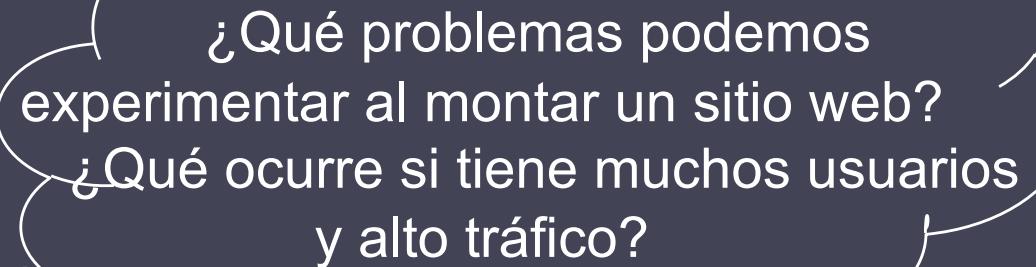


# TEMA 1

## Introducción

SWAP



¿Qué problemas podemos experimentar al montar un sitio web?  
¿Qué ocurre si tiene muchos usuarios y alto tráfico?



José Manuel Soto Hidalgo  
Dpto. Arquitectura y Tecnología de Computadores  
Universidad de Granada

jmsoto@ugr.es

# Índice



[ Introducción ]

Conceptos básicos

Un sitio web mal planificado

Un sitio web de éxito

# 1. Introducción

Supongamos que nos encargan el despliegue de un servidor web para una empresa...

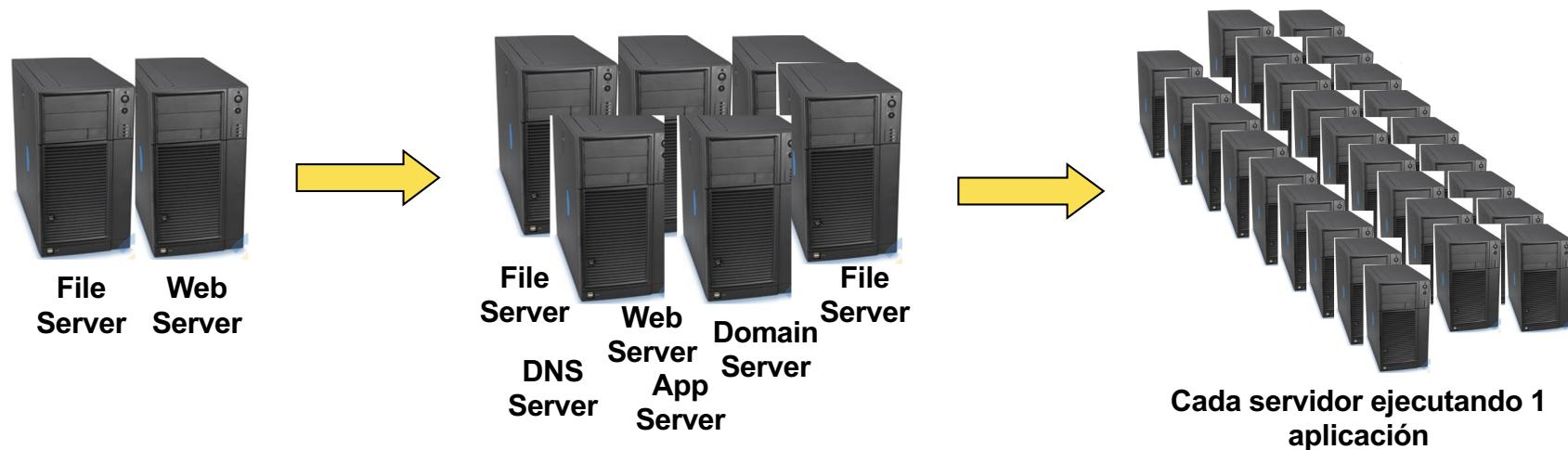
Trabajo inicial de configuración

VS.

Trabajo de mantenimiento

# Introducción

- **Un poco de historia (Servidores en los 90)**
  - Los servidores Intel/AMD (“x86” servers) son baratos
  - Cada servidor tiene un sistema operativo diferente
  - **UN** sistema operativo y **UNA** aplicación en cada servidor
  - 2 servidores pasan a ser 6, quizás 50 o más servidores!
  - El **espacio, la refrigeración y la alimentación** se convierten en un problema....



# Introducción

- **Un poco de historia (Servidores del 2000 en adelante)**
  - Los fabricantes al “rescate”
  - Se centran en crear servidores más pequeños
  - Reducen el tamaño de los chassis (6-20 servers por rack)
  - Crean los servidores “Blade” (30-60 servidores por rack)
  - Problema de espacio solucionado... más o menos
  - La alimentación y la refrigeración siguen siendo un problema



