

## **Introducción:**

El presente proyecto tiene como objetivo desarrollar un sistema de simulación de líneas de colectivos urbanas que comparten paradas de pasajeros y recorridos. El sistema permitirá cargar la información de las líneas y paradas, así como también de los colectivos utilizados en la simulación. A medida que se ejecuta la simulación, se mostrará el recorrido de cada colectivo, las paradas en las que se detiene y los pasajeros que suben y bajan.

El proyecto se divide en incrementos, con el fin de abordar gradualmente la complejidad del problema y permitir entregas incrementales en consenso con la cátedra. En el primer incremento, se enfoca en la carga de información y generación de pasajeros en las paradas. Los datos de líneas, paradas y colectivos se cargarán desde archivos de texto, y se generarán los pasajeros distribuidos en las diferentes paradas. Durante la simulación, se mostrarán las paradas por las que pasa cada colectivo y los cambios en la cantidad de pasajeros.

En el segundo incremento, se agregará a la simulación el cálculo del índice de satisfacción del cliente y la ocupación promedio de los colectivos. Estas métricas permitirán evaluar la calidad del servicio y la eficiencia en el transporte de pasajeros. Además de la aplicación del sistema, se agregará toda la documentación solicitada, que incluirá análisis de las estructuras de datos utilizadas, diagramas de clases, manual de funcionamiento y otros documentos relevantes.

## **Planteo del problema:**

En las ciudades, el transporte público juega un papel fundamental en la movilidad de los ciudadanos. Uno de los medios de transporte más utilizados son los colectivos urbanos, los cuales siguen rutas preestablecidas y realizan paradas en distintos puntos de la ciudad para recoger y dejar pasajeros. Sin embargo, el funcionamiento eficiente de las líneas de colectivos puede ser un desafío debido a la complejidad de gestionar múltiples líneas, paradas compartidas y la demanda variable de pasajeros.

El problema planteado consiste en desarrollar un sistema de simulación de líneas de colectivos urbanas que aborde estos desafíos y permita evaluar el funcionamiento de las mismas. Se busca crear un entorno virtual que reproduzca el comportamiento de las líneas de colectivos, considerando aspectos como la carga de información de líneas y paradas, la generación de pasajeros, el recorrido de los colectivos y la interacción con los pasajeros en cada parada.

El sistema debe abordar los siguientes aspectos:

**Gestión de líneas y paradas:** Se debe proporcionar una manera de cargar la información de las líneas de colectivos y las paradas que forman parte de la red. Esto implica definir las rutas de cada línea, las paradas en cada ruta y los horarios de llegada y salida de los colectivos en cada parada.

**Carga de datos de colectivos:** Se requiere la posibilidad de cargar los datos de los colectivos que participarán en la simulación. Estos datos incluyen información sobre el número de colectivos, capacidad de pasajeros y otros detalles relevantes.

**Generación de pasajeros:** Durante la simulación, se deben generar pasajeros distribuidos en las distintas paradas de la red. Estos pasajeros deberán tener destinos aleatorios y serán recogidos por los colectivos en las paradas correspondientes.

**Recorrido de los colectivos:** Los colectivos deberán seguir las rutas definidas para cada línea y detenerse en las paradas programadas. Se debe proporcionar un mecanismo para simular el movimiento de los colectivos, indicando las paradas por las que pasan y los pasajeros que suben y bajan en cada parada.

**Cálculo de métricas:** Para evaluar la calidad del servicio, se debe calcular el índice de satisfacción del cliente y la ocupación promedio de los colectivos. Estas métricas permitirán medir la eficiencia en el transporte de pasajeros y la capacidad de satisfacer la demanda de manera adecuada.

### **Análisis de las estructuras seleccionadas:**

**Clases:** El proyecto utiliza varias clases para representar diferentes entidades y componentes del sistema, como Colectivo, Linea, Parada, Pasajero y SimuladorColectivos. Estas clases están bien definidas y encapsulan la funcionalidad relacionada en cada una de ellas.

**Listas:** El proyecto hace uso de listas para almacenar colecciones de objetos de diferentes tipos, como la lista de colectivos, líneas, paradas y pasajeros. Las listas permiten almacenar y acceder eficientemente a los elementos, y se utilizan métodos de las listas, como add(), remove(), size(), entre otros, para manipular los datos.

