

INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO



TAREA 2 IMPLEMENTACIÓN DE UN
TOKEN-RING

MATERIA: DISEÑO DE SISTEMAS
DISTRIBUIDOS

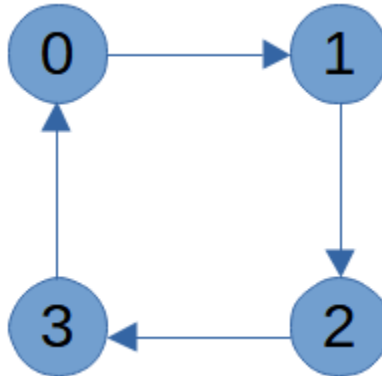
GRUPO: 4CV13

ALUMNO: MORA GUZMÁN JOSE ANTONIO

FECHA ENTREGA: VIERNES 10 SEPTIEMBRE
2021

DESCRIPCIÓN DE LA TAREA

Desarrollar un programa en Java, el cual implementará un token que se enviará de un nodo a otro nodo mediante **sockets seguros**, en una topología lógica de anillo. El anillo constará de cuatro nodos:



El token será un número entero de 64 bits.

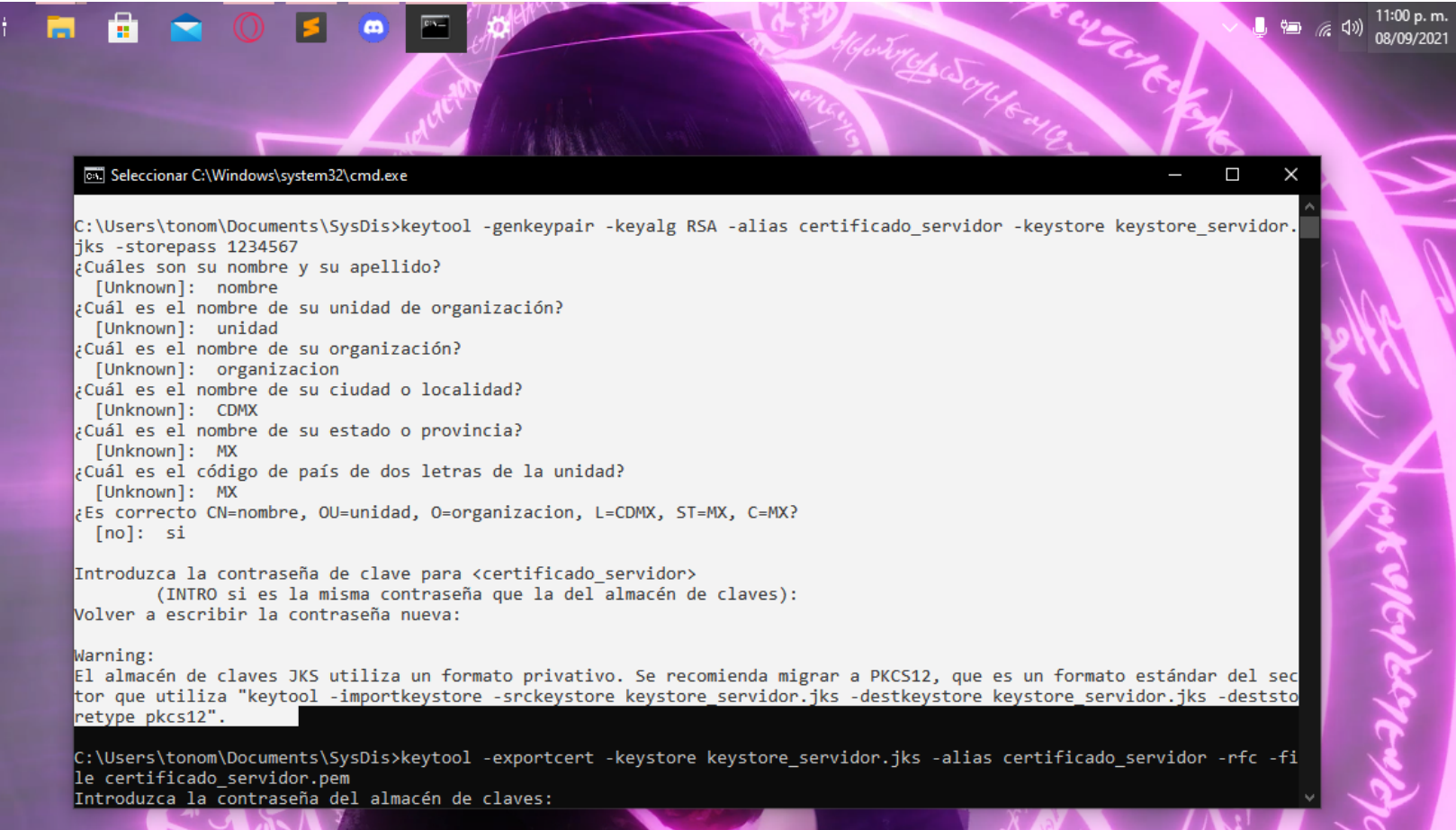
1. Al principio el nodo 0 enviará el token al nodo 1.
2. El nodo 1 recibirá el token y lo enviará al nodo 2.
3. El nodo 2 recibirá el token y lo enviará al nodo 3.
4. El nodo 3 recibirá el token y lo enviará al nodo 0.
5. El nodo 0 recibirá el token y lo enviará al nodo 1.
6. Ir al paso 2.

