

# **Protocolos de Capa de Aplicación, Plataforma Base y Capa De Red**

Santiago Botero García

Laura Natalia Perilla Quintero

Escuela Colombiana de Ingeniería Julio Garavito

**AYSR-1L:** Arquitectura y Servicios de Red

Ing. Jhon Alexander Pachón Pinzón

Noviembre 08, 2025

## **Resumen**

El propósito de este laboratorio es profundizar en la comprensión de las capas de infraestructura y comunicación que sustentan los entornos de TI empresariales. Se centra en la instalación, configuración e interconexión de servidores web, enrutadores y sistemas de monitoreo, así como en la implementación de enrutamiento estático y dinámico (EIGRP y OSPF) para garantizar la conectividad total de la red.

Cada uno de los integrantes realizaron ejercicios prácticos utilizando herramientas como Packet Tracer, Wireshark y software de virtualización para simular y analizar el tráfico de datos y el comportamiento de los protocolos en las capas de aplicación, transporte y red.

Además, el laboratorio refuerza la configuración de servicios web (Apache, Nginx, IIS), la resolución DNS y el alojamiento virtual, al tiempo que introduce la gestión y monitoreo de redes en Azure mediante SNMP y Application Insights.

*Palabras clave.* Servidores Web, Enrutamiento, Capa de Aplicación, Gestión de Redes, TCP/IP, OSPF, EIGRP, DNS, Azure, SNMP, Wireshark.

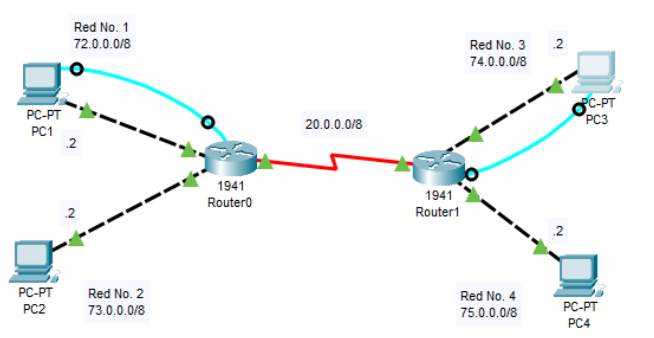
## Contenido

Resumen.....	2
Simulación de Routers .....	4
Implementación de Santiago .....	4
Implementación de Natalia.....	11
Instalación de Software Básico .....	16
Instalación de servicios web.....	16
Servicio de Aplicación Web .....	30
Instalación de Software Base .....	35
Administración de Redes en Azure .....	35
Capa de Transporte (Layer 4) - Confiabilidad y Entrega de Proceso a Proceso .....	40
Capa de Aplicación (Layer 7) - Interpretación Correcta .....	41
Experimentos .....	41
Uso de mensajes ICMP .....	41
Algunas preguntas de comandos para Routers.....	43
Acceso y configuración básica de Routers.....	46
Conclusiones .....	54
Bibliografía .....	56

## Simulación de Routers

En esta sección se llevó a cabo la implementación completa de una red Cisco, replicada por cada uno de los integrantes del grupo de manera independiente, siguiendo los mismos pasos planificados de configuración y conexión. Cada uno configuró sus propios routers y PCs, incluyendo conexiones físicas, asignación de direcciones IPv4, configuración de contraseñas, mensajes del día, descripciones de interfaces, rutas estáticas y pruebas de conectividad entre dispositivos. Esta doble implementación permitió comprobar de forma práctica que las configuraciones y la comunicación entre las distintas redes funcionaban correctamente en ambos casos.

### Implementación de Santiago

Acción Realizada	Captura de Pantalla
<p>Se implementaron las conexiones planificadas de la siguiente manera: las PC01 y PC02 se conectaron al Router0 mediante cables directos, mientras que las PC03 y PC04 se conectaron al Router01 (modelo 1941) también por medio de cables directos. Además, la PC01 se conectó al Router0 mediante un cable de consola, y entre el Router0</p>	 <p>The diagram illustrates a network topology with two routers, Router0 and Router1, connected by a red line representing a serial link. Router0 is connected to PC1 and PC2, and Router1 is connected to PC3 and PC4. The diagram includes IP addresses for each network and the routers.</p> <ul style="list-style-type: none"><li>Red No. 1: 72.0.0.0/8 (connected to Router0)</li><li>Red No. 2: 73.0.0.0/8 (connected to Router0)</li><li>Red No. 3: 74.0.0.0/8 (connected to Router1)</li><li>Red No. 4: 75.0.0.0/8 (connected to Router1)</li><li>Serial link: 20.0.0.0/8 (between Router0 and Router1)</li></ul> <p>Each PC is labeled with its name (PC1, PC2, PC3, PC4) and a '.2' suffix, indicating its IP address. The routers are labeled '1941 Router0' and '1941 Router1'.</p>

<p>y el Router01 se estableció una conexión utilizando un cable serial.</p>	
<p>Se accedió al router desde el PC mediante la consola, usando la terminal con los parámetros por defecto. Se ingresó al modo privilegiado y de configuración global, se asignó de nombre al router Santiago, se estableció un mensaje del día y se configuraron las contraseñas de acceso por consola (ClaveC) y remoto (ClaveT). Finalmente, se deshabilitó la búsqueda de dominios con el comando no ip domain-lookup.</p>	<pre>Router&gt;enable Router#configure terminal Enter configuration commands, one per line. End with CNTRL/Z. Router(config)#hostname Santiago Santiago(config)#banner motd #For exclusive use by RECO students# Santiago(config)#line console 0 Santiago(config-line)#logging synchronous Santiago(config-line)#password ClaveC Santiago(config-line)#login Santiago(config-line)#exit Santiago(config)#line vty 0 15 Santiago(config-line)#logging synchronous Santiago(config-line)#password ClaveT Santiago(config-line)#login Santiago(config-line)#exit Santiago(config)#no ip domain-lookup</pre>
<p>Se configuraron descripciones en las interfaces del router para identificar su función. En cada interfaz utilizada, se accedió al modo de configuración de interfaz y se añadió una descripción correspondiente.</p>	<pre>Santiago(config)#interface GigabitEthernet0/0 Santiago(config-if)#description "LAN Connection to PC01" Santiago(config-if)#exit Santiago(config)#interface GigabitEthernet0/1 Santiago(config-if)#description "LAN Connection to PC02" Santiago(config-if)#exit Santiago(config)#</pre>

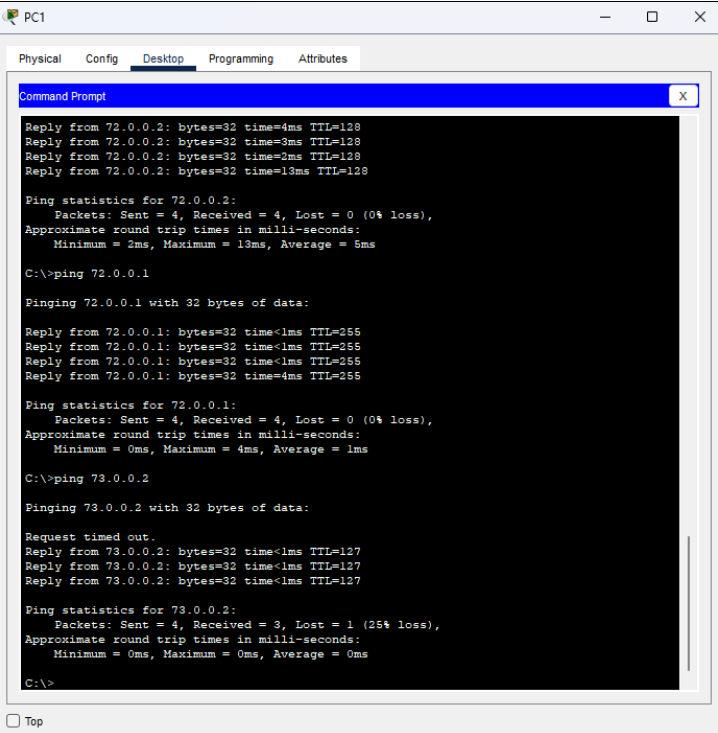
<p>Se configuró la contraseña para el modo privilegiado usando enable secret ClaveE. Luego, se verificó la configuración actual con show running-config y, finalmente, se guardaron los cambios en la memoria de inicio con copy running-config startup-config, asegurando que la configuración se mantenga tras reiniciar el router.</p>	<pre>Santiago(config)#enable secret ClaveE Santiago(config)#exit Santiago# %SYS-5-CONFIG_I: Configured from console by console  Santiago#show running-config Building configuration...  Current configuration : 1253 bytes ! version 15.1 no service timestamps log datetime msec no service timestamps debug datetime msec no service password-encryption ! hostname Santiago ! ! ! enable secret 5 \$1\$mER\$uxvWtuZlBdkSDibNl0/BD. ! ! ! ! ! ! no ip cef no ipv6 cef ! ! ! ! ! ! license udi pid CISCO1941/K9 sn FTX15244W05- ! ! ! ! ! ! no ip domain-lookup ! ! spanning-tree mode pvst ! ! ! ! ! ! interface GigabitEthernet0/0 description "LAN Connection to PC01" no ip address duplex auto speed auto !</pre>
<p>Se configuraron rutas estáticas para permitir la comunicación con redes no directamente conectadas. En la interfaz GigabitEthernet0/0 se añadió la ruta a la red 72.0.0.0/8 y en GigabitEthernet0/1 la ruta a la red 73.0.0.0/8.</p>	<pre>Santiago#configure terminal Enter configuration commands, one per line. End with CNTL/Z. Santiago(config)#ip route 72.0.0.0 255.0.0.0 GigabitEthernet0/0 %Default route without gateway, if not a point-to-point interface, may impact performance Santiago(config)#ip route 73.0.0.0 255.0.0.0 GigabitEthernet0/1 %Default route without gateway, if not a point-to-point interface, may impact performance Santiago(config)#</pre>
<p>Para verificar que las rutas estáticas se habían configurado</p>	<pre>Santiago#copy running-config startup-config Destination filename [startup-config]? Building configuration... [OK] Santiago#show running-config   include ip route ip route 72.0.0.0 255.0.0.0 GigabitEthernet0/0 ip route 73.0.0.0 255.0.0.0 GigabitEthernet0/1 Santiago#</pre>

correctamente, se utilizó el comando show running-config | include ip route

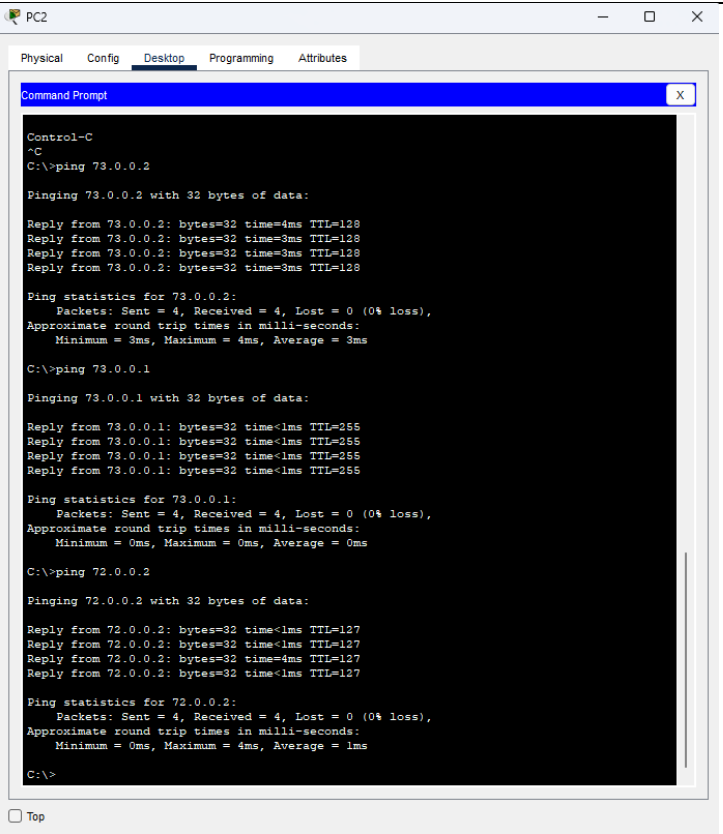
Se asignó la dirección IPv4 correspondiente a cada interfaz, configurando el gateway de cada red. Para verificar que las direcciones se habían aplicado correctamente, se ejecutó el comando show ip interface brief

```
Santiago#show ip interface brief
Interface      IP-Address      OK? Method Status  Protocol
GigabitEthernet0/0  72.0.0.1        YES manual up      up
GigabitEthernet0/1  73.0.0.1        YES manual up      up
Serial0/0/0        unassigned      YES unset  up      up
Serial0/0/1        unassigned      YES unset  administratively down down
FastEthernet0/1/0  unassigned      YES unset  up      down
FastEthernet0/1/1  unassigned      YES unset  up      down
FastEthernet0/1/2  unassigned      YES unset  up      down
FastEthernet0/1/3  unassigned      YES unset  up      down
Vlan1           unassigned      YES unset  administratively down down
Santiago#
```

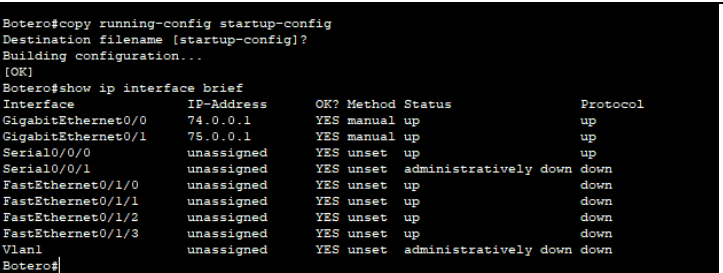
En la PC01 se configuró la dirección IPv4 72.0.0.2 con máscara 255.0.0.0 y gateway 72.0.0.1. Posteriormente, se realizaron pruebas de conectividad mediante ping, tanto hacia su gateway como hacia la PC02, cuya dirección IP es 73.0.0.2, para comprobar la comunicación entre dispositivos.



En la PC02 se configuró la dirección IPv4 73.0.0.2 con máscara 255.0.0.0 y gateway 73.0.0.1. Posteriormente, se realizaron pruebas de conectividad mediante ping, tanto hacia su gateway como hacia la PC01, cuya dirección IP es 72.0.0.2, para comprobar la comunicación entre dispositivos.