HP “Blade” Servers



Dell “Rack” Servers

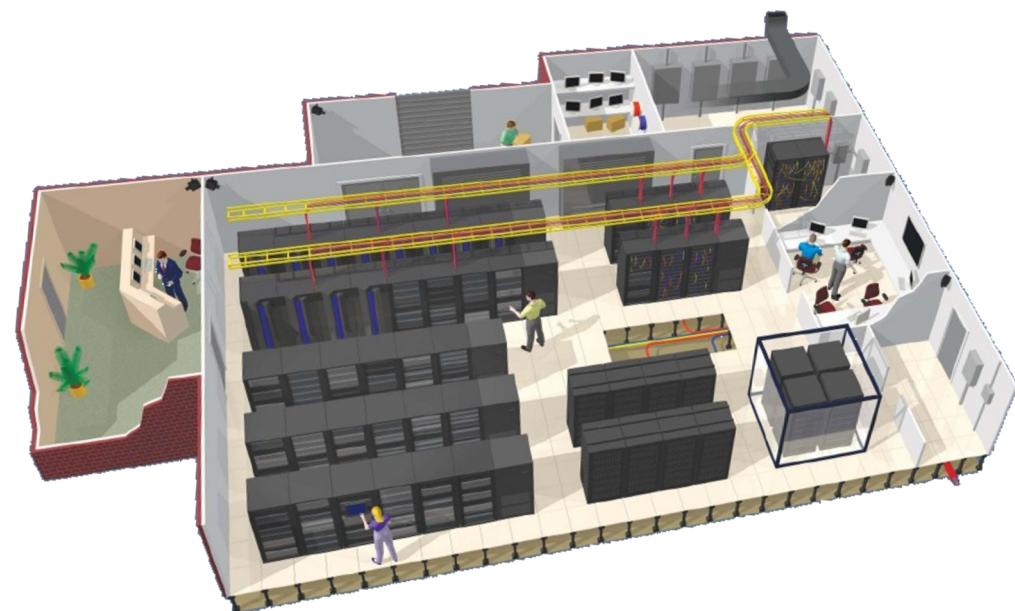


Consumo  
eléctrico



# Introducción

- **Centros de Procesamiento de Datos (CPD)**
  - Un Centro de Procesamiento de Datos (CPD) es una ubicación donde se concentran los recursos necesarios para el procesamiento de la información de una organización.
  - Normativas TIA 942, ISO 27001, EN 1047-2, ISO14644, ASHRAE, Uptime Institute, 24x7
  - Características:
    - Armarios
    - Infraestructura interior
    - Sistema de alimentación
    - Ventilación
    - Cableado



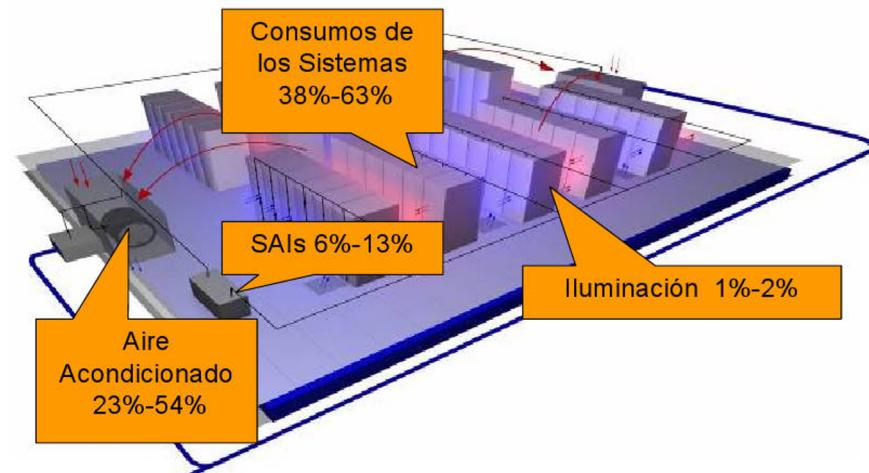
# Introducción

- **Centros de Procesamiento de Datos (CPD)**

## Consumo (**Green IT**)

- Prácticamente el consumo de un CPD proviene del consumo del **equipamiento del CPD** y de la **climatización**.
  - 38 - 63% corresponde a los sistemas (servidores y equipos)
  - 23 - 54% corresponde a la refrigeración
  - 6 - 13% a los sistemas complementarios (sistemas electrógenos, sistemas de alimentación ininterrumpida...)
  - 1 - 2% a iluminación

“Tecnologías Verdes son aquellas que contribuyen a la reducción en el consumo de energía o emisión de dióxido de carbono”



# Introducción

- **Consumo (Green IT).** Reducir consumo equipamiento
  - Optimización del consumo energético generado por los propios servidores y sistemas instalados.
    - Técnicas de virtualización
      - Virtualización
      - Consolidación de servidores
    - Técnicas de utilización dinámica de recursos
      - Herramientas de Red
      - Tecnología Grid
      - El modelo Cloud
        - Modelo de servicios (SaaS, PaaS, IaaS)

