

# Anexo TP1: Implementación Física de la Red Propuesta

## Resumen de Características - Cisco Catalyst 2950 Series

### 1. Concepto General

- **Switches fijos y gestionados de capa 2:** Conmutación 10/100 Mbps con uplinks Gigabit.
- **Cisco IOS Standard Image (SI):** Funcionalidades básicas de datos, voz y video.
- **Administración web integrada:** Cisco Device Manager y Cisco Network Assistant.

### 2. Modelos Principales

Modelo	Puertos Fast Ethernet	Uplinks
Cisco Catalyst 2950SX-48	48 × 10/100 Mbps	2 × 1000BASE-SX (fibra)
Cisco Catalyst 2950T-48	48 × 10/100 Mbps	2 × 10/100/1000BASE-T (cobre)
Cisco Catalyst 2950SX-24	24 × 10/100 Mbps	2 × 1000BASE-SX (fibra)
Cisco Catalyst 2950-24	24 × 10/100 Mbps	No
Cisco Catalyst 2950-12	12 × 10/100 Mbps	No

### 3. Rendimiento y Disponibilidad

- **Ancho de banda de conmutación:** Hasta **13.6 Gbps**.
- **Wire-speed switching** en todos los puertos.
- **Spanning Tree Protocol (STP, RSTP, MSTP, PVST+):** Redundancia sin bucles.
- **EtherChannel / LACP:** Agregación de enlaces para mayor ancho de banda.
- **Soporte de RPS (fuente de alimentación redundante).**

### 4. Funciones de Seguridad

- **802.1X (autenticación por puerto).**
- **Port Security:** Control de acceso por MAC address.
- **SSHv2:** Administración segura por CLI.
- **TACACS+ / RADIUS:** Integración con servidores de autenticación.
- **Private VLAN Edge:** Aislamiento de puertos en el switch.
- **SNMPv3 (no criptográfico):** Monitoreo seguro.

### 5. Calidad de Servicio (QoS)

- **802.1p (CoS) y clasificación de tráfico por puerto.**
- **4 colas de prioridad por puerto (Strict Priority / WRR).**
- **Soporte para priorización de tráfico de voz y video.**

### 6. Administración

- **Cisco Device Manager:** Configuración vía navegador.

- **Cisco Express Setup:** Inicio rápido sin CLI.
- **Cisco Network Assistant:** Gestión centralizada de múltiples switches.
- **SNMP (v1, v2, v3) y RMON:** Monitoreo y estadísticas.
- **CiscoWorks:** Gestión avanzada de red.

## 7. Características Físicas y Consumo

- **Factor de forma:** 1U, rackable.
- **Dimensiones:** De **9.52"** a **13"** de profundidad (según modelo).
- **Peso:** Entre **6.5 lb** y **10.5 lb**.
- **Temperatura de operación:** 0°C a 45°C.
- **Consumo:** Entre **30W** y **45W** según el modelo.

## Procedimientos realizados para la Configuración y Administración de la Red

### a. Conectar una PC al puerto de consola del switch Cisco a 9600 baudios utilizando PuTTY

- ☒ **Preparar conexión física:**
  - ☒ Conectar el cable de consola (RJ-45 a DB-9 o USB a RJ-45) entre la PC y el switch.
  - ☒ Verificar el puerto COM asignado en la PC (p. ej., usando el Administrador de Dispositivos).
- ☒ **Configurar y abrir PuTTY:**
  - ☒ Ejecutar PuTTY.
  - ☒ Seleccionar el modo **Serial**.
  - ☒ Ingresar los parámetros:
    - **Puerto COM:** (ejemplo: COM3)
    - **Baud rate:** 9600
    - **Data bits:** 8
    - **Parity:** Ninguna
    - **Stop bits:** 1
    - **Flow control:** Ninguno
  - ☐ (Opcional) Iniciar PuTTY desde la línea de comandos

**Fue realizado mediante la GUI**

```
putty.exe -serial COM3 -sercfg 9600,8,n,1,N
```

### b. Acceder a las opciones de administración del switch y modificar claves de acceso

*Para esta parte de la experiencia y la configuración en general, se utilizó una cuarta PC, perteneciente a uno de los integrantes de Los Peladitos, aunque fue hecha de manera inter-grupal entre Los Peladitos, NoLoSonIEEE y Taylor Switch.*

- ☒ **Acceder al switch:**
  - ☒ Conectar la PC al switch (vía consola o mediante conexión Ethernet a la IP de administración).
  - ☒ Iniciar sesión usando PuTTY (o navegador si es vía web).

