

# CONFIGURACIÓN Y DESARROLLO DE APLICACIONES EN REDES



## **UNIVERSIDAD**

Universidad Nacional del Centro de la Provincia de Buenos Aires

## **INTEGRANTES**

Cordeiro, Federico (federico.cordeiro95@gmail.com)

Elis, Abigail (abigailelis.s@hotmail.com)

Piliavsky, Pablo (pablopiliavsky21@gmail.com)

## **PROFESOR/A**

Cristian Mateos Diaz

Jorge Hernández Gauna

## **MATERIA**

Configuración y Desarrollo de Aplicaciones en Redes

## **FECHA DE ENTREGA**

16/10/23



# **UNICEN**

Universidad Nacional del Centro  
de la Provincia de Buenos Aires

- 4** **INTRO Y TOPOLOGIA**  
Introduccion, estructura y descripcion
- 5** **VLSM - GENERAL**  
VLSM, tabla y descripcion
- 6** **VLSM - FABRICA A**  
LSM, tabla y descripcion
- 7** **VLSM - FABRICA B**  
LSM, tabla y descripcion
- 9** **TABLAS DE RUTEO**  
Tablas de ruteo, de los distintos routers
- 12** **MINIMIZACION**  
Minimizacion del router seleccionado (R2)
- 13** **TRACEROUTE Y PING**  
Traceroute y Ping del router seleccionado (R2)
- 14** **COMANDOS**  
Comandos utilizados en el router  
seleccionado (R2)

## INTRODUCCIÓN

Antes de comenzar con el desarrollo del informe, aclararemos algunas especificaciones sobre las decisiones tomadas a la hora de llevar a cabo este proyecto:

- En algunos casos, como los servers del datacenter, decidimos seleccionar aquellas IPs debido a la posibilidad de agregar futuros routers, y tener la posibilidad de utilizar las primeras IPs disponibles de dicho rango.
- En relación a los routers R3 y R4, tomamos la decisión de seleccionar a R3 como router de entrada a las distintas fábricas. Y, por otro lado, seleccionamos a R4 como router de salida para enviar al datacenter.
- En cuanto a las fábricas optamos por realizar una sub división dentro de la misma subred, previendo futuras expansiones y solapamiento de direcciones.