# Introducción. Virtualización

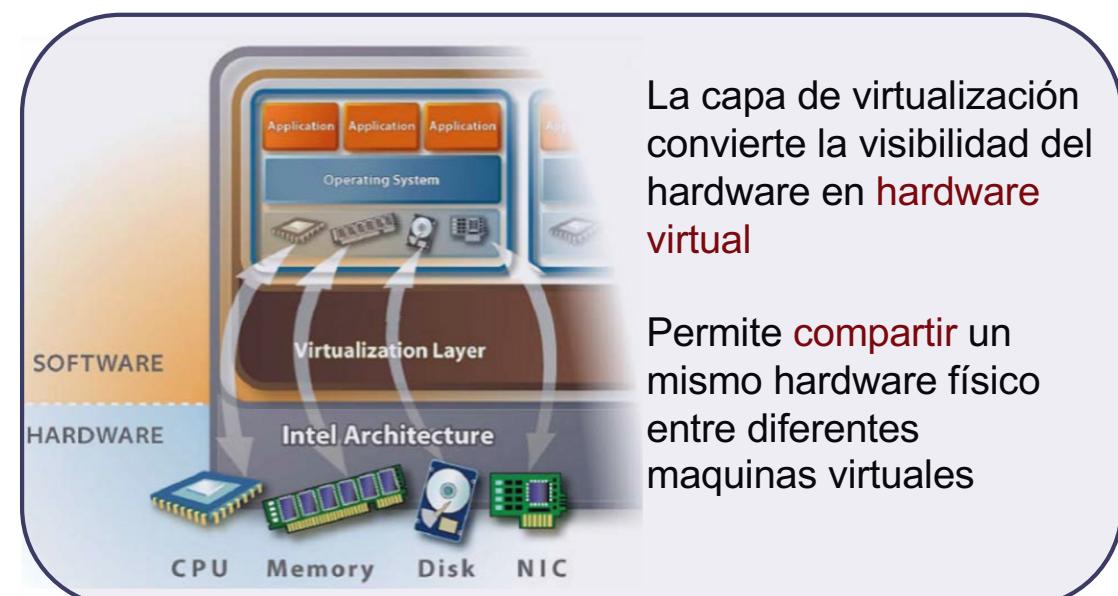
## ■ Consumo (Green IT). Reducir consumo equipamiento

- Creación -a través de software- de una versión virtual de algún recurso tecnológico (plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento, etc.).

