

ALGORITMO

NAT



Proyecto final de la EE Sistemas Operativos
Facultad de Estadística e Informática
Alumnos:

- **Carim Velázquez Chicuellar**
- **David Carrión Romero**
- **Emmanuel Pale Molina**

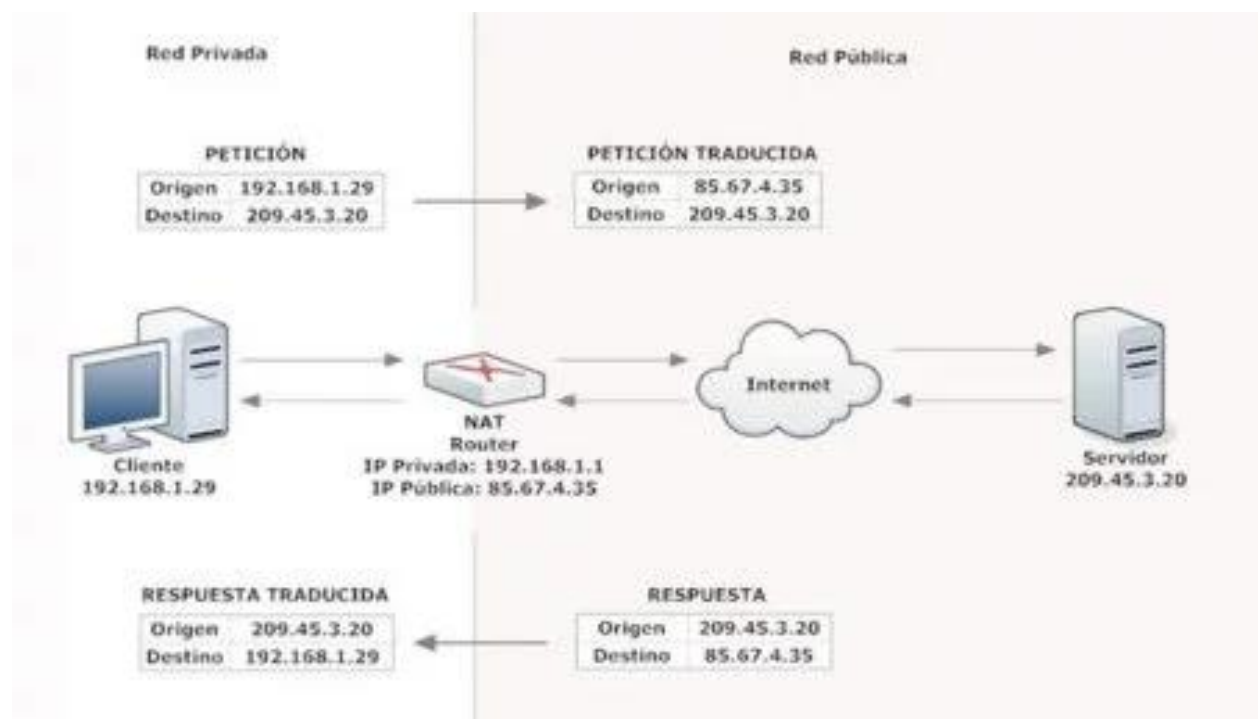


Algoritmo NAT

NAT o Network Address Translation fue desarrollado después de que el número de máquinas conectadas a Internet creciera exponencialmente y las direcciones IP se comenzaran a agotar, ya que están compuestas por 32 bits (equivalente a 4,294,967,296 direcciones únicas).

Funciona de la siguiente manera:

1. Una red de ordenadores utiliza un rango de direcciones IP privadas.
2. Para conectarse a Internet utilizan una única dirección IP (IP pública) proporcionada por el enrutador de Internet al que se conectan.



De esta manera disminuye considerablemente el uso de IP's públicas ya que, en el caso de una grande empresa, solo necesita utilizar una dirección IP pública y no tantas como máquinas esté utilizando.

Desarrollo del proyecto.

Lenguaje: C

Plataforma: Unix

El desarrollo de un algoritmo informático que sea capaz de representar el funcionamiento de NAT debe constar de tres elementos representativos fundamentales:

1. Computadoras.
2. Router de internet
3. Servidor

Cada uno ha sido representado como un proceso distinto, por lo tanto, fue necesario implementar un método para la comunicación entre procesos, el cual fue Sockets.

Computadoras

Archivo: pc.c

Envía un mensaje (cadena de caracteres) al proceso Router porque ese mensaje necesita ser manipulado por el proceso servidor y ser enviado de vuelta al proceso que lo mandó inicialmente, representando una petición de conexión entre la computadora y el Router.

Router de internet

Archivo: router.c

Después de iniciar las conexiones necesarias, espera a un cliente (proceso PC) que mande un mensaje, seguido de ello manda el mensaje al servidor, el cual lo modifica y lo manda de regreso al Router, y éste lo manda directamente de regreso al cliente que mandó el mensaje.