## TOPOLOGÍA DE RED

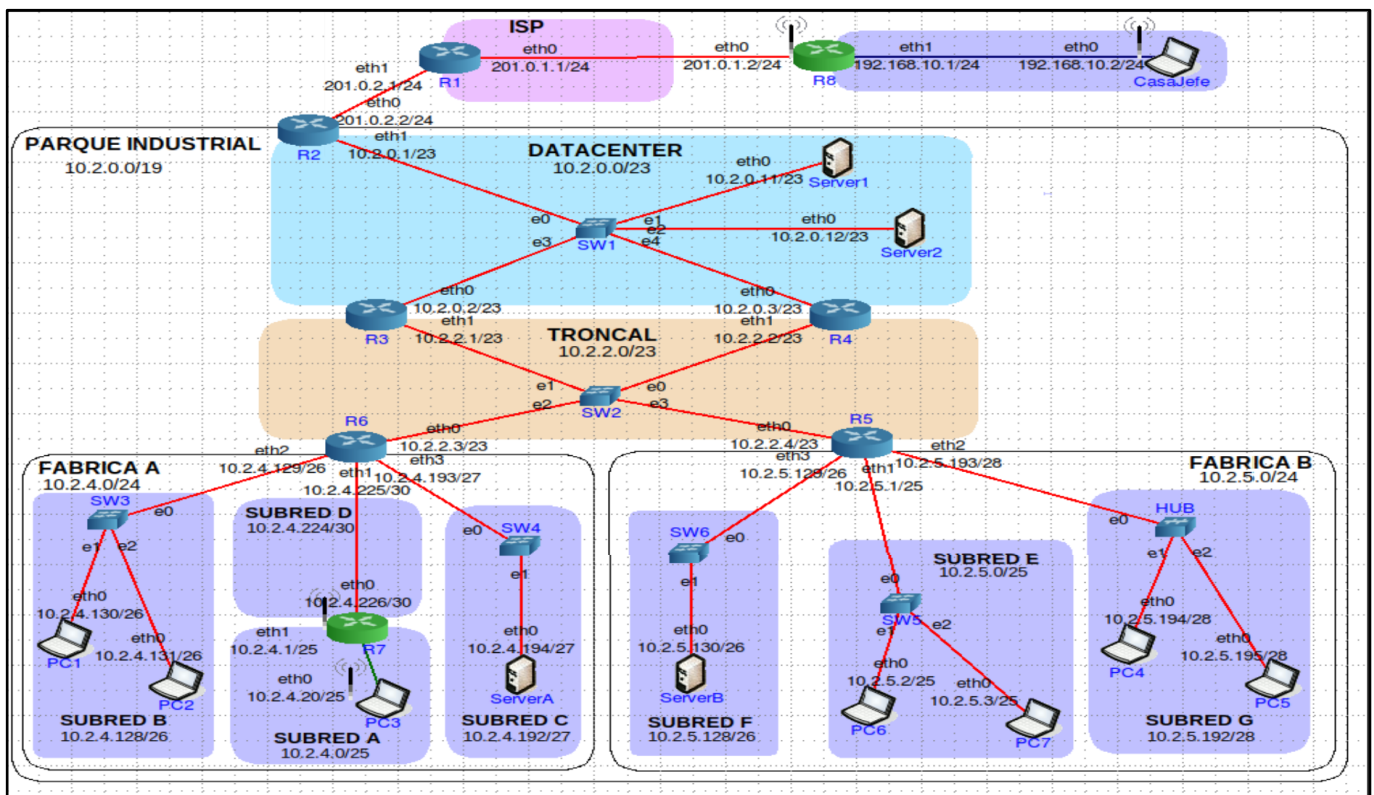


Imagen 1

En esta imagen, se puede observar las distintas subredes, tanto aquellas que están ubicadas dentro del parque industrial, como las que están por fuera del mismo.

- **ISP**
- **Particular casa jefe**
- **Parque industrial**
  - Datacenter
  - Troncal
  - Fabrica A (Subred A, Subred B, Subred C, Subred D)
  - Fabrica B (Subred E, Subred F, Subred G)