No Virtualizado

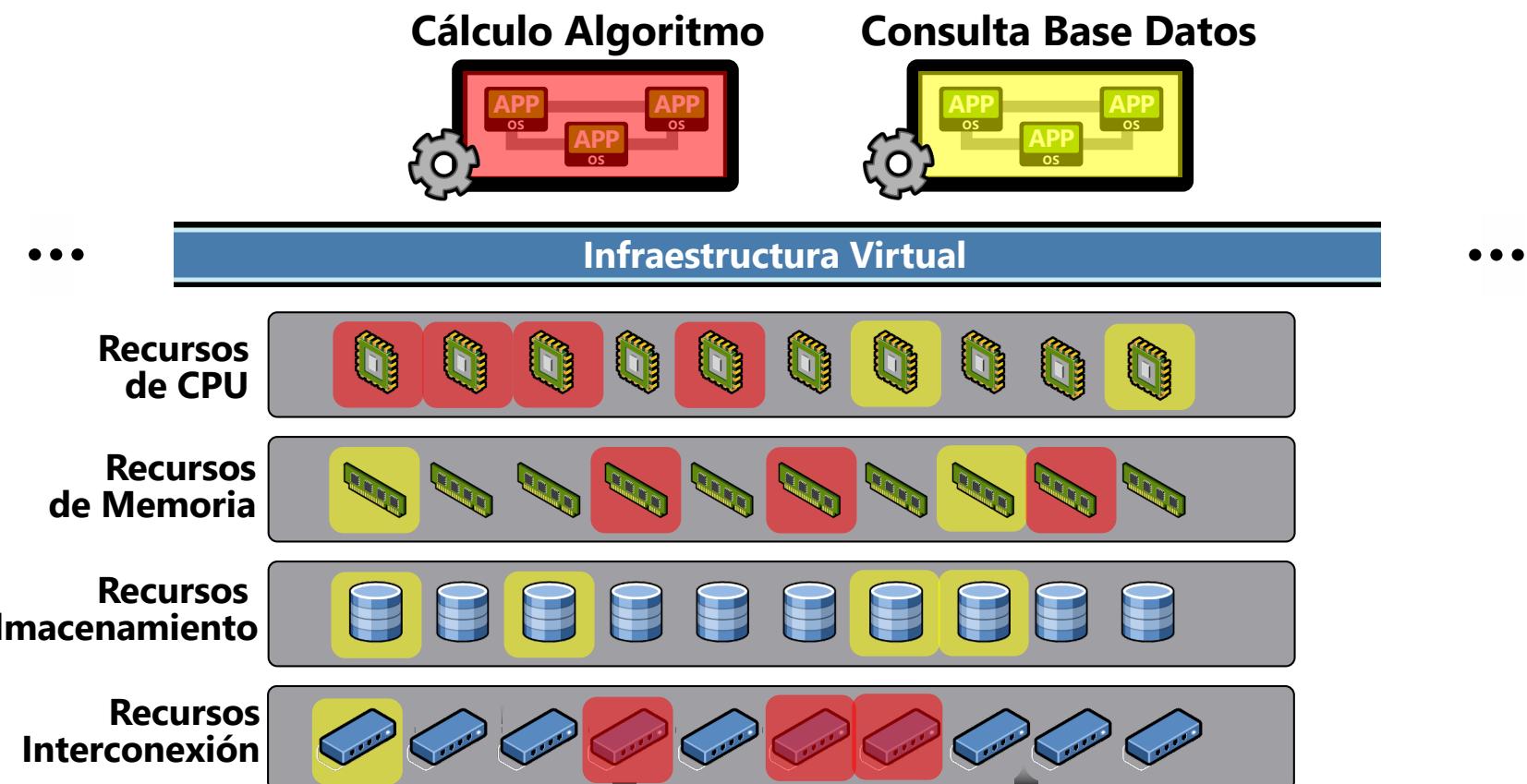


Virtualizado



# Introducción. Virtualización

- **Consumo (Green IT).** Reducir consumo equipamiento
  - Ejemplo.



# Introducción. Virtualización

- **Consumo (Green IT).** Reducir consumo equipamiento
  - **Tipos**
    - **Virtualización de plataformas:** consiste en separar un sistema operativo de los recursos de la plataforma subyacente.
      - Virtualización nativa o completa
      - Para-virtualización
      - Virtualización a nivel de Sistema Operativo
      - Virtualización de aplicaciones
    - **Virtualización de recursos:** consiste en la virtualización de recursos específicos del sistema, como la memoria virtual, el almacenamiento virtual.
      - Memoria Virtual
      - Almacenamiento NAS
      - Almacenamiento SAN

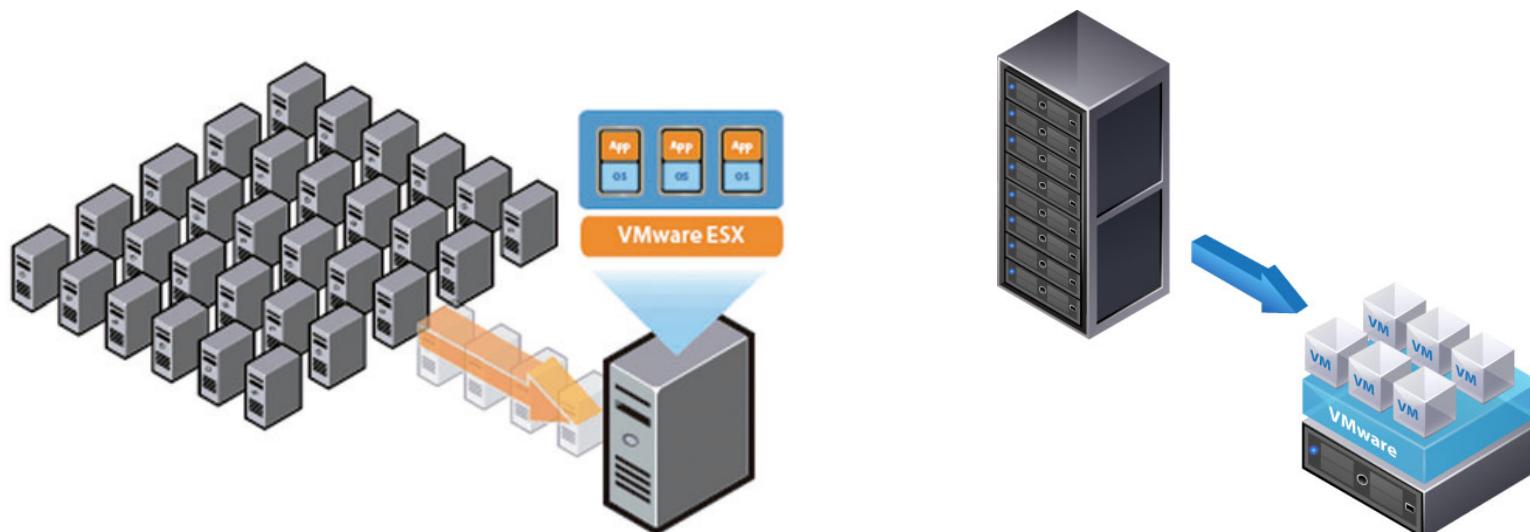
# Introducción. Virtualización

- **Consumo (Green IT).** Reducir consumo equipamiento
  - Software de Virtualización
    - VMWare
    - VirtualBox
    - Parallels
    - Etc.



# Introducción. Virtualización

- **Consumo (Green IT). Reducir consumo equipamiento**
  - **Consolidación de servidores.**
    - La consolidación de servidores es la reestructuración de la infraestructura del CPD con el fin de reducir costes y mejorar el control mediante la optimización de los requisitos de recursos.



# Introducción. Cloud Computing

- **Consumo (Green IT).** Reducir consumo equipamiento
  - Técnicas de utilización dinámica de recursos
    - El modelo Cloud
      - Modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizado y responder a las necesidades del negocio, de forma **flexible** y **adaptativa**, [...] pagando únicamente por el consumo efectuado.
      - Es un modelo que proporciona de manera conveniente, acceso por demanda a un conjunto compartido y configurable de recursos informáticos (redes, servidores, almacenamiento, aplicaciones, etc) que pueden ser rápidamente dispuestos con un esfuerzo mínimo por parte del proveedor de estos recursos.

