# Subnettingpara alumnos con profesores que no saben explicar

## Introducción

Este manual está desarrollado a partir de material recopilado de internet, bibliografía y con algunas aportaciones propias. Los conceptos, técnicas y explicaciones expuestas a continuación son típicas cuando se trabaja con redes de computadores, sobre todo si se trata de realizar direccionamiento IP. Dicho de otro modo, si quieres saber cómo dar direcciones a los ordenadores de tu red local, tienes dos opciones, o bien lo dejas todo en manos de DHCP con lo que las direcciones se asignan de forma automática, o bien continúas leyendo este manual.

Antes de culpar a tu profesor por no saber explicar o por explicarse como el cul……., es interesante que trates de comprender los conceptos y técnicas del siguiente apartado, ya que sin esos conocimientos que son la base, es imposible poder continuar con garantías de éxito. Probablemente tu profesor te haya explicado todo o parte de lo que a continuación podrás leer, pero en vez de hacerte el enteradillo, trata de leer todo con detenimiento, y comprueba si ya sabes estos conceptos.

Al finalizar la lectura de este manual, si logras entender todo, y eres capaz de hacer los ejercicios, podrás demostrar, que en efecto tú tenías razón y tu profesor explica fatal. No tendrás que hacer autocrítica y todo lo que tu profesor ya ha sembrado en tu cerebro sin que lo sepas no tendrá sentido alguno. Podrás decir a tus amigos, compañeros y profesores que aprendes más con un manual que en las clases de tu profesor.

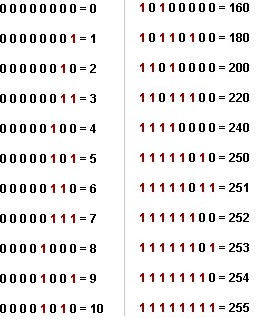
## Conceptos previos

**Convertir valores de binario a decimal**

Las direcciones IP están formadas por 32 bits binarios, pero para que sea más sencillo trabajar con ellas se dividen esos 32 bits en 4 grupos de 8 que se pasan a decimal. Por lo tanto es importante saber convertir números binarios de 8 bits a decimal y viceversa. Probablemente te hayan explicado cómo hacer estas conversiones con un método general, pero como aquí sólo se necesitan 8bits, vamos a ver un método simplificado, que es suficiente para entender y realizar el direccionamiento.

La combinación de 8 bits permite un total de 256 (28=2x2x2x2x2x2x2x2=256) combinaciones posibles que cubre todo el rango de numeración decimal desde el 0 (00000000) hasta el 255 (11111111). A continuación puedes ver algunos ejemplos

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |



101101000=128+32+16+4=180

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

11111110=128+64+32+16+8+2=250

Al pasar de binario a decimal, si te fijas un poco, verás que cada posición determina un valor a sumar al resultado total. Así por ejemplo si tenemos 11111110 lo que hacemos es rellenar la tabla con esos 8 bits y sumar los valores que estén a uno, como en el primer ejemplo. Como siempre vamos a trabajar con número de 8bits, sólo hay que saberse los valores a sumar (128,64,32,16,8,4,2,1) que provienen de calcular 27=128, 26=64,….., 21 =2 y 20=1. *Practica con dos o tres casos para comprobar que has entendido bien el procedimiento*.

**Convertir valores de decimal a binario**

Los números decimales con los que vamos a trabajar siempre van a tener 8bits en binario y van a estar entre 255 y 0. La forma más sencilla de pasar a binario es seguir el proceso contrario al visto en el apartado anterior, es decir, restando. Lo vamos a explicar un ejemplo, ya verás que sencillo es, no hace falta ni calculadora, ¿o es que no sabes restar? Jejeje.

