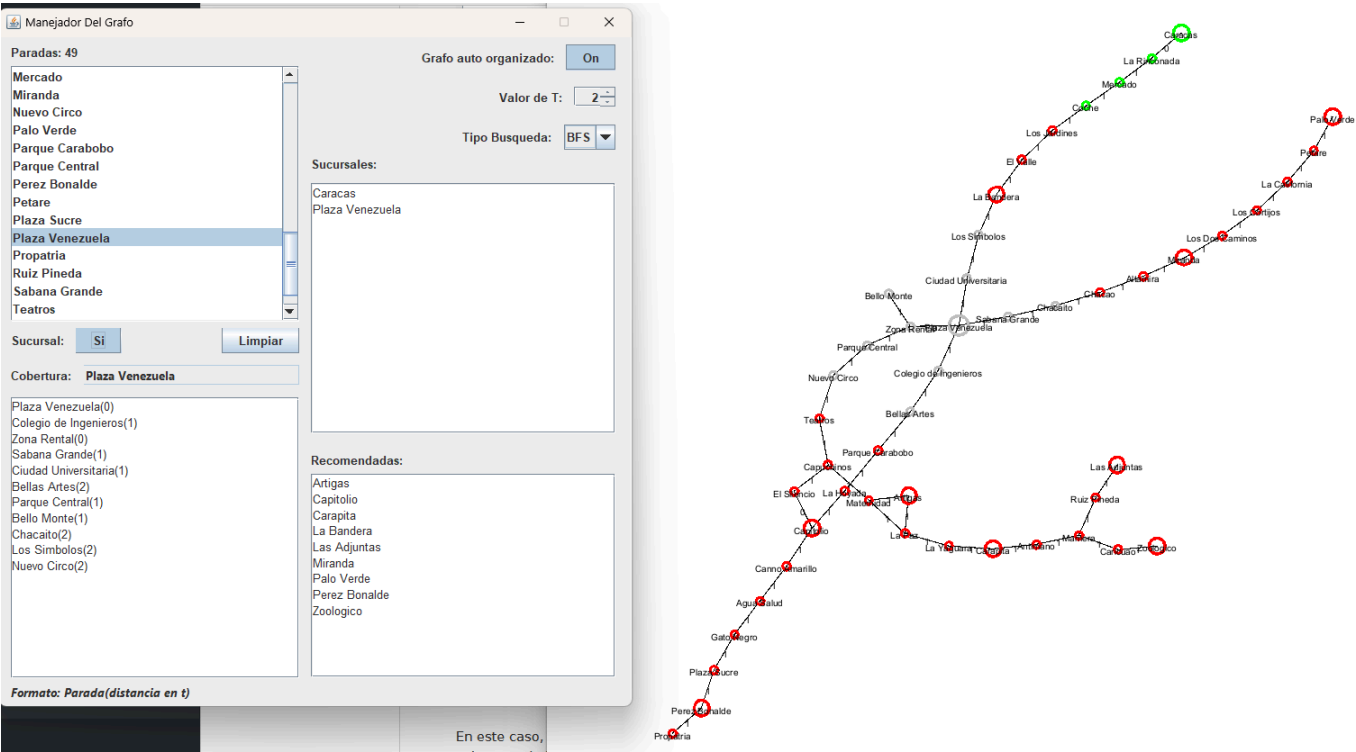


Como podemos observar, la paradas que no se encuentran en la zona de cobertura de Caracas se muestran en rojo. Podemos observar que en la lista de Sucursales Recomendadas se encuentran paradas como Agua Salud, Bellas Artes, Bello Monte, etc... Si observamos el grafo, podemos ver que dichas paradas no solo están en rojo sino que también poseen un mayor tamaño de su punto, indicando que son sucursales recomendadas para cubrir la cobertura de toda la red.