En el Router1 se configuró el nombre de host Botero y se aplicaron las mismas configuraciones básicas que en Router0. Para las interfaces, se asignaron las redes correspondientes:  
  
GigabitEthernet0/0 con 74.0.0.0/8 y  
  
GigabitEthernet0/1 con 75.0.0.0/8.

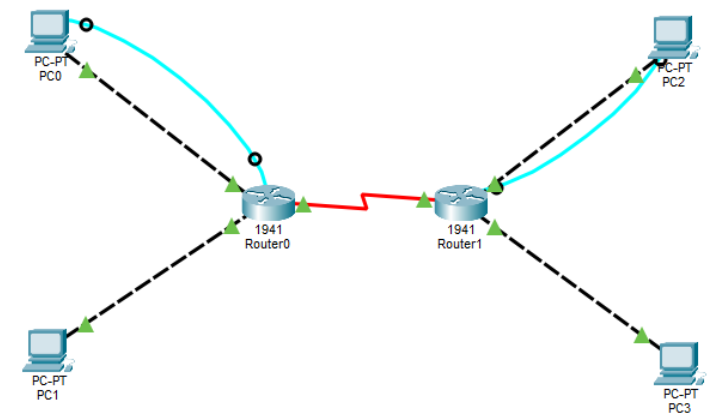




<p>En el Router Santiago se configuró la interfaz Serial0/0/0 asignándole la dirección IPv4 20.0.0.1 con máscara 255.0.0.0. Además, se estableció la velocidad del reloj con clock rate 64000 y se activó la interfaz mediante el comando no shutdown para habilitar la comunicación serial.</p>	<pre>Santiago#configure terminal Enter configuration commands, one per line. End with CNTL/Z. Santiago(config)#interface Serial0/0/0 Santiago(config-if)#ip address 20.0.0.1 255.0.0.0 Santiago(config-if)#clock rate 64000 This command applies only to DCE interfaces Santiago(config-if)#no shutdown Santiago(config-if)#</pre>
<p>En el Router Botero se configuró la interfaz Serial0/0/0 con la dirección IPv4 20.0.0.2 y máscara 255.0.0.0. Finalmente, se activó la interfaz utilizando el comando no shutdown para permitir la comunicación serial con otros routers.</p>	<pre>Botero(config)#interface Serial0/0/0 Botero(config-if)#ip address 20.0.0.2 255.0.0.0 Botero(config-if)#no shutdown Botero(config-if)#</pre>
<p>En el Router Santiago se configuraron rutas estáticas para las redes remotas conectadas a Botero. Se añadió la ruta hacia la red 74.0.0.0/8 y la red 75.0.0.0/8</p>	<pre>Botero#configure terminal Enter configuration commands, one per line. End with CNTL/Z. Botero(config)#ip route 72.0.0.0 255.0.0.0 20.0.0.1 Botero(config)#ip route 73.0.0.0 255.0.0.0 20.0.0.1 Botero(config)#exit Botero# %SYS-5-CONFIG_I: Configured from console by console  Botero#show running-config   include ip route ip route 74.0.0.0 255.0.0.0 GigabitEthernet0/0 ip route 75.0.0.0 255.0.0.0 GigabitEthernet0/1 ip route 72.0.0.0 255.0.0.0 20.0.0.1 ip route 73.0.0.0 255.0.0.0 20.0.0.1 Botero#</pre>

<p>utilizando como próximo salto la dirección 20.0.0.2.</p>	
<p>En el Router Botero se configuraron rutas estáticas hacia las redes remotas conectadas a Santiago. Se añadió la ruta hacia la red 72.0.0.0/8 y la red 73.0.0.0/8 con próximo salto 20.0.0.1.</p>	<pre> Santiago#configure terminal Enter configuration commands, one per line. End with CNTL/Z. Santiago(config)#ip route 74.0.0.0 255.0.0.0 20.0.0.2 Santiago(config)#ip route 75.0.0.0 255.0.0.0 20.0.0.2 Santiago(config)#exit Santiago# %SYS-5-CONFIG_I: Configured from console by console  Santiago#show running-config   include ip route ip route 72.0.0.0 255.0.0.0 GigabitEthernet0/0 ip route 73.0.0.0 255.0.0.0 GigabitEthernet0/1 ip route 74.0.0.0 255.0.0.0 20.0.0.2 ip route 75.0.0.0 255.0.0.0 20.0.0.2 Santiago# </pre>
<p>Se realizaron pruebas de conectividad desde la PC01 mediante el comando ping hacia las PC02, PC03 y PC04, utilizando las direcciones IP 73.0.0.2, 74.0.0.2 y 75.0.0.2, respectivamente. Todos los pings se ejecutaron correctamente, confirmando que la comunicación entre las distintas redes y dispositivos está funcionando de manera adecuada.</p>	 <p>The screenshot shows a Windows PC window titled 'PC1' with tabs for Physical, Config, Desktop, Programming, and Attributes. The 'Desktop' tab is active, displaying a Command Prompt window. The Command Prompt shows the following output:</p> <pre> C:\&gt;ping 74.0.0.2  Pinging 74.0.0.2 with 32 bytes of data:  Reply from 74.0.0.2: bytes=32 time=10ms TTL=126 Reply from 74.0.0.2: bytes=32 time=5ms TTL=126 Reply from 74.0.0.2: bytes=32 time=6ms TTL=126 Reply from 74.0.0.2: bytes=32 time=5ms TTL=126  Ping statistics for 74.0.0.2:     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),     Approximate round trip times in milli-seconds:         Minimum = 5ms, Maximum = 10ms, Average = 6ms  C:\&gt;ping 75.0.0.2  Pinging 75.0.0.2 with 32 bytes of data:  Reply from 75.0.0.2: bytes=32 time=12ms TTL=126 Reply from 75.0.0.2: bytes=32 time=7ms TTL=126 Reply from 75.0.0.2: bytes=32 time=6ms TTL=126 Reply from 75.0.0.2: bytes=32 time=6ms TTL=126  Ping statistics for 75.0.0.2:     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),     Approximate round trip times in milli-seconds:         Minimum = 5ms, Maximum = 12ms, Average = 7ms  C:\&gt;ping 73.0.0.2  Pinging 73.0.0.2 with 32 bytes of data:  Reply from 73.0.0.2: bytes=32 time&lt;1ms TTL=127 Reply from 73.0.0.2: bytes=32 time&lt;1ms TTL=127 Reply from 73.0.0.2: bytes=32 time&lt;1ms TTL=127 Reply from 73.0.0.2: bytes=32 time&lt;1ms TTL=127  Ping statistics for 73.0.0.2:     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),     Approximate round trip times in milli-seconds:         Minimum = 0ms, Maximum = 0ms, Average = 0ms  C:\&gt; </pre>

## Implementación de Natalia

Acción Realizada	Captura de Pantalla
<p>Las computadoras PC01 y PC02 se conectaron al Router0 mediante cables de cobre directo, mientras que las PC03 y PC04 se enlazaron al Router1 utilizando el mismo tipo de cable. Además, la PC01 se conectó al Router0 a través de un cable de consola para realizar la configuración inicial, y entre Router0 y Router1 se estableció un enlace serial empleando un cable DCE/DTE.</p>	
<p>Se asignó el nombre Laura al dispositivo, se creó un mensaje de advertencia (MOTD) para uso exclusivo de estudiantes, y se establecieron contraseñas para el acceso local mediante la línea de consola y para el acceso remoto en las líneas VTY, habilitando la autenticación con el comando login.</p>	<pre>Router#config t Enter configuration commands, one per line. End with CNTL/Z. Router(config)#hostname Laura Laura(config)#banner motd #For exclusive use by AYSR students# Laura(config)#line console 0 Laura(config-line)#logging synchronous Laura(config-line)#password ClaveC Laura(config-line)#login Laura(config-line)#exit Laura(config)#line vty 0 15 Laura(config-line)#logging synchronous Laura(config-line)#password ClaveT Laura(config-line)#login Laura(config-line)#exit Laura(config)#no ip domain-lookup</pre>

<p>Además, se activó la sincronización de mensajes para evitar interrupciones durante la escritura y se deshabilitó la búsqueda de dominios con no ip domain-lookup para prevenir retrasos por errores tipográficos.</p>	
<p>Se ingresó al modo de configuración de cada interfaz GigabitEthernet (0/0 y 0/1) y se asignaron descripciones que indican la conexión hacia los equipos PC01 y PC02 respectivamente.</p>	<pre>Laura(config)#interface GigabitEthernet0/0 Laura(config-if)#description "LAN Connection to PC01" Laura(config-if)# Laura(config-if)#exit Laura(config)#interface GigabitEthernet0/1 Laura(config-if)#description "LAN Connection to PC02" Laura(config-if)#exit Laura(config)#</pre>
<p>Se ingresó al modo de configuración global y se añadieron dos rutas para las redes 132.18.0.0 y 132.19.0.0, especificando como salida las interfaces GigabitEthernet0/0 y GigabitEthernet0/1 respectivamente.</p>	<pre>Laura#configure terminal Enter configuration commands, one per line. End with CNTL/Z. Laura(config)#ip route 132.18.0.0 255.255.0.0 GigabitEthernet0/0 !Default route without gateway, if not a point-to-point interface, may impact performance Laura(config)#ip route 132.19.0.0 255.255.0.0 GigabitEthernet0/1</pre>
<p>Primero se ejecutó el comando copy running-config startup-config para almacenar la configuración activa en</p>	<pre>Laura#copy running-config startup-config Destination filename [startup-config]? Building configuration... [OK] Laura#show running-config   include ip route ip route 132.18.0.0 255.255.0.0 GigabitEthernet0/0 ip route 132.19.0.0 255.255.0.0 GigabitEthernet0/1 Laura#</pre>

<p>la memoria de inicio, asegurando que se mantenga tras reinicios. Luego, se utilizó show running-config   include ip route para confirmar que las rutas estáticas previamente configuradas están correctamente registradas.</p>	
<p>Se ingresó al modo de configuración global y luego a la interfaz Serial0/0/0, donde se asignó la dirección IP 20.0.0.1 con máscara 255.0.0.0. Además, se estableció el clock rate de 64000, necesario en la interfaz DCE para sincronizar la comunicación en el enlace serial.</p>	<pre>Laura#configure terminal Enter configuration commands, one per line. End with CNTL/Z. Laura(config)#interface Serial0/0/0 Laura(config-if)#ip address 20.0.0.1 255.0.0.0 Laura(config-if)#clock rate 64000 This command applies only to DCE interfaces Laura(config-if)#no shutdown Laura(config-if)#</pre>

<p>Se muestra la configuración inicial del Router1, donde se asignó el nombre Perilla y se estableció un mensaje de advertencia (MOTD) para uso exclusivo de estudiantes. Se deshabilitó la búsqueda de dominios con no ip domain-lookup y se configuraron descripciones en las interfaces GigabitEthernet0/0 y 0/1, indicando las conexiones hacia PC3 y PC2, así como en la interfaz serial para identificar el enlace con el Router Laura. Además, se establecieron contraseñas para el modo privilegiado, la consola y el acceso remoto por VTY, habilitando la autenticación con login.</p>	<pre>Router&gt;enable Router#configure terminal Enter configuration commands, one per line. End with CNTL/Z. Router(config)#hostname Perilla Perilla(config)#banner motd "For exclusive use by RECO students" Perilla(config)#line console 0 Perilla(config-line)#logging synchronous Perilla(config-line)#exit Perilla(config)#no ip domain-lookup Perilla(config)#interface GigabitEthernet0/0 Perilla(config-if)#description to PC3 Perilla(config-if)#exit Perilla(config)#interface GigabitEthernet0/1 Perilla(config-if)#description to PC2 Perilla(config-if)#exit Perilla(config)#interface serial0/1/0 %Invalid interface type and number Perilla(config)#interface serial0/0/0 Perilla(config-if)#description to connect to Router Perilla(config-if)#description to connect to Router Laura Perilla(config-if)#exit Perilla(config)#enable secret ClaveE Perilla(config)#line console 0 Perilla(config-line)#password ClaveC Perilla(config-line)#login Perilla(config-line)#exit Perilla(config)#line vty 0 15 Perilla(config-line)#password ClaveT Perilla(config-line)#login Perilla(config-line)#exit Perilla(config)#exit Perilla# %SYS-5-CONFIG_I: Configured from console by console  Perilla#copy running-config startup-config Destination filename [startup-config]? Building configuration... [OK] Perilla#config t</pre>
<p>Se ingresó al modo de configuración global y se añadieron dos rutas para las redes 132.20.0.0 y 132.21.0.0, indicando como salida la interfaz serial Serial0/0/0. Esto permite que el</p>	<pre>Perilla#config t Enter configuration commands, one per line. End with CNTL/Z. Perilla(config)#ip route 132.20.0.0 255.255.0.0 serial 0/0/0 %Default route without gateway, if not a point-to-point interface, may impact performance Perilla(config)#ip route 132.21.0.0 255.255.0.0 serial 0/0/0 %Default route without gateway, if not a point-to-point interface, may impact performance Perilla(config)#exit Perilla# %SYS-5-CONFIG_I: Configured from console by console  Perilla#show ip route static Perilla#copy running-config startup-config Destination filename [startup-config]? Building configuration... [OK] Perilla#show running-config   include ip route ip route 132.20.0.0 255.255.0.0 Serial0/0/0 ip route 132.21.0.0 255.255.0.0 Serial0/0/0</pre>

router conozca cómo alcanzar las redes remotas a través del enlace serial con el Router Laura.

Se evidencia la prueba de conectividad desde PC2 hacia diferentes equipos de la red mediante el comando ping. Los resultados muestran que la PC2 logra comunicarse correctamente con las direcciones 132.21.0.2, 132.19.0.2 y 132.18.0.2, confirmando que las rutas estáticas y la configuración de IPs permiten la interacción entre redes distintas.