Cuando un nodo reciba el token lo incrementará en 1 y desplegará el valor del token.

Cuando la cuenta en el nodo 0 llegue a 1000, el nodo 0 deberá terminar su ejecución.

Vamos a probar el programa en una sola computadora utilizando cuatro ventanas de comandos de Windows o cuatro terminales de Linux o MacOS, cada ventana ejecutará una instancia del programa.

GENERACIÓN DE LOS KEYSTORE CLIENTE Y SERVIDOR



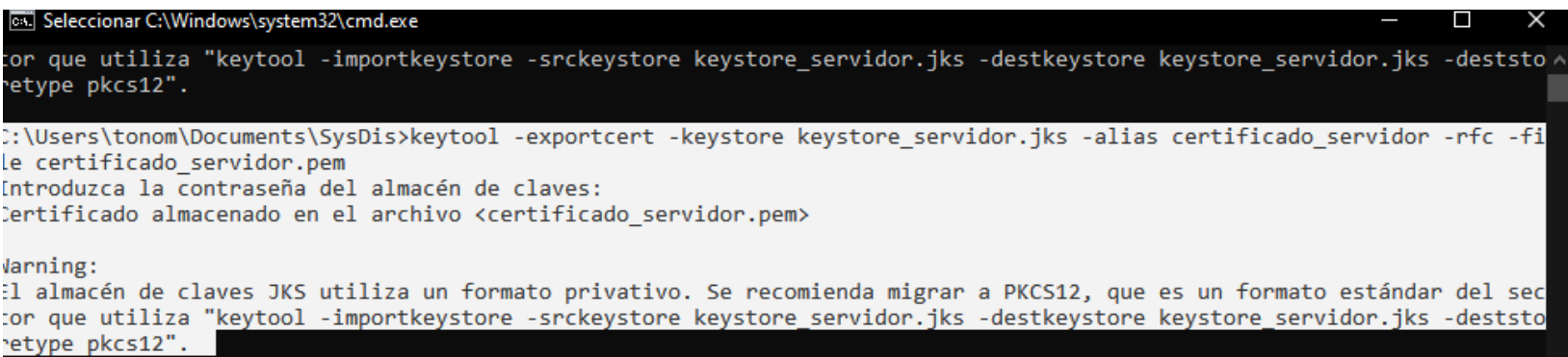
A screenshot of a Windows command prompt window titled "Seleccionar C:\Windows\system32\cmd.exe". The window shows the execution of the following commands and their output:

```
C:\Users\tonom\Documents\SysDis>keytool -genkeypair -keyalg RSA -alias certificado_servidor -keystore keystore_servidor.jks -storepass 1234567
¿Cuáles son su nombre y su apellido?
[Unknown]: nombre
¿Cuál es el nombre de su unidad de organización?
[Unknown]: unidad
¿Cuál es el nombre de su organización?
[Unknown]: organizacion
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: CDMX
¿Cuál es el nombre de su estado o provincia?
[Unknown]: MX
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: MX
¿Es correcto CN=nombre, OU=unidad, O=organizacion, L=CDMX, ST=MX, C=MX?
[no]: si

Introduzca la contraseña de clave para <certificado_servidor>
(INTRO si es la misma contraseña que la del almacén de claves):
Volver a escribir la contraseña nueva:

Warning:
El almacén de claves JKS utiliza un formato privativo. Se recomienda migrar a PKCS12, que es un formato estándar del sector que utiliza "keytool -importkeystore -srckeystore keystore_servidor.jks -destkeystore keystore_servidor.jks -deststoretype pkcs12".

C:\Users\tonom\Documents\SysDis>keytool -exportcert -keystore keystore_servidor.jks -alias certificado_servidor -rfc -file certificado_servidor.pem
Introduzca la contraseña del almacén de claves:
```