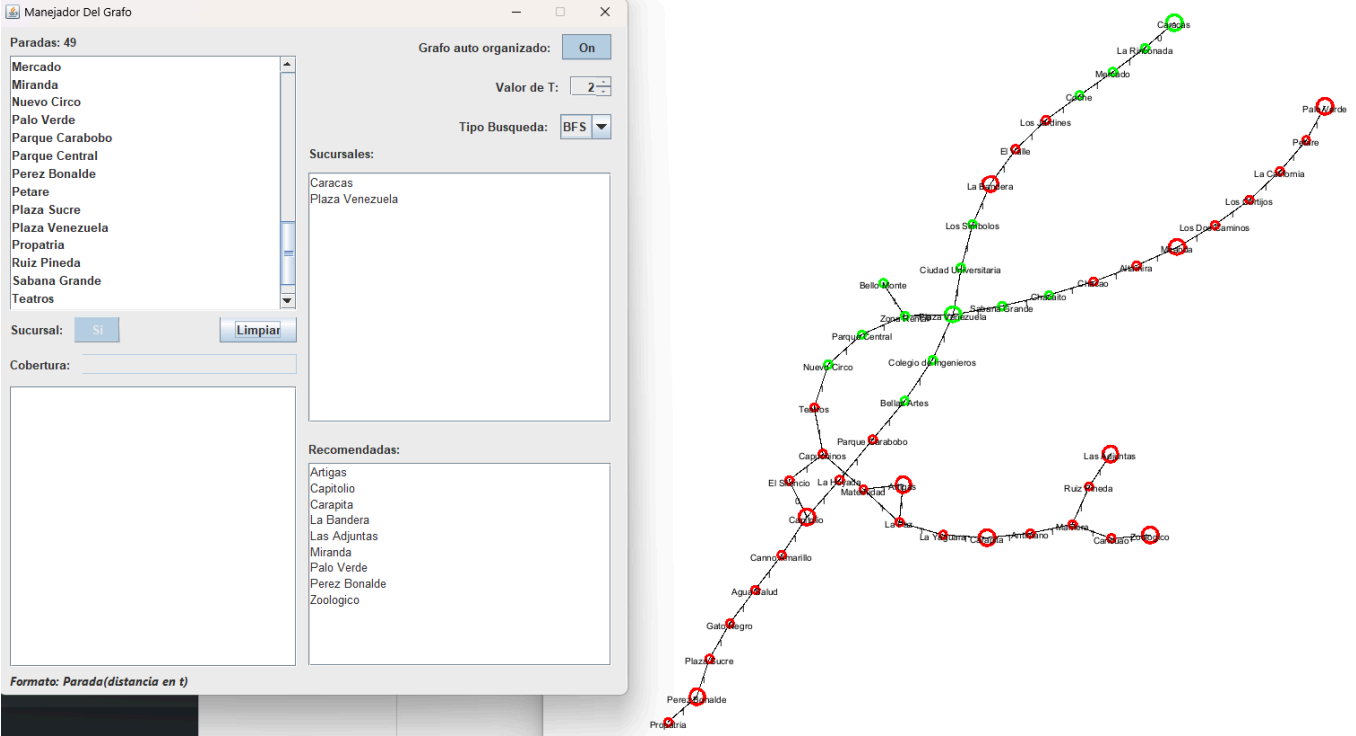
***Cabe resaltar que las paradas que se encuentran en la cobertura de Caracas siguen con un color gris porque está seleccionada la parada. Si cambiamos de parada o le damos a la opción Limpiar, el grafo cambiará su color a verde como se muestra a continuación:***