## VLSM - GENERAL

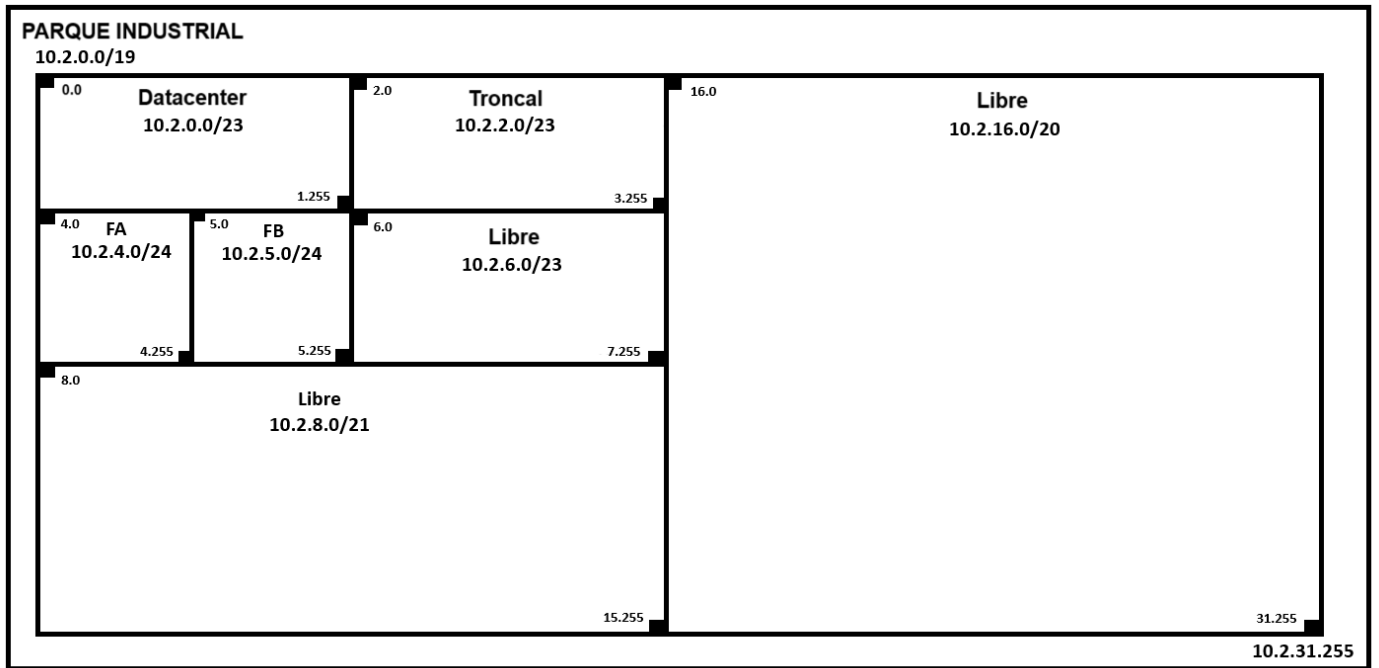


Imagen 2

NOMBRE DE RED	DIRECCIÓN BASE	MASCARA	BROADCAST	RANGO ASIGNABLE
<b>Parque industrial</b>	10.2.0.0	/19	10.2.31.255	10.2.0.1 10.2.31.254
<b>Datacenter</b>	10.2.0.0	/23	10.2.1.255	10.2.0.1 10.2.1.254
<b>Troncal</b>	10.2.2.0	/23	10.2.3.255	10.2.2.1 10.2.3.254
<b>Fabrica A</b>	10.2.4.0	/24	10.2.4.255	10.2.4.1 10.2.4.254
<b>Fabrica B</b>	10.2.5.0	/24	10.2.5.255	10.2.5.1 10.2.5.254

Al realizar las divisiones correspondientes en el VLSM, tanto para el parque industrial en general, como también para las fábricas individuales, realizamos las siguientes consideraciones teniendo en cuenta que nuestras direcciones privadas se encuentran dentro de la red 10.2.0.0/19:

**Datacenter:** Se debe considerar, los 500 posibles servidores, 3 routers de frontera DMZ, y las IPs correspondientes a la red base y el broadcast. Con un total de 505 direcciones necesarias para dicha subred.

Para calcular la cantidad de bits necesarios para los hosts, buscamos la potencia de 2 más cercana a 505, que es  $2^9 = 512$ .

Por lo tanto, le asignamos un bloque de 512 direcciones IP, con una máscara /23. Quedando libres 7 IPs asignables, para futuros routers.

**Troncal:** Se debe considerar, las 300 posibles fábricas, 2 routers (R3 y R4), y las IPs correspondientes a la red base y el broadcast. Con un total de 304 direcciones necesarias para dicha subred.

Para calcular la cantidad de bits necesarios para los hosts, buscamos la potencia de 2 más cercana a 304, que es  $2^9 = 512$ .

Por lo tanto, le asignamos un bloque de 512 direcciones IP, con una máscara /23. Quedando libres 7 IPs asignables, para futuros routers.