**Propiedades y archivos de configuración:** Se utiliza la clase Properties para cargar la configuración del sistema desde un archivo de propiedades (config.properties). Esta configuración incluye rutas de archivos de datos para colectivos, líneas y paradas, así como información adicional, como el número total de pasajeros y recorridos.

**Lectura de archivos:** Se implementa la lectura de archivos de texto (colectivoFilePath, lineaFilePath, paradaFilePath) para cargar los datos de colectivos, líneas y paradas respectivamente. Se utilizan las clases FileReader y BufferedReader para leer los archivos línea por línea y procesar los datos.

**Relaciones entre objetos:** Las clases Colectivo, Linea, Parada y Pasajero están relacionadas entre sí. Por ejemplo, un colectivo puede tener una línea asociada y una parada actual. Además, una parada puede tener una lista de pasajeros que están esperando en ella.

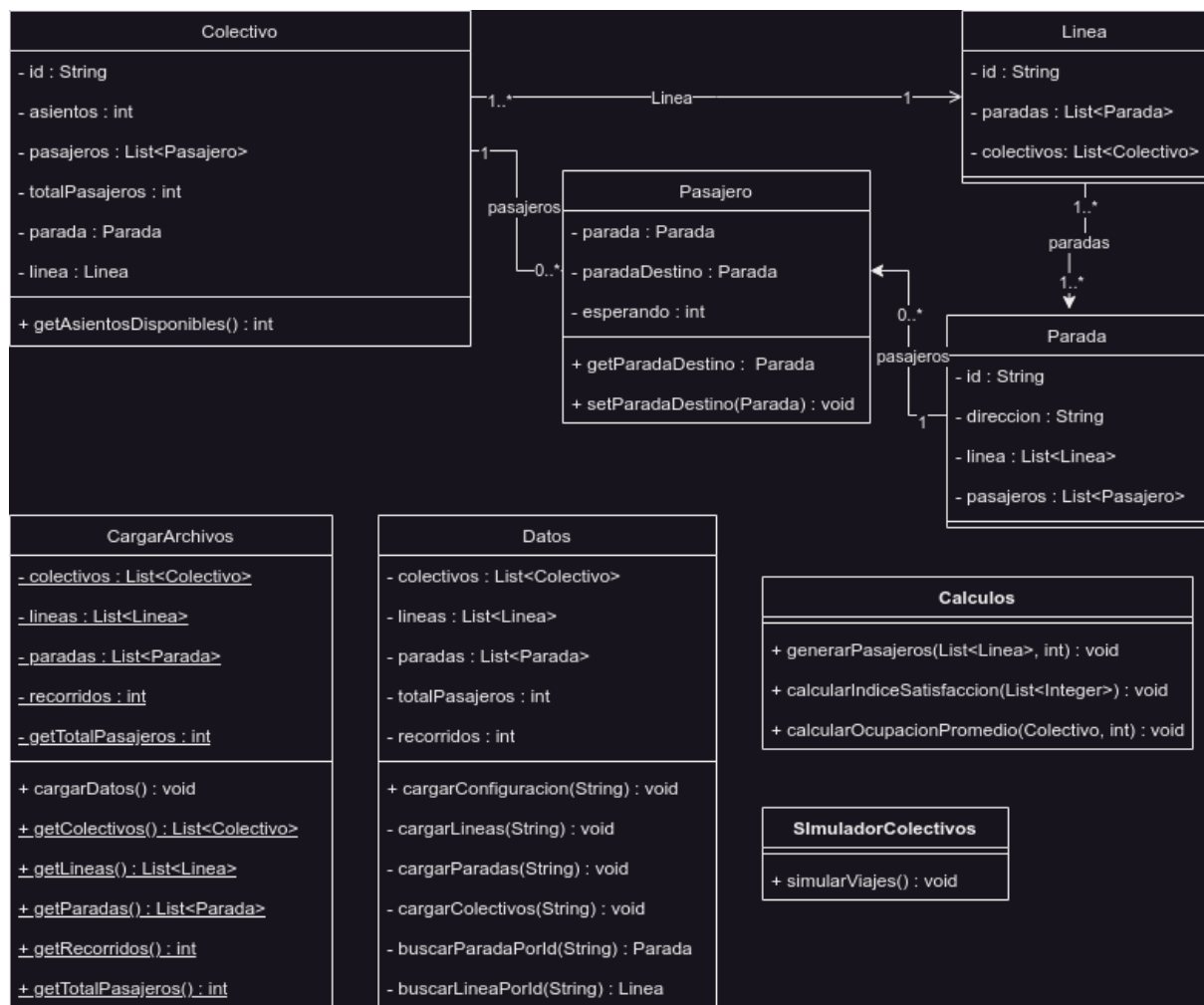
**Generación aleatoria de pasajeros:** Se utiliza la clase Random para generar un número aleatorio de pasajeros en cada parada y asignarlos a la lista de pasajeros de esa parada. La

cantidad de pasajeros generados se basa en el número total de pasajeros definido en la configuración.