- ☒ **Ingresar al modo privilegiado y de configuración:**

- ☒ Ingresar el comando para modo privilegiado:

```
enable
```

- ☒ Entrar al modo de configuración global:

```
configure terminal
```

- ☒ **Modificar claves de acceso:**

Una vez concluida la actividad, la contraseña fue reestablecida a la original ilustrada en el rótulo del dispositivo conmutador.

- ☒ Cambiar la contraseña del modo privilegiado:

```
enable secret [nueva_contraseña]
```

- ☐ (Opcional) Configurar contraseñas para acceso remoto (línea VTY):

**Omisión debido al hecho de que utilizamos la Consola mediante el conector Rs 232.**

```
line vty 0 15
password [nueva_contraseña_vty]
login
exit
```

- ☒ **Guardar la configuración:**

- ☒ Ejecutar:

```
copy running-config startup-config
```

c. Conectar dos computadoras al switch, configurar una red y testear conectividad

- ☒ **Conexión física:**

- ☒ Conectar cada computadora a un puerto del switch usando cables Ethernet (Verificado mediante la Figura a1).
- ☒ Verificar que las luces de enlace en los puertos estén activas.

```

Switch>enable
Password:
Switch#
00:39:35: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
00:39:37: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
Switch#show ip interface brief
00:40:15: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
00:40:17: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to up
% Unknown command or computer name, or unable to find computer address
Switch#show ip interface brief

```

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan1	unassigned	YES	NVRAM	administratively down	down
FastEthernet0/1	unassigned	YES	unset	up	up
FastEthernet0/2	unassigned	YES	unset	up	up
FastEthernet0/3	unassigned	YES	unset	down	down
FastEthernet0/4	unassigned	YES	unset	down	down
FastEthernet0/5	unassigned	YES	unset	down	down
FastEthernet0/6	unassigned	YES	unset	down	down
FastEthernet0/7	unassigned	YES	unset	down	down
FastEthernet0/8	unassigned	YES	unset	down	down
FastEthernet0/9	unassigned	YES	unset	down	down
FastEthernet0/10	unassigned	YES	unset	down	down
FastEthernet0/11	unassigned	YES	unset	down	down

Figura a1. Demostración de los puertos del Switch Up & Running desde PC 4 (PC de Configuración).

- ☒ **Configurar direcciones IP en cada computadora:**
  - ☒ En PC1 (es decir la perteneciente al grupo nuestro, NoLoSonIEEE), asignar la IP estática (192.168.1.10/24).
  - ☒ En PC2 (correspondiente al grupo de Taylor Switch), asignar una IP estática (192.168.1.15/24).

Ambos ítems fueron realizados mediante los comandos:

```
sudo ip addr add 192.168.1.10/24 dev eno1
```

```
sudo ip addr add 192.168.1.15/24 dev eno1
```

- ☒ **Testear conectividad:**
  - ☒ En PC1, abrir la terminal y ejecutar:
 

```
ping 192.168.1.15
```
  - ☒ Verificar que se reciben respuestas de PC2 (Figura a2).

```
francisco-javier-vasquez@francisco-javier-vasquez-Latitude-3410:~$ ping 192.168.1.15
PING 192.168.1.15 (192.168.1.15) 56(84) bytes of data.
64 bytes from 192.168.1.15: icmp_seq=1 ttl=64 time=0.518 ms
64 bytes from 192.168.1.15: icmp_seq=2 ttl=64 time=1.18 ms
64 bytes from 192.168.1.15: icmp_seq=3 ttl=64 time=0.340 ms
64 bytes from 192.168.1.15: icmp_seq=4 ttl=64 time=0.937 ms
64 bytes from 192.168.1.15: icmp_seq=5 ttl=64 time=1.23 ms
64 bytes from 192.168.1.15: icmp_seq=6 ttl=64 time=0.268 ms
64 bytes from 192.168.1.15: icmp_seq=7 ttl=64 time=0.531 ms
64 bytes from 192.168.1.15: icmp_seq=8 ttl=64 time=1.10 ms
64 bytes from 192.168.1.15: icmp_seq=9 ttl=64 time=1.24 ms
64 bytes from 192.168.1.15: icmp_seq=10 ttl=64 time=1.31 ms
64 bytes from 192.168.1.15: icmp_seq=11 ttl=64 time=1.17 ms
64 bytes from 192.168.1.15: icmp_seq=12 ttl=64 time=1.12 ms
^C
--- 192.168.1.15 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11134ms
rtt min/avg/max/mdev = 0.268/0.912/1.313/0.368 ms
```

Figura a2. Ping exitoso realizado desde PC1 (NoLoSonIEEE) a PC2 (TaylorSwitch).

d. Configurar un puerto del switch en modo mirroring y monitorear, con una tercera computadora, el tráfico entre las dos computadoras conectadas en el procedimiento c.

