

### **PARTE 3: Ejercicio de programación JAVA**

**Duración: 60 minutos**

Este ejercicio comprende **3** partes:

1. Desarrollo de las funcionalidades especificadas más abajo
2. Desarrollo del programa ejecutable con datos de ejemplo provistos y entrega de todo el Proyecto **Maven** en la tarea correspondiente de la webasignatura.
3. Desarrollo de los casos de prueba para las funcionalidades requeridas

## **ESCENARIO**

En el contexto de una red de transporte urbano, se enfrenta el desafío de gestionar eficientemente las conexiones entre dos tipos de estaciones: estaciones de autobuses y estaciones de tren. Es crucial para la operación fluida y eficiente del sistema de transporte que las transferencias entre estaciones de autobuses ocurran exclusivamente a través de estaciones de tren, en lugar de conexiones directas entre estaciones de autobuses. Esta estructura está diseñada no solo para optimizar el flujo de tráfico dentro de la ciudad, reduciendo la congestión en áreas urbanas densamente pobladas, sino también para aprovechar al máximo la capacidad y la eficiencia de la infraestructura ferroviaria disponible.

El objetivo de este proyecto es desarrollar un algoritmo robusto que pueda calcular todas las rutas posibles desde una estación de autobuses de origen especificada hasta una estación de autobuses de destino especificada dentro de la red de transporte urbano. Estas rutas deben cumplir estrictamente con la restricción de que todas las conexiones entre estaciones de autobuses deben realizarse a través de estaciones de tren intermedias. Este enfoque no solo garantiza una mejor gestión del tráfico y una distribución más eficiente de los recursos de transporte, sino que también promueve un uso más sostenible de la infraestructura urbana al fomentar el uso de modos de transporte más eficientes y menos contaminantes.

**Se desea:**

**Programar un algoritmo capaz de calcular todas las rutas posibles desde una estación de autobuses de origen especificada hasta una estación de autobuses de destino especificada dentro de la red de transporte. Las rutas deben cumplir con la restricción de que las conexiones** entre estaciones de autobuses se realicen exclusivamente a través de estaciones de tren

La firma de dicho algoritmo debe ser:

TCaminos<TEstacionDeLaRed> caminosDesdeHasta(Comparable nodoOrigen,  
Comparable nodoDestino);

## **PARTE 2: PROGRAMA (vale 30%)**

La clase principal se denomina “**MainParcial2**”, y tiene su correspondiente método “**main**”. En éste, implementa lo necesario para aplicar los TDA y métodos desarrollados.

## **PARTE 3: TEST CASES (vale 25%).**

Implementa el o los Casos de Prueba necesarios para verificar el correcto funcionamiento del método implementado.

## **NOTAS IMPORTANTES:**

- Se proveen las interfaces y clases necesarias. Deben implementarse los métodos necesarios (respetando las firmas indicadas). NO SE DEBEN ALTERAR LAS INTERFACES provistas, ni agregar otros métodos que los requeridos en las interfases.
- **NO SE DEBEN CREAR NUEVAS CLASES Y NO SE DEBE INCLUIR NINGUN METODO NO SOLICITADO O INNECESARIO.**

**ENTREGA:** Debes entregar TODO el proyecto y los archivos de salida solicitados, en un archivo comprimido “Parcial2.zip” en la tarea “PARCIAL2-PARTE3” publicada en la webassignatura, hasta la hora indicada.