**Métodos auxiliares:** El proyecto incluye varios métodos auxiliares para realizar operaciones específicas, como buscar una línea o parada por su ID, subir y bajar pasajeros de un colectivo, calcular el índice de satisfacción del cliente y el promedio de ocupación, entre otros.

Las estructuras seleccionadas en el proyecto, como clases, listas, propiedades y métodos auxiliares, permiten una representación adecuada de los componentes del sistema y el procesamiento de los datos necesarios para simular los viajes de los colectivos.

### Diagrama de clases:



## **Implementación de la solución:**

A continuación se muestra un resumen de la implementación:

**Aplicación:** Esta clase contiene el método principal que sirve como punto de entrada de la aplicación. En el método main, se crea una instancia del SimuladorColectivos y se llama al método simularViajes para iniciar la simulación de los viajes de los colectivos.

**Cálculos:** Esta clase contiene métodos para realizar cálculos relacionados con el sistema de colectivos. En el método generarPasajeros, se generan pasajeros aleatoriamente y se asignan a las paradas de las líneas. El método calcularIndiceSatisfaccion calcula el índice de satisfacción del cliente a partir de las calificaciones recibidas. El método calcularOcupacionPromedio calcula la ocupación promedio del colectivo en función de los pasajeros transportados.

**SimuladorColectivos:** Esta clase simula los viajes de los colectivos y realiza los cálculos correspondientes. En el método simularViajes, se carga la información necesaria llamando al método cargarDatos de la clase CargarArchivos. Luego, se crea una instancia de la clase Cálculos y se inicializan variables para almacenar los resultados de los cálculos. A continuación, se itera sobre la lista de colectivos y se simulan los viajes de cada uno. Dentro del bucle, se itera sobre las paradas de la línea del colectivo y se realiza el proceso de subida y bajada de pasajeros. Se registran las estadísticas de cada parada y se actualiza el contador de pasajeros transportados. Al finalizar cada recorrido, se calcula la ocupación promedio del colectivo llamando al método calcularOcupacionPromedio, y se calcula el índice de satisfacción del cliente llamando al método calcularIndiceSatisfaccion.

**Datos:** Esta clase es un contenedor de datos estáticos que almacena la información cargada desde los archivos de configuración. Contiene listas de objetos Colectivo, Línea, Parada, un entero totalPasajerosConfig que representa el número total de pasajeros configurados y otro entero llamado recorridos que son los recorridos configurados.

**CargarArchivos:** Esta clase se encarga de cargar los datos desde los archivos de configuración y almacenarlos en la clase Datos. El método cargarDatos se encarga de leer los archivos y llenar las listas de objetos con la información correspondiente. Los métodos getters devuelven los datos cargados.

**Colectivo:** Esta clase representa un colectivo en el sistema. Tiene un identificador id, una línea a la que pertenece y una capacidad total de pasajeros. Además, tiene una lista de pasajeros para almacenar los pasajeros que se encuentran en el colectivo.

**Parada:** Esta clase representa una parada en la línea de un colectivo. Tiene un identificador id, una dirección y una lista de pasajeros que representa los pasajeros que están esperando en esa parada.

**Línea:** Esta clase representa una línea de colectivos. Tiene un identificador id y una lista de paradas que representa las paradas en el recorrido de la línea. También mantiene una lista de colectivos asociados a la línea.

Pasajero: Esta clase representa un pasajero en el sistema. Tiene una parada de origen y una parada de destino. También tiene una variable esperando que indica el tiempo que ha estado esperando en la parada de origen.