PC2

Physical Config Desktop Programming Attributes

Command Prompt

```
C:\>ping 132.21.0.2

Pinging 132.21.0.2 with 32 bytes of data:

Request timed out.
Reply from 132.21.0.2: bytes=32 time<1ms TTL=127
Reply from 132.21.0.2: bytes=32 time<1ms TTL=127
Reply from 132.21.0.2: bytes=32 time<1ms TTL=127

Ping statistics for 132.21.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
C:\>ping 132.19.0.2

Pinging 132.19.0.2 with 32 bytes of data:

Reply from 132.19.0.2: bytes=32 time=1ms TTL=126
Reply from 132.19.0.2: bytes=32 time=1ms TTL=126
Reply from 132.19.0.2: bytes=32 time=1ms TTL=126

Ping statistics for 132.19.0.2:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

Control-C
^C
C:\>ping 132.18.0.2

Pinging 132.18.0.2 with 32 bytes of data:


Reply from 132.18.0.2: bytes=32 time=1ms TTL=126
Reply from 132.18.0.2: bytes=32 time=10ms TTL=126

Ping statistics for 132.18.0.2:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 10ms, Average = 5ms
```

## Instalación de Software Básico

### Instalación de servicios web

Se llevará a cabo la instalación y configuración de servicios web en diferentes sistemas operativos con el objetivo de comprender cómo se implementan y administran en entornos heterogéneos. Se instalará el servidor Apache en una máquina virtual con Solaris, el servidor Nginx en una máquina con Slackware Linux, y se configurará el servidor web integrado en Windows Server (IIS). Posteriormente, se crearán páginas web simples para verificar el funcionamiento de cada servicio y se configurará su inicio automático al arrancar el sistema. Además, se realizarán pruebas de acceso desde otro equipo para confirmar la conectividad y se implementará un servicio DNS que permitirá acceder a los servidores mediante nombres de dominio en lugar de direcciones IP.

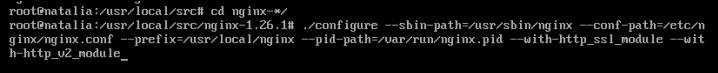
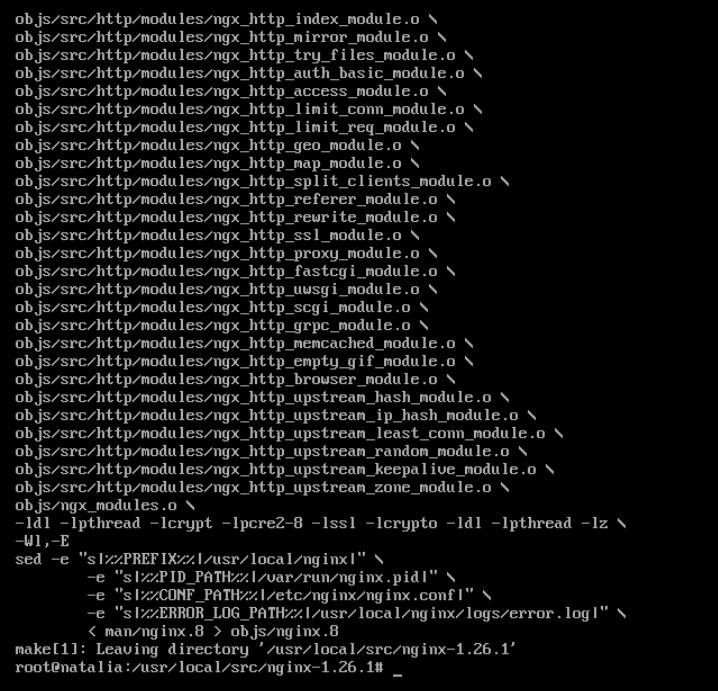
Acción Realizada	Captura de Pantalla
En Solaris, se sincroniza la información local con los repositorios remotos para asegurarse de que la lista de paquetes disponibles esté actualizada.	



<p>Se ejecuta el comando <code>pkg install apache-24</code>, el cual indica al sistema que descargue e instale el paquete Apache 2.4, luego se utiliza el comando <code>pkg info   grep apache</code>, el cual sirve para verificar la información de los paquetes relacionados con Apache que se encuentran instalados en el sistema, esta verificación confirma que la instalación se realizó correctamente y que el servidor Apache está listo para ser configurado y utilizado.</p>	<pre> root@solaris:~# pkg install apache-24 No hay actualizaciones para esta imagen. root@solaris:~# pkg info   grep apache URL de proyecto: https://apr.apache.org/ URL de origen: https://archive.apache.org/dist/apr/apr-1.6.3.tar.gz URL de proyecto: https://apr.apache.org/ URL de origen: https://archive.apache.org/dist/apr/apr-util-1.6.1.tar.gz Nombre: service/system/apache-stats-24 Descripción: Provides statistics about running apache instances to the FMRI: pkg://solaris/service/system/apache-stats-24@11.4-11.4.0.0 .1.15.0:20180817T002718Z Nombre: web/server/apache-24 FMRI: pkg://solaris/web/server/apache-24@2.4.33-11.4.0.0.1.14.0: 20180814T173250Z URL de proyecto: https://httpd.apache.org/ URL de origen: https://archive.apache.org/dist/httpd/httpd-2.4.33.tar.gz Nombre: web/server/apache-24/module/apache-ssl FMRI: pkg://solaris/web/server/apache-24/module/apache-ssl@2.4.3 3-11.4.0.0.1.14.0:20180814T173244Z URL de proyecto: https://httpd.apache.org/ URL de origen: https://archive.apache.org/dist/httpd/httpd-2.4.33.tar.gz Nombre: web/server/apache-24/module/apache-wsgi-27 FMRI: pkg://solaris/web/server/apache-24/module/apache-wsgi-27@ .5.1-11.4.0.0.1.14.0:20180814T173245Z root@solaris:~# </pre>
<p>Para habilitar el servicio de Apache en Solaris, se ejecuta <code>svcs -a   grep -i http</code> para listar todos los servicios del sistema y filtra aquellos relacionados con HTTP, la salida muestra que el servicio <code>svc:/network/http:apache24</code> está deshabilitado, luego se utiliza: <code>svcadm enable svc:/network/http:apache24</code> para</p>	<pre> root@solaris:~# svcs -a   grep -i http disabled 10:35:59 svc:/network/http:apache24 root@solaris:~# svcadm enable svc:/network/http:apache24 root@solaris:~# </pre>

habilitar el servicio de Apache y lo inicia, permitiendo que el servidor web comience a funcionar.	
Se usa el comando slackpkg update para descargar y sincronizar los archivos de listas de paquetes desde los servidores oficiales de Slackware.	<pre>/PACKAGES.TXT... --2025-11-05 11:59:14-- https://mirrors.slackware.com/slackware/slackware64-15.0/pasture/PACKAGES.TXT Resolving mirrors.slackware.com (mirrors.slackware.com)... 207.223.116.213 Connecting to mirrors.slackware.com (mirrors.slackware.com) 207.223.116.213 :443... connected. HTTP request sent, awaiting response... 200 OK Length: 2163 (2.1K) [text/plain] Saving to: #/tmp/slackpkg.zJunJZ/pasture-PACKAGES.TXT#  /tmp/slackpkg.zJunJZ/pas 100%[=====] 2.11K --.-KB/s in 0s  2025-11-05 11:59:14 (597 MB/s) - #/tmp/slackpkg.zJunJZ/pasture-PACKAGES.TXT# saved [2163/2163]  Downloading https://mirrors.slackware.com/slackware/slackware64-15.0/testing/PACKAGES.TXT /PACKAGES.TXT... --2025-11-05 11:59:14-- https://mirrors.slackware.com/slackware/slackware64-15.0/testing/PACKAGES.TXT Resolving mirrors.slackware.com (mirrors.slackware.com)... 207.223.116.213 Connecting to mirrors.slackware.com (mirrors.slackware.com) 207.223.116.213 :443... connected. HTTP request sent, awaiting response... 302 Found Location: https://mirror.cedia.org.ec/slackware/slackware64-15.0/testing/PACKAGES.TXT [following] --2025-11-05 11:59:14-- https://mirror.cedia.org.ec/slackware/slackware64-15.0/testing/PACKAGES.TXT Resolving mirror.cedia.org.ec (mirror.cedia.org.ec)... 201.159.221.67, 2000:68:0:bebe::3 Connecting to mirror.cedia.org.ec (mirror.cedia.org.ec) 201.159.221.67 :443... connected. HTTP request sent, awaiting response... 200 OK Length: 20561 (20K) [text/plain] Saving to: #/tmp/slackpkg.zJunJZ/testing-PACKAGES.TXT#  /tmp/slackpkg.zJunJZ/tes 100%[=====] 20.00K --.-KB/s in 0s  2025-11-05 11:59:15 (150 MB/s) - #/tmp/slackpkg.zJunJZ/testing-PACKAGES.TXT# saved [20561/20561]  Formatting lists to slackpkg style... Package List: using CHECKSUMS.md5 as source Package descriptions  root@natalia:~#</pre>
Se instalan las herramientas y librerías necesarias para compilar e instalar Nginx en Slackware.	<pre>slackpkg install gcc make pcre-devel zlib-devel openssl-devel_</pre>
El sistema presenta una lista de componentes relacionados con GCC.	

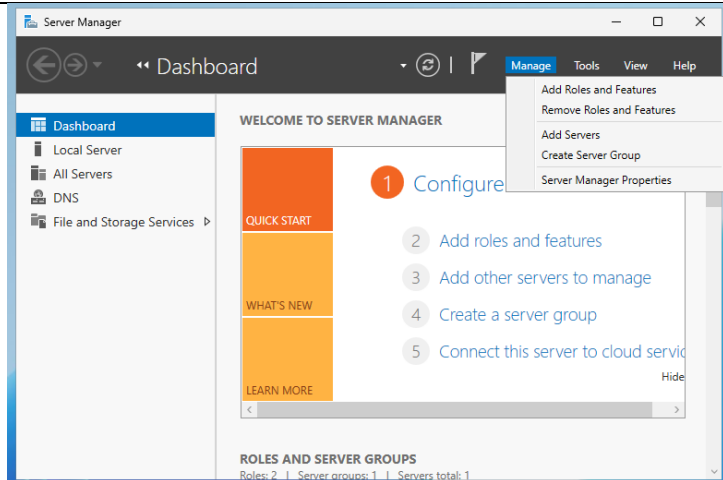
<p>Durante el proceso, se verifica cada paquete, se confirma que no es necesario descargarlo nuevamente y se procede a instalarlo.</p>	<pre># # Executing install script for gcc-gcobol-15.2.0-x86_64-1_slack15.0.txz. Package gcc-gcobol-15.2.0-x86_64-1_slack15.0.txz installed. # Package gcc-gm2-15.2.0-x86_64-1_slack15.0.txz is already in cache - not downloading # Installing gcc-gm2-15.2.0-x86_64-1_slack15.0.txz. Verifying package gcc-gm2-15.2.0-x86_64-1_slack15.0.txz. Installing package gcc-gm2-15.2.0-x86_64-1_slack15.0.txz: PACKAGE DESCRIPTION: # gcc-gm2 (Modula-2 support for GCC) # # Modula-2 support for the GNU Compiler Collection. # # Modula-2 language support covering ISO/IEC 10514-1, PIM2, PIM3, and # PIM4 dialects together with a complete set of ISO/IEC 10514-1 and PIM # libraries. # # Executing install script for gcc-gm2-15.2.0-x86_64-1_slack15.0.txz. Package gcc-gm2-15.2.0-x86_64-1_slack15.0.txz installed. # Package gcc-rust-15.2.0-x86_64-1_slack15.0.txz is already in cache - not downloading # Installing gcc-rust-15.2.0-x86_64-1_slack15.0.txz. Verifying package gcc-rust-15.2.0-x86_64-1_slack15.0.txz. Installing package gcc-rust-15.2.0-x86_64-1_slack15.0.txz: PACKAGE DESCRIPTION: # gcc-rust (Rust support for GCC) # # Rust support for the GNU Compiler Collection. # # gccrs is a full alternative implementation of the Rust language on top # of GCC with the goal to become fully upstream with the GNU toolchain. # Please note, the compiler is in a very early stage and not usable yet # for compiling real Rust programs. # # Package gcc-rust-15.2.0-x86_64-1_slack15.0.txz installed. Searching for NEW configuration files... No .new files found.  root@natalia:~#</pre>
<p>Para descargar el código fuente de Nginx en la máquina con Slackware, se cambia al directorio <code>cd /usr/local/src</code>, luego se ejecuta <code>wget http://nginx.org/download/nginx-1.26.1.tar.gz</code>, el cual descarga el archivo comprimido que contiene la versión 1.26.1 de Nginx directamente desde el sitio oficial.</p>	<pre>root@natalia:~# cd /usr/local/src root@natalia:/usr/local/src# wget http://nginx.org/download/nginx-1.26.1.tar.gz --2025-11-05 12:23:43-- http://nginx.org/download/nginx-1.26.1.tar.gz Resolving nginx.org (nginx.org)... 3.125.197.172, 52.58.199.22, 2a05:a014:5c0:2600::6, ... Connecting to nginx.org (nginx.org) 3.125.197.172 :80... connected. HTTP request sent, awaiting response... 200 OK Length: 1244738 (1.2M) [application/octet-stream] Saving to: 'nginx-1.26.1.tar.gz'  nginx-1.26.1.tar.gz 100%[=====] 1.19M 865KB/s in 1.4s  2025-11-05 12:23:44 (865 KB/s) - 'nginx-1.26.1.tar.gz' saved [1244738/1244738]  root@natalia:/usr/local/src#</pre>
<p>Este comando se utiliza para descomprimir y extraer el contenido del archivo comprimido que contiene el código fuente de Nginx.</p>	<pre>/usr/local/src# tar zxvf nginx-*.tar.gz_</pre>