## VLSM – FABRICA A

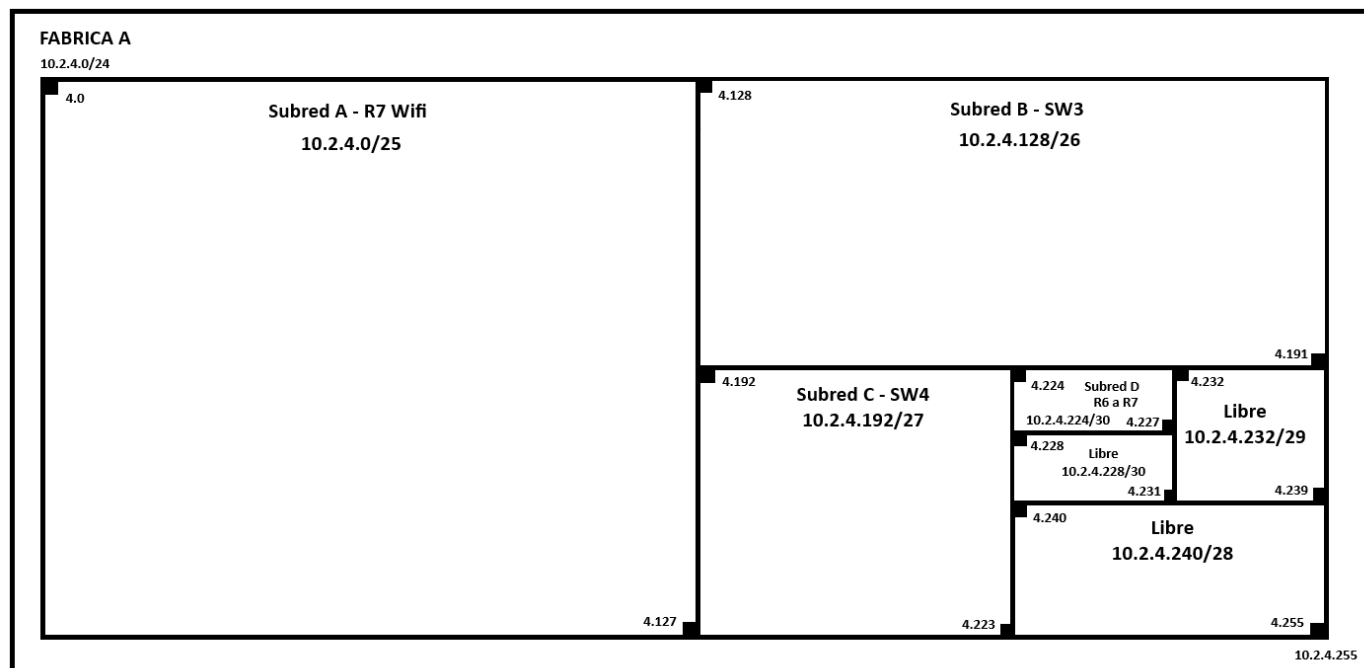


Imagen 3

NOMBRE DE RED	DIRECCIÓN BASE	MASCARA	BROADCAST	RANGO ASIGNABLE
Subred A R7 Wifi	10.2.4.0	/25	10.2.4.127	10.2.4.1 10.2.4.126
Subred B SW3	10.2.4.128	/26	10.2.4.191	10.2.4.129 10.2.4.190
Subred C SW4	10.2.4.192	/27	10.2.4.223	10.2.4.193 10.2.4.222
Subred D R6 a R7	10.2.4.224	/30	10.2.4.227	10.2.4.225 10.2.4.226

**Fabrica A:** Para realizar el VLSM de la fábrica A. Primero se realizaron los cálculos necesarios para cada subred de dicha fabrica.

**En la Subred A (R7 Wifi):** Debemos contemplar 80 posibles direcciones como mínimo, sumado al R7, la red base y el broadcast. Dando una totalidad de 83 direcciones necesarias. Para calcular la cantidad de bits necesarios para los hosts, buscamos la potencia de 2 más cercana a 83, que es  $2^7 = 128$ .

Por lo tanto, le asignamos un bloque de 128 direcciones IP, con una máscara /25.

**En la Subred B (SW3):** Debemos contemplar 45 posibles direcciones como mínimo, sumado al R6, la red base y el broadcast. Dando una totalidad de 48 direcciones necesarias. Para calcular la cantidad de bits necesarios para los hosts, buscamos la potencia de 2 más cercana a 48, que es  $2^6 = 64$ .

Por lo tanto, le asignamos un bloque de 64 direcciones IP, con una máscara /26.

**En la Subred C (SW4):** Debemos contemplar 16 posibles direcciones como mínimo, sumado al R6, la red base y el broadcast. Dando una totalidad de 19 direcciones necesarias. Para calcular la cantidad de bits necesarios para los hosts, buscamos la potencia de 2 más cercana a 19, que es  $2^5 = 32$ .

Por lo tanto, le asignamos un bloque de 32 direcciones IP, con una máscara /27.

**En la Subred D (R6 a R7):** Debemos contemplar ambos routers, más la dirección red base y el broadcast. Dando una totalidad de 4 direcciones necesarias. Para calcular la cantidad de bits necesarios para los hosts, buscamos la potencia de 2 más cercana a 4, que es  $2^2 = 4$ .

Por lo tanto, le asignamos un bloque de 4 direcciones IP, con una máscara /30.

**Para calcular el rango necesario de IPs para la fábrica A, realizamos:**

(Subred A (128 IPs) + Subred B (64 IPs) + Subred C (32 IPs) + Subred D (4 IPs)) = 228 IPs

La potencia de 2, más cercana a 228, es  $2^8 = 256$ . Por lo tanto, le asignamos un bloque de 256 direcciones IP, con una máscara /24.