### **Manual de funcionamiento (ingreso de datos, pruebas, resultados de salida):**

Ingreso de datos: Los datos pueden ser ingresados desde los archivos de texto que se encuentran fuera de la carpeta Colectivo. Los archivos se pueden modificar de la siguiente manera:

- colectivo.txt:

Respetando el orden y la nomenclatura utilizada, se puede modificar el id del colectivo, la línea a que pertenece, cuantos asientos dispone, y cuántos pasajeros puede llevar. Solo puedes editar los cuatro campos.

id del colectivo;línea a la que pertenece;cantidad de asientos;total de pasajeros

C1;L1;30;60

C2;L1;30;60

C3;L2;30;60

- parada.txt

Respetando el orden y la nomenclatura utilizada, se puede modificar el id de la parada, y la dirección a la que pertenece. Solo puedes editar los dos campos.

id de la parada;dirección

P1;1 De Marzo, 405

P2;25 De Mayo, 299

P3;25 De Mayo, 693

- linea.txt

Respetando el orden y la nomenclatura utilizada, se puede modificar el id de la línea, y las paradas asociadas. Puedes agregar las paradas que usted quiera, siempre y cuando estas existan en parada.txt.

id de la línea;id de las paradas

L5;P33;P47;P40;P38;P37;P17;P2;P3;P31;P64;P28;P44;P46

L6;P1;P5;P51;P52;P25;P27;P45

- config.properties

Antes de ejecutar la simulación, asegúrate de tener el archivo de configuración que contenga la información necesaria para el funcionamiento del sistema. Este archivo puede incluir detalles como los colectivos, líneas y paradas, número de pasajeros, cantidad de recorridos, etc. Asegúrate de que la estructura del archivo de configuración sea compatible con el formato esperado por el código.

nombre de archivos

linea=linea.txt

parada=parada.txt

colectivo=colectivo.txt

cantidad de pasajeros  
pasajeros=1000

número de recorridos de cada colectivo  
recorridos=1

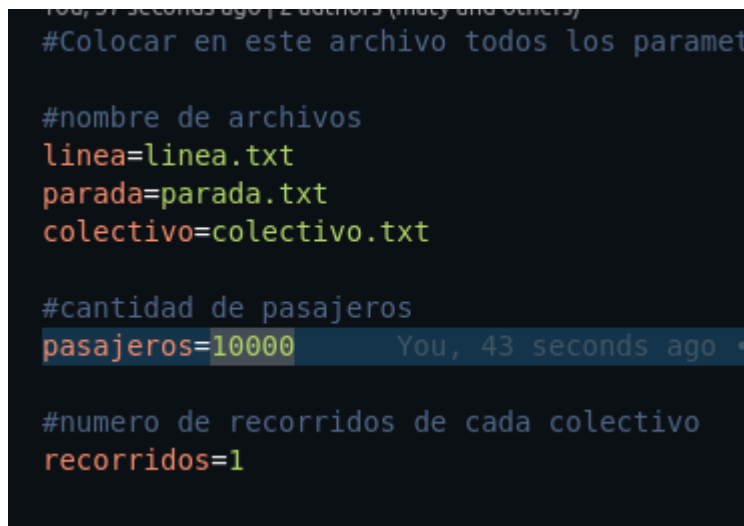
Datos de pasajeros: Si el sistema permite la generación aleatoria de pasajeros en las paradas, asegúrate de tener un mecanismo para proporcionar estos datos, ya sea ingresándolos manualmente o generándolos automáticamente a partir de ciertos criterios.

Carga de configuración: El primer paso es cargar la configuración del sistema desde el archivo correspondiente. Esto puede implicar leer el archivo de configuración y extraer la información necesaria para crear colectivos, líneas y paradas, así como establecer las conexiones entre ellos.

Inicio de la simulación: Una vez cargada la configuración, puedes iniciar la simulación llamando al método o función correspondiente. La simulación puede tener una duración predeterminada o continuar hasta que se cumpla una condición de salida específica.

Monitoreo y seguimiento: Durante la simulación, puedes monitorear el estado de los colectivos, líneas y paradas para asegurarte de que el sistema funcione correctamente. Esto puede implicar imprimir o registrar información relevante en la consola o en un archivo de registro.

Resultados de salida y pruebas: Algunas capturas de las salidas.