A continuation of the command prompt window showing the migration of the keystore to PKCS12 format:

```
Introduzca la contraseña del almacén de claves:
Certificado almacenado en el archivo <certificado_servidor.pem>

Warning:
El almacén de claves JKS utiliza un formato privativo. Se recomienda migrar a PKCS12, que es un formato estándar del sector que utiliza "keytool -importkeystore -srckeystore keystore_servidor.jks -destkeystore keystore_servidor.jks -deststoretype pkcs12".
```

```
C:\Users\tonom\Documents\SysDis>keytool -import -alias certificado_servidor -file certificado_servidor.pem -keystore key  
store_cliente.jks -storepass 123456  
Propietario: CN=nombre, OU=unidad, O=organizacion, L=CDMX, ST=MX, C=MX  
Emisor: CN=nombre, OU=unidad, O=organizacion, L=CDMX, ST=MX, C=MX  
Número de serie: 60cf7008  
Válido desde: Wed Sep 08 22:04:18 CDT 2021 hasta: Tue Dec 07 21:04:18 CST 2021  
Huellas digitales del certificado:  
SHA1: 06:44:1C:36:5B:63:B1:14:76:C8:2B:BE:EE:F9:60:DC:B3:38:ED:C3  
SHA256: FE:9B:9C:32:63:30:94:07:05:1E:AF:7A:E6:41:40:F8:73:D3:88:41:7F:C1:E9:56:9F:F5:AB:15:13:7F:F2:9F  
Nombre del algoritmo de firma: SHA256withRSA  
Algoritmo de clave pública de asunto: Clave RSA de 2048 bits  
Versión: 3  
  
Extensiones:  
  
#1: ObjectId: 2.5.29.14 Criticality=false  
SubjectKeyIdentifier [  
KeyIdentifier [  
0000: FA 52 C1 A9 C7 EF 9E EE A8 34 83 77 A4 BF 05 2C .R.....4.w...,  
0010: 54 A8 52 6A T.Rj  
]  
]  
  
¿Confiar en este certificado? [no]: si  
Se ha agregado el certificado al almacén de claves  
  
C:\Users\tonom\Documents\SysDis>
```

COMPILACIÓN

```
C:\Windows\system32\cmd.exe  
  
C:\Users\tonom\Documents\SysDis>javac Token.java  
  
C:\Users\tonom\Documents\SysDis>java Token 0 localhost  
nodo: 0 token: 5
```

EJECUCIÓN

```
Seleccionar C:\Windows\system32\cmd.exe
nodo: 0 token: 953
nodo: 0 token: 957
nodo: 0 token: 961
nodo: 0 token: 965
nodo: 0 token: 969
nodo: 0 token: 973
nodo: 0 token: 977
nodo: 0 token: 981
nodo: 0 token: 985
nodo: 0 token: 989
nodo: 0 token: 993
nodo: 0 token: 997
nodo: 0 token: 1001
C:\Users\tonom\Documents\SysDis>
```

C:\Windows\system32\cmd.exe

nodo: 1 token: 954

nodo: 1 token: 958

nodo: 1 token: 962

nodo: 1 token: 966

nodo: 1 token: 970

nodo: 1 token: 974

nodo: 1 token: 978

nodo: 1 token: 982

nodo: 1 token: 986

nodo: 1 token: 990

nodo: 1 token: 994

nodo: 1 token: 998

Exception in thread "main" java.io.EOFException
 at java.io.DataInputStream.readFully(Unknown Source)
 at java.io.DataInputStream.readLong(Unknown Source)
 at Token.main(Token.java:83)

