

# Actividad: Implementación de un token-ring

Instituto Politécnico Nacional  
Escuela Superior de Computo  
Desarrollo de Sistemas Distribuidos  
Adrian González Pardo  
4CV1  
21/01

22 de octubre de 2020

## 1. Código fuente:

```
1 import java.io.DataOutputStream;
2 import java.io.DataInputStream;
3 import java.net.Socket;
4 import java.net.ServerSocket;
5 import java.lang.Thread;
6
7 /*
8  * @author Adrian Gonzalez Pardo
9  */
10
11
12 class Token{
13     static DataInputStream entrada;
14     static DataOutputStream salida;
15     static boolean primera_vez = true;
16     static String ip;
17     static long token = 0;
18     static int nodo;
19
20     static class Worker extends Thread{
21         public void run(){
22             //Algoritmo 1
23             try{
24                 ServerSocket servidor=new ServerSocket(50000);
25                 Socket conexion=servidor.accept();
26                 entrada=new DataInputStream(conexion.getInputStream());
27             }catch(Exception e){
28                 System.err.println(e.getMessage());
29             }
30         }
31     }
32
33     public static void main(String[] args) throws Exception{
34         if (args.length != 2){
35             System.err.println("Se debe pasar como parametros el numero de nodo y la
36             IP del siguiente nodo");
37             System.exit(1);
38         }
39
40         nodo = Integer.valueOf(args[0]); // el primer parametro es el numero de
41         ip = args[1]; // el segundo parametro es la IP del siguiente nodo en el
42         anillo
```

```

41 //Algoritmo 2
42 Worker w=new Worker();
43 w.start();
44 Socket conexion=null;
45 while(true){
46     try{
47         conexion=new Socket(ip,50000);
48         break;
49     }catch (Exception e){
50         Thread.sleep(500);
51     }
52 }
53 salida=new DataOutputStream(conexion.getOutputStream());
54 w.join();
55 for(;;){
56     if(nodo==0){
57         if(primeravez){ /*True*/
58             primera_vez=!primera_vez;
59         }else{
60             token=entrada.readLong();
61         }
62     }else{
63         token=entrada.readLong();
64     }
65     token++;
66     System.out.println("Nodo:"+nodo+" Token: "+token);
67     salida.writeLong(token);
68     break;
69 }
70 }
71 }

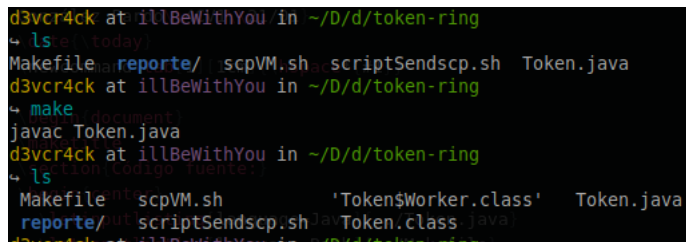
```

```

1 # Archivo Makefile
2 # @author Adrian Gonzalez Pardo
3 JVC=javac
4 SRCC=$(wildcard *.java)
5 OBJS=$(SRCC:.java=.class)
6 all: ${OBJS}
7
8 %.class: %.java
9     ${JVC} $<
10
11 .PHONY: clean
12
13 clean:
14     rm *.class

