

**Informática II**  
**Practica No. 4**

**Sesiones 1 y 2**

**Objetivos:**

- Adquirir destreza en el modelamiento e implementación objetos.
- Adquirir destreza en el manejo de los contenedores implementados en la STL.
- Emplear la programación orientada a objetos para la solución de problemas asociados a redes de computadores.

**Simulación de una red de enrutadores**

Muchas de las actividades que realizamos actualmente implican el acceso a recursos en sitios remotos. Consultar una página web, enviar un correo o participar de una sesión de juego en línea, son actividades que necesitan conectar al menos dos equipos separados por metros o por miles de kilómetros. Para facilitar dicha conectividad, los computadores, celulares, consolas y otros terminales, se agrupan en “redes”, las cuales deben ser alcanzables entre sí. Así, por ejemplo, desde un navegador web en un computador ubicado en una red de una casa se puede alcanzar la página de algún periódico que esté alojada en un servidor en una red en Estados Unidos.

Los enrutadores son los equipos intermedios que se encargan de proveer la conectividad entre las diferentes redes. Un enrutador puede conocer una o varias redes y puede compartir esta información con otros enrutadores con el fin de construir bases de datos consistentes y, así, poder direccionar la información desde un origen a un destino. En esas bases de datos se guarda la información para encontrar el camino más corto (usualmente conocido como “el mejor camino”) a todas las redes de una topología (más adelante se verá que en ocasiones no se necesita conocer un camino a todas las redes). Para entender esto se puede revisar el siguiente ejemplo:

Consideremos una red de 4 enrutadores (A, B, C, D), conectados a través de enlaces con costos como se dispone en la Figura 1. Se puede suponer que cada enrutador conoce sólo una red, lo que reduce el problema a encontrar el camino más corto desde cada enrutador a todos los enrutadores de la topología. Si tomamos un estado inicial de la red, podemos asumir que cada enrutador sólo conoce los costos de los enlaces con los vecinos directamente conectados.

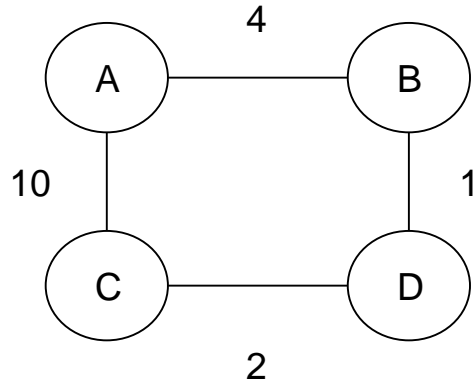


Figura 1. Red de enrutadores

En el enrutador **A** tendríamos la siguiente tabla, con los costos conocidos a diferentes nodos:

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>A</b>	0	4	10	-

Después de intercambiar información con los otros nodos, **A** ejecuta un algoritmo donde encuentra una ruta a **D** a través de **B** y una mejor ruta a **C** (que la directa) a través de **B**:

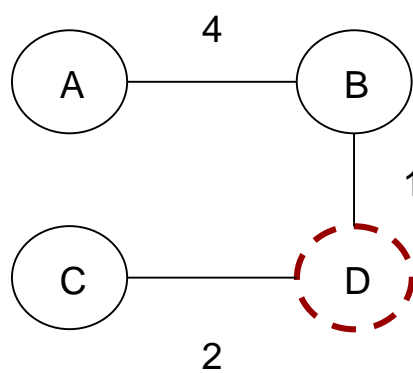
	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>A</b>	0	4	7	5

Este procedimiento se realiza de manera “simultánea” en todos los nodos, obteniendo al final lo siguiente:

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>A</b>	0	4	7	5
<b>B</b>	4	0	3	1
<b>C</b>	7	3	0	2
<b>D</b>	5	1	2	0

Sin embargo, esta tabla sólo informa de los costos de cada enrutador al resto de enrutadores en la topología, no entrega información, por ejemplo, de la ruta que debe tomar **A** para llegar a **C** (que no es la directamente conectada). Así, cuando **A** debe enviar información hacia **C** lo hace a través del enlace con **B** (conocido como siguiente salto).

En esta práctica utilizaremos una generalización de una de las bases de datos que utiliza un enrutador para llegar a diferentes destinos, conocida como la “tabla de enrutamiento”. Para construirla se necesitará que cada enrutador conozca la topología completa de la red y los costos de cada enlace (incluidos los enlaces de los enrutadores restantes). A partir de esta información, cada enrutador debe calcular el camino más corto (con él mismo como fuente) a todos los nodos de la red. Por ejemplo, tomando a **D** en el ejemplo anterior, el árbol que dibuja las rutas más cortas a todos los nodos:



La tabla de enrutamiento almacenará entonces los recorridos completos desde **D** hacia los otros nodos, con sus respectivos costos.

### Ejemplos

El profesor realizará varios ejemplos del uso de contenedores tales como *vector*, *list*, etc; implementados en la STL.

### Ejercicio

Con base en el conocimiento adquirido a partir de la sesión teórica acerca de enrutamiento, la información de esta guía, y utilizando los conceptos de programación orientada a objetos, escriba un programa que permita simular una red de enrutadores. Para eso tenga en cuenta lo siguiente:

- A. **Modelar un enrutador.** El modelamiento debe tener en cuenta que el enrutador debe estar en capacidad de almacenar su tabla de enrutamiento, la cual puede ser actualizada en tiempo de ejecución ya sea porque el costo de los enlaces cambia o los enlaces entre enrutadores cambian (se eliminan o se agregan).
- B. **Modelar una red** que permita:
  - a. **Agregar/Remover tantos enrutadores** como se desee.

- b. Actualizar la tabla de enrutamiento de cada enrutador conforme se va actualizando la topología de la red.
  - c. Se debe permitir configurar una red a partir de una tabla de topología de una red específica que sea cargada desde un archivo.
- C. Ofrecer la posibilidad de saber cuánto cuesta enviar un paquete desde un enrutador origen a uno destino.
- D. Decir cuál es el camino que debe seguir un paquete para ser enviado de forma eficiente desde un enrutador origen a uno destino.
- E. Generar redes de forma aleatoria para facilitar la realización de pruebas. Para eso podría hacer uso de la capacidad de cargar redes desde archivos y así facilitar la generación aleatoria.

La implementación deberá incluir el uso de contenedores implementados en la STL.

Esta guía fue desarrollada por los profesores Jaime Vergara y Augusto Salazar.