<p>En esta imagen se muestra el paso donde se configura la compilación de Nginx antes de instalarlo.</p>	 <pre> root@natalia:/usr/local/src# cd nginx-* root@natalia:/usr/local/src/nginx-1.26.1# ./configure --sbin-path=/usr/sbin/nginx --conf-path=/etc/nginx/nginx.conf --prefix=/usr/local/nginx --pid-path=/var/run/nginx.pid --with-http_ssl_module --with-http_v2_module </pre>
<p>En esta imagen se muestra el resultado de ejecutar el comando make, este comando compila el código fuente de Nginx utilizando las configuraciones definidas previamente en el archivo Makefile.</p>	 <pre> objs/src/http/modules/nginx_http_index_module.o \ objs/src/http/modules/nginx_http_mirror_module.o \ objs/src/http/modules/nginx_http_try_files_module.o \ objs/src/http/modules/nginx_http_auth_basic_module.o \ objs/src/http/modules/nginx_http_access_module.o \ objs/src/http/modules/nginx_http_limit_conn_module.o \ objs/src/http/modules/nginx_http_limit_req_module.o \ objs/src/http/modules/nginx_http_geo_module.o \ objs/src/http/modules/nginx_http_map_module.o \ objs/src/http/modules/nginx_http_split_clients_module.o \ objs/src/http/modules/nginx_http_referer_module.o \ objs/src/http/modules/nginx_http_rewrite_module.o \ objs/src/http/modules/nginx_http_ssl_module.o \ objs/src/http/modules/nginx_http_proxy_module.o \ objs/src/http/modules/nginx_http_fastcgi_module.o \ objs/src/http/modules/nginx_http_uwsgi_module.o \ objs/src/http/modules/nginx_http_scgi_module.o \ objs/src/http/modules/nginx_http_grpc_module.o \ objs/src/http/modules/nginx_http_memcached_module.o \ objs/src/http/modules/nginx_http_empty_gif_module.o \ objs/src/http/modules/nginx_http_browser_module.o \ objs/src/http/modules/nginx_http_upstream_hash_module.o \ objs/src/http/modules/nginx_http_upstream_ip_hash_module.o \ objs/src/http/modules/nginx_http_upstream_least_conn_module.o \ objs/src/http/modules/nginx_http_upstream_random_module.o \ objs/src/http/modules/nginx_http_upstream_keepalive_module.o \ objs/src/http/modules/nginx_http_upstream_zone_module.o \ objs/nginx_modules.o \ -lld -lpthread -lcrypt -lpcrc2-8 -lssl -lcrypto -ldl -lpthread -lz \ -Wl,-E sed -e "s %%PREFIX%% /usr/local/nginx " \ -e "s %%PID_PATH%% /var/run/nginx.pid " \ -e "s %%CONF_PATH%% /etc/nginx/nginx.conf " \ -e "s %%ERROR_LOG_PATH%% /usr/local/nginx/logs/error.log " \ &lt; man/nginx.8 &gt; objs/nginx.8 make[11]: Leaving directory '/usr/local/src/nginx-1.26.1' root@natalia:/usr/local/src/nginx-1.26.1# _ </pre>

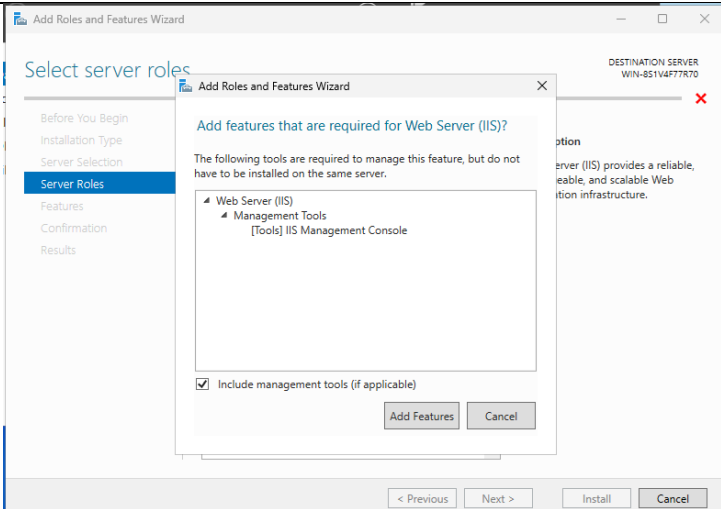
En esta imagen se muestra el resultado del comando make install, el cual copia los archivos compilados y configura la estructura final de instalación de Nginx en el sistema.

```
cp objs/nginx /usr/sbin/nginx'
test -d '/etc/nginx' \
    || mkdir -p '/etc/nginx'
cp conf/koi-win '/etc/nginx'
cp conf/koi-utf '/etc/nginx'
cp conf/win-utf '/etc/nginx'
test -f '/etc/nginx/mime.types' \
    || cp conf/mime.types '/etc/nginx'
cp conf/mime.types '/etc/nginx/mime.types.default'
test -f '/etc/nginx/fastcgi_params' \
    || cp conf/fastcgi_params '/etc/nginx'
cp conf/fastcgi_params \
    '/etc/nginx/fastcgi_params.default'
test -f '/etc/nginx/fastcgi.conf' \
    || cp conf/fastcgi.conf '/etc/nginx'
cp conf/fastcgi.conf '/etc/nginx/fastcgi.conf.default'
test -f '/etc/nginx/uwsgi_params' \
    || cp conf/uwsgi_params '/etc/nginx'
cp conf/uwsgi_params \
    '/etc/nginx/uwsgi_params.default'
test -f '/etc/nginx/scgi_params' \
    || cp conf/scgi_params '/etc/nginx'
cp conf/scgi_params \
    '/etc/nginx/scgi_params.default'
test -f '/etc/nginx/nginx.conf' \
    || cp conf/nginx.conf '/etc/nginx/nginx.conf'
cp conf/nginx.conf '/etc/nginx/nginx.conf.default'
test -d '/var/run' \
    || mkdir -p '/var/run'
test -d '/usr/local/nginx/logs' \
    || mkdir -p '/usr/local/nginx/logs'
test -d '/usr/local/nginx/html' \
    || cp -R html '/usr/local/nginx'
test -d '/usr/local/nginx/logs' \
    || mkdir -p '/usr/local/nginx/logs'
make[1]: Leaving directory '/usr/local/src/nginx-1.26.1'
root@natalia:/usr/local/src/nginx-1.26.1#
```

Se inicia el proceso para agregar roles y características, este paso es fundamental para instalar el rol de Servidor Web (IIS) en Windows Server.



Se confirma la instalación de las características necesarias para el rol Web Server (IIS).



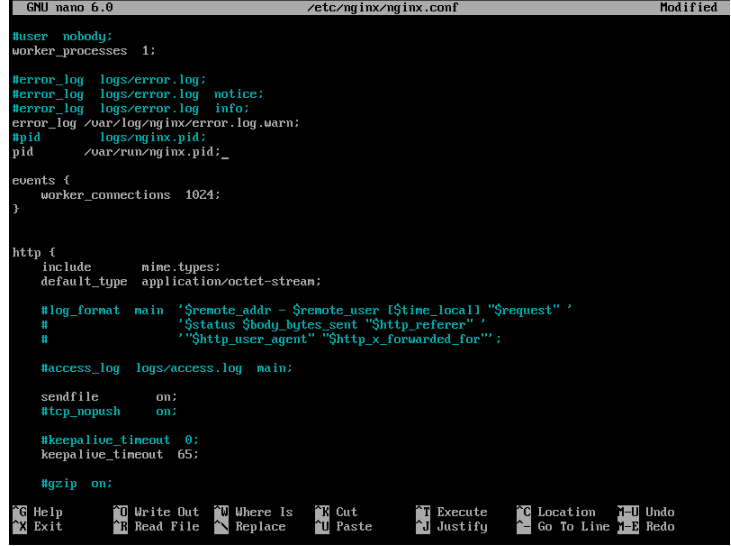
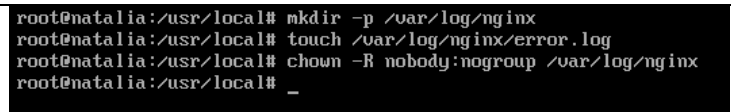
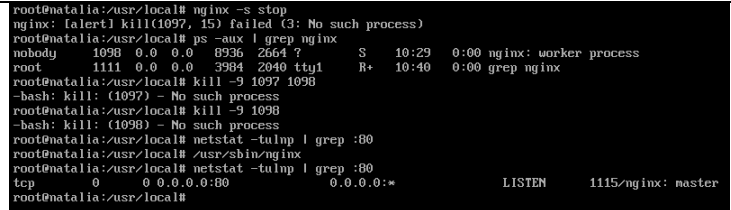
Se realiza la creación de una página web sencilla en el servidor Apache en Solaris. Se usa mkdir -p /var/www/html para generar la ruta donde se almacenarán los archivos HTML que servirá Apache y se crea el archivo index.html con contenido básico

```
root@solaris:~# mkdir -p /var/www/html
root@solaris:~# echo "<html><body><h1>Apache en Solaris - 0K</h1></body></html>" | tee /var/www/html/index.html
root@solaris:~#
```

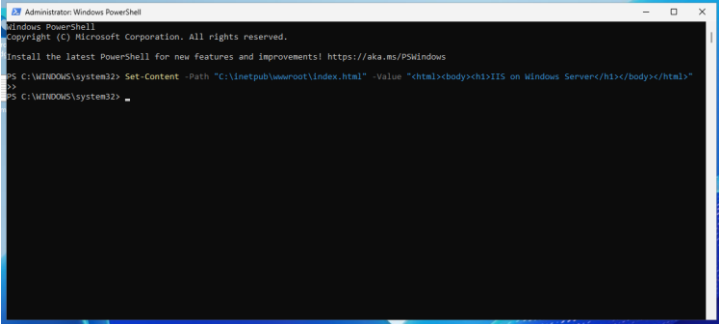
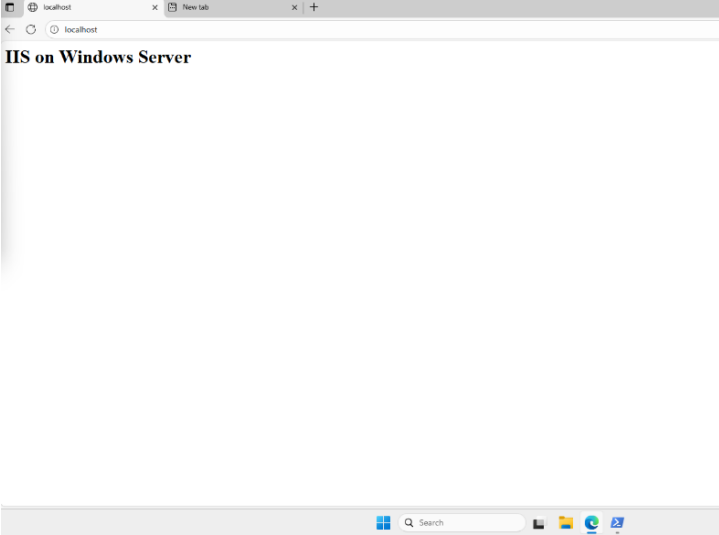
Se ejecuta chown -R root:root /var/www/html, este comando cambia el propietario y el grupo del directorio /var/www/html a root, asegurando que el administrador del sistema tenga control total sobre los archivos, luego se ejecuta chmod -R 755

```
root@solaris:~# chown -R root:root /var/www/html
root@solaris:~# chmod -R 755 /var/www/html
root@solaris:~#
```

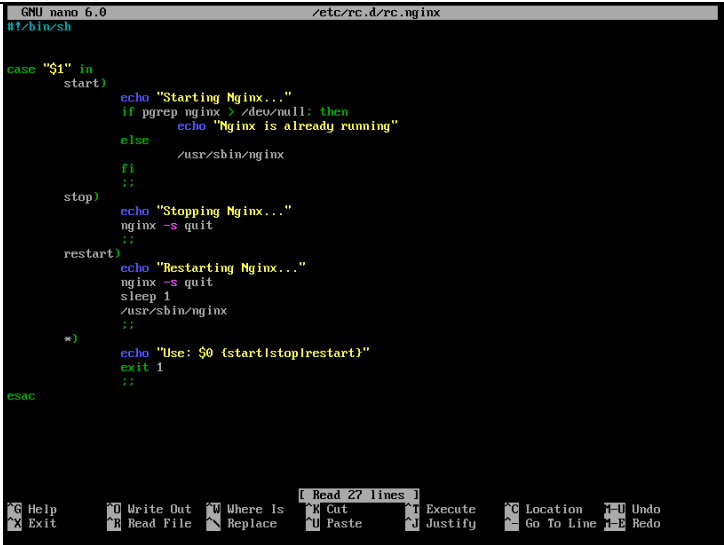
<p>/var/www/html para otorgar permisos de lectura y ejecución para todos los usuarios, y permisos de escritura únicamente al propietario.</p>	
<p>Se muestra la verificación del funcionamiento del servidor Apache en Solaris utilizando el comando curl -i http://localhost, este comando realiza una petición HTTP al servidor local y muestra tanto los encabezados de la respuesta como el contenido.</p>	<pre>root@solaris:~# curl -i http://localhost HTTP/1.1 200 OK Date: Wed, 05 Nov 2025 15:56:46 GMT Server: Apache/2.4.33 (Unix) Last-Modified: Fri, 17 Aug 2018 02:07:27 GMT ETag: "2d-57398056ff5c0" Accept-Ranges: bytes Content-Length: 45 Content-Type: text/html  &lt;html&gt;&lt;body&gt;&lt;h1&gt;It works!&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt; root@solaris:~#</pre>
<p>Nos aseguramos de que Nginx tenga una página inicial para probar su funcionamiento y que la configuración pueda ajustarse según las necesidades.</p>	<pre>root@natalia:~# mkdir -p /usr/local/nginx/html root@natalia:~# echo "&lt;html&gt;&lt;body&gt;&lt;h1&gt;Nginx en Slackware - OK&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt;"   tee / tee: /: Is a directory &lt;html&gt;&lt;body&gt;&lt;h1&gt;Nginx en Slackware - OK&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt; root@natalia:~# echo "&lt;html&gt;&lt;body&gt;&lt;h1&gt;Nginx en Slackware - OK&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt;"   tee /usr/local/n ginx/html/index.html &lt;html&gt;&lt;body&gt;&lt;h1&gt;Nginx en Slackware - OK&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt; root@natalia:~# cd /usr/local root@natalia:~# nano /etc/nginx/nginx.conf</pre>

<p>Se añade error_log</p> <p>/var/log/nginx/error.log.warn para monitorear el funcionamiento del servidor y diagnosticar problemas</p> <p>pid /var/run/nginx.pid para indicar la ubicación del archivo PID, que almacena el identificador del proceso principal de Nginx.</p>	
<p>Este paso es importante porque garantiza que Nginx pueda escribir en los archivos de log sin problemas y que la seguridad del sistema se mantenga.</p>	
<p>Este paso asegura que el servidor web Nginx está funcionando correctamente después de la configuración.</p>	
<p>Se muestra la verificación final del funcionamiento del servidor Nginx en Slackware mediante el comando curl http://localhost, la salida confirma que Nginx está activo, escuchando en el</p>	



<p>puerto 80 y sirviendo correctamente la página web desde la ruta configurada.</p>	
<p>Se muestra la creación de una página web sencilla para el servidor IIS en Windows Server.</p>	 <pre>Administrator: Windows PowerShell Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.  Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  PS C:\WINDOWS\system32&gt; Set-Content -Path "C:\inetpub\wwwroot\index.html" -Value "&lt;html&gt;&lt;body&gt;&lt;h1&gt;IIS on Windows Server&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt;" PS C:\WINDOWS\system32&gt;</pre>
<p>Se muestra la verificación del servidor IIS en Windows Server desde un navegador web.</p>	 <p>localhost    New Tab    +</p> <p>←    →    localhost</p> <h3>IIS on Windows Server</h3> <p>Search</p>
<p>Se comprueba que Apache se ejecuta de forma estable y que está listo para atender solicitudes HTTP.</p>	<pre>root@solaris:~# svcs -a   grep -i apache disabled      10:35:59    svc:/system/apache-stats-24:default online        10:47:48    svc:/network/http:apache24</pre>