```

## 2. Capturas y descripción del programa

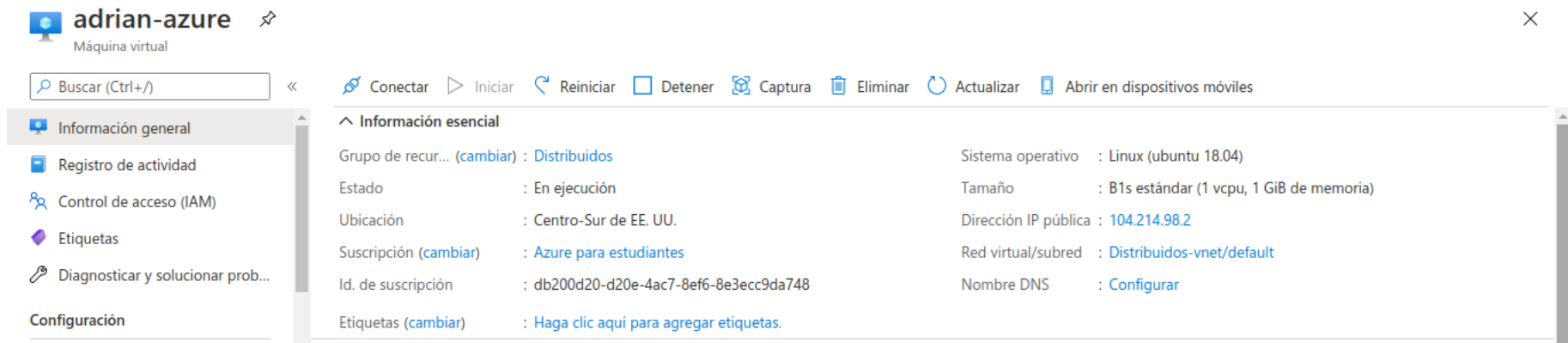


```

d3vcr4ck at illBeWithYou in ~/D/d/token-ring
↳ ls
Makefile  reporte/  scpVM.sh  scriptSendscp.sh  Token.java
d3vcr4ck at illBeWithYou in ~/D/d/token-ring
↳ make
javac Token.java
d3vcr4ck at illBeWithYou in ~/D/d/token-ring
↳ ls
Makefile  scpVM.sh  'Token$Worker.class'  Token.java
reporte/  scriptSendscp.sh  Token.class  java:
d3vcr4ck at illBeWithYou in ~/D/d/token-ring

```

Figura 1: Compilación



**adrian-azure** Máquina virtual

Buscar (Ctrl+/) «

Conectar ▶ Iniciar ↺ Reiniciar □ Detener 📷 Captura 🗑 Eliminar ↻ Actualizar 📱 Abrir en dispositivos móviles

**Información general**

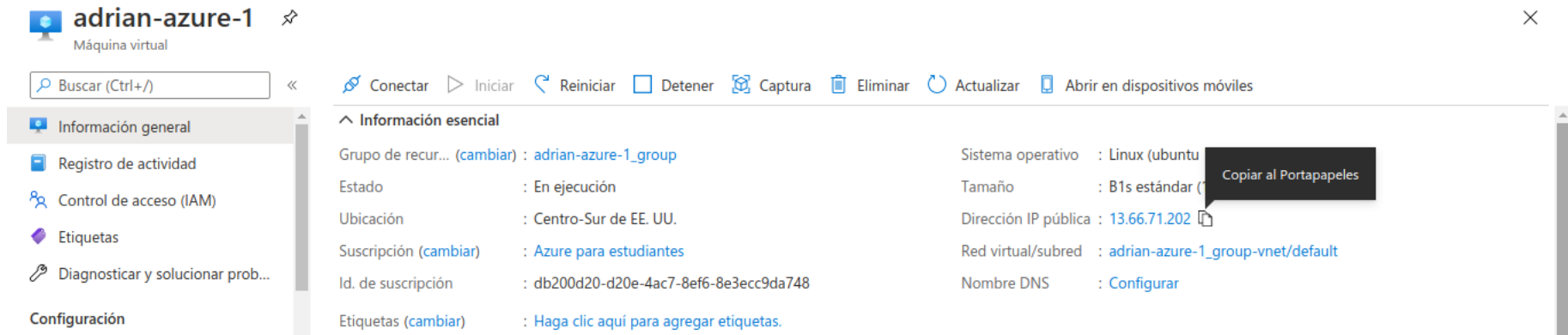
- Registro de actividad
- Control de acceso (IAM)
- Etiquetas
- Diagnosticar y solucionar prob...