## VLSM – FABRICA B

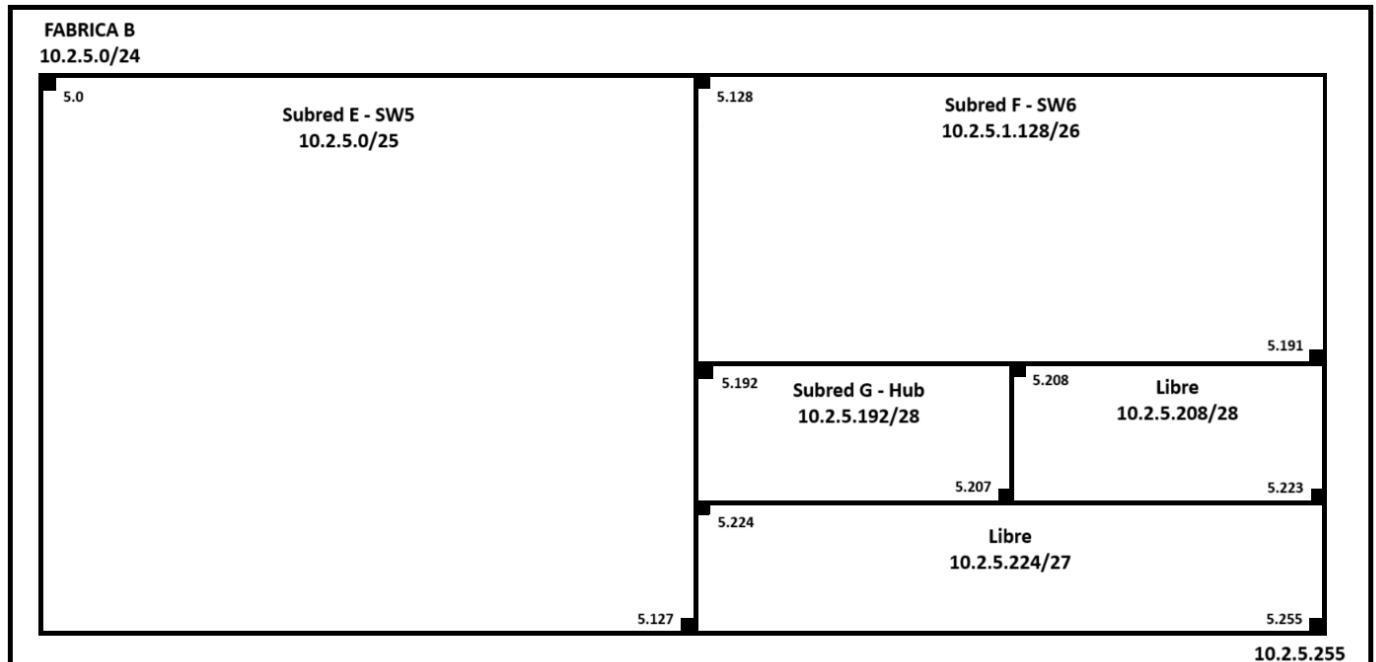


Imagen 4

NOMBRE DE RED	DIRECCIÓN BASE	MASCARA	BROADCAST	RANGO ASIGNABLE
<b>Subred E SW5</b>	10.2.5.0	/25	10.2.5.127	10.2.5.1 10.2.5.126
<b>Subred F SW6</b>	10.2.5.128	/26	10.2.5.191	10.2.5.129 10.2.5.190
<b>Subred G Hub</b>	10.2.5.192	/28	10.2.5.207	10.2.5.193 10.2.5.206

**Fabrica B:** Para realizar el VLSM de la fábrica B. Primero se realizaron los cálculos necesarios para cada subred de dicha fabrica.

**En la Subred E (SW5):** Debemos contemplar 65 posibles direcciones como mínimo, sumado al R5, la red base y el broadcast. Dando una totalidad de 68 direcciones necesarias. Para calcular la cantidad de bits necesarios para los hosts, buscamos la potencia de 2 más cercana a 68, que es  $2^7 = 128$ .

Por lo tanto, le asignamos un bloque de 128 direcciones IP, con una máscara /25.

**En la Subred F (SW6):** Debemos contemplar 32 posibles direcciones como mínimo, sumado al R5, la red base y el broadcast. Dando una totalidad de 35 direcciones necesarias. Para calcular la cantidad de bits necesarios para los hosts, buscamos la potencia de 2 más cercana a 35, que es  $2^6 = 64$ .

Por lo tanto, le asignamos un bloque de 64 direcciones IP, con una máscara /26

**En la Subred G (Hub):** Debemos contemplar 8 posibles direcciones como mínimo, sumado al R5, la red base y el broadcast. Dando una totalidad de 11 direcciones necesarias. Para calcular la cantidad de bits necesarios para los hosts, buscamos la potencia de 2 más cercana a 11, que es  $2^4 = 16$ .

Por lo tanto, le asignamos un bloque de 16 direcciones IP, con una máscara /28

**Para calcular el rango necesario de IPs para la fábrica B, realizamos:**

(Subred E (128 IPs) + Subred F (64 IPs) + Subred G (16 IPs)) = 208 IPs

La potencia de 2, más cercana a 208, es  $2^8 = 256$ . Por lo tanto, le asignamos un bloque de 256 direcciones IP, con una máscara /24.