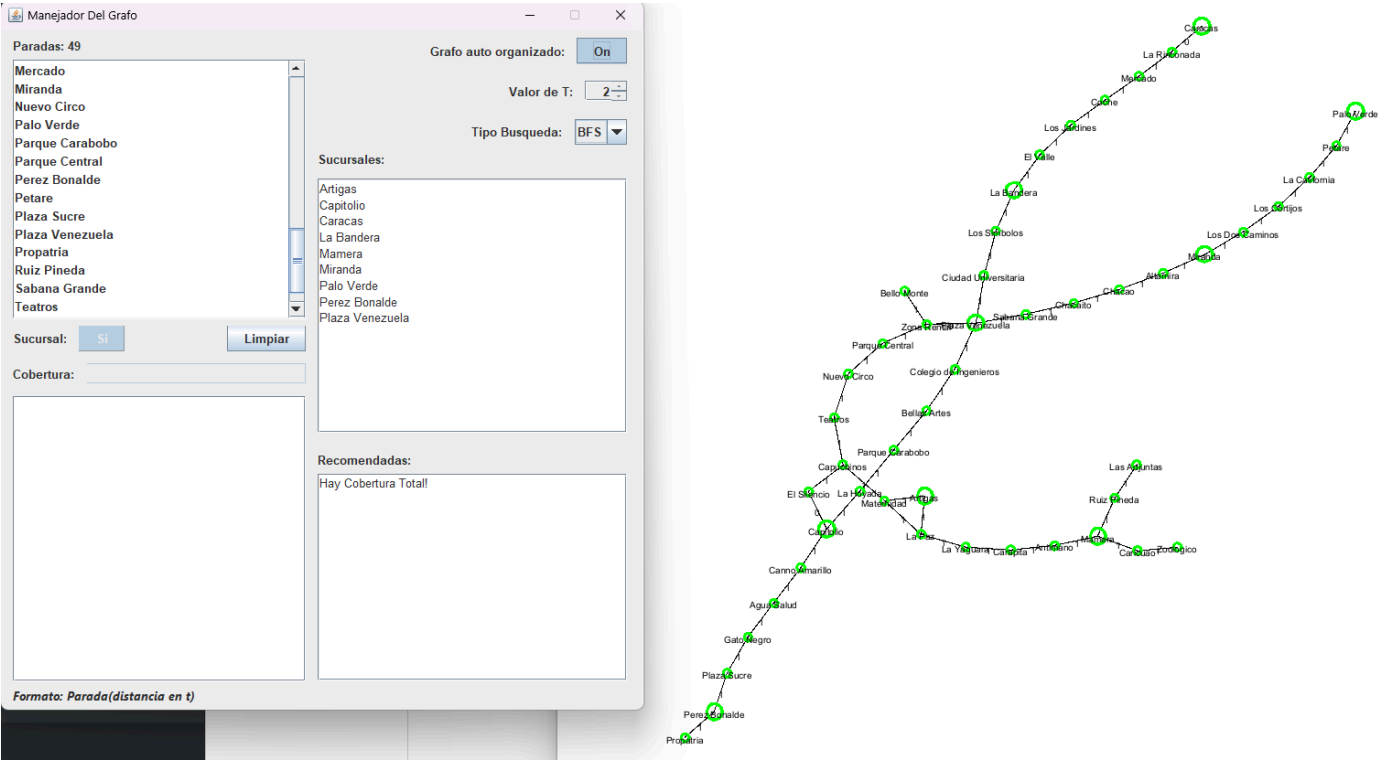
Ahora supongamos que seleccionamos otra parada. Seleccionaremos la parada Plaza Venezuela:



En este caso, Plaza Venezuela posee 10 paradas que se encuentran en su cobertura (si no la contamos a sí misma). Se puede observar que la cobertura de Plaza Venezuela se encuentra en gris porque está seleccionada, la cobertura de Caracas se encuentra en verde y las demás paradas se encuentran en rojo. Sumado en eso, podemos observar un cambio en las sucursales recomendadas porque elegimos una parada que no se encontraba entre las recomendaciones. Al Limpiar la selección, el grafo nos queda de la siguiente manera:



Si seleccionamos todas las sucursales recomendadas restantes, sucederá lo siguiente:




Todas las paradas se encontrarán en verde, lo que significa que la cobertura total de la red está completa.

**\*\* ) Agregar una línea**

Cuando se crea una nueva línea con una lista de paradas, el Visualizador del Grafo mostrará distintos resultados dependiendo de la línea y las paradas.

Ejemplo #1: Supongamos que creamos la siguiente línea:



—

□

×

**Nombre Línea:**

Línea Prueba 1

**Paradas:**

Parada1  
Parada 2  
Parada 3  
Parada 4  
Parada 5

*Nombres de Paradas: separadas por coma o por una nueva línea.*

*Transferencias en formato Parada1:Parada2*

*Los nombres son sencibles a Mayúsculas y Minúsculas.*

Agregar

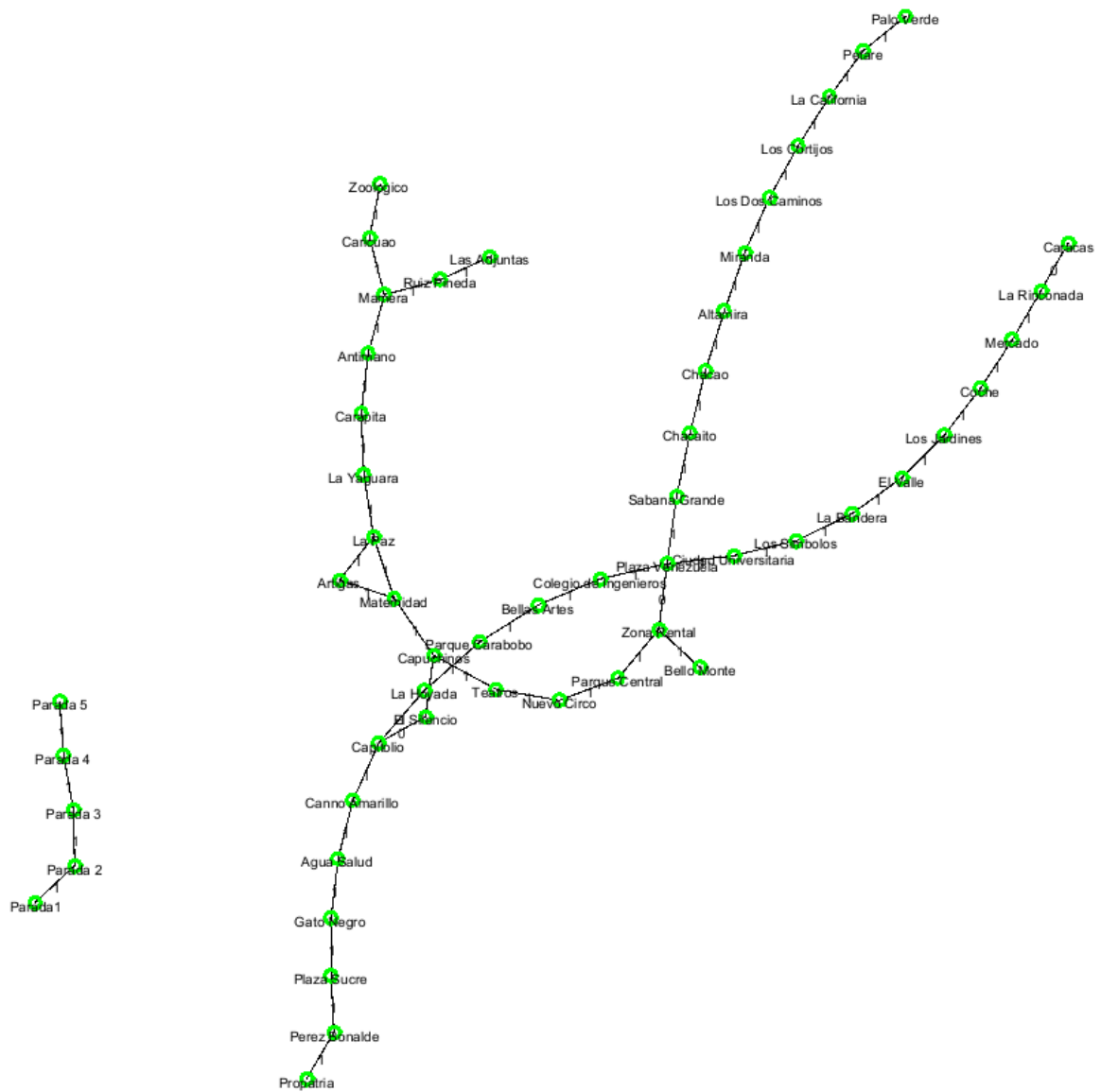
**Información de la línea a agregar:**

Línea: Línea Prueba 1  
Paradas:  
Parada1  
Parada 2  
Parada 3  
Parada 4  
Parada 5

