

# Listas de Acceso

Ing. José Restaino

Ing. Álvaro Sanchez

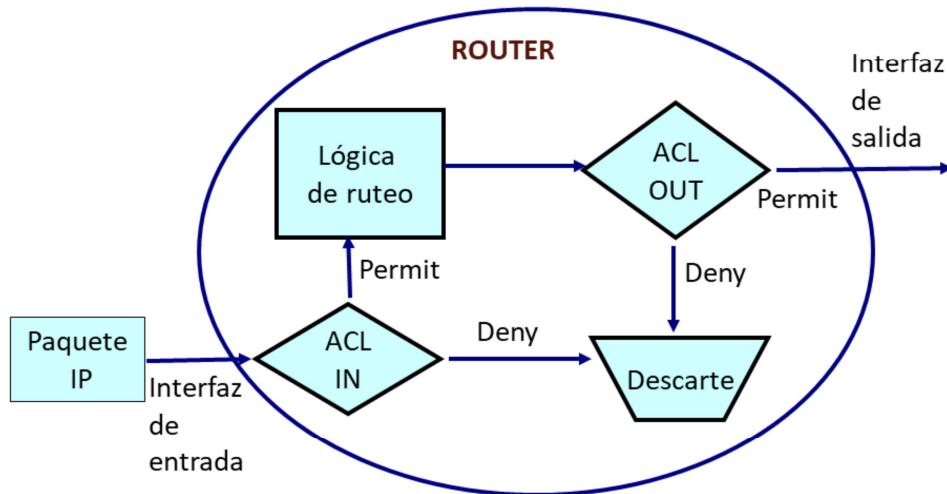
## Listas de Acceso

Las listas de acceso pueden ser utilizadas para realizar una variedad importante de funciones tales como:

- Filtrado de tráfico IP (ej. no permitir que paquetes IP provenientes de ciertas IP orígenes lleguen a ciertas IP destino).
- Restringir acceso a equipos.
- Filtrado de *update* de ruteos (ej. puedo decidir cuáles redes publicar y cuáles no).
- Marcado de paquetes para priorizar (útil para voz sobre IP)
- NAT

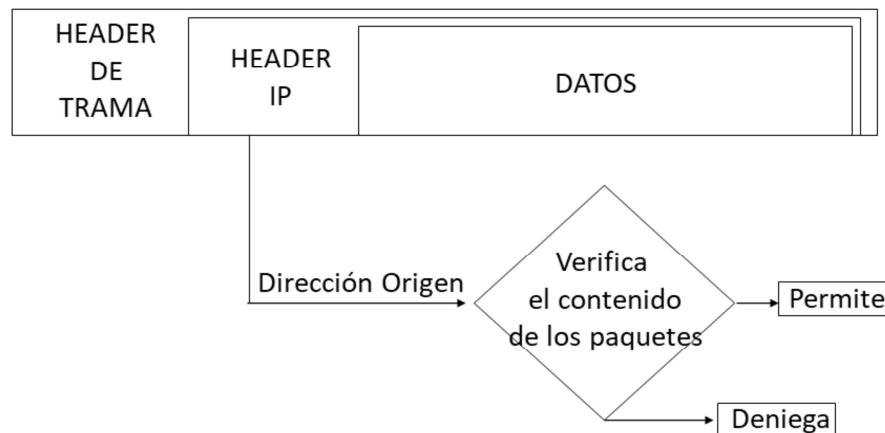
Se estudiarán en este capítulo los dos primeros puntos.

## Filtrado de tráfico IP



Nota: ACL IN: Lista de acceso aplicada a la entrada  
ACL OUT: Lista de acceso aplicada a la salida

## Análisis de paquetes con listas de acceso Estándar



## Listas de Acceso Estándar

```
R1(config)# access-list access-list-number {permit | deny} source [mask  
(wildcard)]
```