## TABLAS DE RUTEO

R7 Wifi			
Red	Directa/Indirecta	Interfaz	Próximo router
10.2.4.0/25	Directa	Eth1	-
10.2.4.224/30	Directa	Eth0	-
Default (0.0.0.0/0)	Indirecta	Eth0	10.2.4.225

```
root@R7:/tmp/pycore.35705/R7.conf# ip route
default via 10.2.4.225 dev eth0
10.2.4.0/25 dev eth1 proto kernel scope link src 10.2.4.1
10.2.4.224/30 dev eth0 proto kernel scope link src 10.2.4.226
root@R7:/tmp/pycore.35705/R7.conf#
```

Imagen 5

R6			
Red	Directa/Indirecta	Interfaz	Próximo router
10.2.2.0/23	Directa	Eth0	-
10.2.4.128/26	Directa	Eth2	-
10.2.4.192/27	Directa	Eth3	-
10.2.4.224/30	Directa	Eth1	-
10.2.4.0/25	Indirecta	Eth1	10.2.4.226
10.2.5.0/24	Indirecta	Eth0	10.2.2.4
Default (0.0.0.0/0)	Indirecta	Eth0	10.2.2.2

```
root@R6:/tmp/pycore.35705/R6.conf# ip route
default via 10.2.2.2 dev eth0
10.2.2.0/23 dev eth0 proto kernel scope link src 10.2.2.3
10.2.4.0/25 via 10.2.4.226 dev eth1
10.2.4.128/26 dev eth2 proto kernel scope link src 10.2.4.129
10.2.4.192/27 dev eth3 proto kernel scope link src 10.2.4.193
10.2.4.224/30 dev eth1 proto kernel scope link src 10.2.4.225
10.2.5.0/24 via 10.2.2.4 dev eth0
root@R6:/tmp/pycore.35705/R6.conf#
```

Imagen 6

R5			
Red	Directa/Indirecta	Interfaz	Próximo router
10.2.2.0/23	Directa	Eth0	-
10.2.5.0/25	Directa	Eth1	-
10.2.5.128/26	Directa	Eth3	-
10.2.5.192/28	Directa	Eth2	-
10.2.4.0/24	Indirecta	Eth0	10.2.2.3
Default (0.0.0.0/0)	Indirecta	Eth0	10.2.2.2

```
root@R5:/tmp/pycore.35705/R5.conf# ip route
default via 10.2.2.2 dev eth0
10.2.2.0/23 dev eth0 proto kernel scope link src 10.2.2.4
10.2.4.0/24 via 10.2.2.3 dev eth0
10.2.5.0/25 dev eth1 proto kernel scope link src 10.2.5.1
10.2.5.128/26 dev eth3 proto kernel scope link src 10.2.5.129
10.2.5.192/28 dev eth2 proto kernel scope link src 10.2.5.193
root@R5:/tmp/pycore.35705/R5.conf#
```

Imagen 7

R4			
Red	Directa/Indirecta	Interfaz	Próximo router
10.2.0.0/23	Directa	Eth0	-
10.2.2.0/23	Directa	Eth1	-
10.2.4.0/24	Indirecta	Eth1	10.2.2.3
10.2.5.0/24	Indirecta	Eth1	10.2.2.4
Default (0.0.0.0/0)	Indirecta	Eth0	10.2.0.1

```

root@R4:/tmp/pycore.35705/R4.conf# ip route
default via 10.2.0.1 dev eth0
10.2.0.0/23 dev eth0 proto kernel scope link src 10.2.0.3
10.2.2.0/23 dev eth1 proto kernel scope link src 10.2.2.2
10.2.4.0/24 via 10.2.2.3 dev eth1
10.2.5.0/24 via 10.2.2.4 dev eth1
root@R4:/tmp/pycore.35705/R4.conf#

```

Imagen 8

R3			
Red	Directa/Indirecta	Interfaz	Próximo router
10.2.0.0/23	Directa	Eth0	-
10.2.2.0/23	Directa	Eth1	-
10.2.4.0/24	Indirecta	Eth1	10.2.2.3
10.2.5.0/24	Indirecta	Eth1	10.2.2.4
Default (0.0.0.0/0)	Indirecta	Eth0	10.2.0.1

```

root@R3:/tmp/pycore.35705/R3.conf# ip route
default via 10.2.0.1 dev eth0
10.2.0.0/23 dev eth0 proto kernel scope link src 10.2.0.2
10.2.2.0/23 dev eth1 proto kernel scope link src 10.2.2.1
10.2.4.0/24 via 10.2.2.3 dev eth1
10.2.5.0/24 via 10.2.2.4 dev eth1
root@R3:/tmp/pycore.35705/R3.conf#

```

Imagen 9

R2			
Red	Directa/Indirecta	Interfaz	Próximo router
10.2.0.0/23	Directa	Eth1	-
201.0.2.0/24	Directa	Eth0	-
10.2.0.0/19	Indirecta	Eth1	10.2.0.2
Default (0.0.0.0/0)	Indirecta	Eth0	201.0.2.1

```

root@R2:/tmp/pycore.35705/R2.conf# ip route
default via 201.0.2.1 dev eth0
10.2.0.0/23 dev eth1 proto kernel scope link src 10.2.0.1
10.2.0.0/19 via 10.2.0.2 dev eth1
201.0.2.0/24 dev eth0 proto kernel scope link src 201.0.2.2
root@R2:/tmp/pycore.35705/R2.conf#