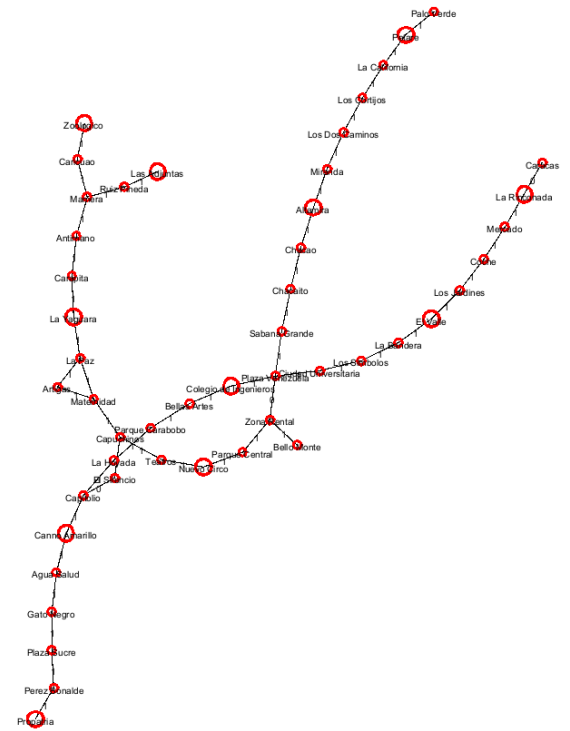
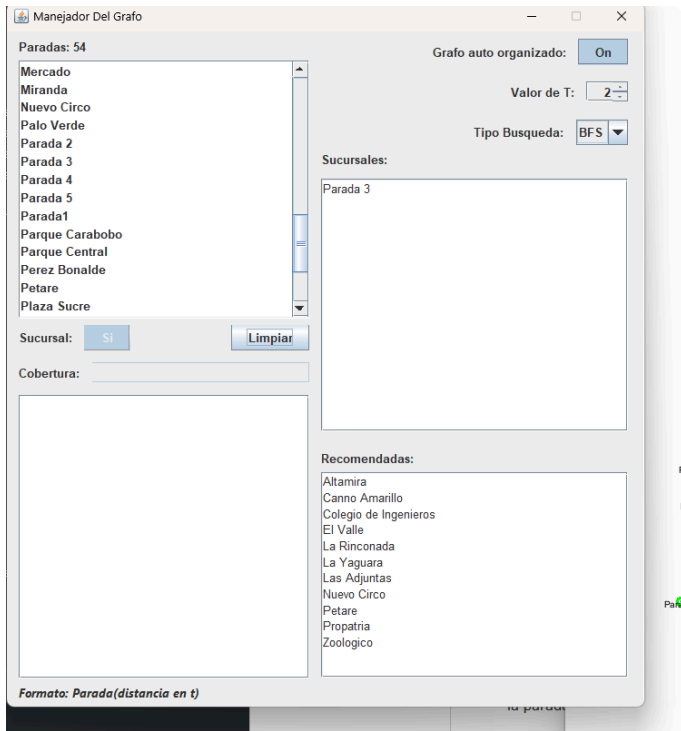
Aceptar

Cancelar

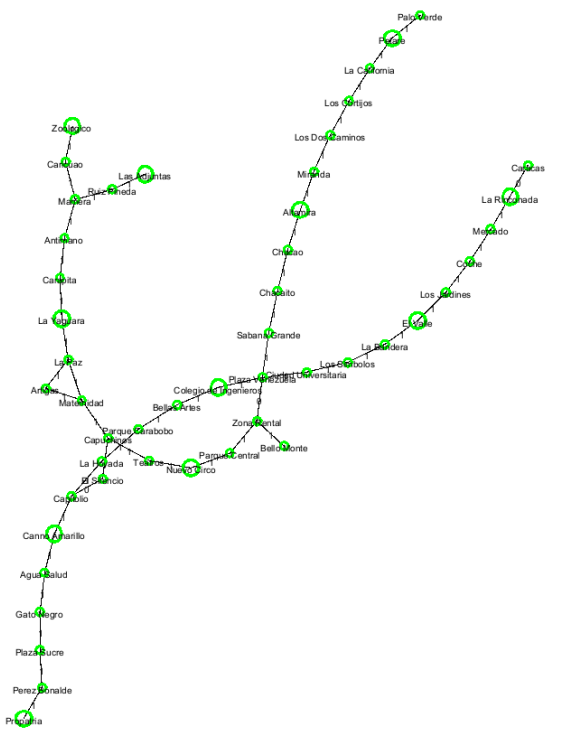
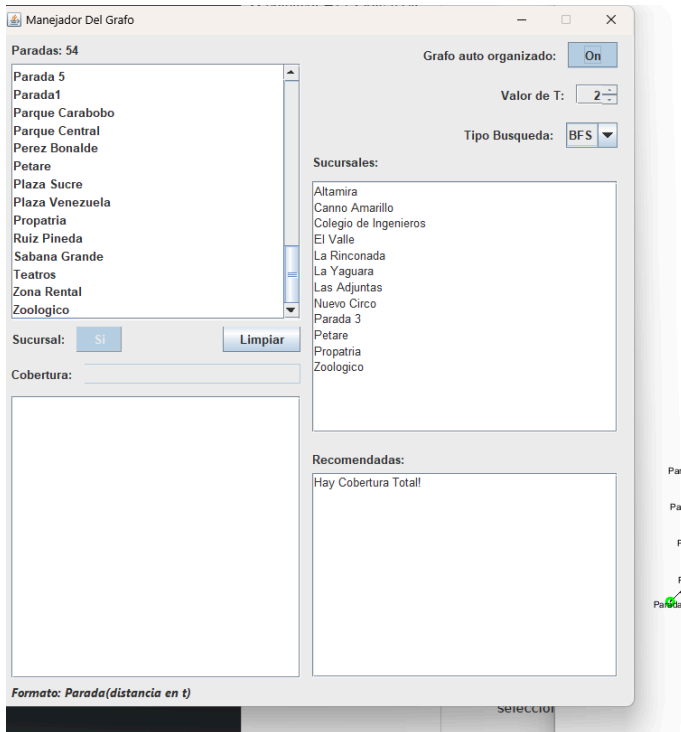
Al visualizar el grafo, se mostrará lo siguiente:



Como se puede observar, la línea que creamos se encuentra apartada de la red del Metro de Caracas. Igualmente podemos cubrir la cobertura total de la red si seleccionamos una parada de esa línea como sucursal y todas las demás paradas de esa misma línea se encuentran en la cobertura de la parada seleccionada. Supongamos que elegimos como sucursal la Parada 3 y tenemos que  $t = 2$ :



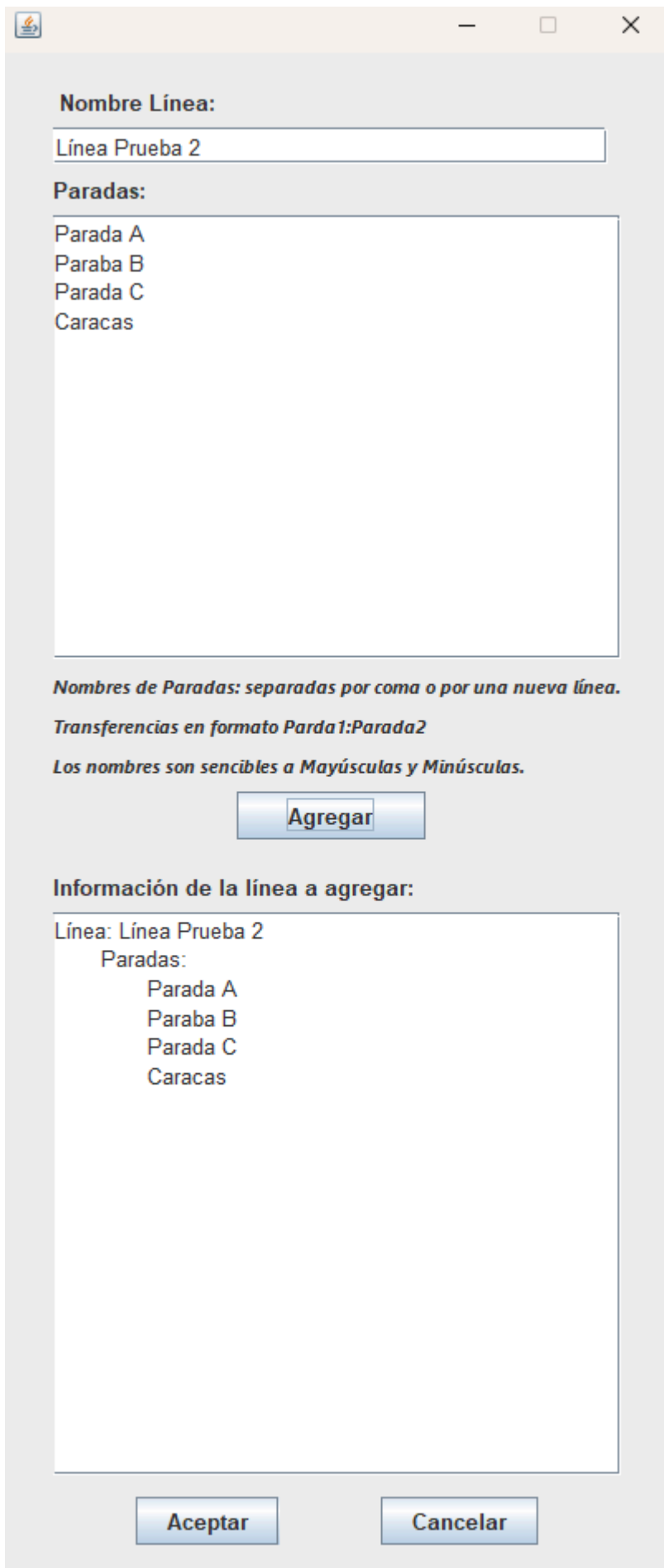
Como se puede observar, la Parada 3 tiene en su zona de cobertura a todas las paradas de la Línea Prueba que se creó. Si terminamos de seleccionar todas las sucursales recomendadas, ocurrirá lo siguiente:



Si observamos, la cobertura total se logra así la línea no este conectada a la red.

Ejemplo #2: Supongamos que creamos la siguiente línea:





**Nombre Línea:**

Línea Prueba 2

**Paradas:**

Parada A  
Paraba B  
Parada C  
Caracas

*Nombres de Paradas: separadas por coma o por una nueva línea.*  
*Transferencias en formato Parada1:Parada2*  
*Los nombres son sencibles a Mayúsculas y Minúsculas.*

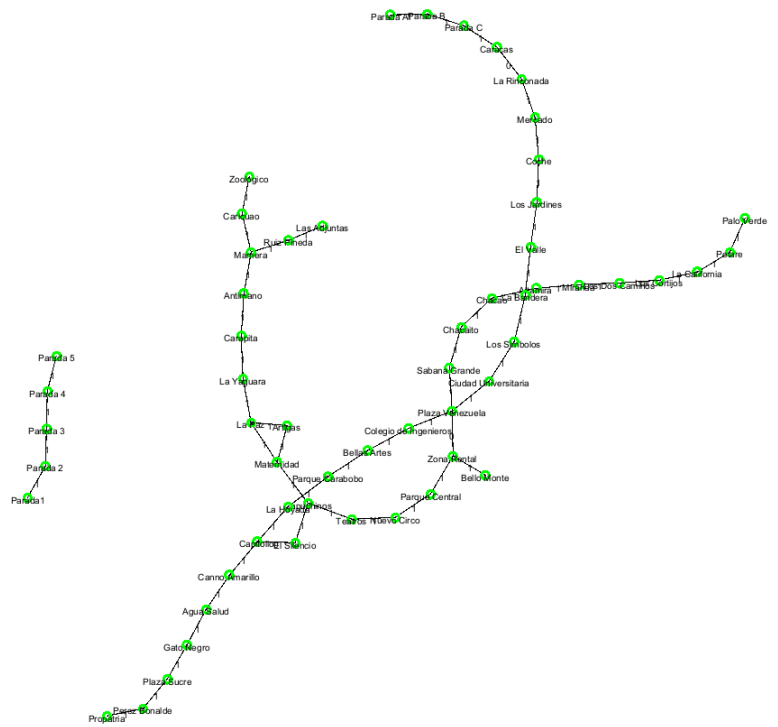
Agregar

**Información de la línea a agregar:**

Línea: Línea Prueba 2  
Paradas:  
Parada A  
Paraba B  
Parada C  
Caracas

Aceptar Cancelar

Como se puede observar, la Línea Prueba 2 tiene 4 paradas pero una de ellas es una que ya se encuentra en la red la cual es Caracas. Si la agregamos y vemos el grafo, se mostrará lo siguiente:



Ejemplo #3: Supongamos que creamos la siguiente línea:

—

□

×

**Nombre Línea:**

Línea Prueba 3

**Paradas:**

Propatria  
Palo Verde

*Nombres de Paradas: separadas por coma o por una nueva línea.*  
*Transferencias en formato Parada1:Parada2*  
*Los nombres son sencibles a Mayúsculas y Minúsculas.*

Agregar

**Información de la línea a agregar:**

Línea: Línea Prueba 3  
Paradas:  
Propatria  
Palo Verde

Aceptar


Cancelar

Si observamos, la Línea Prueba 3 posee tan solo 2 paradas y además, ambas paradas ya se encuentran en la red. Si la agregamos y vemos el grafo, se mostrará lo siguiente:



tienes una serie de acciones repetitivas que realizas en tu computadora, como abrir varias aplicaciones, mover archivos o ejecutar comandos. En lugar de hacer esto manualmente cada vez, puedes crear un archivo .bat para que tu computadora lo haga por ti.