Para pasar a binario 210 repetimos estos pasos hasta llegar exactamente a 0:

* Buscamos el mayor múltiplo de dos que podemos restar sin pasarnos 🡪 128
* Hacemos la resta. 210-128= 82
* Buscamos el mayor múltiplo de 2 que podemos restar a 82 sin pasarnos 🡪 64
* Hacemos la resta 82-64=18
* Buscamos el mayor múltiplo de 2 que podemos restar a 18 sin pasarnos 🡪 16
* Hacemos la resta 18-16=2
* Buscamos el mayor múltiplo de 2 que podemos restar a 2 sin pasarnos 🡪 2
* Hacemos la resta 2-2 = 0 **FIN**

Para obtener el número binario que codifica el 210 con 8 dígitos binarios, sólo tendremos que volver a usar nuestra maravillosa tabla, poniendo a 1 los valores que hemos restado, para el ejemplo, 128, 64,16 y 2. El resto se dejan a 0.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |

210= 11010010

**Dirección IP**

La dirección IP es un número de 32 bits o dígitos binarios, que se agrupan en 4 grupos de 8 bits cada uno. Sirve para identificar de forma única a un equipo dentro de una red. Por ejemplo 192.168.10.73

**Máscara de red**

La máscara de red, es un número de 32 bits o 32 dígitos binarios, que igual que las direcciones IP se agrupan en 4 grupos de 8 bits cada uno. La máscara de red tiene la peculiaridad de que está formada por 1s en la parte izquierda y 0s en la parte derecha y nunca se pueden mezclar entre sí. Dependiendo del número de 1s y 0s tendremos una máscara u otra, eso sí cuantos más 1s menos 0 y al revés, ya que siempre tenemos 32bits. La máscara de red sirve para dividir la dirección IP en dos zonas o partes: la parte de red (1s) y la parte de host (0s). Como se llevan muy mal entre sí, donde terminan los 1s y empiezan 0s a menudo se suele poner una barra que los separa. Ejemplo de una máscara válida y otra invalida:

11111111.11111111.00000000.00000000 🡪 255.255.0.0

11111111.11111111.11101010.00000000 ERRRRRORRRR 🡪 mezclados 0s y 1s

## Clases de direcciones IP

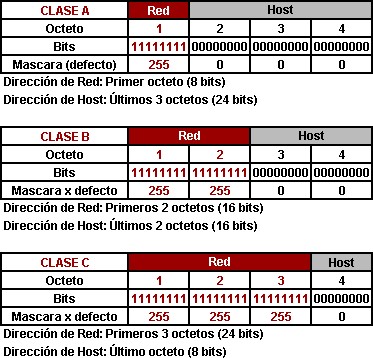
## Dirección IP Clase A, B y C

Recuerda que las direcciones IP y las máscaras de red están compuestas por 32 bits divididos en 4 octetos (8 bits cada uno), es decir, 4 números decimales y cada número está entre 0 y 255. Si te ponen 120.240.250.260, eso no es una IP válida porque el último número es 260 y es mayor que 255. El valor del primer número de una dirección IP va a determinar su clase(A, B, C, D, E) y cada clase va a tener una máscara por defecto, un rango de IPs, cantidad de redes y de hosts por red.Es importante conocer donde empieza y donde acaba cada clase, para poder identificar a la clase que pertenece una IP y obtener así su máscara de red por defecto.

El primer número de la IP identifica su clase: **A(0-127), B(128-191), C(192-223).**



Cada Clase tiene una máscara de red por defecto, la Clase A 255.0.0.0, la Clase B 255.255.0.0 y la Clase C 255.255.255.0. Al direccionamiento que utiliza la máscara de red por defecto, se le denomina “direccionamiento con clase” (classfulladdressing). En otro caso se dice que es un direccionamiento sin clase.

Siempre que se hace subnetting es a partir de una dirección de red Clase A, B, o C y ésta se adapta según las subredes y hosts por subred que se necesiten.

Se suele usar una notación simplificada para expresar las máscaras de red que consiste en poner una / seguido del número de 1s que hay en la máscara, el resto hasta 32 bits serán 0s.

Por ejemplo /8, /16, /24 son las máscaras por defecto de las Clases A, B y C.