```

Imagen 10

R1			
Red	Directa/Indirecta	Interfaz	Próximo router
10.2.0.0/19	Indirecta	Eth1	201.0.2.2
192.168.10.0/24	Indirecta	Eth0	201.0.1.2
201.0.1.0/24	Directa	Eth0	-
201.0.2.0/24	Directa	Eth1	-

```

root@R1:/tmp/pycore.35705/R1.conf# ip route
10.2.0.0/19 via 201.0.2.2 dev eth1
192.168.10.0/24 via 201.0.1.2 dev eth0
201.0.1.0/24 dev eth0 proto kernel scope link src 201.0.1.1
201.0.2.0/24 dev eth1 proto kernel scope link src 201.0.2.1
root@R1:/tmp/pycore.35705/R1.conf# █

```

Imagen 11

R8			
Red	Directa/Indirecta	Interfaz	Próximo router
192.168.10.0/24	Directa	Eth1	-
201.0.1.0/24	Directa	Eth0	-
10.2.0.0/19	Indirecta	Eth0	201.0.1.1
Default (0.0.0.0/0)	Indirecta	Eth0	201.0.1.1

```

root@R8:/tmp/pycore.35705/R8.conf# ip route
default via 201.0.1.1 dev eth0
10.2.0.0/19 via 201.0.1.1 dev eth0
192.168.10.0/24 dev eth1 proto kernel scope link src 192.168.10.1
201.0.1.0/24 dev eth0 proto kernel scope link src 201.0.1.2
root@R8:/tmp/pycore.35705/R8.conf# █

```

Imagen 12

Con el comando `ip route`, se puede observar (*imagen 5 a 12*) las tablas de ruteo IP que se muestran en consola, correspondientes a cada tabla de enrutamiento establecidas estratégicamente.



## R2 ANTES DE MINIMIZACIÓN

```
root@R2:/tmp/pycore.36917/R2.conf# route
Tabla de rutas IP del núcleo
Destino          Pasarela          Genmask           Indic Métric Ref       Uso Interfaz
10.2.0.0          0.0.0.0           255.255.254.0     U      0      0         0 eth1
10.2.2.0          10.2.0.2          255.255.254.0     UG     2      0         0 eth1
10.2.4.0          10.2.0.2          255.255.255.128   UG     4      0         0 eth1
10.2.4.128        10.2.0.2          255.255.255.192   UG     3      0         0 eth1
10.2.4.192        10.2.0.2          255.255.255.224   UG     3      0         0 eth1
10.2.4.224        10.2.0.2          255.255.255.252   UG     3      0         0 eth1
10.2.5.0          10.2.0.2          255.255.255.128   UG     3      0         0 eth1
10.2.5.128        10.2.0.2          255.255.255.192   UG     3      0         0 eth1
10.2.5.192        10.2.0.2          255.255.255.240   UG     3      0         0 eth1
192.168.10.0      201.0.2.1         255.255.255.0     UG     3      0         0 eth0
201.0.1.0         201.0.2.1         255.255.255.0     UG     2      0         0 eth0
201.0.2.0         0.0.0.0           255.255.255.0     U      0      0         0 eth0
root@R2:/tmp/pycore.36917/R2.conf#
```

Imagen 13

## R2 DESPUÉS DE MINIMIZACIÓN

```
root@R2:/tmp/pycore.35705/R2.conf# ip route
default via 201.0.2.1 dev eth0
10.2.0.0/23 dev eth1 proto kernel scope link src 10.2.0.1
10.2.0.0/19 via 10.2.0.2 dev eth1
201.0.2.0/24 dev eth0 proto kernel scope link src 201.0.2.2
root@R2:/tmp/pycore.35705/R2.conf#
```

Imagen 14

## MINIMIZACIÓN DE R2



Imagen 15

Como se observa en la (imagen 13), en un comienzo el R2 tenía designada por defecto, comunicación con todas las subredes posibles, tanto directas como indirectas. Por esto, cambiamos la configuración de los servicios que permitían la asignación de dichas rutas, dejando activas únicamente aquellas que mediante los comandos (imagen 15), fueron agregadas de forma manual y estratégica, para la minimización y optimización del ruteo de dicho dispositivo.