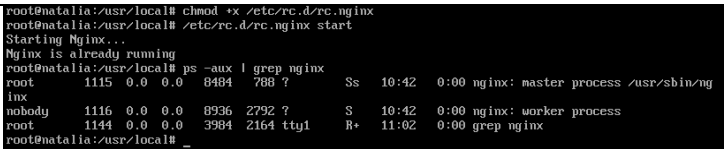
Se realiza la creación de un script de inicio para gestionar el servicio Nginx en Slackware de una forma sencilla.



```
GNU nano 6.0 /etc/rc.d/rc.nginx
#!/bin/sh

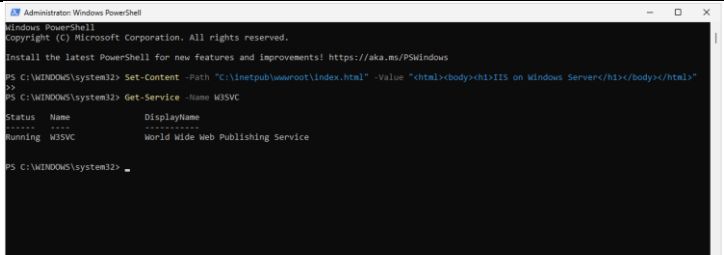
case "$1" in
  start)
    echo "Starting Nginx..."
    if pgrep nginx > /dev/null; then
      echo "Nginx is already running"
    else
      /usr/sbin/nginx
    fi
    ;;
  stop)
    echo "Stopping Nginx..."
    nginx -s quit
    ;;
  restart)
    echo "Restarting Nginx..."
    nginx -s quit
    sleep 1
    /usr/sbin/nginx
    ;;
  *)
    echo "Use: $0 {start|stop|restart}"
    exit 1
    ;;
esac
```

Este paso asegura que el script funciona correctamente y que Nginx puede ser gestionado de forma automática, cumpliendo con el requisito de que el servicio se inicie al arrancar el sistema.



```
root@natalia:/usr/local# chmod +x /etc/rc.d/rc.nginx
root@natalia:/usr/local# /etc/rc.d/rc.nginx start
Starting Nginx...
Nginx is already running
root@natalia:/usr/local# ps -aux | grep nginx
root      1115  0.0  0.0  8484  788 ?        Ss   10:42   0:00 nginx: master process /usr/sbin/ng
lnx
nobody    1116  0.0  0.0   8936 2792 ?        S    10:42   0:00 nginx: worker process
root      1144  0.0  0.0   3984 2164 tty1    R+   11:02   0:00 grep nginx
root@natalia:/usr/local#
```

Se ejecuta el comando Get-Service -Name W3SVC para verificar el estado del servicio World Wide Web Publishing Service, que es el responsable de publicar páginas web en IIS.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> Set-Content -Path "C:\inetpub\wwwroot\index.html" -Value "<html><body><h1>IIS on Windows Server</h1></body></html>"
PS C:\WINDOWS\system32> Get-Service -Name W3SVC

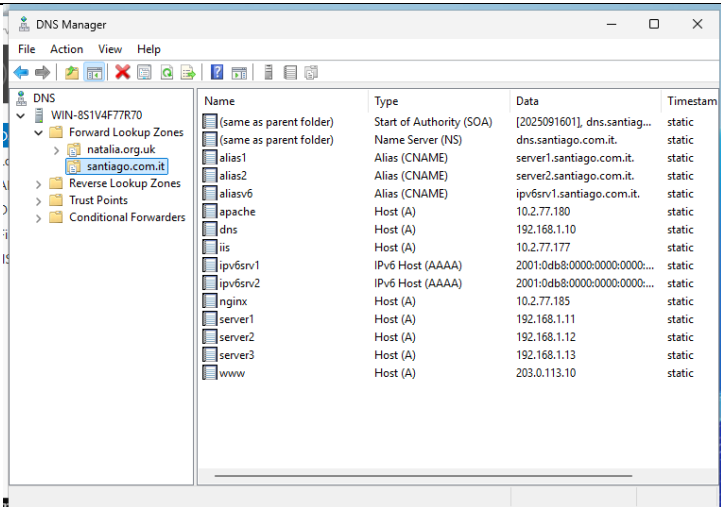
Status Name      DisplayName
-----
Running W3SVC     World Wide Web Publishing Service

PS C:\WINDOWS\system32>
```

<p>Se realiza la prueba de conectividad y funcionamiento de los servidores web instalados en diferentes sistemas operativos. Desde un equipo cliente, se utiliza el comando curl para enviar solicitudes HTTP a las direcciones IP de cada servidor.</p>	<pre>C:\Users\Redes&gt;curl http://10.2.77.185 &lt;html&gt;&lt;body&gt;&lt;h1&gt;Nginx en Slackware - OK&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt;  C:\Users\Redes&gt;curl http://10.2.77.177 &lt;html&gt;&lt;body&gt;&lt;h1&gt;IIS on Windows Server&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt;  C:\Users\Redes&gt;curl http://10.2.77.180 &lt;html&gt;&lt;body&gt;&lt;h1&gt;It works!&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt;</pre>
<p>Con esta configuración, los clientes podrán acceder a cada servidor web utilizando su nombre en lugar de la dirección IP, lo que facilita la administración y mejora la usabilidad.</p>	<pre>\$TTL 86400 @   IN  SOA dns.santiago.com.it. root.santiago.com.it. (       2025091601 ; Serial       3600       ; Refresh       1800       ; Retry       1209600    ; Expire       86400      ; Minimum TTL  ; IN NS  dns.santiago.com.it.  ; Servidor DNS dns   IN  A      192.168.1.10  ; Servidores con IPv4 server1 IN A      192.168.1.11 server2 IN A      192.168.1.12 server3 IN A      192.168.1.13  ; Servidores con IPv6 ipv6srv1 IN AAAA  2001:db8::11 ipv6srv2 IN AAAA  2001:db8::12  ; New Web Servers apache.santiago.com.it. IN  A      10.2.77.180 nginx.santiago.com.it.  IN  A      10.2.77.185 iis.santiago.com.it.    IN  A      10.2.77.177</pre>
<p>Se verifica que el archivo de zona DNS configurado previamente no contenga errores.</p>	<pre>root@solaris:~# named-checkzone santiago.com.it /etc/named.santiago.com.it zone santiago.com.it/IN: loaded serial 2025091601 OK root@solaris:~# █</pre>
<p>Este reinicio es necesario después de modificar el archivo de zona y validar su sintaxis, asegurando que los</p>	<pre>root@solaris:~# svcadm restart svc:/network/dns/server:default root@solaris:~# svcadm restart svc:/network/dns/client:default</pre>

cambios entren en vigor y que el servicio DNS funcione correctamente

Esta configuración permite que los clientes accedan a los servicios web utilizando nombres en lugar de direcciones IP, lo que mejora la administración y la experiencia del usuario.



The screenshot shows the Windows DNS Manager interface. The left pane displays the hierarchy: DNS > WIN-8S1V4F77R70 > Forward Lookup Zones > natalia.org.uk > santiago.com.it. The right pane shows a list of DNS records for the santiago.com.it zone.

Name	Type	Data	Timestamp
(same as parent folder)	Start of Authority (SOA)	[2025091601], dns.santiag...	static
(same as parent folder)	Name Server (NS)	dns.santiago.com.it.	static
alias1	Alias (CNAME)	server1.santiago.com.it.	static
alias2	Alias (CNAME)	server2.santiago.com.it.	static
aliasv6	Alias (CNAME)	ipv6srv1.santiago.com.it.	static
apache	Host (A)	10.2.77.180	static
dns	Host (A)	192.168.1.10	static
iis	Host (A)	10.2.77.177	static
ipv6srv1	IPv6 Host (AAAA)	2001:0db8:0000:0000:0000:...	static
ipv6srv2	IPv6 Host (AAAA)	2001:0db8:0000:0000:0000:...	static
nginx	Host (A)	10.2.77.185	static
server1	Host (A)	192.168.1.11	static
server2	Host (A)	192.168.1.12	static
server3	Host (A)	192.168.1.13	static
www	Host (A)	203.0.113.10	static

Con esto se confirma que los clientes pueden acceder a los servidores web utilizando nombres en lugar de direcciones IP.

```
PS C:\Users\USUARIO> nslookup nginx.santiago.com.it
Server: UnKnown
Address: 10.2.77.180

Name:    nginx.santiago.com.it
Address: 10.2.77.185

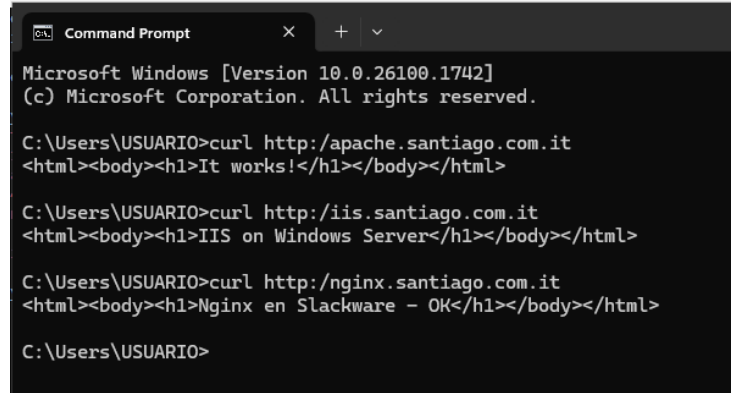
PS C:\Users\USUARIO> nslookup apache.santiago.com.it
Server: UnKnown
Address: 10.2.77.180

Name:    apache.santiago.com.it
Address: 10.2.77.180

PS C:\Users\USUARIO> nslookup iis.santiago.com.it
Server: UnKnown
Address: 10.2.77.180

Name:    iis.santiago.com.it
Address: 10.2.77.177
```

Esta prueba demuestra que los clientes pueden acceder a los servicios web utilizando nombres de dominio en lugar de direcciones IP, lo que confirma la correcta integración entre los servidores web y el servicio DNS.



```
Command Prompt
Microsoft Windows [Version 10.0.26100.1742]
(c) Microsoft Corporation. All rights reserved.

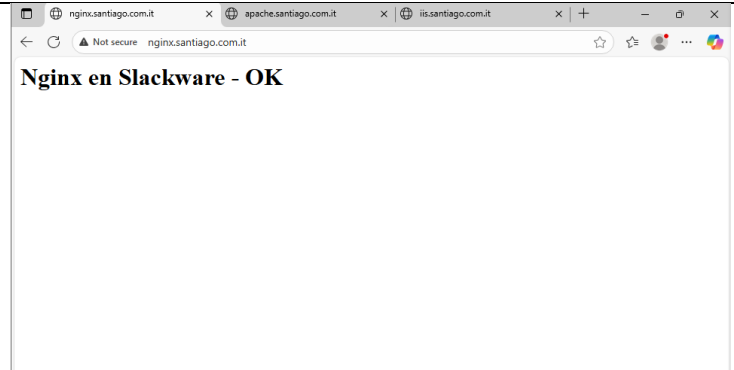
C:\Users\USUARIO>curl http://apache.santiago.com.it
<html><body><h1>It works!</h1></body></html>

C:\Users\USUARIO>curl http://iis.santiago.com.it
<html><body><h1>IIS on Windows Server</h1></body></html>

C:\Users\USUARIO>curl http://nginx.santiago.com.it
<html><body><h1>Nginx en Slackware - OK</h1></body></html>