- ☒ **Conexión física:**
  - ☒ Conectar la tercera computadora al switch usando un cable Ethernet.
- ☒ **Configurar port mirroring en el switch: (Figura a3)**

```
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface FastEthernet0/3
Switch(config-if)#no shutdown
Switch(config-if)#exit
Switch(config)#monitor session 1 source interface FastEthernet0/1
Switch(config)#monitor session 1 source interface FastEthernet0/2
Switch(config)#monitor session 1 destination interface FastEthernet0/3
Switch(config)#end
```

Figura a3. Configuración de Mirroring desde PC 4 (PC de Configuración).

- ☒ Acceder al modo de configuración:

```
configure terminal
```

- ☒ Configurar la sesión de monitorización (ejemplo: sesión 1):
  - ☒ Establecer el puerto fuente (GigabitEthernet0/2):

```
monitor session 1 source interface GigabitEthernet0/2
```

- ☒ Establecer el puerto de destino (GigabitEthernet0/3, conectado a la PC de monitoreo):

```
monitor session 1 destination interface GigabitEthernet0/3
```

- ☒ **Guardar la configuración:**

- ☒ Ejecutar:

```
copy running-config startup-config
```

- ☒ **Monitorear tráfico ICMP:**

- ☒ En la tercera computadora, abrir una herramienta de análisis de red (En este caso se utilizó Wireshark).
- ☒ Iniciar la captura en la interfaz correspondiente.
- ☒ Mientras se ejecuta el comando **ping** entre PC1 y PC2 (Figura a2) y viceversa, observar el tráfico capturado (Figura a4).

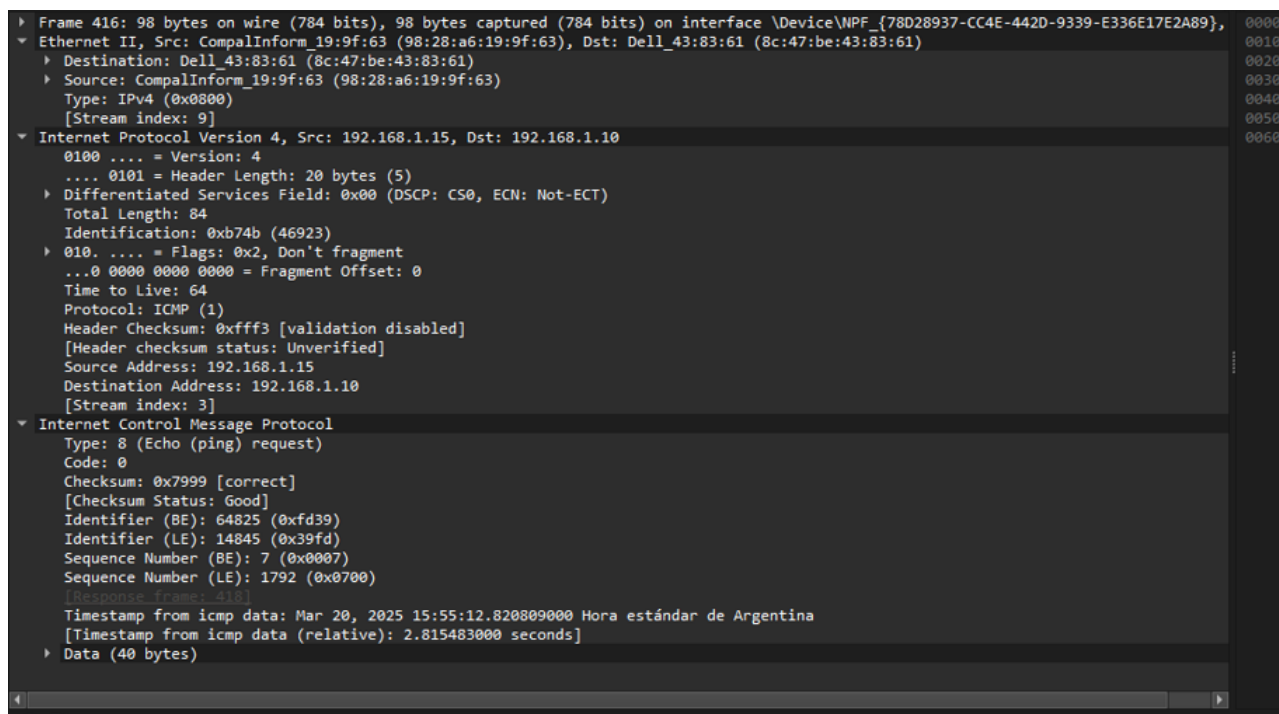


Figura a4. Ping exitoso siendo realizado desde PC2 (TaylorSwitch) hacia PC1 (NoLoSonIEEE). Interceptado mediante Wireshark por PC3 (Los Peladitos).