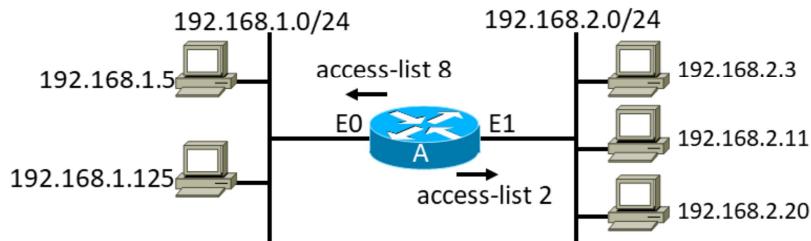
```
R1(config-if)# ip access-group access-list-number {in | out}
```

Significado:

- Access-list-number: Aquí se ingresa un numero entre 1 y 99 que indica que la lista de acceso es estándar.
- Permit | Deny: Esta es la acción a tomar en caso de coincidencia (bloquea o permite tráfico)
- Source: Dirección IP origen
- Mask: Indica cuáles bits de la dirección IP serán verificados en búsqueda de coincidencias.

Nota: La segunda línea de configuración permite asociar esta lista de acceso a una interfaz ya sea a la salida o a la entrada

## Ejemplo de Listas de Acceso Estándar



Ej. 1: Permitir el acceso a la red 192.168.2.0/24 solo al host 192.168.1.5

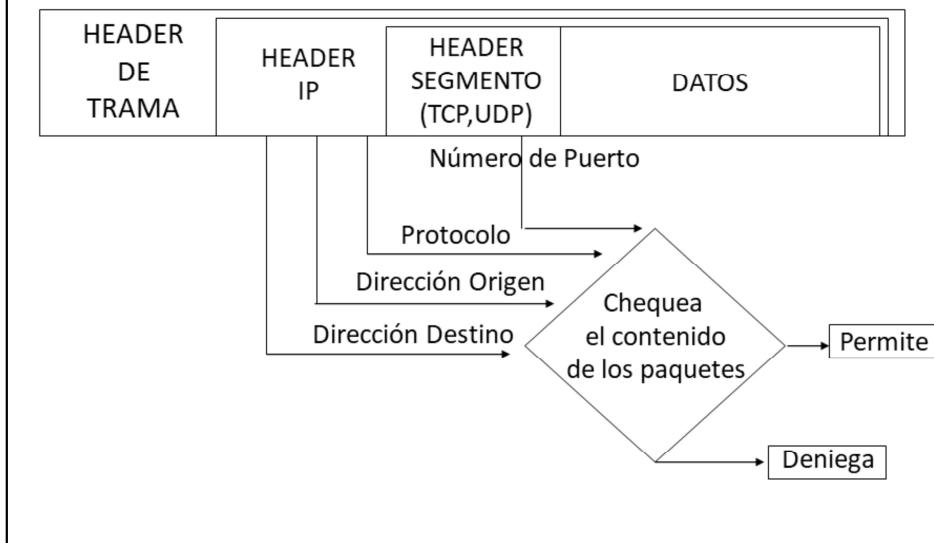
```
Router(config)# access-list 2 permit host 192.168.1.5
Router(config)# interface ethernet 1
Router(config-if)# ip access-group 2 out
```

Ej. 2: Denegar el acceso a la red 192.168.1.0/24 al host 192.168.2.20 y 192.168.2.3

```
Router(config)# access-list 8 deny host 192.168.2.20
Router(config)# access-list 8 deny host 192.168.2.3
Router(config)# access-list 8 permit any
Router(config)# interface ethernet 0
Router(config-if)# ip access-group 8 out
```

Nota: al final de cada lista hay implícito un **access-list n deny any**

## Análisis de paquetes con Listas de Acceso Extendidas



## Listas de Acceso Extendidas

```
R1(config)# access-list access-list-number {permit | deny} protocol
          source source-wildcard [operator port] destination
          destination-wildcard [operator port] [established] [log]
```

```
R1(config-if)# ip access-group access-list-number {in | out}
```

Significado:

- Access-list: Aquí se ingresa un número entre 100 y 199 que indica que la lista de acceso es extendida.
- Permit | Deny: Esta es la acción a tomar en caso de coincidencia (bloquea o permite tráfico)