**Configuración**

**Información esencial**

Grupo de recur... (cambiar) :	Distribuidos	Sistema operativo :	Linux (ubuntu 18.04)
Estado :	En ejecución	Tamaño :	B1s estándar (1 vcpu, 1 GiB de memoria)
Ubicación :	Centro-Sur de EE. UU.	Dirección IP pública :	104.214.98.2
Suscripción (cambiar) :	Azure para estudiantes	Red virtual/subred :	Distribuidos-vnet/default
Id. de suscripción :	db200d20-d20e-4ac7-8ef6-8e3ecc9da748	Nombre DNS :	Configurar
Etiquetas (cambiar) :	Haga clic aquí para agregar etiquetas.		

Figura 2:Maquina virtual 1 en Azure de Ubuntu Server 18



**adrian-azure-1** Máquina virtual

Buscar (Ctrl+/) «

Conectar ▶ Iniciar ↺ Reiniciar □ Detener 📷 Captura 🗑 Eliminar ↻ Actualizar 📱 Abrir en dispositivos móviles

**Información general**

- Registro de actividad
- Control de acceso (IAM)
- Etiquetas
- Diagnosticar y solucionar prob...

**Configuración**

**Información esencial**

Grupo de recur... (cambiar) :	adrian-azure-1_group	Sistema operativo :	Linux (ubuntu 18.04)
Estado :	En ejecución	Tamaño :	B1s estándar (1 vcpu, 1 GiB de memoria)
Ubicación :	Centro-Sur de EE. UU.	Dirección IP pública :	13.66.71.202 📋
Suscripción (cambiar) :	Azure para estudiantes	Red virtual/subred :	adrian-azure-1_group-vnet/default
Id. de suscripción :	db200d20-d20e-4ac7-8ef6-8e3ecc9da748	Nombre DNS :	Configurar
Etiquetas (cambiar) :	Haga clic aquí para agregar etiquetas.		