C:\Users\USUARIO>
```

En esta imagen se muestra la prueba visual en el navegador para confirmar que los servidores web son accesibles mediante los nombres configurados en DNS.



## Servicio de Aplicación Web

Acción Realizada	Captura de Pantalla
<p>Se instaló PHP 7.4 en el sistema Solaris mediante la búsqueda del paquete disponible con pkg search php y su posterior instalación usando pkg install web/php-74, dejando el entorno listo para integrarse con bases de datos como PostgreSQL.</p>	 <pre> pkg:/web/php-73@7.3.32-11.4.42.0.0.111.0 com.oracle.info.name set php pkg:/web/php-74@7.4.25-11.4.42.0.0.111.0 com.oracle.info.name set php pkg:/web/php-83@8.3.14-11.4.81.0.0.193.0 com.oracle.info.name set php pkg:/web/php-84@8.4.1-11.4.81.0.0.193.0 com.oracle.info.name set php pkg:/web/php@8.4.1-11.4.81.0.0.193.0 com.oracle.info.name set php pkg:/web/php-81@8.1.31-11.4.81.0.0.193.0 com.oracle.info.name set php pkg:/web/php-82@8.2.26-11.4.81.0.0.193.0 pkg.fMRI set solaris/web/php pkg:/web/php@8.4.1-11.4.81.0.0.193.0 basename file usr/php/7.1/bin/php pkg:/web/php-71@7.1.17-11.4.0.0.1.14.0 basename file usr/php/5.6/bin/php pkg:/web/php-56@5.6.36-11.4.0.0.1.14.0 basename file usr/php/8.0/bin/php pkg:/web/php-80@8.0.12-11.4.42.0.0.111.0 basename file usr/php/7.3/bin/php pkg:/web/php-73@7.3.32-11.4.42.0.0.111.0 basename file usr/php/7.4/bin/php pkg:/web/php-74@7.4.25-11.4.42.0.0.111.0 basename file usr/php/8.3/bin/php pkg:/web/php-83@8.3.14-11.4.81.0.0.193.0 basename file usr/php/8.4/bin/php pkg:/web/php-84@8.4.1-11.4.81.0.0.193.0 basename file usr/php/8.1/bin/php pkg:/web/php-81@8.1.31-11.4.81.0.0.193.0 basename file usr/php/8.2/bin/php pkg:/web/php-82@8.2.26-11.4.81.0.0.193.0 root@solaris:~# </pre>
<p>Se configuró Apache para habilitar el soporte de PHP 7.4 modificando el archivo de configuración /etc/apache2/2.4/httpd.conf, cargando el módulo php7_module, asociando los archivos .php al intérprete de PHP mediante AddHandler, y definiendo index.php como prioridad en la directiva DirectoryIndex, asegurando que las páginas PHP se</p>	 <pre> # directory and module as required. # IncludeOptional /etc/apache2/2.4/conf.d/*.conf  # # Note: The following must must be present to support # starting without SSL on platforms with no /dev/random equivalent # but a statically compiled-in mod_ssl. # &lt;IfModule ssl_module&gt; SSLRandomSeed startup builtin SSLRandomSeed connect builtin &lt;/IfModule&gt;  LoadModule php7_module libexec/libphp7.so AddHandler php7-script .php DirectoryIndex index.php index.html </pre>

<p>procesen correctamente al acceder al servidor web.</p>	
<p>Se actualizó la variable de entorno PATH del usuario para incluir la ruta de PHP (/usr/local/bin/php) editando el archivo ~/.bashrc y aplicando los cambios con source ~/.bashrc, lo que permite ejecutar comandos de PHP desde cualquier ubicación en la terminal.</p>	
<p>Se descargó el código fuente de PHP versión 7.4.33 desde el sitio oficial utilizando el comando wget https://www.php.net/distributions/php-7.4.33.tar.gz, luego se accedió al directorio temporal con cd /tmp y se ejecutó dicho comando para obtener el archivo comprimido; posteriormente, se descomprimió con tar -xzf php-7.4.33.tar.gz, lo que generó el directorio php-7.4.33, al cual se ingresó mediante cd php-</p>	

<p>7.4.33 para preparar el entorno de compilación e instalación desde la fuente.</p>	
<p>Se instalaron las dependencias necesarias para compilar PHP utilizando el gestor de paquetes del sistema con los comandos <code>pkg install pkg-config</code> y <code>pkg install library/oniguruma</code>. Luego, se configuró la variable de entorno <code>PKG_CONFIG_PATH</code> con el comando <code>export</code></p> <p><code>PKG_CONFIG_PATH=/ruta/del/archivo:\$PKG_CONFIG_PATH</code>. A continuación, se ejecutó el script de configuración con <code>./configure --with-apxs2=/usr/apache2/2.4/bin/apxs --with-zlib --enable-mbstring</code>.</p> <p>Finalmente, se reinició el servicio Apache con los comandos <code>svcadm clear apache24</code> y <code>svcadm restart</code></p>	<pre> root@solaris:~# svcadm clear apache24 root@solaris:~# svcadm restart apache24 root@solaris:~# svcs apache24 STATE      STIME     FMRI online     14:11:40   svc:/network/http:apache24 root@solaris:~# █ </pre>



apache24, y se verificó su estado con `svcs apache24` para confirmar que el servidor web estuviera funcionando correctamente y que PHP estuviera habilitado.

Se creó un archivo de prueba `info.php` en el directorio raíz de Apache con el contenido `<?php phpinfo(); ?>`, permitiendo verificar la instalación y configuración de PHP accediendo a `http://10.2.77.185/info.php` desde un navegador, lo que muestra información detallada del entorno PHP en el servidor.

The screenshot shows a web browser window with the address bar displaying `10.2.77.180/info.php`. The page title is "PHP Version 7.4.33". The main content area displays a table of system and configuration information. Below this, there is a "Configuration" section with a sub-section "apache2handler" which contains a table with the following data:

Configuration	Value
Apache Version	Apache/2.4.33 (Unix) PHP/7.4.33
Apache API Version	20120211

Se creó una base de datos en el servidor PostgreSQL de la máquina Slackware mediante el comando `CREATE DATABASE estudiantes OWNER santiago;`. Una vez dentro de esta base de datos, se definió la tabla `notas` utilizando el comando

```
postgres@natalia:~$ psql
Password for user postgres:
psql (14.19)
Type "help" for help.

postgres=# CREATE DATABASE estudiantes OWNER santiago;
postgres=# \c
postgres=# CREATE DATABASE estudiantes OWNER santiago;
postgres=# \c
postgres@natalia:~$ psql -U santiago -d estudiantes -U
Password:
psql (14.19)
Type "help" for help.

estudiantes=# CREATE TABLE notas ( id SERIAL PRIMARY KEY, nombre VARCHAR(100), nota_1 INT, nota_2 INT, nota_3 INT);
estudiantes=# CREATE TABLE
estudiantes=#
```

<pre>CREATE TABLE notas ( id SERIAL PRIMARY KEY, nombre VARCHAR(100), nota_1 INT, nota_2 INT, nota_3 INT );, para registrar las calificaciones correspondientes.</pre>	
--	--

## **Instalación de Software Base**

### **Administración de Redes en Azure**

#### **¿Qué observa al actualizar el sitio web repetidamente?**

Según AarDavMax (n.d.), al refrescar repetidamente la aplicación web, lo que se observa en los paneles de Live Metrics y Overview de Application Insights es una respuesta instantánea del sistema de telemetría, operando con una latencia de menos de un segundo.

- Incremento de Tasas: Se verá un aumento inmediato en métricas clave como:
  - Tasa de Solicitudes (Request Rate): Muestra el número de solicitudes HTTP entrantes por unidad de tiempo.
  - Tasa de Dependencias (Dependency Rate): Si la web app realiza llamadas a una base de datos o un servicio externo (que es probable en una app real), veremos un aumento en las llamadas de dependencia.
  - Tasa de Entrada/Salida de Datos (Data In/Out Rate): Muestra el volumen de tráfico de red.
- Métricas de Rendimiento: La tasa de Tiempo de Respuesta del Servidor (Server Response Time) puede fluctuar ligeramente, y las métricas de rendimiento del host como el Uso de CPU y el Uso de Memoria también podrían mostrar picos momentáneos, dependiendo de la carga.
- Muestra de Telemetría: En la sección inferior de Live Stream, aparecerán muestras detalladas de solicitudes exitosas, mostrando el tiempo de duración, el nombre de la solicitud y el código de respuesta (probablemente 200 OK).

## ¿Qué representa cada uno de los elementos que se muestran?

Según AarDavMax (n.d.), cada ítem en el panel de Live Metrics representa una pieza crucial de telemetría agregada en tiempo real:

- Request Rate (Tasa de Solicitudes): Indicador primario de tráfico. Mide la demanda del servicio.
- Failed Requests (Solicitudes Fallidas): Indicador directo de fallos. Muestra fallos de aplicación (códigos 5xx) o cliente (códigos 4xx) al instante.
- Server Response Time (Tiempo de Respuesta): Indicador de experiencia de usuario (UX).  
Tiempo promedio que tarda el servidor en procesar la solicitud.
- Dependency Rate/Duration (Tasa de Dependencias): Indicador de salud de sistemas externos. Mide llamadas a bases de datos, APIs de terceros, o colas.
- CPU Usage / Memory Usage (Uso de CPU/Memoria): Indicadores de salud del host. Muestran la utilización de recursos de la máquina virtual o el contenedor subyacente.
- Sample Telemetry (Muestra de Telemetría): Eventos detallados a nivel de transacción.  
Proporciona rastros de pila (stack traces) y detalles de fallos para depuración en producción sin necesidad de reiniciar o modificar logs.

## **¿Qué otras funcionalidades ofrece el sistema “Application Insights” para servicios web, bases de datos y otros sistemas implementados en la nube?**

Según AarDavMax (n.d.), Application Insights se extiende mucho más allá del monitoreo en vivo, ofreciendo un conjunto de herramientas de Observabilidad (APM) para el ciclo de vida completo del software:

- Para Servicios Web y Aplicaciones:
  - Mapa de Aplicaciones (Application Map): Visualiza la arquitectura de la aplicación y las interacciones entre componentes (servicios web, bases de datos, colas) y sus dependencias externas, mostrando métricas de salud y rendimiento en cada nodo.
  - Búsqueda de Transacciones (Transaction Search): Permite rastrear transacciones individuales de extremo a extremo para diagnósticos detallados, correlacionando telemetría de solicitud, dependencia y registros.
  - Análisis de Uso (Usage Analysis): Herramientas para entender el comportamiento del usuario, incluyendo análisis de usuarios, sesiones, eventos y embudos de conversión (funnels).
  - Disponibilidad (Availability): Monitoreo sintético y de ping web que prueba la disponibilidad del endpoint de la aplicación desde múltiples ubicaciones geográficas.
- Para Bases de Datos y Otros Sistemas Cloud:

- Application Insights rastrea el rendimiento de las llamadas de dependencia a bases de datos (SQL, NoSQL), Azure Storage, Azure Service Bus, y otros. Esto permite identificar consultas lentas o cuellos de botella en la capa de datos.
- Integración con Logs (Log Analytics), usando el lenguaje de consulta KQL, para realizar análisis profundos y personalizados en los datos de telemetría retenidos.

### **¿Cómo podría ser beneficiosa esta funcionalidad en un entorno corporativo?**

Según IngridAtMicrosoft (n.d.), la utilidad de esta funcionalidad en un entorno corporativo se articula en términos de Riesgo, Eficiencia y Experiencia del Cliente:

1. Reducción del Tiempo Medio de Reparación (MTTR): El monitoreo en tiempo real y la telemetría correlacionada permiten a los equipos de DevOps y SRE identificar y diagnosticar la causa raíz de un incidente (por ejemplo, una dependencia lenta o un error en el código) en minutos en lugar de horas.
2. Optimización Proactiva de Costos y Rendimiento: Al correlacionar el uso de CPU/Memoria con la tasa de solicitudes y el tiempo de respuesta, la empresa puede ajustar de manera precisa las políticas de Autoescalado (escalamiento horizontal) o dimensionar correctamente los recursos (escalamiento vertical), evitando el sobredimensionamiento (reduciendo costos) o el subdimensionamiento (mejorando el rendimiento).
3. Análisis de Impacto Empresarial: El uso del Análisis de Uso (funnels, cohortes) transforma los datos técnicos en métricas de negocio, permitiendo a la gerencia y a los

equipos de producto tomar decisiones informadas sobre las prioridades de desarrollo. Por ejemplo, identificar dónde los usuarios abandonan el proceso de compra.

4. Cumplimiento del Acuerdo de Nivel de Servicio (SLA): La función de Disponibilidad asegura que la empresa pueda medir y demostrar el cumplimiento de su SLA de tiempo de actividad a sus clientes.

### **¿Cómo contribuye la capa de red a la transmisión de datos entre la aplicación web y los clientes en diferentes ubicaciones geográficas?**

Según Sillam (2024), la capa de red es el eje central de la comunicación inter-red, fundamental para conectar la aplicación web a clientes en diferentes ubicaciones geográficas.

- **Direccionamiento Lógico (IP):** La Capa de Red, a través del Protocolo de Internet (IP), asigna una dirección lógica única a la aplicación web (el servidor) y a cada cliente. Esto permite que los paquetes de datos sean dirigidos de forma precisa a través de la vasta interconexión de redes.
- **Encaminamiento (Routing):** Los dispositivos de esta capa (principalmente routers) utilizan algoritmos de enrutamiento para determinar la mejor ruta (la más eficiente, no siempre la más corta) para que un paquete de datos viaje desde el servidor en una región de Azure (por ejemplo, Este de EE. UU.) hasta un cliente en otra parte del mundo (por ejemplo, Europa). El router examina la dirección IP de destino y utiliza su tabla de enrutamiento para reenviar el paquete hacia el siguiente salto de red.

- Fragmentación: La Capa de Red puede fragmentar los paquetes IP si exceden la Unidad de Transmisión Máxima (MTU) de cualquier enlace físico intermedio, asegurando que la comunicación sea compatible con redes de distintas capacidades.

**¿De qué manera las capas de aplicación y transporte garantizan una transferencia de datos fiable y una correcta interpretación de las respuestas de la aplicación web?**

#### ***Capa de Transporte (Layer 4) - Confiabilidad y Entrega de Proceso a Proceso***

La Capa de Transporte (predominantemente utilizando TCP para web apps) es la responsable de la entrega confiable de datos entre el proceso de la aplicación en el servidor y el proceso en el cliente.

- Segmentación y Reensamblaje: El TCP divide los datos del mensaje en segmentos en el origen y los reensambla en el destino.
- Control de Errores (Checksums): TCP incluye un checksum para detectar la corrupción de datos en tránsito. Si se detecta un error, el segmento se descarta y se solicita una retransmisión.
- Control de Flujo y Congestión: TCP utiliza ventanas deslizantes para gestionar la velocidad de envío de datos y evitar abrumar al receptor o a la red.
- Conexión y Acuse de Recibo (ACK): TCP establece una conexión lógica (three-way handshake), numera los segmentos de datos y requiere que el receptor envíe un acuse de recibo (ACK) para cada segmento. Si no se recibe un ACK dentro de un tiempo de espera, el segmento se retransmite, garantizando que todos los datos lleguen y lo hagan en el orden correcto.



### ***Capa de Aplicación (Layer 7) - Interpretación Correcta***

La Capa de Aplicación (usando HTTP/HTTPS para la web app) asegura que la respuesta del servidor se interprete correctamente por el cliente (el navegador).

- **Definición de Protocolo:** HTTP define el formato de las solicitudes (requests) y respuestas (responses), incluyendo verbos (GET, POST), encabezados (headers) y códigos de estado (200 OK, 404 Not Found, 500 Internal Server Error).
- **Formato de Datos:** Esta capa maneja la codificación y el formato de los datos (e.g., HTML, JSON, XML), a menudo con la ayuda de la Capa de Presentación (Layer 6). Por ejemplo, un encabezado Content-Type: text/html le indica al navegador que interprete la carga útil como una página web.
- **Autenticación y Sesión:** Los protocolos de la capa de aplicación gestionan los mecanismos de inicio de sesión, cookies y tokens de sesión, asegurando que el cliente reciba la respuesta de la aplicación correcta y personalizada.

(Transport Layer & Application Layer, n.d.)

## **Experimentos**

### **Uso de mensajes ICMP**

Usando el comando tracert en la consola de Windows (CMD), se consultó la ruta de red hacia el sitio web francés de ropa <https://www.lahalle.com/>, con el fin de evaluar la conectividad y los nodos intermedios que participan en el acceso al servidor.

## Figura 1

### *Análisis de ruta de red hacia el sitio web de La Halle*

```
C:\Windows\System32>tracert www.lahalle.com

Tracing route to commcloud.prod-bchm-lahalle-com.cc-ecdn.net.cdn.cloudflare.net [104.18.41.11]
over a maximum of 30 hops:

  1  <1 ms    <1 ms    <1 ms    192.168.1.1
  2   3 ms     3 ms     3 ms     152.201.96.1
  3   5 ms     4 ms     4 ms     telefonica1-nap.ccit.org.co [206.223.124.156]
  4   5 ms     5 ms     5 ms     internexa1-nap.ccit.org.co [206.223.124.154]
  5   *        *        *        Request timed out.
  6   6 ms     5 ms     5 ms     179.1.95.127
  7   5 ms     4 ms     4 ms     104.18.41.11

Trace complete.