- En el primer comando (imagen 15), configuramos y habilitamos la interfaz eth0, con la IP/MASK publica 201.0.2.2/24.
- En el segundo comando (imagen 15), configuramos y habilitamos la interfaz eth1, con la IP/MASK privada 10.2.0.1/23.
- En el tercer comando (imagen 15), configuramos la ruta por defecto para todas aquellas comunicaciones que no pertenezcan a la red del parque industrial (10.2.0.0/19) o alguna de sus subredes locales, delegando la responsabilidad al ISP.
- En el último comando (imagen 15), configuramos y agregamos una ruta general de acceso al parque industrial, por medio de R3 (10.2.0.2) el cuál designamos como router de entrada a las distintas fábricas y a la red troncal.

## COMPROBACIÓN DE RUTAS ASIGNADAS PARA EL TRÁFICO DE DATOS PARA R2

```
root@R2:/tmp/pycore.35705/R2.conf# traceroute 10.2.5.194
traceroute to 10.2.5.194 (10.2.5.194), 30 hops max, 60 byte packets
 1  10.2.0.2 (10.2.0.2)  0.053 ms  0.003 ms  0.002 ms
 2  10.2.2.4 (10.2.2.4)  0.036 ms  0.003 ms  0.003 ms
 3  10.2.5.194 (10.2.5.194)  0.020 ms  0.005 ms  0.005 ms
root@R2:/tmp/pycore.35705/R2.conf#
```

Imagen 16

En esta imagen, podemos observar la ruta que recorre el paquete enviado con dicho comando y los saltos intermedios hasta su destino, desde R2 hasta la PC4 (10.2.5.194), la cual se encuentra dentro de la fábrica B.

### Recorrido (imagen 16)

0. R2 - Origen
1. R3 (10.2.0.2) - Intermedio
2. R5 (10.2.2.4) - Intermedio
3. PC4 (10.2.5.194) - Destino

## ENVIÓ DE PAQUETE DE PROTOCOLO DE INTERNET (IP)

```
root@R2:/tmp/pycore.35705/R2.conf# ping 10.2.4.131
PING 10.2.4.131 (10.2.4.131) 56(84) bytes of data.
64 bytes from 10.2.4.131: icmp_seq=1 ttl=62 time=0.118 ms
64 bytes from 10.2.4.131: icmp_seq=2 ttl=62 time=0.073 ms
64 bytes from 10.2.4.131: icmp_seq=3 ttl=62 time=0.070 ms
64 bytes from 10.2.4.131: icmp_seq=4 ttl=62 time=0.072 ms
^C
--- 10.2.4.131 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3049ms
rtt min/avg/max/mdev = 0.070/0.083/0.118/0.021 ms
root@R2:/tmp/pycore.35705/R2.conf#
```

Imagen 17

En esta imagen, podemos observar que al momento de realizar un ping desde R2 a PC2 (10.2.4.131), que se encuentra dentro de la fábrica A, el canal tiene un 100% de estabilidad y el estado de la comunicación es óptimo, ya que, de los paquetes transmitidos hacia dicha dirección, se recibe la totalidad de los mismos.

## COMANDOS EN R2

### Route / Ip route

Con estos comandos, visualizamos la tabla de ruteo IP (ej: imagen 13 y 14).

### Traceroute <ip-dir>

Con este comando, se observan las rutas necesarias que recorre un paquete de protocolo de internet (IP) para llegar a destino (ej: imagen 16).

### Ping <ip-dir>

Empleamos este comando, para comprobar el estado de la comunicación entre el dispositivo actual, con otro equipo al que le corresponde la IP de destino, por medio del envío de paquetes ICMP de solicitud y de respuesta.

### Ifconfig <interface> <dir-ip/mask> <action>

Utilizamos este comando, para cambiar la configuración de la interfaz, indicando la dirección IP/MASK correspondiente. Con la opción, de poder emplear la acción “up” para habilitarla y “down” para deshabilitarla al momento de iniciar la sesión.

### Ip route add default via <dir-ip-destino> dev <output-interface>

Utilizamos este comando, para configurar la tabla de enrutamiento, agregando como ruta por defecto la dirección IP del Gateway, en la interfaz de salida establecida.

### ip route add <red/mask> via <dir-ip-destino> dev <output-interface>

Utilizamos este comando, para configurar la tabla de ruteo, con el objetivo de que el router actual, pueda acceder a una subred indirecta, via IP del próximo router, indicando la interfaz de salida.

## CONCLUSIÓN

En conclusión, este proyecto no solo ha sido una oportunidad para consolidar los conocimientos adquiridos en la asignatura relacionados con hardware y software, sino que también nos ha permitido aplicarlos en situaciones de la vida real. Esto no solo fomentó la discusión y el intercambio de ideas entre nosotros y nuestros compañeros, sino que también nos habilitó para vislumbrar futuras posibilidades de optimización en los procesos y las relaciones entre los diversos sistemas.