



Instituto Politécnico Nacional.  
Escuela Superior De Cómputo.



Materia:  
Desarrollo de Sistemas Distribuidos.

Tema:  
Token-Ring.  
(Tarea 04)

Profesor:  
Carlos Pineda Guerrero.

Alumno:  
Mario Alberto Miranda Sandoval.

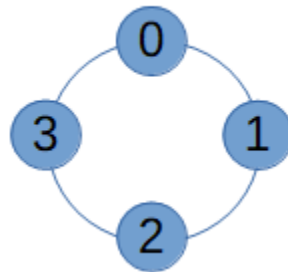
Grupo:  
4CM5

## Objetivo.

Desarrollar un programa en Java, el cual implementará un token que pasará de un nodo a otro nodo, en una topología lógica de anillo. El anillo constará de cuatro nodos, del 0 al 3.

## Desarrollo.

Para desarrollar la siguiente práctica se usará la siguiente topología.



Su funcionamiento es el siguiente, el nodo 0 inicializa el token cuando es la primera vez, posteriormente lo incrementará y lo mandará al nodo 1, este volverá a incrementar el token y lo enviará al nodo 2, el nodo 2 repite el mismo paso, enviando el token incrementado al nodo 3 y el nodo 3 incrementa y envía el token al nodo 0.

Se debe aclarar que para probar el token-ring cada nodo en la topología es una máquina virtual, accediendo a ellas mediante RPD, una vez creadas las máquinas virtuales, iremos al apartado de redes y ahí en el botón de agregar regla de seguridad de entrada en donde dice "Intervalos de puertos de destino" colocaremos el valor 50000, protocolo TCP y de nombre "Port\_50000" como se ve en la siguiente imagen.

La imagen muestra la interfaz de configuración de una regla de seguridad de entrada en Azure. El título de la ventana es "Agregar regla de seguridad de entrada" con un icono de escudo y una "X" para cerrar. Debajo del título, se indica "nodo-4-nsg".

La configuración visible es:

- Básica:** Seleccionado.
- Destino:** Any.
- Intervalos de puertos de destino:** 50000 (con un icono de verificación verde).
- Protocolo:** Any (seleccionado), con opciones para TCP, UDP y ICMP.
- Acción:** Permitir (seleccionado), con opción para Denegar.
- Prioridad:** 320.
- Nombre:** Port\_50000.

Posteriormente se podrá observar en redes como esta la nueva regla que nosotros añadimos.

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción	
300	⚠ RDP	3389	TCP	Cualquiera	Cualquiera	✔ Permitir	***
310	Port_50000	50000	TCP	Cualquiera	Cualquiera	✔ Permitir	***
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	✔ Permitir	***
65001	AllowAzureLoadBalancer...	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	✔ Permitir	***
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	✖ Denegar	***

Este proceso se debe hacer para cada máquina virtual.

Posteriormente, descargamos el archivo .rpd de cada máquina virtual, para acceder a el creamos una carpeta y disco lógico con subst <unidad>: <nombre carpeta>, una vez hecho esto, seleccionando él .rpd, iremos a modificar, recursos locales, más, seleccionamos la unidad lógica donde lo queremos montar, ahora para acceder al escritorio remoto le damos a conectar, introducimos el usuario y contraseña que colocamos al crear la máquina virtual, le damos a aceptar y luego a "si", se nos dará acceso al escritorio remoto, este proceso se hace para todas las máquinas.

Una vez hecho esto se puede observar de la siguiente manera.



Cabe destacar que fue necesario instalar y configurar el jdk en cada máquina virtual. Posteriormente compilamos cada programa en cada máquina virtual y lo ejecutamos añadiendo el número de nodo y la IP siguiente de cada máquina virtual.

```
Administrator: C:\windows\system32\cmd.exe - java Token 0 13.65.202.136
C:\Users\mario\Desktop>javac Token.java
C:\Users\mario\Desktop>java Token 0 13.65.202.136

Administrator: C:\windows\system32\cmd.exe - java Token 1 65.52.36.119
C:\Users\mario\Desktop>javac Token.java
C:\Users\mario\Desktop>java Token 1 65.52.36.119

Administrator: C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.1518]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\mario>cd Desktop
C:\Users\mario\Desktop>javac Token.java
C:\Users\mario\Desktop>
```

```
Administrator: C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.1518]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\mario>cd Desktop

C:\Users\mario\Desktop>javac Token.java

C:\Users\mario\Desktop>_
```

Ahora la topología no funcionara en los siguientes dos casos, si no están conectadas las máquinas entre sí y si el programa está en ejecución y una máquina deja de ejecutar el programa el resto de las máquinas terminan su ejecución con error.

Entonces, cuando se ejecuta la primera máquina, se queda a la escucha ya que esta esperando a conectarse a la IP de la siguiente máquina, ahora una vez que se conectó, lo que evita que el programa siga de largo es que el socket queda a la espera de la entrada del valor del token, mientras la última máquina no se conecte a la primera no podrá iniciar esta y el proceso no iniciara.

Una vez conectadas las máquinas el token se comienza a enviar y se puede observar como el token se va incrementando.

```
Administrator: C:\windows\system32\cmd.exe - java Token 0 13.65.202.136
2125
2129
2133
2137
2141
2145
2149
2153
2157
2161
2165
2169
2173
2177
2181
2185
2189
2193
2197
2201
2205
2209
2213
2217
2221
2225
2229
2233
2237
```

Ejecución en el nodo 0.

```
Administrator: C:\windows\system32\cmd.exe - java Token 1 65.52.36.119
1502
1506
1510
1514
1518
1522
1526
1530
1534
1538
1542
1546
1550
1554
1558
1562
1566
1570
1574
1578
1582
1586
1590
1594
1598
1602
1606
1610
1614
```

Ejecución en el nodo 1.

```
Administrator: C:\windows\system32\cmd.exe - java Token 2 40.124.52.99
1671
1675
1679
1683
1687
1691
1695
1699
1703
1707
1711
1715
1719
1723
1727
1731
1735
1739
1743
1747
1751
1755
1759
1763
1767
1771
1775
1779
1783
```

Ejecución en el nodo 2.

```
Administrator: C:\windows\system32\cmd.exe - java Token 3 13.65.224.87
2032
2036
2040
2044
2048
2052
2056
2060
2064
2068
2072
2076
2080
2084
2088
2092
2096
2100
2104
2108
2112
2116
2120
2124
2128
2132
2136
2140
2144
```

Ejecución en el nodo 3.