

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

Este ejercicio comprende **3** partes:

1. Desarrollo de la 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 test para las funcionalidades requeridas

### **ESCENARIO**

En el corazón de la ciudad inteligente de Algorville, donde la tecnología y la conectividad son fundamentales para la vida cotidiana, se plantea un desafío crucial relacionado con la evaluación de la conectividad en su vasta red de dispositivos interconectados. La municipalidad, en su búsqueda por optimizar la eficiencia de la comunicación en esta Smart City, busca determinar no sólo la existencia de una conexión directa, sino también la posibilidad de conectividad a través de saltos, permitiendo así una comunicación eficaz entre un Nodo X y un Nodo Y.

La red de dispositivos en Algorville está en constante expansión para abordar las demandas de una ciudad inteligente. Los participantes deben desarrollar un algoritmo innovador que no sólo evalúe la conectividad entre nodos específicos, sino que también considere la capacidad de establecer conexiones a través de nodos intermedios, facilitando así una comunicación eficiente en esta infraestructura tecnológicamente avanzada.

En el contexto de una Smart City, donde la interconexión de dispositivos es esencial para optimizar servicios públicos, gestión de tráfico, y eficiencia energética, la solución propuesta deberá adaptarse dinámicamente a las complejidades de la red. Esto incluye enlaces intermitentes, dispositivos temporalmente fuera de servicio y cambios en la topología de la red, elementos que son fundamentales en el desarrollo de una ciudad inteligente.

Uno de los dispositivos conectados a la red Algorville realiza mediciones de cierta magnitud y estas deben ser procesadas on-device por criterios de privacidad. De las mediciones tomadas en cierto período de tiempo, se necesita obtener la instancia de medición de mayor magnitud.

### **PARTE 1: Funcionalidad a desarrollar (vale 45%):**

Descargar de la webasignatura el archivo “Parcial2-2023-2.zip” que contiene el Proyecto **Maven** a ser completado.

#### **Se desea:**

1. Implementa en la clase apropiada, un método que, recibiendo por parámetro el nombre de 2 nodos de la red, determine si existe conexión entre ambos nodos.
2. Implementa en la clase apropiada, un método que, dadas las mediciones de un dispositivo en cierto período de tiempo, indique la medición de mayor magnitud.

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

La clase principal se denomina “**Parcial2**”, y tiene su correspondiente método “**main**”. En éste, implementa lo necesario para aplicar los TDA y métodos desarrollados.

1. Leer y cargar los datos de dispositivos y conexiones a partir de los archivos dispositivos.txt y conexiones.txt (el parsing de las líneas ya está dado)

2. Invocar el método **conectados** para los siguientes dispositivos:
  - a. CS10 a CS60
  - b. CS30 a CS80
  - c. CS80 a CS90
3. Leer y cargar los datos de las mediciones a partir de un archivo mediciones.txt
4. Invocar el método **obtenerMayorMedición** para los datos del paso anterior.
5. Escribir un archivo de texto "**salida.txt**", con los datos de conexión de los dispositivos.

### **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 interfaces.
- **NO SE DEBEN CREAR NUEVAS CLASES Y NO SE DEBE INCLUIR NINGUN METODO NO SOLICITADO O INNECESARIO.**
- Debe prevenirse, tanto como sea posible utilizando solamente las interfaces y clases provistas, que el árbol resultante del punto 2 tenga una estructura extremadamente mala (la altura del árbol resultante no debe ser comparable al tamaño del mismo).

**ENTREGA:** Debes entregar TODO el proyecto Netbeans 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.