- ☒ **Monitorear tráfico de un paquete UDP personalizado:**

- ☒ Entre PC1 y PC2 se envía, mediante un script de python, el payload "Hola mundo desde NoLoSonIEEE para Taylor Switch. Escucha Los Peladitos" \$^1\$ (en formato binario; Recepción visualizada en la Figura a5).

```
# Código de Emisión
import socket
```

```

ip_destino = '192.168.1.15'
puerto = 5005

mensaje = b'Hola Mundo, desde NoLoSonIEEE, para Taylor Switch. Escucha
Los Peladitos.'

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(mensaje, (ip_destino, puerto))
sock.close()

```

```

# Código de Recepción
import socket

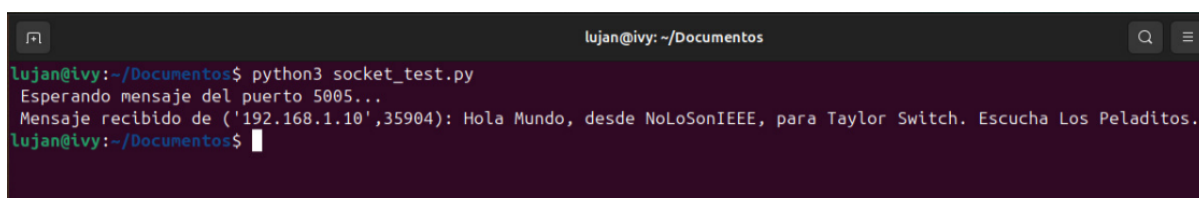
# Configurar socket UDP
ip_local = "192.168.1.10"
puerto = 5005

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((ip_local, puerto))

print(f"Esperando mensaje del puerto {puerto}...")

while True:
    data, addr = sock.recvfrom(1024) # Recibir hasta 1024 bytes
    print(f"Mensaje recibido de {addr}: {data.decode()}")

```



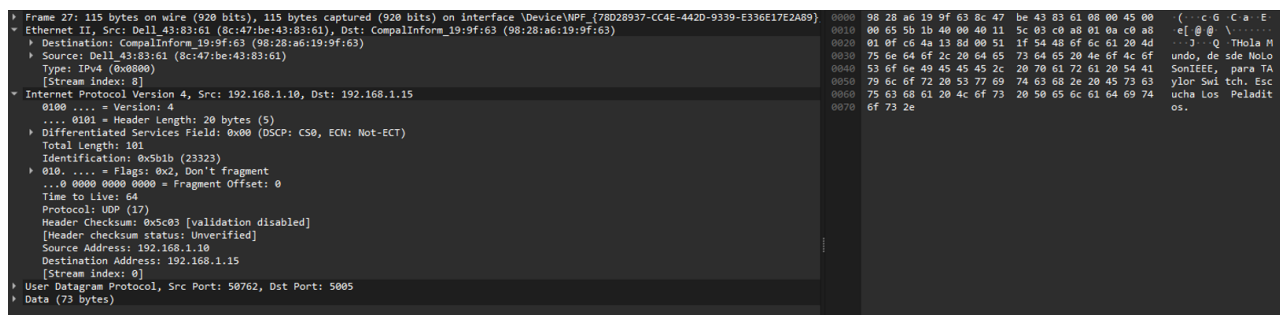
```

lujan@ivy: ~/Documentos
lujan@ivy:~/Documentos$ python3 socket_test.py
Esperando mensaje del puerto 5005...
Mensaje recibido de ('192.168.1.10',35904): Hola Mundo, desde NoLoSonIEEE, para Taylor Switch. Escucha Los Peladitos.
lujan@ivy:~/Documentos$

```

Figura a5. Recpección del tráfico UDP por Taylor Switch, enviado por NoLoSonIEEE.