Este proceso establece dos conexiones:

1. Cuando recibe el mensaje del
2. La segunda vez es cuando recibe el mensaje manipulado por el proceso servidor y lo hace llegar nuevamente al proceso PC

Servidor

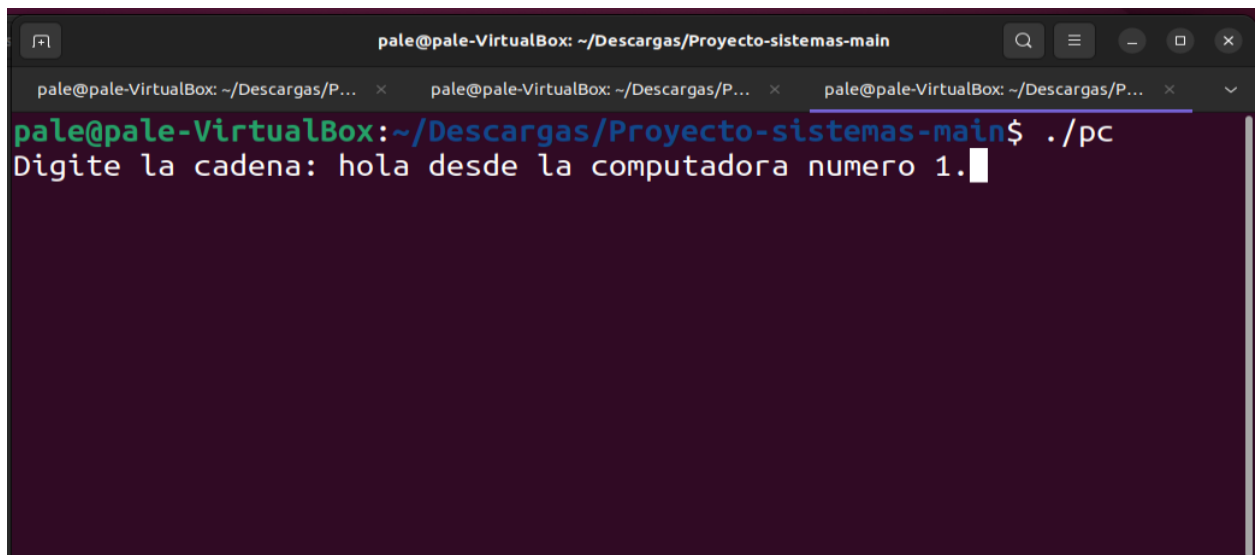
Archivo: server.c

El proceso servidor se encarga de manipular los mensajes recibidos por el Router (sin importar de que proceso PC fue enviado). Contiene una función que convierte a mayúsculas toda la cadena recibida. Después de manipular la cadena se regresa nuevamente al proceso Router.

En un escenario real representa el momento donde la petición fue aprobada por el Router y la hizo llegar al servidor, para trabajar con los datos enviados y nuevamente regresarlos.