Figura 3:Maquina virtual 2 en Azure de Ubuntu Server 18

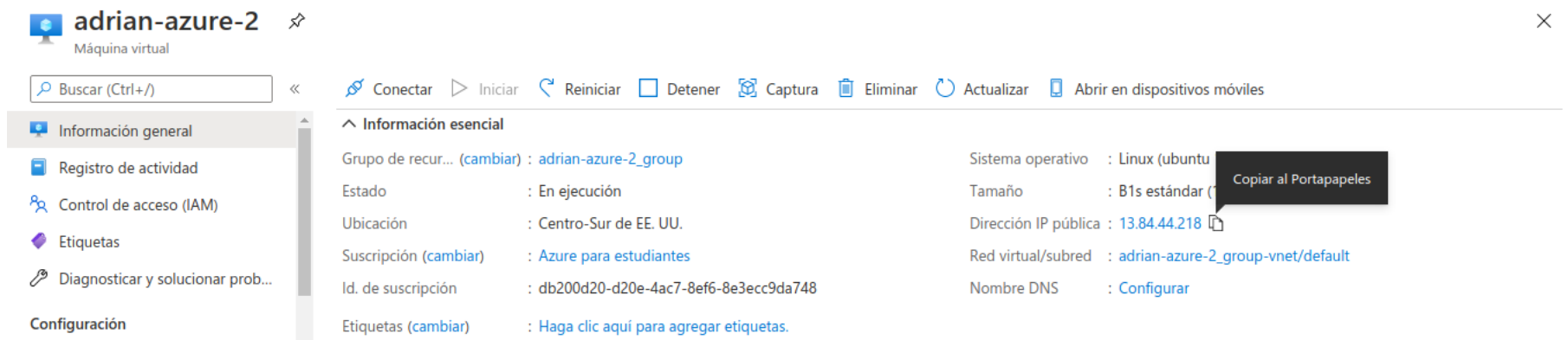


Figura 4: Máquina virtual 3 en Azure de Ubuntu Server 18

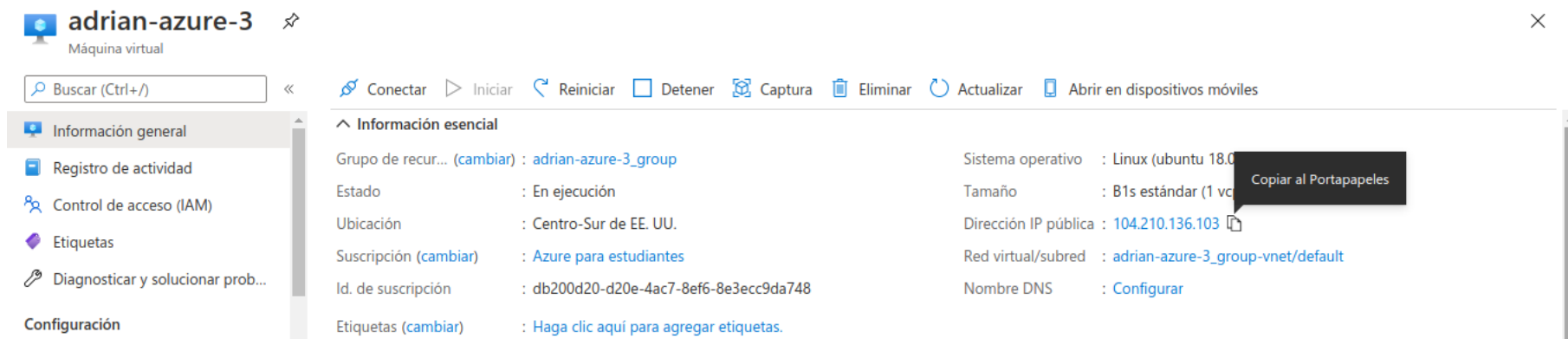


Figura 5: Máquina virtual 4 en Azure de Ubuntu Server 18

Para las máquinas virtuales es necesario conocer las IP de las máquinas virtuales y con ello evitamos escribir muchas veces el comando scp ejecutando el siguiente script, al igual que es necesaria la instalación de los paquetes openjdk-11-jdk y cmake

```

1 #!/usr/bin/env bash
2 # @author Adrian Gonzalez Pardo
3 usr=adrian
4
5 sshpass -p "adrianPardo_1" scp *.java Makefile $usr@104.214.98.2:~/token-ring
6 sshpass -p "adrianPardo_1" scp *.java Makefile $usr@13.66.71.202:~/token-ring
7 sshpass -p "adrianPardo_1" scp *.java Makefile $usr@13.84.44.218:~/token-ring
8 sshpass -p "adrianPardo_1" scp *.java Makefile $usr@104.210.136.103:~/token-ring

```

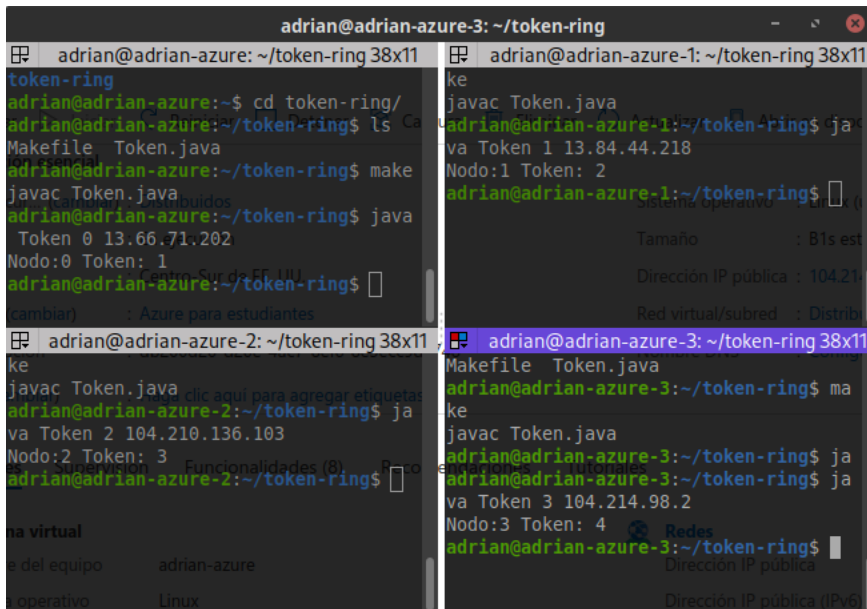


Figura 6: Muestra de la ejecución en cada VM vía ssh

Ahora bien en caso de querer realizar una implementación a nivel red Local con al menos 4 equipos, se puede hacer lo siguiente (destacando que ya hay un emparejamiento de llaves ssh)