- ☒ Debido al Mirroring, Los Peladitos escucharán el mensaje y serán capaces de interceptar el payload (Ver Figura a6).



```

Frame 27: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) on interface \Device\NPF{78028937-CC4E-442D-9339-E336E17E2A09}
  Ethernet II, Src: Dell_43:83:61 (8c:47:be:43:83:61), Dst: CompalInform_19:9f:63 (98:28:a6:19:9f:63)
    Destination: CompalInform_19:9f:63 (98:28:a6:19:9f:63)
    Source: Dell_43:83:61 (8c:47:be:43:83:61)
    Type: IPv4 (0x0000)
  [Stream index: 0]
  Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.15
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total length: 101
    Identification: 0x5b1b (23323)
    010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0x5c03 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.10
    Destination Address: 192.168.1.15
    [Stream index: 0]
  User Datagram Protocol, Src Port: 50762, Dst Port: 5005
  Data (73 bytes)
    0000  98 28 a6 19 0f 63 8c 47 be 43 83 61 08 00 45 00  ( . . . G C a E
    0010  00 65 5b 1b 40 00 40 11 5c 03 c0 a8 01 0a c0 a8  e [ @ @ \ . . .
    0020  01 0f c6 4a 13 8d 00 51 1f 54 48 6f 6c 61 20 4d  . . J . . Q Thola M
    0030  75 6e 64 6f 2c 20 64 65 73 64 65 20 4e 6f 4c 6f  und, de sde NoLo
    0040  53 6f 6e 49 45 45 45 2c 20 70 61 72 61 20 54 41  SonIEEE, para TA
    0050  79 6c 6f 72 20 53 77 69 74 63 68 2e 20 45 73 63  ylor Swi tch, Esc
    0060  75 63 68 61 20 4c 6f 73 20 50 65 6c 61 64 69 74  ucha Los Peladit
    0070  6f 73 2e  os.

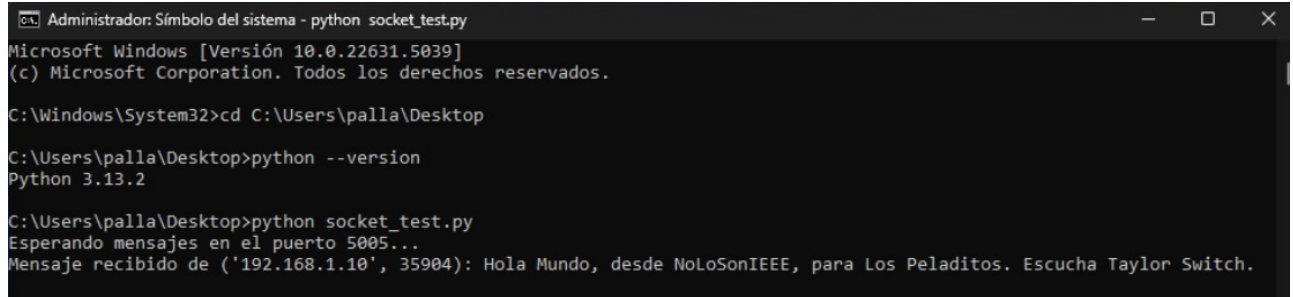
```

Figura a6. Intercepción del tráfico UDP desde PC 3 con el mensaje "Hola Mundo, desde NoLoSonIEEE, para Taylor Switch. Escucha Los Peladitos." (Se aprecia a la derecha)



## Observaciones

- \$^1\$ En el medio del procedimiento se generó una fe de erratas con respecto a la capitalización de la letra *a* en el nombre del grupo "Taylor Switch". Esto fue corregido en instancias dispares entre las capturas (se realizaron dos intentos exitosos con esa diferencia), de ahí la discrepancia tipográfica de esta letra entre el mensaje obtenido (Figura a5) y el mensaje interceptado (Figura a6).
- El procedimiento se repitió parcialmente bajo la dinámica "envía NoLoSonIEEE, recibe Los Peladitos, escucha Taylor Switch". Sin embargo, el tiempo se acabó antes de poder culminar con la tarea repetida. De todas maneras, la Figura a7 ilustra el evento de recepción.



```
Administrador: Símbolo del sistema - python socket_test.py
Microsoft Windows [Versión 10.0.22631.5039]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>cd C:\Users\palla\Desktop

C:\Users\palla\Desktop>python --version
Python 3.13.2

C:\Users\palla\Desktop>python socket_test.py
Esperando mensajes en el puerto 5005...
Mensaje recibido de ('192.168.1.10', 35904): Hola Mundo, desde NoLoSonIEEE, para Los Peladitos. Escucha Taylor Switch.
```

*Figura a7. Envío alternativo del mensaje UDP, desde NoLoSonIEEE hacia Los Peladitos (Grupo que en la experiencia anterior escuchó en el puerto de mirroring), usando la dirección IP de recepción 192.168.1.11/24*