La cantidad de bits "0" en la porción de host de la máscara, indican que parte de la dirección de IP se usa para asignar direcciones de host. Por ejemplo si tenemos una dirección 172.16.0.0/24, como la IP es de clase B, su máscara por defecto es /16, y por tanto 24-16=8, en la máscara que nos dan hay 8bits que se usan para crear las diferentes subredes.

**Dirección de red (poner 0s)**

Es una dirección especial que sirve para hacer referencia a toda la red. Se obtiene a partir una dirección IP y una máscara de red, pasando ambas direcciones a binario, y poniendo a 0 en la IP todos los bits que estén a cero en la máscara

10.10.10.10/8 🡪 IP en binario 00001010.00001010.00001010.00001010

255.0.0.0 = /8 🡪 Máscara en binario 11111111.00000000.00000000.00000000

La dirección de red será 10.0.0.0 🡪 00001010.00000000.00000000.00000000

*Da igual el valor que tuvieran en la IP porque en la máscara está a 0, por eso se pasan a 0s*

*Se queda igual que en la IP porque en la máscara está a 1*

**Dirección de broadcast (poner 1s)**

Es una dirección especial que sirve para mandar un mensaje a todos los ordenadores o host que pertenezcan a esa misma red. Se obtiene de una forma muy parecida a la dirección de red, pero en lugar de poner todo a 0s se pone todo a 1s.

10.10.10.10/8 🡪 IP en binario 00001010.00001010.00001010.00001010

255.0.0.0 = /8 🡪 Máscara en binario 11111111.00000000.00000000.00000000

Dirección de broadcast será 🡪 00001010.11111111.11111111.11111111

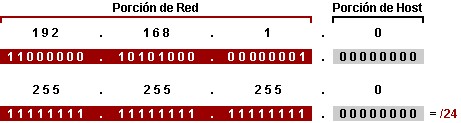
*Da igual el valor que tuvieran en la IP porque en la máscara está 0, por eso se pasan a 1s*

*Se queda igual que en la IP porque en la máscara está a 1*

10.255.255.255

**Ejemplos:**

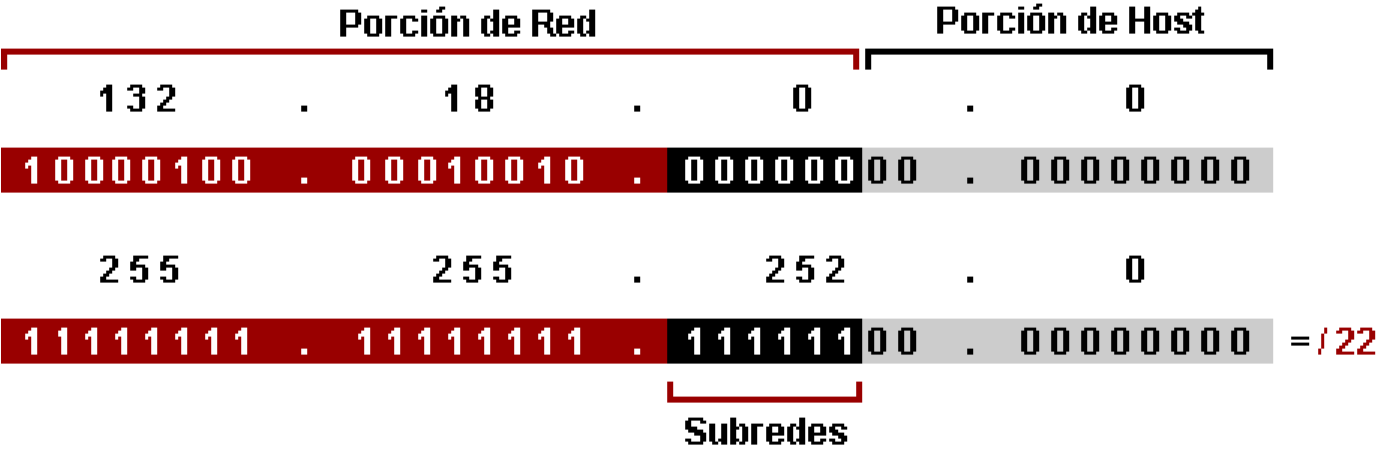
Si tenemos la dirección IP Clase C 192.168.1.0/24 y la pasamos a binario, los primeros 3 octetos, que coinciden con los bits “1” de la máscara de red, es la dirección de red, que va a ser común a todos los hosts que sean asignados en el último octeto. Con este mismo criterio, si tenemos una dirección Clase B, los 2 primeros octetos son la dirección de red que va a ser común a todos los hosts que sean asignados en los últimos 2 octetos, y si tenemos una dirección Clase A, el 1 octeto es la dirección de red que va a ser común a todos los hosts que sean asignados en los últimos 3 octetos.