C:\Users\tonom\Documents\SysDis>

C:\Windows\system32\cmd.exe

nodo: 2 token: 999

```
Exception in thread "main" javax.net.ssl.SSLException: java.net.SocketException: Connection reset
    at sun.security.ssl.Alert.createSSLException(Unknown Source)
    at sun.security.ssl.TransportContext.fatal(Unknown Source)
    at sun.security.ssl.TransportContext.fatal(Unknown Source)
    at sun.security.ssl.TransportContext.fatal(Unknown Source)
    at sun.security.ssl.SSLSocketImpl.handleException(Unknown Source)
    at sun.security.ssl.SSLSocketImpl.access$400(Unknown Source)
    at sun.security.ssl.SSLSocketImpl$AppInputStream.read(Unknown Source)
    at java.io.DataInputStream.readFully(Unknown Source)
    at java.io.DataInputStream.readLong(Unknown Source)
    at Token.main(Token.java:83)
    Suppressed: java.net.SocketException: Connection reset by peer: socket write error
        at java.net.SocketOutputStream.socketWrite0(Native Method)
        at java.net.SocketOutputStream.socketWrite(Unknown Source)
        at java.net.SocketOutputStream.write(Unknown Source)
        at sun.security.ssl.SSLSocketOutputRecord.encodeAlert(Unknown Source)
        ... 9 more
Caused by: java.net.SocketException: Connection reset
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at sun.security.ssl.SSLSocketInputRecord.read(Unknown Source)
    at sun.security.ssl.SSLSocketInputRecord.readHeader(Unknown Source)
    at sun.security.ssl.SSLSocketInputRecord.bytesInCompletePacket(Unknown Source)
    at sun.security.ssl.SSLSocketImpl.readApplicationRecord(Unknown Source)
    at sun.security.ssl.SSLSocketImpl.access$300(Unknown Source)
    ... 4 more
```

C:\Users\tonom\Documents\SysDis>

C:\Windows\system32\cmd.exe

nodo: 3 token: 1000

```
Exception in thread "main" javax.net.ssl.SSLException: java.net.SocketException: Connection reset
    at sun.security.ssl.Alert.createSSLException(Unknown Source)
    at sun.security.ssl.TransportContext.fatal(Unknown Source)
    at sun.security.ssl.TransportContext.fatal(Unknown Source)
    at sun.security.ssl.TransportContext.fatal(Unknown Source)
    at sun.security.ssl.SSLSocketImpl.handleException(Unknown Source)
    at sun.security.ssl.SSLSocketImpl.access$400(Unknown Source)
    at sun.security.ssl.SSLSocketImpl$AppInputStream.read(Unknown Source)
    at java.io.DataInputStream.readFully(Unknown Source)
    at java.io.DataInputStream.readLong(Unknown Source)
    at Token.main(Token.java:83)
    Suppressed: java.net.SocketException: Connection reset by peer: socket write error
        at java.net.SocketOutputStream.socketWrite0(Native Method)
        at java.net.SocketOutputStream.socketWrite(Unknown Source)
        at java.net.SocketOutputStream.write(Unknown Source)
        at sun.security.ssl.SSLSocketOutputRecord.encodeAlert(Unknown Source)
        ... 9 more
Caused by: java.net.SocketException: Connection reset
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at sun.security.ssl.SSLSocketInputRecord.read(Unknown Source)
    at sun.security.ssl.SSLSocketInputRecord.readHeader(Unknown Source)
    at sun.security.ssl.SSLSocketInputRecord.bytesInCompletePacket(Unknown Source)
    at sun.security.ssl.SSLSocketImpl.readApplicationRecord(Unknown Source)
    at sun.security.ssl.SSLSocketImpl.access$300(Unknown Source)
    ... 4 more
```

C:\Users\tonom\Documents\SysDis>

CONCLUSIÓN

Esta practica me sirvió mucho de para aprender un poco mas a manejar los sockets seguros que si bien hacer el programa sin tokens seguros fue sencillo tuve varios problemas al hacerlo con tokens seguros debido a que no agregaba los parámetros del cliente y servidor en el código si no solo al ejecutar y así no me funcionaba. Pero después de estar intentando me salió