## Listas de Acceso Extendidas

- Protocol: Indica el numero de protocolo <0-255> (Ej IP, EIGRP, OSPF, ICMP, TCP, UDP, etc)
- Source y Destination: Indican dirección IP origen y destino
- Source-wildcard y destination-wildcard: Indican rango de chequeo de la dirección origen y destino ("0" Chequea y "1" No Chequea)
- Operator port: Operación sobre puertos (Ej: "lt" menor que, "gt" mayor que, "eq" igual que, "nq" distinto a, "range" rango de puertos, etc)
- Log: Registra en el log las coincidencia de la lista de acceso

## Listas de Acceso Nombradas

```
R1(config)# ip access-list {standard | extended} name
R1(config-xxx-nacl)# {permit | deny} protocol source source-wildcard
                                [operator port] destination destination-wildcard
                                [operator port] [established] [log]
R1(config-if)# ip access-group name {in | out}
```

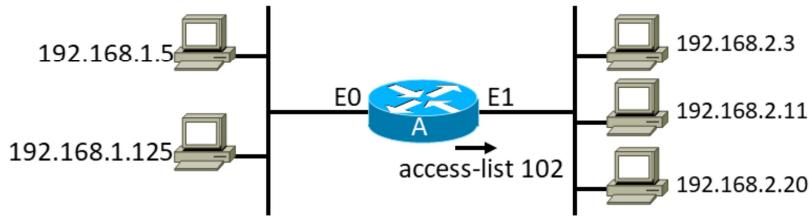
Significado:

- Las líneas de “permit” y de “deny” se ingresan como subcomandos del comando “ip access-list”, en el prompt se indica si la lista es estándar o extendida xxx = std o xxx = ext
- Permit | Deny: Esta es la acción a tomar en caso de coincidencia (bloquea o permite tráfico)

## Listas de Acceso Nombradas

- Protocol: Indica el numero de protocolo <0-255> (Ej IP, EIGRP, OSPF, ICMP, TCP, UDP, etc)
- Source y Destination: Indican dirección IP origen y destino
- Source-wildcard y destination-wildcard: Indican rango de chequeo de la dirección origen y destino (“0” Chequea y “1” No Chequea)
- Operator port: Operación sobre puertos (Ej: “lt” menor que, “gt” mayor que, “eq” igual que, “nq” distinto a, “range” rango de puertos, etc)
- Log: Registra en el log las coincidencia de la lista de acceso

## Ejemplo de Listas de Acceso Extendidas



Ej: Permitir que los host de la red 192.168.1.0/24 puedan acceder al puerto 53 (UDP) de los host de la red 192.168.2.0/24 y solo puedan acceder al puerto 23 (TCP) del host 192.168.2.3

```
Router(config)# access-list 102 permit udp 192.168.1.0 0.0.0.255 192.168.2.0  
0.0.0.255 eq 53
```

```
Router(config)# access-list 102 permit tcp 192.168.1.0 0.0.0.255 host 192.168.2.3  
eq 23
```

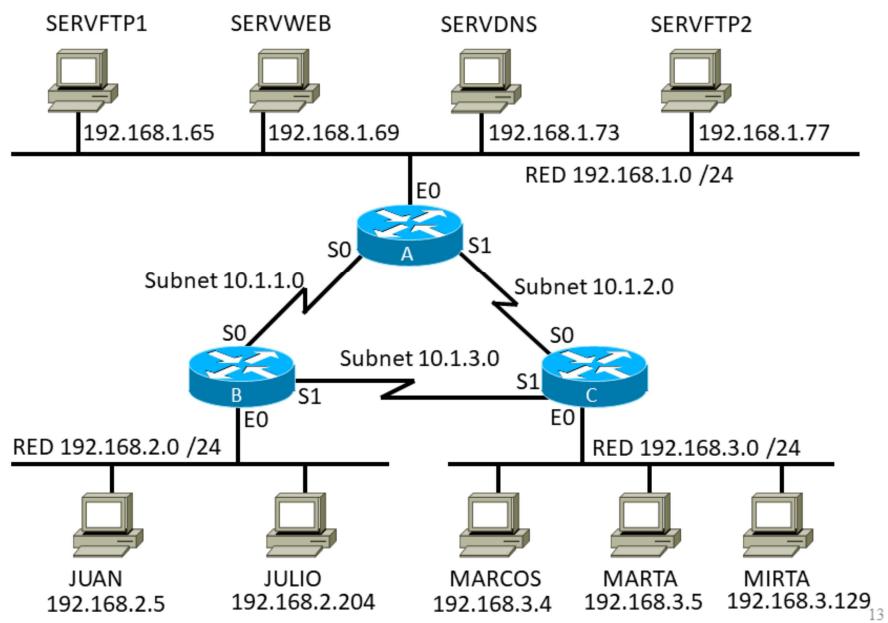
```
Router(config)# interface ethernet 1
```

```
Router(config-if)# ip access-group 102 out
```