# Introducción. Cloud Computing



## Desventajas

- Conexión constante
- Puede ser lento
- Privacidad



## Prestaciones

- Adaptación elástica
- Capacidad almacenamiento ilimitada
- Disponibilidad

## Reducción de costes

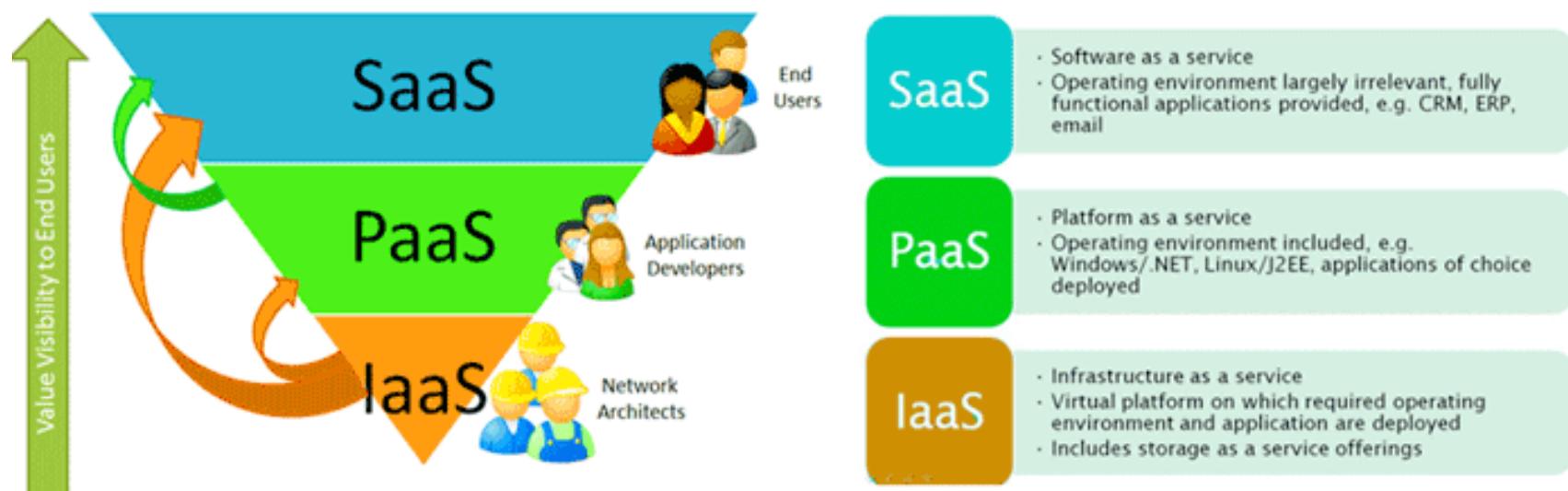
- Infraestructura
- Licencias de software
- Energía
- Personal

## Gestión

- Menos incidentes
- Actualizaciones de software
- Automatización en gestión

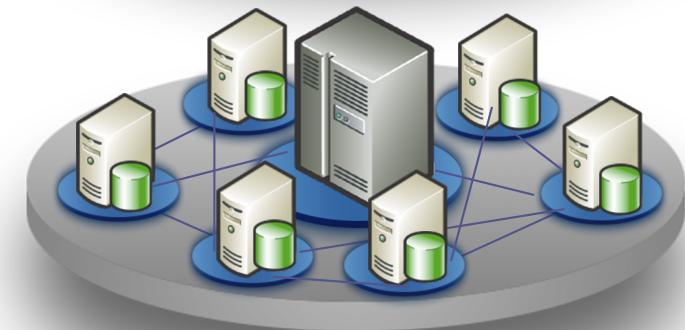
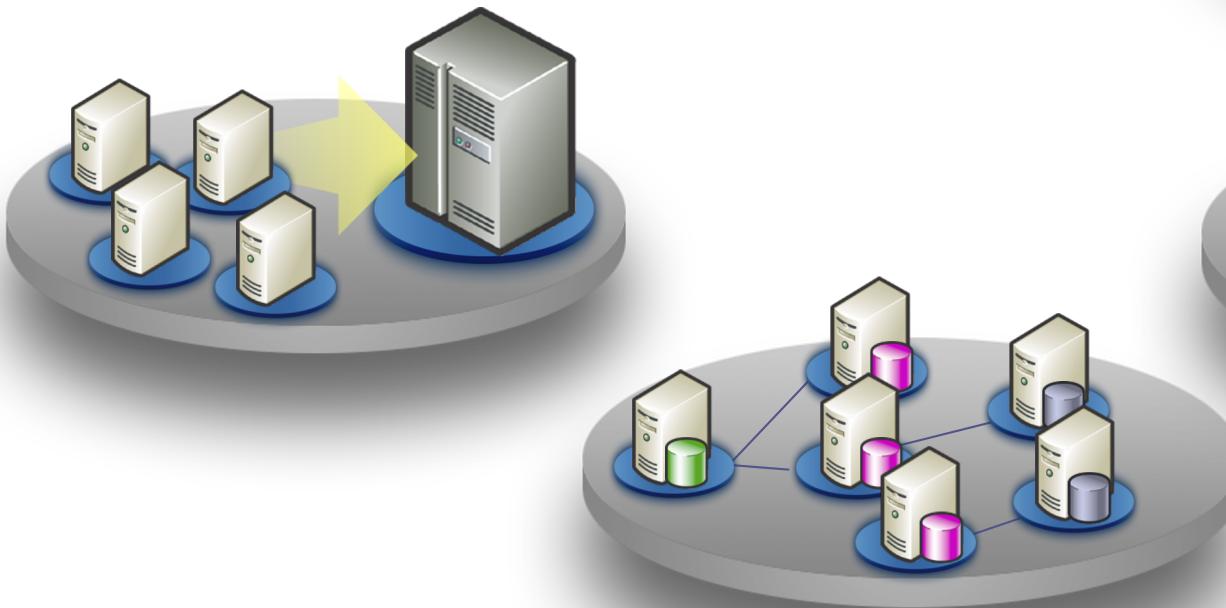
# Introducción. Cloud Computing