A nivel local se conoce la lista de los siguientes equipos:

- Lenovo IP: 192.168.100.69
- Acer IP: 192.168.100.3
- Raspberry Pi 3B IP: 192.168.100.194
- Raspberry Pi 4B IP: 192.168.100.103

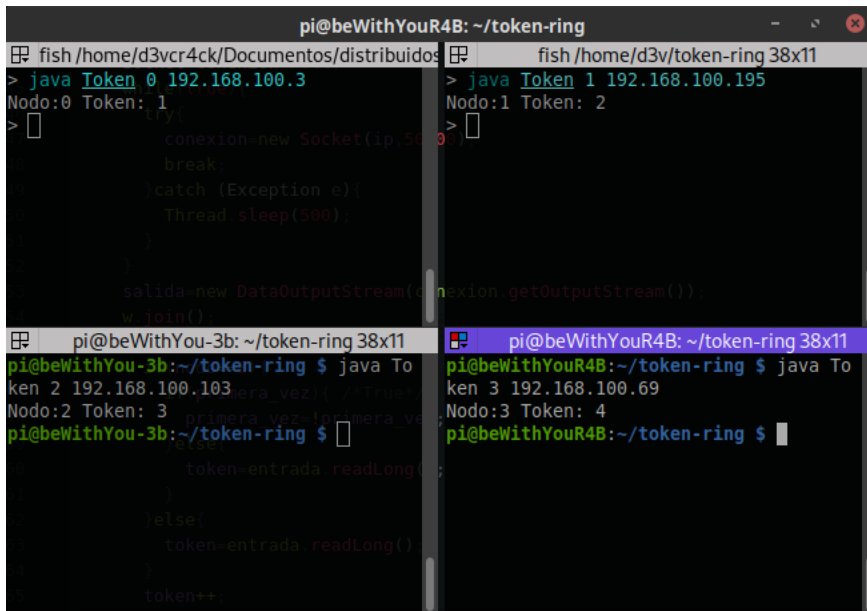
Para el cual se envío los datos con el siguiente script:

```

1 #!/usr/bin/env bash
2 # @author Adrian Gonzalez Pardo
3
4 scp *.java Makefile pi@192.168.100.103:~/token-ring
5 scp *.java Makefile pi@192.168.100.194:~/token-ring
6 scp *.java Makefile d3v@192.168.100.3:~/token-ring

```

Y finalmente se visualiza así:



```
pi@beWithYouR4B: ~/token-ring
fish /home/d3vcr4ck/Documents/distribuidos
> java Token 0 192.168.100.3
Nodo:0 Token: 1
>
    try {
        conexion=new Socket(ip,5000);
        break;
    } catch (Exception e){
        Thread.sleep(500);
    }
}

salida=new DataOutputStream(conexion.getOutputStream());
w.join();

pi@beWithYou-3b: ~/token-ring 38x11
pi@beWithYou-3b:~/token-ring $ java To
ken 2 192.168.100.103
Nodo:2 Token: 3
pi@beWithYou-3b:~/token-ring $
    primera_vez=true;
    primera_vez=false;
    token=entrada.readLong();
}
else{
    token=entrada.readLong();
}
token++;

pi@beWithYouR4B: ~/token-ring 38x11
fish /home/d3v/token-ring 38x11
> java Token 1 192.168.100.195
Nodo:1 Token: 2
>
    try {
        conexion=new Socket(ip,5000);
        break;
    } catch (Exception e){
        Thread.sleep(500);
    }
}

salida=new DataOutputStream(conexion.getOutputStream());
w.join();

pi@beWithYouR4B:~/token-ring $ java To
ken 3 192.168.100.69
Nodo:3 Token: 4
pi@beWithYouR4B:~/token-ring $
```

Figura 7: Muestra de la ejecución en cada Nodo a nivel local vía ssh