Nota: al final de cada lista hay implícito un **access-list n deny any**

12

## Topología de prueba



## ¿Qué hacen estas ACLs?

**Router A:**

```
interface ethernet 0
ip access-group 101 out
access-list 101 permit tcp 192.168.3.4 0.0.0.1 192.168.1.69 0.0.0.0 eq 80
access-list 101 permit tcp host 192.168.2.204 host 192.168.1.69 eq 80
access-list 101 permit udp 192.168.2.128 0.0.0.127 host 192.168.1.73 eq 53
```

**Router B:**

```
interface ethernet 0
ip access-group 102 in
access-list 102 deny ip any 192.168.3.0 0.0.0.255
access-list 102 permit ip any any
```

**Router C:**

```
interface ethernet 0
ip access-group 103 in
access-list 103 permit ip any 192.168.1.64 0.0.0.15
access-list 103 permit tcp host 192.168.3.5 host 192.168.1.77 lt 1023
```

## Resumen de potencialidad (ACL extendidas)

Tipo de Lista de Acceso	Qué puede ser combinado?
IP Extendida	Dirección IP Origen
	Porciones de recurso dirección IP, usando una máscara wildcard
	Dirección IP Destino
	Porciones de dirección IP destino, usando una máscara wildcard
	Tipo protocolo (TCP, UDP, ICMP, IGMP, y otros)
	Puerto origen
	Puerto destino
	Combinaciones establecidas para todos los flujos TCP excepto el primero
	IP TOS
	IP precedencia

## Ejemplo de lista de Acceso y su significado

```
access-list 123 permit udp host 192.168.5.5 eq 85 192.156.24.5 eq www  
access-list 123 deny tcp any host 10.1.2.101 eq 23  
access-list 123 deny udp 192.168.2.0 0.0.0.255 host 1.2.3.4 eq dns  
access-list 123 permit udp 192.168.6.20 0.0.255.255 lt 1023 3.0.0.0 0.255.255.255  
access-list 101 deny tcp 1.0.0.0 0.255.255.255 20.21.2.3 0.0.0.255 range 20 30  
access-list 101 deny tcp 33.1.2.0 0.0.0.255 8.1.59.3 0.255.0.255 eq telnet
```

## Ejemplo de lista de Acceso y su significado

access-list 123 permit udp host 192.168.5.5 eq 85 192.156.24.5 eq www

Permite paquetes udp con origen 192.168.5.5 y puerto 85 lleguen al host 192.156.24.5 puerto 80

access-list 123 deny tcp any host 10.1.2.101 eq 23

Impide que paquete tcp de cualquier origen realicen un telnet a la 10.1.2.101

access-list 123 deny udp 192.168.2.0 0.0.0.255 host 1.2.3.4 eq dns

Impide que paquetes udp con origen la red 192.168.2 realicen consultas DNS al host 1.2.3.4

access-list 123 permit udp 192.168.6.2 0.0.255.255 lt 1023 3.0.0.0 0.255.255.255

Permite que paquetes udp con orígenes las IP comenzadas con 192.168 y puertos menores a 1023 accedan a cualquier equipo con ips que comiencen en 3

access-list 101 deny tcp 1.0.0.0 0.255.255.255 20.21.2.3 0.0.0.255 range 20 30

Impide que conexiones tcp que comiencen con la dirección origen 1, se conecten a equipos con ip comenzando en 20.21.3. y puertos destino tcp entre 20 y 30

access-list 101 deny tcp 33.1.2.0 0.0.0.255 8.1.59.3 0.255.0.255 eq telnet

Impide que conexiones tcp que comiencen con 33.1.2 puedan realizar un telnet a equipos que en su primer octeto tengan un 8 y en el tercero un 59