C:\Windows\System32>
```

*Nota.* Captura de pantalla del resultado del comando tracert mostrando los saltos desde la red local hasta el servidor de Cloudflare que aloja contenido para La Halle.

El resultado del comando tracert evidencia que la conexión desde el equipo local hasta el servidor de destino (104.18.41.11), que corresponde a un nodo de Cloudflare utilizado por el sitio web de La Halle, se realiza de manera eficiente y sin interrupciones críticas. Los tiempos de respuesta son bajos en todos los saltos, lo que indica una buena calidad de conexión, y aunque uno de los nodos intermedios no respondió (salto 5), esto no afectó la llegada exitosa al destino, ya que es común que algunos routers bloqueen las respuestas ICMP por razones de seguridad. En conjunto, el trazado demuestra que no existen problemas de conectividad entre la red local y el servidor final.

## **Algunas preguntas de comandos para Routers**

**¿Cuál es la diferencia entre la contraseña de habilitación y el secreto de habilitación? Si ambos están configurados, ¿cuál tiene prioridad?**

La enable password y la enable secret son como dos cerraduras en la puerta de acceso privilegiado de un dispositivo de red. La enable password es la cerradura original: almacena la contraseña en texto plano o con un cifrado débil (Tipo 7), fácilmente reversible y poco segura. La enable secret es la versión moderna: usa hashing unidireccional (MD5 o SHA-256), lo que la hace mucho más robusta.

Si ambas están configuradas, enable secret siempre tiene precedencia. Esta evolución refleja cómo la seguridad de acceso al dispositivo ha pasado de una protección básica a un mecanismo confiable y resistente a ataques.

(User Security Configuration Guide, Cisco IOS Release 15MT - Configuring Security With Passwords, Privileges, and Logins [Support], 2020)

**¿Cuál es la diferencia entre consola y VTY?**

Segun Cisco Systems (2008), console y VTY son dos formas de acceder a la gestión de un dispositivo de red. La línea console es un acceso fuera de banda (OOB): requiere conexión física al puerto de consola, ideal para configuración inicial, recuperación de contraseñas o solución de problemas cuando la red no funciona. La línea VTY es un acceso dentro de banda (In-Band): permite administración remota vía red (Telnet o SSH) y se usa para la gestión diaria.

En otros términos, la consola se conecta como “hardware directo”, mientras que VTY funciona como un “socket de red” que abstrae el hardware. La consola suele tener una línea (0) y requiere cable especial y terminal, mientras que VTY ofrece múltiples líneas (0 a 4 o más) y depende de una red IP funcional. Para seguridad, VTY debe usar SSH, mientras que la consola confía en login local o password.

### **¿Cuál es el proceso de arranque de los routers del Laboratorio de Redes?**

El proceso de arranque de un router Cisco es una secuencia crítica que asegura que tanto el plano de control como el plano de datos estén listos para operar. Comienza con el POST (Power-On Self-Test), que verifica hardware como CPU, memoria y interfaces. Luego se carga el bootloader (ROMmon), encargado de localizar la imagen del sistema operativo (IOS), generalmente en la memoria Flash.

Una vez localizada, la imagen de IOS se carga en RAM y se ejecuta. Finalmente, el router busca el archivo de configuración de inicio (startup-config) en NVRAM: si existe, se copia a RAM como running-config; si no, el router entra en Setup Mode para configuración inicial. Este proceso garantiza que el router arranque de manera controlada y segura, listo para funcionar.

(CCNA Training » Cisco Router Boot Sequence Tutorial, n.d.)

## **¿Qué tipos de memoria tienen los routers del Laboratorio de Redes?**

Según Dulaney (2021), los routers Cisco utilizan cuatro tipos de memoria, cada una con un rol específico:

- Flash (no volátil): almacena la imagen completa del IOS y otros archivos del sistema.
- RAM/DRAM (volátil): ejecuta el IOS en funcionamiento, el running-config y las tablas de enrutamiento y ARP.
- NVRAM (no volátil): guarda el archivo startup-config, es decir, la configuración permanente.
- ROM (no volátil): contiene el bootloader (ROMmon) y una versión de respaldo del IOS, además de ejecutar el POST al encender el router.

Esta combinación asegura que el sistema operativo pueda ejecutarse eficientemente y que la configuración y el firmware persistan incluso tras un reinicio.

## **¿Cuál es la diferencia entre los archivos de configuración de inicio y de ejecución?**

La running-configuration es la configuración activa del dispositivo, almacenada en RAM; refleja todos los cambios realizados en tiempo real y se pierde al reiniciar. La startup-configuration es la configuración persistente, guardada en NVRAM, que se carga automáticamente al arrancar el dispositivo.

Para que los cambios del running-config sean permanentes, deben copiarse explícitamente al startup-config mediante `copy running-config startup-config` (o `write memory`).

Mantenerlos sincronizados es clave para garantizar que la red arranque con la configuración deseada y evitar pérdida de cambios.

(Managing Configuration Files, 2013)

### Acceso y configuración básica de Routers

Dirección de red	88.0.0.0/10
Representación binaria	01011000.00000000.00000000.00000000
Local	88.0.0.1
Broadcast	88.63.255.255
Max IP	88.63.255.254
Min IP	88.0.0.2
Mask	255.192.0.0

Para los 4000 hosts de la izquierda se realiza el subnetting correspondiente:

$$2^x - 2 \geq 4000, \text{ lo que implica que } x \text{ es } 12, \text{ ya que } 2^{12} - 2 = 4094$$

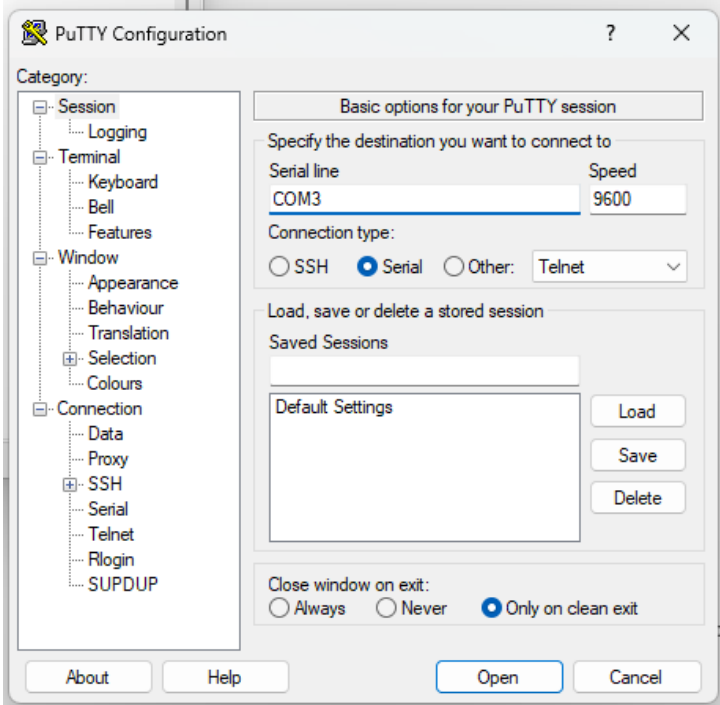
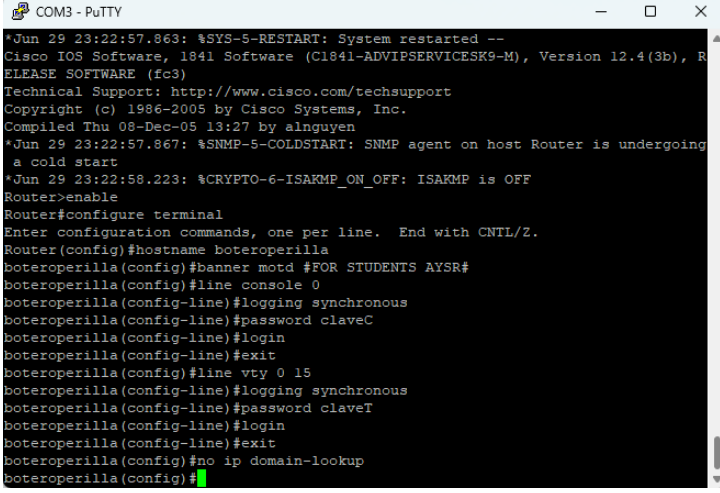
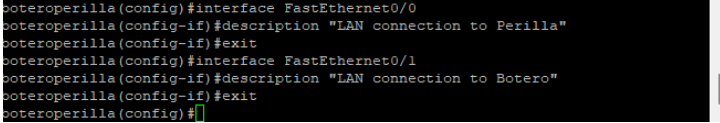
Dirección de red	88.0.16.0
Representación binaria	01011000.00000000.00010000.00000000
Gateway	88.0.16.1
Broadcast	88.0.31.255
Min IP	88.0.16.2

Max IP	88.0.31.254
Mask	255.255.240.0

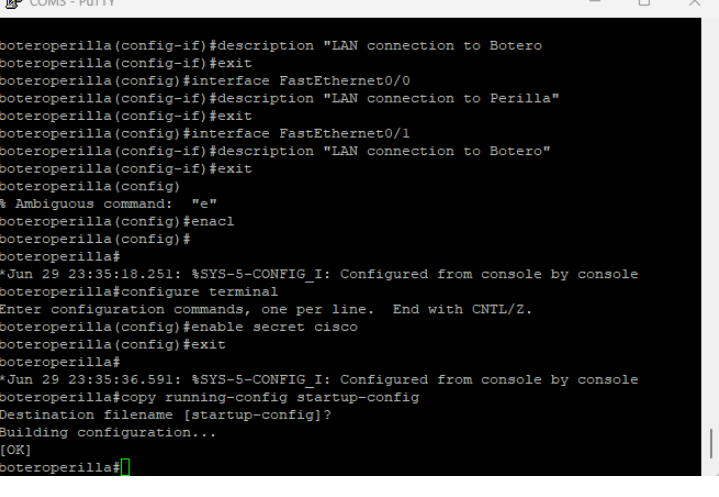
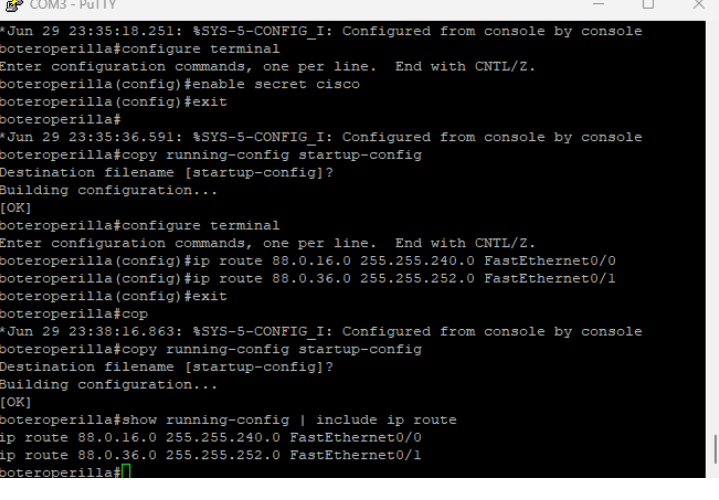
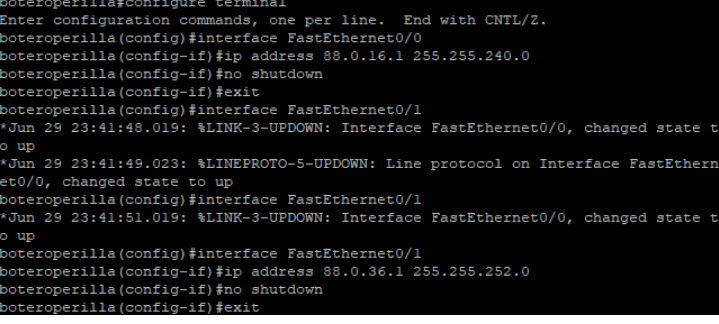
Para los 600 hosts de la derecha se realiza el subnetting correspondiente:

$$2^x - 2 \geq 600, \text{ lo que implica que } x \text{ es } 10, \text{ ya que } 2^{10} - 2 = 1022$$

Dirección de red	88.0.36.0
Representación binaria	01011000.00000000.00100100.00000000
Gateway	88.0.36.1
Broadcast	88.0.39.255
Min IP	88.0.36.2
Max IP	88.0.39.254
Mask	255.255.252.0

Acción Realizada	Captura de Pantalla
<p>Se configuró un router Cisco 1841 utilizando un cable de consola. La conexión se realizó mediante el software PuTTY, seleccionando el tipo de conexión serial, usando el puerto COM3 y una velocidad de transmisión de 9600 baudios.</p>	
<p>Se procedió a cambiar el hostname del router y a configurar las contraseñas de acceso. En la línea console 0 se estableció una contraseña y se habilitó el comando logging synchronous. Además, se desactivó la resolución de nombres DNS mediante el comando no ip domain-lookup.</p>	
<p>Se modificaron las descripciones de las interfaces conectadas,</p>	



<p>identificando correctamente los enlaces correspondientes a los dos equipos conectados al router.</p>	
<p>La configuración actual fue guardada en la memoria no volátil utilizando el comando copy running-config startup-config, y se habilitó el modo privilegiado (enable).</p>	 <pre> boteroperilla(config-if)#description "LAN connection to Botero boteroperilla(config-if)#exit boteroperilla(config)#interface FastEthernet0/0 boteroperilla(config-if)#description "LAN connection to Perilla" boteroperilla(config-if)#exit boteroperilla(config)#interface FastEthernet0/1 boteroperilla(config-if)#description "LAN connection to Botero" boteroperilla(config-if)#exit boteroperilla(config) % Ambiguous command: "e" boteroperilla(config)#enable boteroperilla(config)# boteroperilla# *Jun 29 23:35:18.251: %SYS-5-CONFIG_I: Configured from console by console boteroperilla#configure terminal Enter configuration commands, one per line. End with CNTL/Z. boteroperilla(config)#enable secret cisco boteroperilla(config)#exit boteroperilla# *Jun 29 23:35:36.591: %SYS-5-CONFIG_I: Configured from console by console boteroperilla#copy running-config startup-config Destination filename [startup-config]? Building configuration... [OK] boteroperilla# </pre>
<p>Se configuraron las interfaces del router con las rutas IP correspondientes a las redes 88.0.16.0 con máscara 255.255.240.0 y 88.0.36.0 con máscara 255.255.252.0.</p>	 <pre> boteroperilla#configure terminal Enter configuration commands, one per line. End with CNTL/Z. boteroperilla(config)#enable secret cisco boteroperilla(config)#exit boteroperilla# *Jun 29 23:35:36.591: %SYS-5-CONFIG_I: Configured from console by console boteroperilla#copy running-config startup-config Destination filename [startup-config]? Building configuration... [OK] boteroperilla#configure terminal Enter configuration commands, one per line. End with CNTL/Z. boteroperilla(config)#ip route 88.0.16.0 255.255.240.0 FastEthernet0/0 boteroperilla(config)#ip route 88.0.36.0 255.255.252.0 FastEthernet0/1 boteroperilla(config)#exit boteroperilla#cop *Jun 29 23:38:16.863: %SYS-5-CONFIG_I: Configured from console by console boteroperilla#copy running-config startup-config Destination filename [startup-config]? Building configuration... [OK] boteroperilla#show running-config   include ip route ip route 88.0.16.0 255.255.240.0 FastEthernet0/0 ip route 88.0.36.0 255.255.252.0 FastEthernet0/1 boteroperilla# </pre>
<p>A cada interfaz del router se le asignó una dirección IP que funciona como gateway predeterminado para cada una de las redes configuradas.</p>	 <pre> boteroperilla#configure terminal Enter configuration commands, one per line. End with CNTL/Z. boteroperilla(config)#interface FastEthernet0/0 boteroperilla(config-if)#ip address 88.0.16.1 255.255.240.0 boteroperilla(config-if)#no shutdown boteroperilla(config-if)#exit boteroperilla(config)#interface FastEthernet0/1 *Jun 29 23:41:48.019: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t o up *Jun 29 23:41:49.023: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern et0/0, changed state to up boteroperilla(config)#interface FastEthernet0/1 *Jun 29 23:41:51.019: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t o up boteroperilla(config)#interface FastEthernet0/1 boteroperilla(config-if)#ip address 88.0.36.1 255.255.252.0 boteroperilla(config-if)#no shutdown boteroperilla(config-if)#exit </pre>

Mediante el comando show ip interface brief se verificó la configuración del router, confirmando que las interfaces se encontraban activas y en estado up.