- Utilización dinámica de recursos -> Acceso por demanda a un conjunto compartido y configurable de recursos informáticos.
  - Modelo de negocio cloud**



# Introducción. Arquitectura de servicio

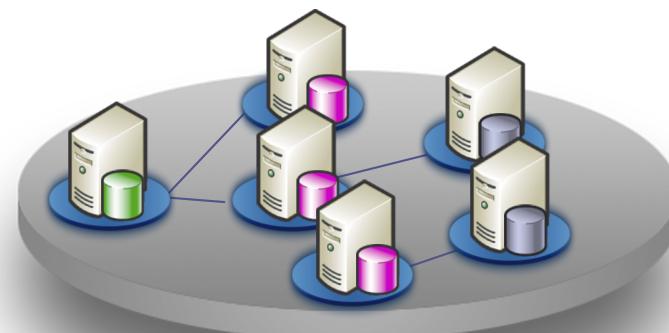
- Sistema aislado
- Arquitectura cliente-servidor
- Arquitectura de  $n$  capas
- Arquitectura Cliente-Cola-Cliente



# Introducción. Arquitectura de servicio

Posibles aproximaciones para servidor web:

- Montar un servidor en una máquina
- Adquirir un segundo servidor para desarrollo
- Adquirir un segundo servidor para sustituir el principal en caso de desastre
- Montar una granja web



# Índice



Introducción

[ Conceptos básicos ]

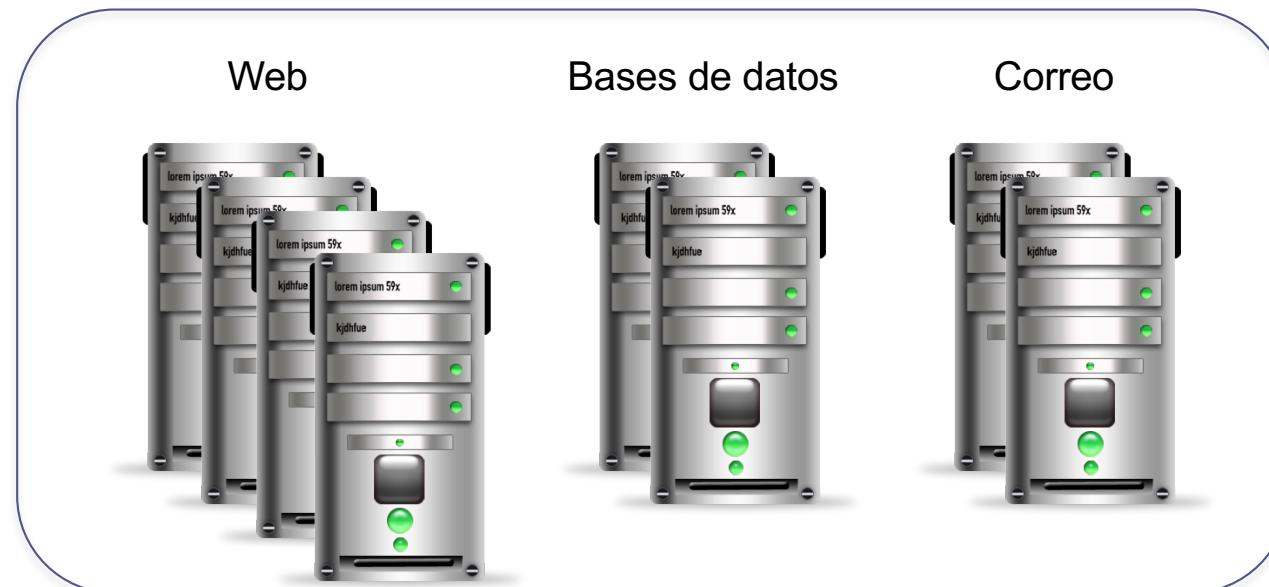
Un sitio web mal planificado

Un sitio web de éxito

## 2. Conceptos básicos

Una granja web es un conjunto de servidores dedicados a servir contenido web a los usuarios finales.

Algunos de esos servidores servirán contenido estático, otros actuarán como servidores de bases de datos, otros para aplicaciones dinámicas, etc.