17

## Ejercicios

Se tiene la ACL:

```
Access-list 100 permit tcp 10.0.0.0 0.0.0.127 any eq telnet
```

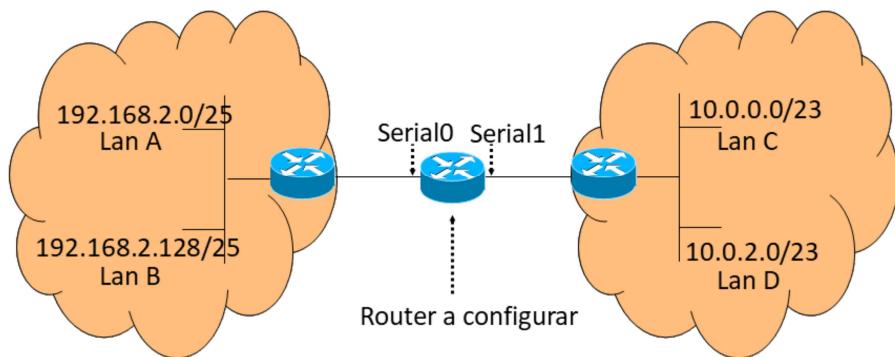
```
Access-list 100 deny udp 10.0.0.64 0.0.0.63 any
```

```
Access-list 100 permit ip 10.0.0.64 0.0.0.63 any
```

En esa situación:

- a. Se descarta todo el tráfico originado en la red 10.0.0.64/26
- b. Se permite todo el tráfico originado en la red 10.0.0.64/26
- c. Se permite todo el tráfico originado en la red 10.0.0.64/26 excepto el udp
- d. Se descarta todo el tráfico originado en la red 10.0.0.64/26 excepto el tcp.

## Ejercicios



- 1- Permitir el tráfico de A a C y a D, pero prohibir el tráfico de B a C y a D.
- 2- Permitir el tráfico de 192.168.2.1 a C y a D y prohibir todo otro tráfico
- 3- Permitir el tráfico de 192.168.2.126 a D y de B a C y a D, y prohibir todo otro tráfico
- 4- Prohibir el spoofing en cada una de las lanes
- 5- Permitir las conexiones TCP establecidas en la LAN B

## Ejercicios

Un router tiene la siguiente ACL:

```
deny udp any 220.20.220.208 0.0.0.7 eq 402
deny tcp any 220.20.220.220 0.0.0.3 gt 30
deny tcp any 220.20.220.192 0.0.0.15 eq 80
deny tcp any 220.20.220.224 0.0.0.31 eq 23
permit ip any any
```

en consecuencia, se puede afirmar que de la subred 220.20.220.192/26  
el rango de direcciones que no será afectado por ningún tipo de  
descartes es:

- a. 220.20.220.200 0.0.0.7
- b. 220.20.220.204 0.0.0.3
- c. 220.20.220.216 0.0.0.3
- d. 220.20.220.216 0.0.0.7

## Mecanismos de control

### Gestión de publicaciones

Empleo de distribute list

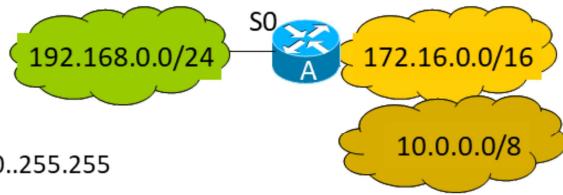
Comandos: Router(config)#router <protocolo> <AS\_number | process\_ID>  
Router(config-router)#distribute-list <ACL number> <in | out> <interface>

Se permiten a lo sumo una DISTRIBUTE-LIST de entrada y una de salida en configuración global por cada proceso de enrutamiento, y una DISTRIBUTE-LIST de entrada y una de salida por cada interfaz dentro un proceso de enrutamiento.

Mediante distributed lists es posible restringir los anuncios de rutas que se publican. Es posible hacerlo en modo global para todo el proceso de ruteo, o de manera de afectar los anuncios entrantes o los anuncios salientes por una interfaz.