Entonces, aprovechando el uso de dicho archivo, creamos el archivo Proyecto1.bat el cual se ve de la siguiente manera:

 Proyecto1	24/10/2024 1:32 p. m.	Archivo por lotes ...	1 KB
---	-----------------------	-----------------------	------

Y finalmente decidimos utilizar una carpeta data para guardar los archivos JSON de Caracas y Bogota que se usan en el proyecto.

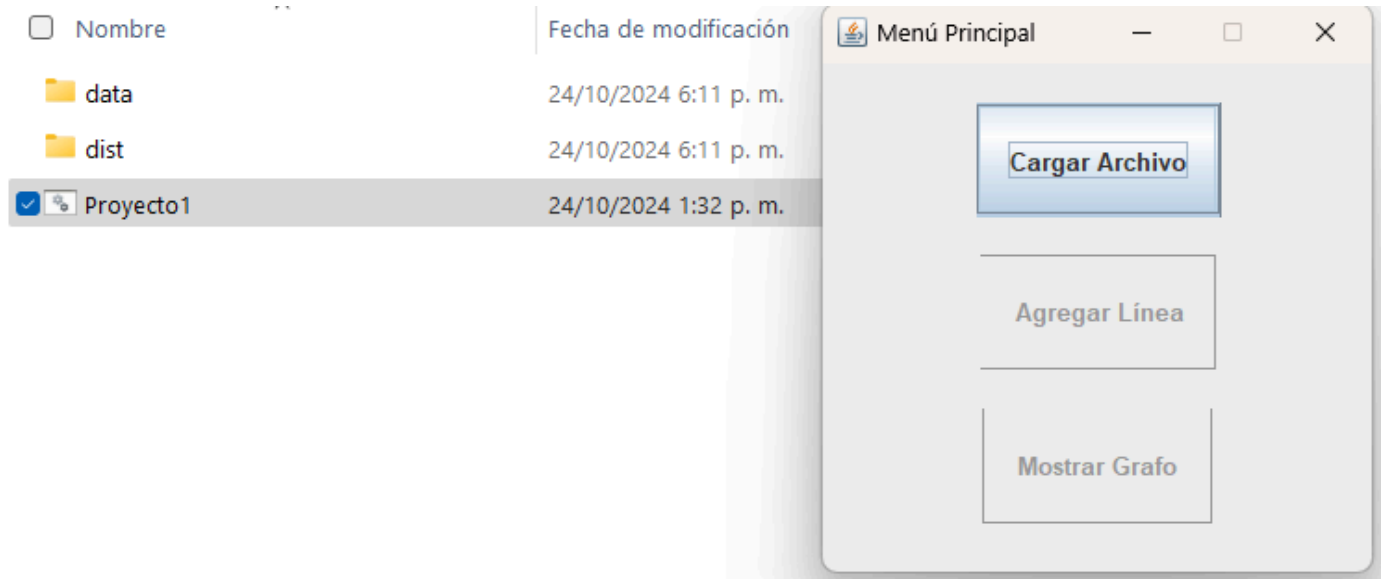
 Bogota	11/10/2024 12:01 p. m.	Archivo de origen ...	5 KB
 Caracas	11/10/2024 12:01 p. m.	Archivo de origen ...	2 KB

En un directorio aparte del proyecto, se mostraría algo así:

 data	24/10/2024 6:11 p. m.	Carpeta de archivos	
 dist	24/10/2024 6:11 p. m.	Carpeta de archivos	
 Proyecto1	24/10/2024 1:32 p. m.	Archivo por lotes ...	1 KB

**Antes de poder ejecutarlo, debemos asegurarnos que nuestra computadora cuenta con un JRE, también conocido como Java Runtime Environment o Entorno de ejecución de Java. Es un conjunto de software esencial para que las aplicaciones desarrolladas en Java puedan ejecutarse en tu computadora. Generalmente las computadoras suelen venir con dicho entorno, y si el usuario o cliente tiene instalado una versión de java muy probablemente lo tenga incluido.**

Para poder ejecutarlo, simplemente le damos doble click a el archivo .bat y el programa se ejecutará de manera idéntica a como se ejecutaria en NetBeans.



Eso ha sido todo lo que debe saber acerca de el proyecto y su debido uso y ejecución.