## SUBNETTING

La función del Subnetting es dividir una red IP física en subredes lógicas (redes más pequeñas) aunque todas pertenezcan a la misma red física. Esta división permite una mejor administración, control del tráfico, seguridad y también mejora el rendimiento de la red al reducir el tráfico de broadcast.

Por ejemplo 132.18.0.0/22. En este caso el 3º octeto está divido en 2, una parte en la porción de red y otra en la de host, la parte de la dirección IP que corresponde a la porción de red, que tienen en la máscara de red los bits “1”, se va a ir modificando según se vayan asignando las subredes y solo va a ser común a los host que son parte de esa subred. Los 2 bits “0” del 3º octeto en la porción de host y todo el último octeto de la dirección IP, van a ser utilizados para asignar direcciones de host.



**Cantidad de Subredes es igual a: 2N**, donde "N" es el número de bits "robados" a la porción de Host.

**Cantidad de Hosts por Subred es igual a: 2M -2**, donde "M" es el número de bits disponible en la porción de host y "-2" es debido a que toda subred debe tener su propia dirección de red y su propia dirección de broadcast.

**Aclaración:** la fórmula para obtener la cantidad de subredes era 2N -2, donde "N" es el número de bits "robados" a la porción de host y "-2" porque la primer subred (subnetzero) y la última subred (subnetbroadcast) no eran utilizables ya que contenían la dirección de la red y broadcast respectivamente. El RFC 1878 afirma que esta es una práctica obsoleta, es mejor usar la fórmula 2N, que permite utilizar tanto la subred zero como la subnetbroadcast para ser asignadas.

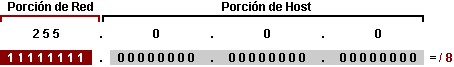
# Subnetting Manual de una Red Clase A para hacer subredes de igual tamaño

Dada la dirección IP Clase A 10.0.0.0/8 para una red, se nos pide que mediante subnetting obtengamos 7 subredes. Este es un ejemplo típico que se nos puede pedir, aunque remotamente nos topemos en la vida real.

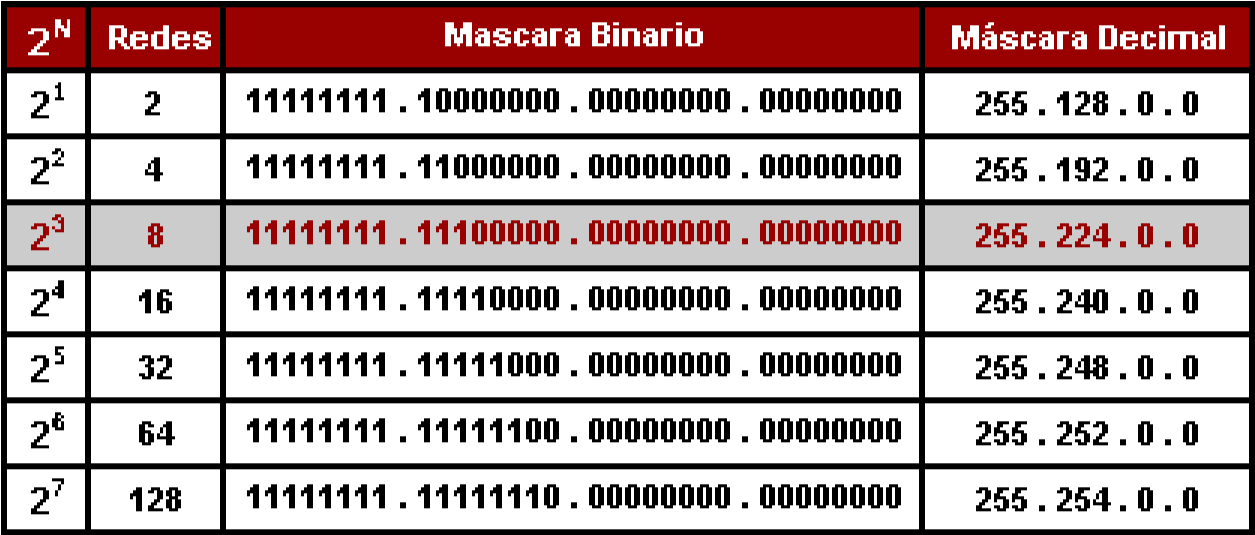
Lo vamos a realizar en 2 pasos:

## Adaptar la Máscara de Red por Defecto a Nuestras Subredes (1)

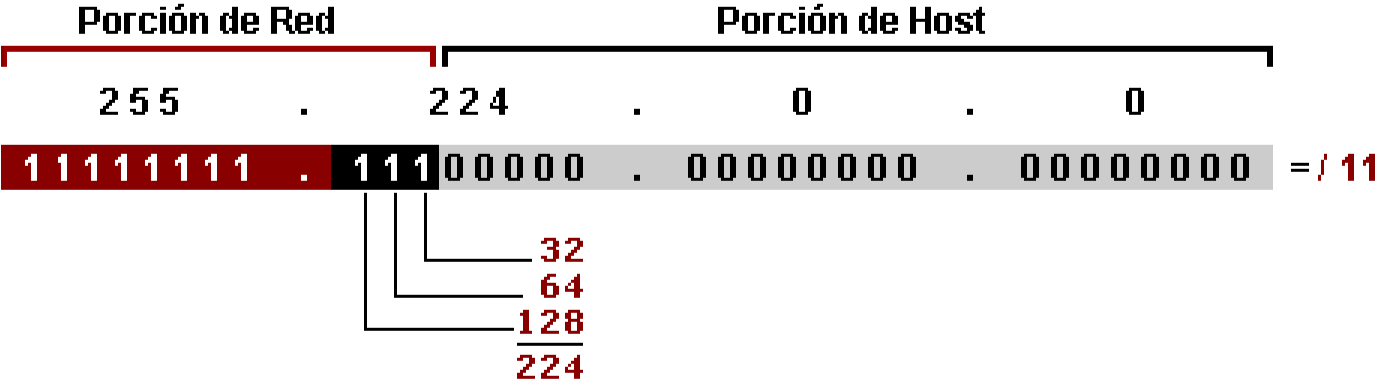
La máscara por defecto para la red 10.0.0.0 es:



Mediante la fórmula **2N**, donde **N** es la cantidad de bits que tenemos que robarle a la porción de host, adaptamos la máscara de red por defecto a la subred.En este caso particular **2N = 7** (o mayor) ya que nos pidieron que hagamos 7 subredes.