En el siguiente ejemplo el router A oculta la red 10.0.0.0/8 al publicar información de ruteo hacia la red 192.168.0.0/24:

```
router rip
network 10.0.0.0
network 172.16.0.0
distribute-list 1 out S0
access-list 1 permit 172.16.0.0 0.0..255.255
```



### Listas de distribución

Las listas de distribución permiten filtrar las publicaciones de anuncios de enrutamiento que se envían. Consiste en un método híbrido que recurre a los automatismos de los protocolos y al mismo tiempo permite agregar manualmente características particulares a la información que se difunde.

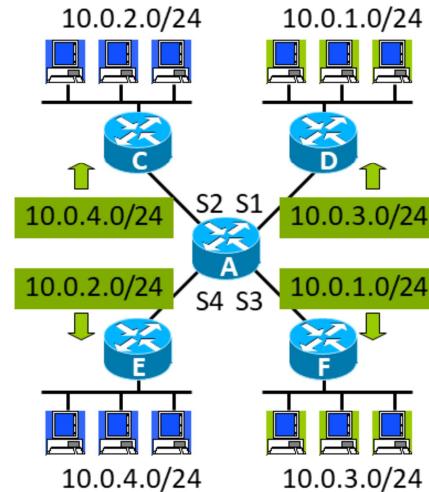
## Mecanismos de control

### Gestión de publicaciones

#### Empleo de distribute list

En el siguiente ejemplo, se emplean en modo saliente en interfaces para obtener el funcionamiento de los PCs en grupos de trabajo (como si fueran VLANs):

```
access-list 1 permit 10.0.3.0 0.0.0.255
access-list 2 permit 10.0.4.0 0.0.0.255
access-list 3 permit 10.0.1.0 0.0.0.255
access-list 4 permit 10.0.2.0 0.0.0.255
router rip
  distribute-list 1 out Serial 1
  distribute-list 2 out Serial 2
  distribute-list 3 out Serial 3
  distribute-list 4 out Serial 4
```



## Mecanismos de control

Gestión de publicaciones

### Prefix list

Los prefix lists son utilizados para seleccionar prefijos a publicar

Sintaxis

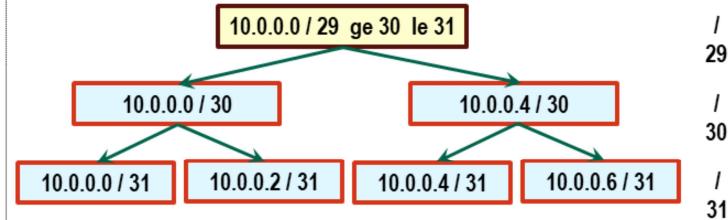
**ip prefix-list list-name [seq sequence-value] deny | permit network/length} [ge ge-value] [le le-value]**

- Si no se utiliza los atributos opcionales ge y le, se selecciona el prefijo específicamente. Con dichos atributos se indica cuales bits estamos analizando a partir del mas significativo.
- Si se utiliza el atributo ge, significa que la mascara debe de ser mayor igual al valor seleccionado (no es valido que el valor sea menor que la longuitud del prefijo)
- Si se utiliza el atributo le, significa que la mascara debe de ser menor o igual al valor seleccionado.
- Un prefix list puede tener varias sentencias, para ver el orden de análisis de los mismos, se toma el orden creciente de los números de secuencias.
- La última linea es implícita y es **deny 0.0.0.0/0 le 32** (análogo a una ACL)

## Mecanismos de control

Gestión de publicaciones

Ejemplo de prefix list



## Mecanismos de control

Gestión de publicaciones

### Prefix list

Ejemplos

**1. ip prefix-list ORT 10 permit 10.0.0.0/8 ge 21**

Dentro del rango de IPs entre las ips 10.0.0.0 y 10.255.255.255, se seleccionaran las redes cuya mascara igual o mas especifica que /21 (quedan excluidas las redes desde la /8 a las /20)