A screenshot of a code editor with a dark background. The code is written in a light blue/cyan color. It shows configuration parameters for a simulation. The parameters are: #nombre de archivos, linea=linea.txt, parada=parada.txt, colectivo=colectivo.txt, #cantidad de pasajeros, pasajeros=10000, and #numero de recorridos de cada colectivo, recorridos=1. The line 'pasajeros=10000' is highlighted with a blue selection bar. There is a faint watermark 'You, 43 seconds ago' in the background of the code editor.

```
#Colocar en este archivo todos los parametros de configuracion

#nombre de archivos
linea=linea.txt
parada=parada.txt
colectivo=colectivo.txt

#cantidad de pasajeros
pasajeros=10000

#numero de recorridos de cada colectivo
recorridos=1
```

```
- Parada ID: P27, Dirección: El Jariyal, 1725
Asientos disponibles: -5
Pasajeros en el colectivo: 20
  Pasajeros que subieron: 9
  Pasajeros que bajaron: 10
Pasajeros esperando en la parada: 0

- Parada ID: P45, Dirección: Los Alamos, 3900
Asientos disponibles: 15
Pasajeros en el colectivo: 0
  Pasajeros que subieron: 0
  Pasajeros que bajaron: 20
Pasajeros esperando en la parada: 0

Pasajeros transportados: 36. Ocupación promedio: 20.0%
Indice de satisfacción del cliente: 1.1587999999999998%
104 pasajeros calificaron con 5 el servicio
287 pasajeros calificaron con 4 el servicio
81 pasajeros calificaron con 3 el servicio
149 pasajeros calificaron con 2 el servicio
9379 pasajeros calificaron con 1 el servicio
```

Al tener 10.000 pasajeros, muchos no pueden entrar a los colectivos por su límite de capacidad. Y así, ir probando las entradas para ver las salidas resultantes.

Estadísticas y métricas: Al finalizar la simulación, puedes recopilar y calcular diversas estadísticas y métricas para evaluar el rendimiento del sistema. Esto puede incluir el número de pasajeros atendidos, el tiempo promedio de espera de los pasajeros, la ocupación promedio de los colectivos, etc.

**Errores detectados (si existe algún error y bajo qué condiciones se produce):**

**Data.java:**

Hay un error de dependencia al cargar las listas desde el archivo. En la línea 37 se repite la función de cargarLineas() porque en la primera llamada no tiene las paradas asociadas. Y si llamase cargarParadas() primero, las paradas no tienen líneas porque no se llamó cargarLineas() primero. Las paradas recién se cargan al llamar cargarParadas() después de las líneas.