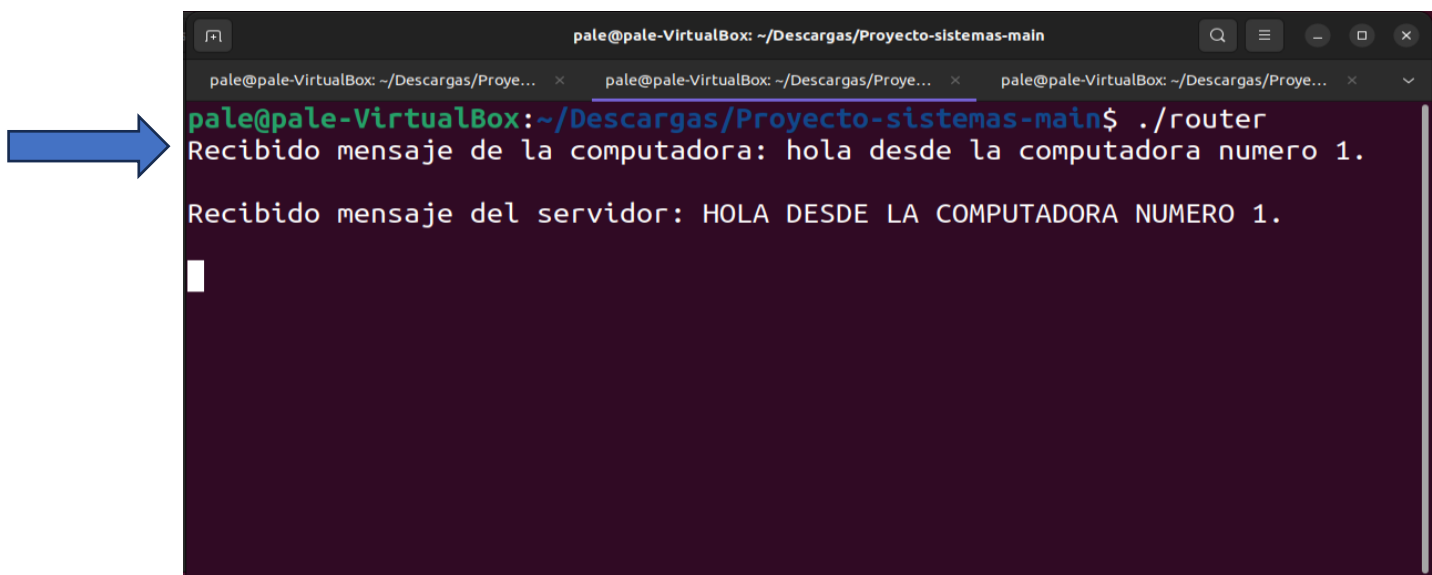
Implementación

1. Se ejecuta el programa Router
2. Se ejecuta el programa Servidor
3. Se ejecuta el programa PC el cual pedirá que introduzca un mensaje



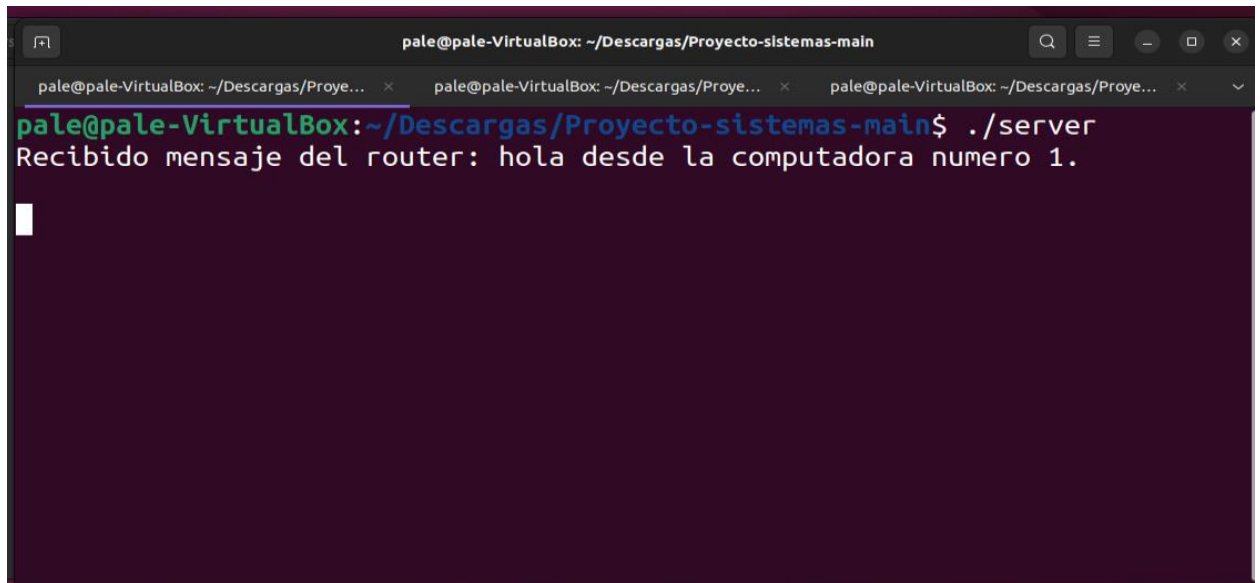
```
pale@pale-VirtualBox: ~/Descargas/Proyecto-sistemas-main
pale@pale-VirtualBox: ~/Descargas/Proyecto-sistemas-main$ ./pc
Digite la cadena: hola desde la computadora numero 1.
```

4. Después de enviar el mensaje se puede dirigir a la ventana donde está ejecutando el proceso Router para verificar que el mensaje lo recibió.



```
pale@pale-VirtualBox: ~/Descargas/Proyecto-sistemas-main
pale@pale-VirtualBox: ~/Descargas/Proyecto-sistemas-main$ ./router
Recibido mensaje de la computadora: hola desde la computadora numero 1.
Recibido mensaje del servidor: HOLA DESDE LA COMPUTADORA NUMERO 1.
```