**2. ip prefix-list ORT 10 permit 10.0.1.0/24 le 30**

Dentro del rango de IPs entre las ips 10.0.1.0 y 10.0.1.255, se seleccionaran las redes cuya mascara igual o menos especifica que /30 (quedan excluidas las redes /31 y /32)

**3. ip prefix-list ORT 10 permit 10.1.0.0/16 ge 21 le 29**

Dentro del rango de IPs entre las ips 10.1.0.0 y 10.1.255.255, se seleccionaran las redes cuya mascara igual o mas especifica que /21 y menor o igual especifica que /30 (quedan excluidas las redes /16 a /20 y /30 a /32)

## Mecanismos de control

Gestión de publicaciones

### Prefix list

Ejemplo para filtrar prefijos en BGP

```
ip prefix-list ort_bgp_in seq 5 deny 10.0.0.0/24 ge 25 le 26
ip prefix-list ort_bgp_in seq 10 permit 0.0.0.0/0 le 32
ip prefix-list ort_bgp_out seq 5 permit 0.0.0.0/0
!
router bgp 65000
neighbor 192.168.1.2 remote-as 65100
neighbor 192.168.1.2 prefix-list ort_bgp_in in
neighbor 192.168.1.2 prefix-list ort_bgp_out out
```

Admito todas las publicaciones salvo la de los prefijos con una especificidad entre /25 y /26 contenidos en el 10.0.0.0/24.

Solamente publico la ruta por defecto.

## Mecanismos de control

Gestión de publicaciones

### Route-Maps

Route maps es una función utilizada en Cisco IOS (otros fabricantes y en otros SO de Cisco existen otras similares con otro nombre) que se utiliza para realizar diferentes cosas, la que vamos a utilizar en este curso es la de modificar políticas de ruteo.

Los route-maps son:

- Una secuencia ordenada de declaraciones (statements)
- cada una de ellas tiene un resultado de permitir o denegar
- Cada declaración contiene condiciones de match, en caso de no existir se asume un match implícito
- Si una de las declaraciones se cumple que lo que se establece el match se cumple, se aplica la/s acción/es definida/s y las restantes declaraciones no van a seguir siendo analizadas

### Route maps

Los route maps son herramientas más generales que las listas de distribución. Permiten encaminar el tráfico tanto por destino como también por origen, y en general, por cualquier característica que pueda especificarse mediante un lista de acceso. Además, es posible emplearlos para marcar paquetes con distintos grados de prioridad, y modificarles diversos parámetros.

## Mecanismos de control

Gestión de publicaciones

### Route-Maps

Cada declaración del ROUTE-MAP se identifica con un número de secuencia. Por defecto la primer instrucción del ROUTE-MAP tendrá 10, la segunda 20 y así sucesivamente.

El subcomando MATCH especifica el criterio de verificación que usaremos en el route-map (por ejemplo comunidades BGP, prefix-list, ACL, etc).

El subcomando SET especifica las acciones que tomaremos.

```
Router(config)#route-map <map-tag> [permit | deny] <sequence_number>
Router(config)#match ip address <ACL_number>
Router(config)#set interface <interfaz>
```

### Route maps

Los route maps son herramientas más generales que las listas de distribución. Permiten encaminar el tráfico tanto por destino como también por origen, y en general, por cualquier característica que pueda especificarse mediante un lista de acceso. Además, es posible emplearlos para marcar paquetes con distintos grados de prioridad, y modificarles diversos parámetros.

## Mecanismos de control

Gestión de publicaciones

### Route-Maps

Qué hace el siguiente route map?

```
router bgp 65000
    neighbor 192.168.0.1 remote-as 65100
    neighbor 192.168.0.1 route-map local-policy in
!
route-map local-policy permit 10
    match community ortComm
    set weight 300
route-map local-policy permit 20
    match ip prefix-list address ort_bgp_in
    set local-preference 125
route-map local-policy permit 30
    set as-path prepend 65000
!
ip community standard ortComm permit 6057:6057
!
ip as-path access-list 1 deny _350_
ip as-path access-list 1 permit .*
!
ip prefix-list ort_bgp_in seq 5 deny 10.0.0.0/24 ge 25 le 26
ip prefix-list ort_bgp_in seq 10 permit 0.0.0.0/0 le 32
```