### Lotes de prueba:

Prueba con solo veinte paradas y mil pasajeros:

```
P1;1 De Marzo, 405
P2;25 De Mayo, 299
P3;25 De Mayo, 693
P4;28 De Julio, 200
P5;28 De Julio, 311
P6;8 De Marzo, 385
P7;Abraham Mathews, 2525
P8;Agustin Pujol, 374
P9;Alto Rio Senguer, 1311
P10;Avenida Domec Garcia Norte, 239
P11;Avenida Juan Muzzio, 258
P12;Avenida Juan Muzzio, 482
P13;Avenida Julio Argentino Roca, 1567
P14;Avenida Julio Argentino Roca, 2251
P15;Bartolome Mitre, 401
P16;Bartolome Mitre, 610
P17;Belgrano, 314
P18;Bouchard, 1285
P19;Bouchard, 880
P20;Bv. Almirante Brown, 2925
```

```
L1;P16;P15;P4;P10
L2;P9;P6;P11;P12;P17;P2;P3;P13;P14;P7;P20
L3;P8;P1;P5
L4;P1;P5;P19;P18
L5;P17;P2;P3
L6;P1;P5
```



```
condt.java/jdt_ws/trabajoIntegrador_7b54e217/bin/collectivo.app.Application
[Parada [id: P16, direccion: Bartolome Mitre, 610, linea: L1, pasajeros: 71]
, Parada [id: P15, direccion: Bartolome Mitre, 401, linea: L1, pasajeros: 57]
, Parada [id: P4, direccion: 28 De Julio, 200, linea: L1, pasajeros: 58]
, Parada [id: P10, direccion: Avenida Domec Garcia Norte, 239, linea: L1, pasajeros: 0]
]
[Parada [id: P9, direccion: Alto Rio Senguer, 1311, linea: L2, pasajeros: 50]
, Parada [id: P6, direccion: 8 De Marzo, 385, linea: L2, pasajeros: 56]
, Parada [id: P11, direccion: Avenida Juan Muzzio, 258, linea: L2, pasajeros: 74]
, Parada [id: P12, direccion: Avenida Juan Muzzio, 482, linea: L2, pasajeros: 45]
, Parada [id: P17, direccion: Belgrano, 314, linea: L2, pasajeros: 99]
, Parada [id: P2, direccion: 25 De Mayo, 299, linea: L2, pasajeros: 71]
, Parada [id: P3, direccion: 25 De Mayo, 693, linea: L2, pasajeros: 89]
, Parada [id: P13, direccion: Avenida Julio Argentino Roca, 1567, linea: L2, pasajeros: 56]
, Parada [id: P14, direccion: Avenida Julio Argentino Roca, 2251, linea: L2, pasajeros: 62]
, Parada [id: P7, direccion: Abraham Mathews, 2525, linea: L2, pasajeros: 45]
, Parada [id: P20, direccion: Bv. Almirante Brown, 2925, linea: L2, pasajeros: 0]
]
[Parada [id: P8, direccion: Agustin Pujol, 374, linea: L3, pasajeros: 21]
, Parada [id: P1, direccion: 1 De Marzo, 405, linea: L3, pasajeros: 118]
, Parada [id: P5, direccion: 28 De Julio, 311, linea: L3, pasajeros: 21]
]
[Parada [id: P1, direccion: 1 De Marzo, 405, linea: L3, pasajeros: 118]
, Parada [id: P5, direccion: 28 De Julio, 311, linea: L3, pasajeros: 21]
, Parada [id: P19, direccion: Bouchard, 880, linea: L4, pasajeros: 12]
, Parada [id: P18, direccion: Bouchard, 1285, linea: L4, pasajeros: 0]
]
[Parada [id: P17, direccion: Belgrano, 314, linea: L2, pasajeros: 99]
, Parada [id: P2, direccion: 25 De Mayo, 299, linea: L2, pasajeros: 71]
, Parada [id: P3, direccion: 25 De Mayo, 693, linea: L2, pasajeros: 89]
]
[Parada [id: P1, direccion: 1 De Marzo, 405, linea: L3, pasajeros: 118]
, Parada [id: P5, direccion: 28 De Julio, 311, linea: L3, pasajeros: 21]
]
```

```
Pasajeros transportados: 0. Ocupación promedio: 0.0%
Indice de satisfacción del cliente: 2.3810000000000002%
80 pasajeros calificaron con 5 el servicio
174 pasajeros calificaron con 4 el servicio
245 pasajeros calificaron con 3 el servicio
49 pasajeros calificaron con 2 el servicio
452 pasajeros calificaron con 1 el servicio
```

Con diez pasajeros

```
Pasajeros transportados: 0. Ocupación promedio: 0.0%
Indice de satisfacción del cliente: 5.0%
10 pasajeros calificaron con 5 el servicio
0 pasajeros calificaron con 4 el servicio
0 pasajeros calificaron con 3 el servicio
0 pasajeros calificaron con 2 el servicio
0 pasajeros calificaron con 1 el servicio
```

Con dos recorridos, y mil pasajeros.

```
Pasajeros transportados: 0. Ocupación promedio: 0.0%
Indice de satisfacción del cliente: 2.74%
80 pasajeros calificaron con 5 el servicio
160 pasajeros calificaron con 4 el servicio
223 pasajeros calificaron con 3 el servicio
494 pasajeros calificaron con 2 el servicio
43 pasajeros calificaron con 1 el servicio
```

Con un solo colectivo.

```
100, 48 segundos ago
C1;L1;30;60
```

```
Pasajeros transportados: 133. Ocupación promedio: 73.888888
Indice de satisfacción del cliente: 1.369%
30 pasajeros calificaron con 5 el servicio
43 pasajeros calificaron con 4 el servicio
60 pasajeros calificaron con 3 el servicio
0 pasajeros calificaron con 2 el servicio
867 pasajeros calificaron con 1 el servicio
```

Con una sola línea y cuatro paradas.

```
L1;P16;P15;P4;P10
```

```
C1;L1;30;60
```

```
C2;L1;30;60
```

```
P4;28 De Julio, 200;L1
P10;Avenida Domec Garcia Norte, 239;L1
P15;Bartolome Mitre, 401;L1
P16;Bartolome Mitre, 610;L1
```

```
Pasajeros transportados: 82. Ocupación promedio: 45
Indice de satisfacción del cliente: 1.422%
30 pasajeros calificaron con 5 el servicio
46 pasajeros calificaron con 4 el servicio
82 pasajeros calificaron con 3 el servicio
0 pasajeros calificaron con 2 el servicio
842 pasajeros calificaron con 1 el servicio
```

```
[Parada [id: P16, direccion: Bartolome Mitre, 610, linea: L1, pasajeros: 353]
, Parada [id: P15, direccion: Bartolome Mitre, 401, linea: L1, pasajeros: 297]
, Parada [id: P4, direccion: 28 De Julio, 200, linea: L1, pasajeros: 350]
, Parada [id: P10, direccion: Avenida Domec Garcia Norte, 239, linea: L1, pasajeros: 0]
]
```