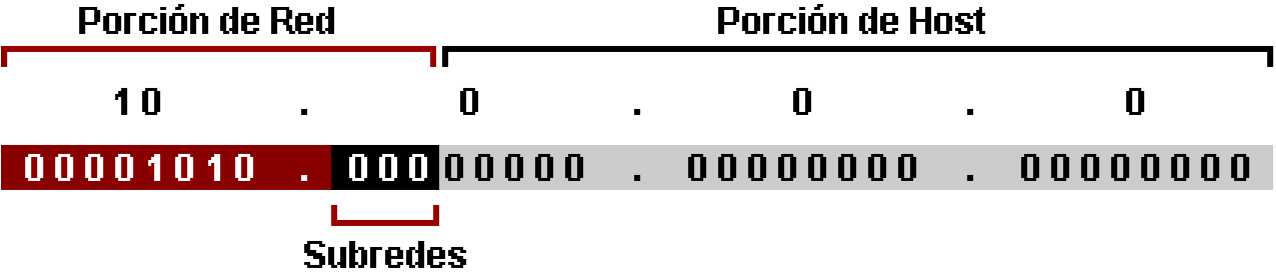


Una vez hecho el cálculo nos da que debemos robar 3 bits a la porción de host para hacer 7 subredes o más y que el total de subredes útiles va a ser de 8, es decir que va a quedar 1 para uso futuro. Tomando la máscara Clase A por defecto, a la parte de red le agregamos los **3 bits** que le robamos a la porción de host reemplazándolos por "**1**" y así obtenemos 255.224.0.0 que es la máscara de subred que vamos a utilizar para todas nuestras subredes y hosts.



**Obtener Rango de Subredes (2)**

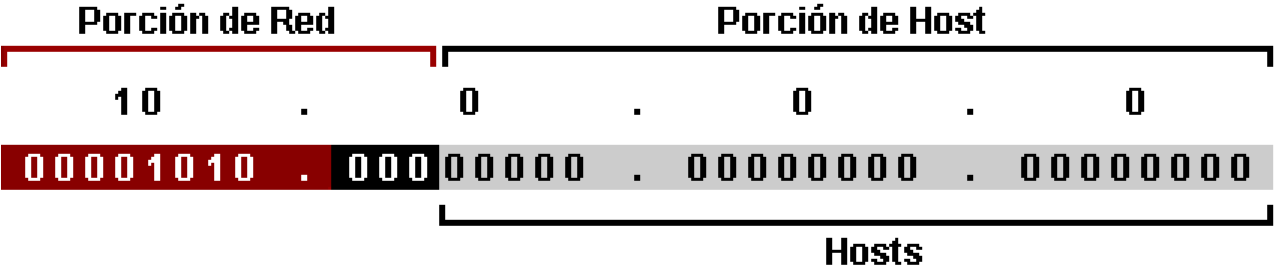
Para obtener las subredes se trabaja únicamente con la dirección IP de la red, en este caso 10.0.0.0. Para esto vamos a modificar el mismo octeto de bits (el segundo) que modificamos anteriormente en la máscara de red pero esta vez en la dirección IP.



Para obtener el rango hay varias formas, la que me parece más sencilla a mí es la de restarle a 256 el número de la máscara de red adaptada. En este caso sería: **256-224=32**, entonces **32** va a ser el rango entre cada subred.



Si queremos calcular cuántos hosts vamos a obtener por subred debemos aplicar la fórmula **2M - 2**, donde **M** es el número de bits "**0**" disponible en la porción de host de la dirección IP de la red y **- 2** es debido a que toda subred debe tener su propia dirección de red y su propia dirección de broadcast.



En este caso particular sería:

**221 - 2 = 2.097.150** hosts utilizables por subred.

Algoritmo resumen subnetting en subredes de igual tamaño dando IP, máscara y número de subredes que se necesita crear:

1. Poner la IP y poner en binario los octetos que están a 0 en la máscara
2. Marcar con una raya en la IP donde terminan los 1s y empiezan los 0s
3. Obtenemos la nueva máscara de red

* Si preguntan por subredes usa la fórmula **2N ≥ número de subredes** y N son los bits que hay que pasar de la parte de hosts a la nueva máscara, márcalo con una línea discontinua, contando M bits de izquierda a derecha a partir de la línea continua.
* Si preguntan por hosts en cada subred usa **2M – 2 ≥ número de hosts** donde M son los bits que se usarán para dar direcciones a los distintos hosts de cada subred. Debemos contar M bits de derecha a izquierda y marcar con una línea discontinua. Los bits que faltan para llegar a la línea continua del apartado 2 son los bits para crear subredes.