```
boteroperilla#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Prot
occol					
FastEthernet0/0	88.0.16.1	YES	manual	up	up
FastEthernet0/1	88.0.36.1	YES	manual	up	up
Serial0/0/0	unassigned	YES	unset	administratively down	down
Serial0/0/1	unassigned	YES	unset	administratively down	down
Serial0/1/0	unassigned	YES	unset	administratively down	down
Serial0/1/1	unassigned	YES	unset	administratively down	down

```
boteroperilla#
```

En la primera máquina se configuró manualmente la dirección IP 88.0.16.2, con máscara 255.255.240.0, puerta de enlace 88.0.16.1 y servidor DNS 88.0.16.1.

### Edit IP settings

Manual

#### IPv4

On

IP address

88.0.16.2

Subnet mask

255.255.240.0

Gateway

88.0.16.1

#### Preferred DNS

88.0.16.1

DNS over HTTPS

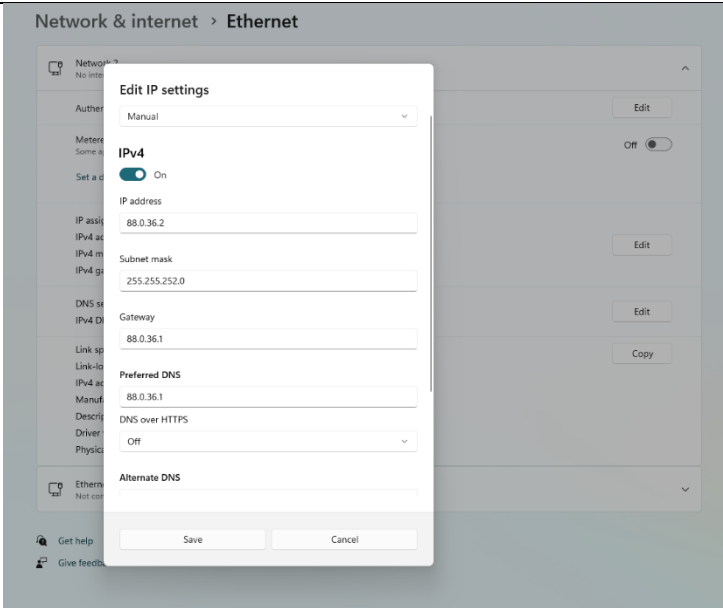
Off

#### Alternate DNS

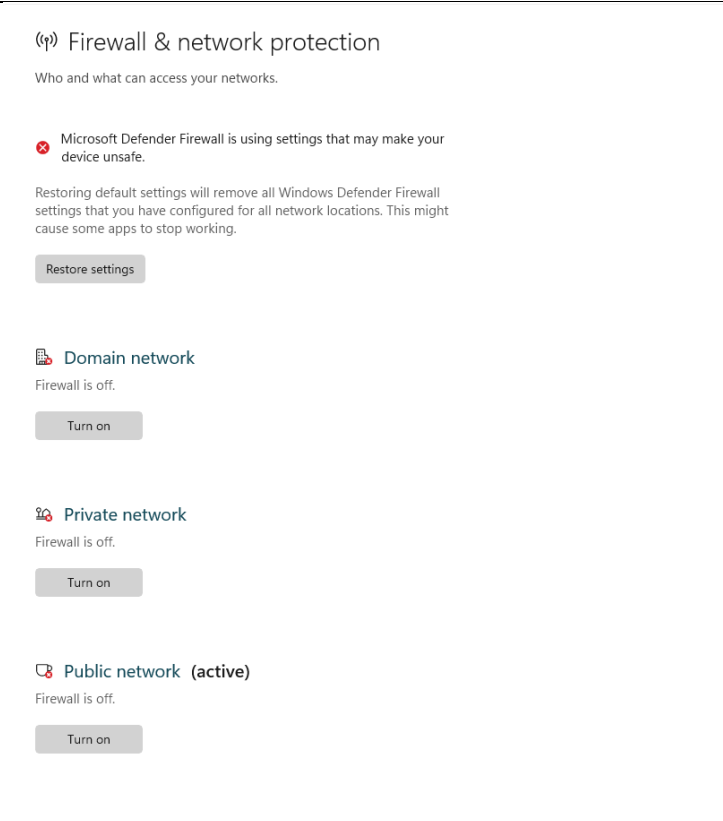
Save

Cancel

En la segunda máquina se realizó la configuración de red asignando la dirección IP 88.0.36.2, máscara 255.255.252.0, puerta de enlace 88.0.36.1 y servidor DNS 88.0.36.1.



Para efectos de prueba, se desactivó temporalmente el firewall del sistema, deshabilitando los perfiles de red de dominio, red privada y red pública.



<p>Como prueba de conectividad, desde la máquina con IP 88.0.16.2 se ejecutó un comando ping hacia la dirección 88.0.36.2, obteniendo un 0% de pérdida de paquetes y un tiempo promedio de 0 ms.</p>	 <pre> C:\Users\Redes&gt;ping 88.0.36.2  Pinging 88.0.36.2 with 32 bytes of data: Reply from 88.0.36.2: bytes=32 time=1ms TTL=127 Reply from 88.0.36.2: bytes=32 time=1ms TTL=127 Reply from 88.0.36.2: bytes=32 time=1ms TTL=127 Reply from 88.0.36.2: bytes=32 time=1ms TTL=127  Ping statistics for 88.0.36.2:     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),     Approximate round trip times in milli-seconds:         Minimum = 0ms, Maximum = 1ms, Average = 0ms  C:\Users\Redes&gt; </pre>
<p>De manera inversa, desde la máquina con IP 88.0.36.2 se realizó un ping hacia la dirección 88.0.16.2, obteniendo igualmente un 0% de pérdida de paquetes y un tiempo promedio de 0 ms.</p>	 <pre> C:\Users\Redes&gt;ping 88.0.16.2  Pinging 88.0.16.2 with 32 bytes of data: Reply from 88.0.16.2: bytes=32 time&lt;1ms TTL=127 Reply from 88.0.16.2: bytes=32 time&lt;1ms TTL=127 Reply from 88.0.16.2: bytes=32 time=1ms TTL=127 Reply from 88.0.16.2: bytes=32 time=1ms TTL=127  Ping statistics for 88.0.16.2:     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),     Approximate round trip times in milli-seconds:         Minimum = 0ms, Maximum = 1ms, Average = 0ms  C:\Users\Redes&gt; </pre>
<p>En este ejercicio se realizó la interconexión con otro router mediante una interfaz serial, a la cual se le asignó la dirección IP correspondiente para habilitar la comunicación entre ambos dispositivos.</p>	 <pre> boteroperilla#configure terminal Enter configuration commands, one per line. End with CNTL/Z. boteroperilla(config)#interface Serial0/0/0 boteroperilla(config-if)#ip address 100.0.24.2 255.255.255.248 boteroperilla(config-if)#no shutdown boteroperilla(config-if)# *Jun 30 00:14:31.295: %LINK-3-UPDOWN: Interface Serial0/0/0, changed state to do wn boteroperilla(config-if)#exit </pre>
<p>Finalmente, se verificó la conectividad entre routers mediante el envío de cinco mensajes ICMP</p>	 <pre> boteroperilla#ping 100.0.24.3  Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 100.0.24.3, timeout is 2 seconds: !!!! Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms boteroperilla# </pre>

Echo de 100 bytes, obteniendo una tasa de éxito del 100% y un tiempo promedio de ida y vuelta de 28 ms.	
---	--

## **Conclusiones**

El desarrollo de este laboratorio permitió consolidar una comprensión integral del funcionamiento de las redes de datos desde la capa física hasta la capa de aplicación, evidenciando cómo cada nivel del modelo OSI y TCP/IP contribuye de manera coordinada a la correcta prestación de servicios en entornos empresariales y de nube.

La implementación y simulación de redes con routers Cisco demostró la importancia de una configuración adecuada de direccionamiento IP, rutas estáticas y enlaces seriales, así como del uso correcto de mecanismos de acceso y seguridad (console, VTY, enable secret). Las pruebas de conectividad exitosas confirmaron que una planeación lógica adecuada garantiza la comunicación eficiente entre redes distintas.

La instalación y configuración de servidores web en entornos heterogéneos (Solaris, Slackware Linux y Windows Server) permitió evidenciar que, aunque los procedimientos varían según el sistema operativo, los principios fundamentales de los servicios web son consistentes. Asimismo, la integración con DNS demostró cómo la abstracción del direccionamiento IP mediante nombres de dominio mejora la usabilidad y la administración de la infraestructura.

El despliegue de una aplicación web con soporte para PHP y PostgreSQL reforzó la relación directa entre la capa de aplicación y los servicios de red subyacentes, mostrando cómo la correcta configuración del servidor, los permisos y las dependencias es crítica para el funcionamiento estable de aplicaciones reales.

La utilización de Application Insights en Azure permitió observar en tiempo real el comportamiento de una aplicación web, resaltando la relevancia de la observabilidad, el monitoreo proactivo y la telemetría en entornos corporativos. Estas herramientas facilitan la

detección temprana de fallos, la optimización del rendimiento y la toma de decisiones informadas tanto técnicas como de negocio.

Finalmente, los experimentos con ICMP, tracer y análisis de rutas, junto con el estudio del rol de las capas de red, transporte y aplicación, evidenciaron cómo la confiabilidad, el enrutamiento eficiente y la correcta interpretación de los datos son esenciales para garantizar una experiencia de usuario estable y segura, incluso cuando los clientes se encuentran en diferentes ubicaciones geográficas.

## Bibliografía

AarDavMax. (n.d.). *Diagnose with live metrics - Application Insights - Azure Monitor*. Microsoft

Learn. <https://learn.microsoft.com/en-us/azure/azure-monitor/app/live-stream?tabs=otel>

AarDavMax. (n.d.-b). *Introducción a la observabilidad de OpenTelemetry de Application*

*Insights - Azure Monitor*. Microsoft Learn. <https://learn.microsoft.com/es-es/azure/azure-monitor/app/app-insights-overview>

CCNA Training » *Cisco Router Boot sequence Tutorial*. (n.d.). [https://www.9tut.com/cisco-](https://www.9tut.com/cisco-router-boot-sequence-tutorial)

[router-boot-sequence-tutorial](https://www.9tut.com/cisco-router-boot-sequence-tutorial)

Cisco Systems, Inc. (2008, 11 de mayo). *Cisco Nexus 5000 Series Switch CLI software*

*configuration guide – Configuring access and trunk interfaces* (Guide No. OL-16597-01).

Recuperado de

<https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/configuration/guide/cli/CLIConfigurationGuide/AccessTrunk.pdf>

Dulaney, E. (2021). *CompTIA Network+ N10-008 Exam Cram* (7.<sup>a</sup> ed.). Pearson.

ISBN 978-0137375622.

[https://ptgmedia.pearsoncmg.com/images/9780137375769/samplepages/9780137375769\\_Sample.pdf](https://ptgmedia.pearsoncmg.com/images/9780137375769/samplepages/9780137375769_Sample.pdf)

IngridAtMicrosoft. (n.d.). *Escalabilidad automática en Azure Monitor - Azure Monitor*.

Microsoft Learn. <https://learn.microsoft.com/es-es/azure/azure-monitor/autoscale/autoscale-overview>



*Managing configuration Files.* (2013, October 30). Cisco.

[https://www.cisco.com/c/en/us/td/docs/ios/ios\\_xe/fundamentals/configuration/guide/TIPs\\_conversion/config\\_mgmt\\_xe\\_3s\\_Book/cf\\_config-files\\_xe.html](https://www.cisco.com/c/en/us/td/docs/ios/ios_xe/fundamentals/configuration/guide/TIPs_conversion/config_mgmt_xe_3s_Book/cf_config-files_xe.html)

Sillam, Y. (2024, November 28). *What is OSI Model | 7 Layers Explained* | Imperva. Learning Center. <https://www.imperva.com/learn/application-security/osi-model/>

*Transport layer & application layer.* (n.d.). <https://pravin-hub-rgb.github.io/BCA/resources/sem3/datatbc301/unit5/index.html>

User Security Configuration Guide, Cisco IOS Release 15MT - Configuring Security with Passwords, Privileges, and Logins [Support]. (2020, August 13). Cisco.

[https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec\\_usr\\_cfg/configuration/15-mt/sec-usr-cfg-15-mt-book/sec-cfg-sec-4cli.html](https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_cfg/configuration/15-mt/sec-usr-cfg-15-mt-book/sec-cfg-sec-4cli.html)