Con veinte recorridos y una sola línea.

```
Pasajeros transportados: 0. Ocupación promedio: 0.0%
Indice de satisfacción del cliente: 2.2652043868394816%
30 pasajeros calificaron con 5 el servicio
51 pasajeros calificaron con 4 el servicio
74 pasajeros calificaron con 3 el servicio
848 pasajeros calificaron con 2 el servicio
0 pasajeros calificaron con 1 el servicio
```

### **Posibles mejoras y extensiones:**

#### **Clase Colectivo:**

- Implementar métodos para permitir la reserva de asientos en el colectivo.
- Agregar un mecanismo para rastrear la ubicación actual del colectivo en tiempo real, por ejemplo, mediante el uso de GPS.
- Incorporar lógica adicional para el cálculo de tarifas y la gestión de pagos de los pasajeros.
- Permitir la programación y gestión de horarios de los colectivos.

#### **Clase SimuladorColectivos:**

- Mejorar la representación visual de la simulación utilizando una interfaz gráfica de usuario (GUI).
- Agregar la capacidad de generar informes detallados sobre el rendimiento de los colectivos y la satisfacción de los pasajeros.
- Implementar un algoritmo más sofisticado para la generación de pasajeros en las paradas, teniendo en cuenta factores como la demanda, la hora del día y las preferencias de los pasajeros.
- Incorporar la posibilidad de simular eventos inesperados, como retrasos en el tráfico o cambios en las condiciones climáticas.

#### **Clase Datos:**

- Mejorar la eficiencia de la carga de datos implementando técnicas de almacenamiento en caché o indexación para agilizar las búsquedas de líneas y paradas.
- Añadir validaciones y manejo de errores para garantizar la integridad de los datos cargados desde los archivos.
- Ampliar la capacidad de cargar datos desde diferentes fuentes, como bases de datos o servicios web.
- Implementar la capacidad de exportar los datos cargados a diferentes formatos, como archivos CSV o JSON.

### **Conclusiones:**

A lo largo de este documento, he presentado un sistema de simulación de un sistema de

transporte público basado en colectivos. Este sistema ha sido diseñado para modelar y evaluar el funcionamiento de las líneas de colectivos, la demanda de pasajeros y los tiempos de viaje en una red de transporte.

Desarrollé un modelo que simula el movimiento de los colectivos entre paradas, la generación de pasajeros en cada parada y la asignación de pasajeros a los colectivos. Además, incorpora factores como la capacidad limitada de los colectivos y la posibilidad de que los pasajeros no puedan abordar debido a la falta de espacio.

Mediante la simulación, he obtenido datos sobre los tiempos de viaje promedio, la ocupación de los colectivos y la cantidad de pasajeros que no pudieron abordar debido a la capacidad limitada. Estos datos nos han permitido evaluar el desempeño del sistema y realizar análisis comparativos entre diferentes escenarios.

En base a los resultados obtenidos, se puede concluir que el sistema de transporte público basado en colectivos presenta ciertas limitaciones y desafíos. Los tiempos de viaje pueden ser significativamente afectados por la capacidad limitada de los colectivos, lo que puede resultar en una mayor espera para los pasajeros y una menor eficiencia del sistema.

Además, he observado que la asignación aleatoria de pasajeros a los colectivos puede resultar en una distribución desigual de la carga entre ellos. Esto puede llevar a una mayor ocupación en algunos colectivos y una menor utilización de otros, lo que puede afectar la comodidad de los pasajeros y la eficiencia global del sistema.

Sin embargo, también he identificado oportunidades de mejora y extensiones para el sistema. Algunas posibles mejoras incluyen la implementación de algoritmos de enrutamiento más eficientes, la consideración de eventos y condiciones especiales, la implementación de estrategias de asignación más sofisticadas y la incorporación de consultas y seguimiento en tiempo real.