1. Dar valor a los bits de subred para ir creando las diferentes subredes. Recuerda que /X + N + M = 32 bits. Para cada subred hacer los siguientes pasos:
   1. Obtener la dirección de subred (a partir de la raya discontinua todo 0s)
   2. Obtener la dirección de broadcast (a partir de la raya discontinua todo 1s)
   3. Obtener la 1ª IP valida (suma uno a la dirección de red)
   4. Obtener la última IP (resta uno a la dirección de broadcast)
   5. Obtener el número de hosts de la subred (**2M – 2**)

**Ejemplo:** crear cuatro subredes a partir de la dirección 192.168.1.0/24

Paso 1: 192.168 .1 . 0 0 0 0 0 0 0 0

Paso 2: 255.255.255 . 0 0 0 0 0 0 0 0 En total tenemos 24 unos y 8 ceros ya que 24+8 = 32. Ahora marcamos en la IP donde terminan los 1s de la máscara.

192.168 .1 . 0 0 0 0 0 0 0 0

Paso 3:**2N ≥ 4 subredes 🡪22 =4 ≥ 4 🡪 N=2** 192.168 .1 . 0 0 0 0 0 0 0 0

Paso 4: dar valores distintos a los bits de subred, para dos bits 00, 01, 10, 11

Subred1 192.168 .1 . 0 0 0 0 0 0 0 0 Broadcast1 192.168 .1 . 0 0 1 1 1 1 1 1

Subred2 192.168 .1 . 0 1 0 0 0 0 0 0 Broadcast2 192.168 .1 . 0 1 1 1 1 1 1 1

Subred3192.168 .1 . 1 0 0 0 0 0 0 0 Broadcast3 192.168 .1 .1 0 1 1 1 1 1 1

Subred4192.168 .1 . 1 1 0 0 0 0 0 0 Broascast4 192.168 .1 . 1 1 1 1 1 1 1 1

Para cada subred tenemos:

* Dirección de subred1: 192.168.1.0 Primera ip: 192.168.1.1
* Dirección de broadcast1:192.168.1.63 Última ip: 192.168.1.62
* Dirección de subred2: 192.168.1.64 Primera ip: 192.168.1.65
* Dirección de broadcast2:192.168.1.127 Última ip: 192.168.1.126
* Dirección de subred3: 192.168.1.128 Primera ip: 192.168.1.129
* Dirección de broadcast3:192.168.1.191 Última ip: 192.168.1.190
* Dirección de subred4: 192.168.1.192 Primera ip: 192.168.1.193
* Dirección de broadcast4:192.168.1.255 Última ip: 192.168.1.254

# Subnetting Manual VLSM

La forma en que se dividen las redes con subnetting, es una forma demasiado rígida y se desperdician muchas direcciones al no poder ajustar el tamaño de las subredes de una forma más ajustada a las necesidades reales. A continuación se recoge un algoritmo para aplicar subnetting con tamaño de máscara variable y que permite un mejor ajuste a la hora de hacer el subnetting.

Normalmente VLSM se aplica sobre un diagrama para una empresa o situación real, donde aparecen routers, switches, impresoras, equipos finales, etc. Vamos a trabajar sobre esa premisa.

Lo normal es que cada interfaz de un router defina un dominio de broadcast y da igual que tenga conectados otros switches o routers. Es importante saber cuántas direcciones ip son necesarias para cada uno de esos dominios y para ello hay que contar todos los hosts, impresoras, teléfonos, ….. y hay que sumar una dirección IP que hay que asignar a la interfaz del router y otras dos para la red y el broadcast.

Cada interfaz de un router define un dominio de broadcast diferente. Si dos o más routers están conectados entre sí o a un switch, todo lo que esté conectado pertenecerá al mismo dominio de broadcast

Algoritmo VLSM:

Paso1: redondea cada dominio de broadcast, para cada uno se creará una subred

Paso 2: cuenta el número de IPs por subred y anotaló junto a cada subred

Paso 3: ordena de mayor a menor las subredes por tamaño

Paso 4: Para cada subred, de mayor a menor tamaño hacer:

4.1 Partir de la dirección libre que quede, coger la menor subred que no se solape con ninguna de las anteriores.

4.2 Hacer el subnetting para esa red aplicando el algoritmo normal

4.3 Una vez obtenida la dirección de red asignar direcciones válidas dentro de la subred a todos los routers y equipos que necesiten dirección IP.