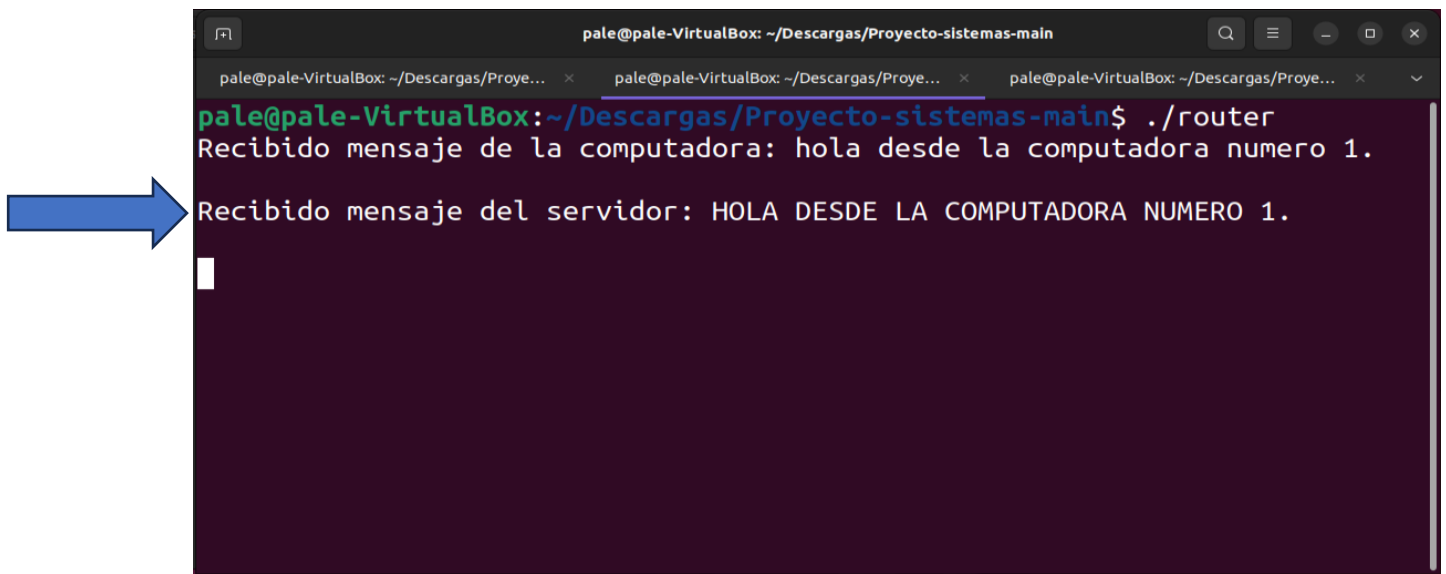
5. Para verificar que el mensaje fue enviado por el proceso Router y recibido por el proceso Servidor se puede dirigir a la ventana donde se ejecuta el proceso Servidor.



A terminal window titled 'pale@pale-VirtualBox: ~/Descargas/Proyecto-sistemas-main'. The prompt is 'pale@pale-VirtualBox:~/Descargas/Proyecto-sistemas-main\$'. The command './server' has been executed, and the output is 'Recibido mensaje del router: hola desde la computadora numero 1.'.

```
pale@pale-VirtualBox: ~/Descargas/Proyecto-sistemas-main$ ./server
Recibido mensaje del router: hola desde la computadora numero 1.
```


6. Después, como el algoritmo de NAT lo indica, ese mensaje debe ser devuelto al Router ya modificado.



A terminal window titled 'pale@pale-VirtualBox: ~/Descargas/Proyecto-sistemas-main'. The prompt is 'pale@pale-VirtualBox:~/Descargas/Proyecto-sistemas-main\$'. The command './router' has been executed, and the output is 'Recibido mensaje de la computadora: hola desde la computadora numero 1.' followed by 'Recibido mensaje del servidor: HOLA DESDE LA COMPUTADORA NUMERO 1.' on the next line. A blue arrow points to the second line of output.

```
pale@pale-VirtualBox:~/Descargas/Proyecto-sistemas-main$ ./router
Recibido mensaje de la computadora: hola desde la computadora numero 1.
Recibido mensaje del servidor: HOLA DESDE LA COMPUTADORA NUMERO 1.
```

7. Por último, el proceso Router manda ese mensaje ya modificado al proceso PC que inicialmente lo mandó



```
pale@pale-VirtualBox: ~/Descargas/Proyecto-sistemas-main
pale@pale-VirtualBox: ~/Descargas/Proye... x pale@pale-VirtualBox: ~/Descargas/Proye... x pale@pale-VirtualBox: ~/Descargas/Proye... x
pale@pale-VirtualBox:~/Descargas/Proyecto-sistemas-main$ ./pc
Digite la cadena: hola desde la computadora numero 1.
Mensaje del router: HOLA DESDE LA COMPUTADORA NUMERO 1.
pale@pale-VirtualBox:~/Descargas/Proyecto-sistemas-main$
```

Consideraciones

El orden de ejecución de los programas es:

1. Router
2. Server
3. PC

Se debe correr el programa PC tantas máquinas diferentes se quieran conectar al mismo Router.