

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 los centros de datos modernos, las redes consisten en una compleja malla de máquinas que incluyen tanto nodos de procesamiento como switches. Los nodos de procesamiento son los encargados de realizar las tareas de computación y procesamiento de datos, mientras que los switches gestionan el enrutamiento de datos entre estos nodos de procesamiento. La comunicación eficiente entre los nodos de procesamiento es crucial para un rendimiento óptimo, pero debe gestionarse cuidadosamente para garantizar que el flujo de datos sea correcto y eficiente a través de la red.

Dado este contexto, es necesario calcular todos los caminos posibles entre dos nodos de procesamiento especificados en la red. Sin embargo, existe una restricción crítica: no se permiten conexiones directas entre los nodos de procesamiento. La comunicación entre los nodos de procesamiento debe enrutarse mediante switches. Esta restricción asegura que la red mantenga un sistema de enrutamiento estructurado y manejable, previniendo la comunicación directa entre máquinas que podría llevar a cuellos de botella o ineficiencias.

Se desea:

Programar un algoritmo capaz de calcular todos los posibles caminos desde un nodo de procesamiento de origen especificado hasta un nodo de procesamiento de destino especificado. Los caminos deben adherirse a las reglas de comunicación de la red, donde los datos solo pueden viajar de un nodo de procesamiento a otro a través de uno o más switches.

La firma de dicho algoritmo debe ser:

```
TCaminos<TNodeDeLaRed> caminosDesdeHasta(Comparable nodoOrigen,  
Comparable nodoDestino);
```

PARTE 2: PROGRAMA (vale 30%)

La clase principal se denomina “Main**Parcial2**”, 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 webasignatura, hasta la hora indicada.