## 2. Conceptos básicos

En cada uno de esos grupos de máquinas se puede instalar diferente tipo de aplicaciones y/o versiones:

- apache
- nginx
- thttpd
- Cherokee
- node.js

Cada uno de esos es más eficiente para un tipo de servicio.

## 2. Conceptos básicos

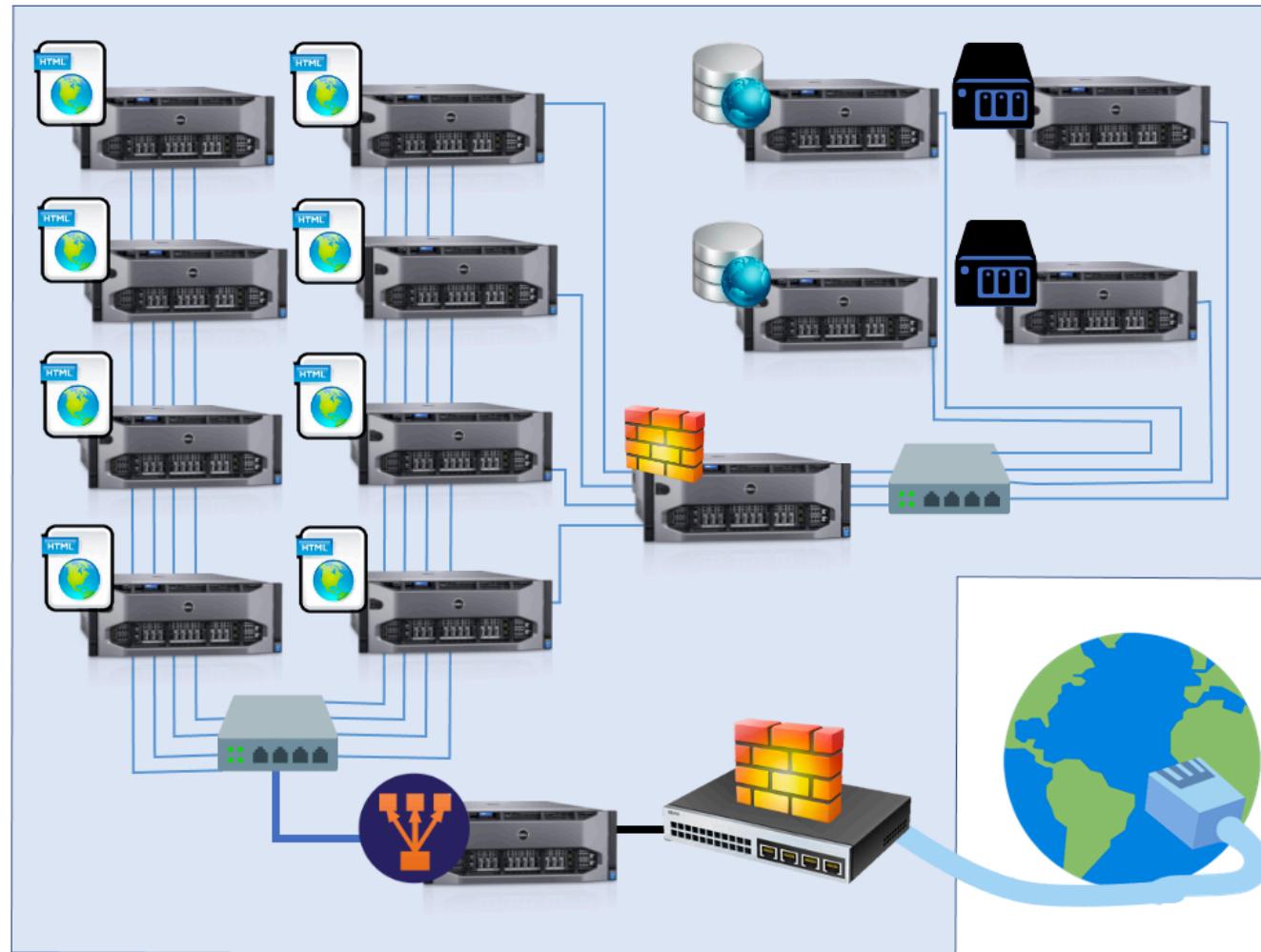
Son sistemas muy complejos pero sumamente flexibles y funcionales.

En muchos casos, una granja web albergará muchos sitios web:

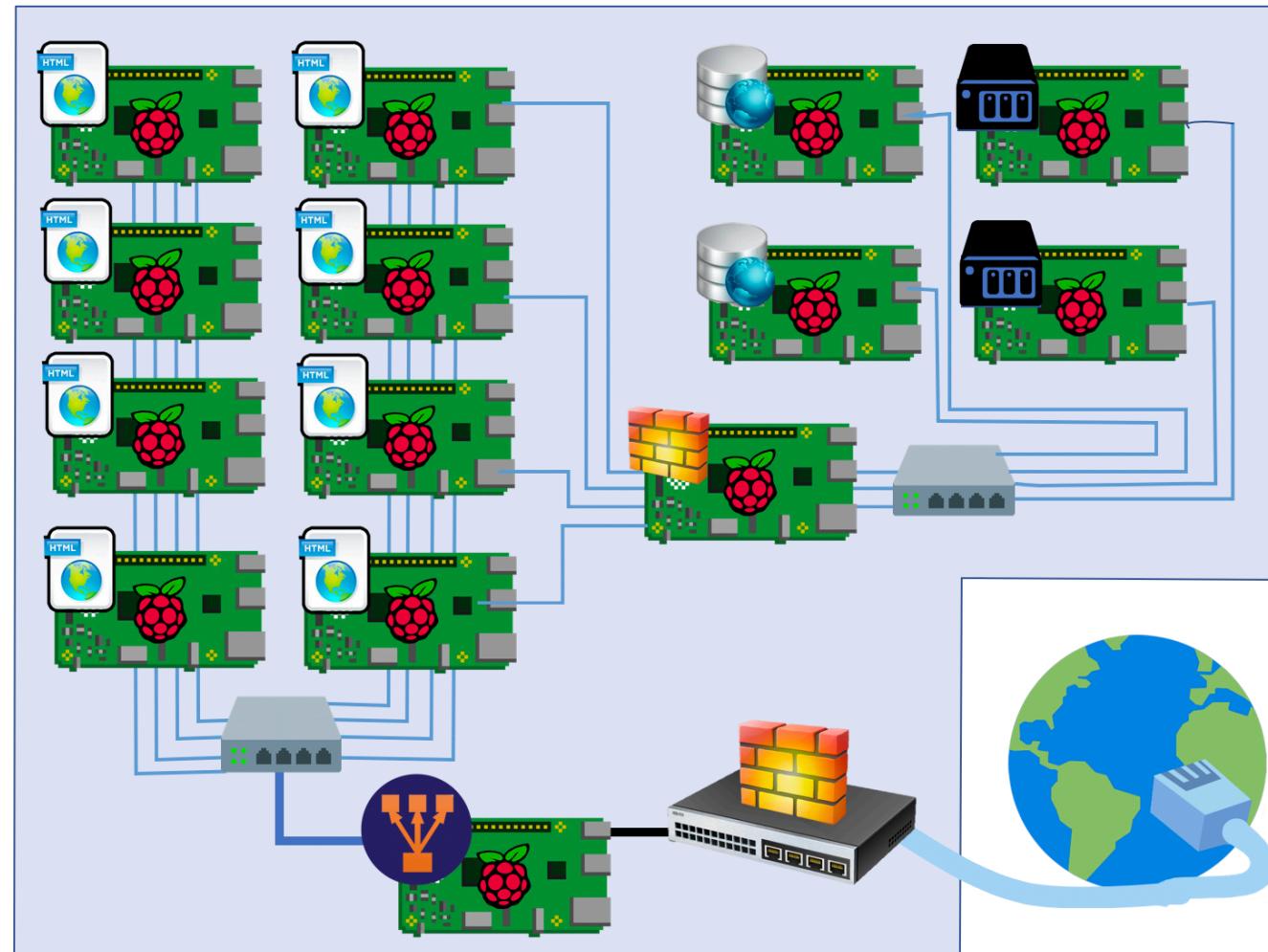
- correspondientes a varias empresas
- servicios bien diferenciados de una empresa

Estos sistemas resuelven los problemas de sistemas más simples para dar servicio a un alto número de usuarios.

## 2. Conceptos básicos



## 2. Conceptos básicos



# Índice



Introducción

Conceptos básicos

[ Un sitio web mal planificado ]

Un sitio web de éxito

# 3. Un sitio web mal planificado

## Introducción

Cuando nos encargan el desarrollo de un sitio web, inicialmente se suelen instalar todos los servicios en una sola máquina.

Opción más rápida y barata.

Si no hay muchos usuarios (poco tráfico), funcionará.

Sólo hace falta una máquina con el sistema operativo bien configurado, una aplicación de servidor web, y una conexión a Internet.

### 3. Un sitio web mal planificado

Ese sitio, con esa configuración tan simple, no tardará en experimentar problemas conforme se incremente el tráfico de red (peticiones).

Un problema puede sobrevenir por el uso de un almacenamiento de datos inadecuado (no usar bases de datos bien configuradas), que redunda en una falta de escalabilidad.

## 3. Un sitio web mal planificado

Primera aproximación:

Montar un servidor en una máquina



fácil y rápido

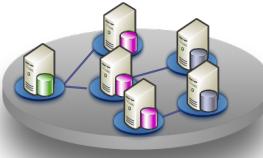


necesitará más trabajo de mantenimiento  
tarde o temprano, tendremos problemas

Las tareas derivadas de problemas hardware y software,  
para dar el servicio adecuado, acaba siendo un trabajo  
demasiado costoso en un sistema mal planificado.

### 3. Un sitio web mal planificado

Necesidad de controlar posibles problemas:



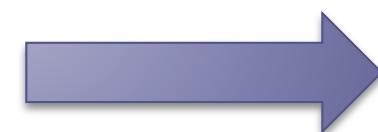
con la arquitectura del sistema



con la disponibilidad, y



con la carga del sistema





### 3. Un sitio web mal planificado

#### Problemas de la arquitectura del sistema

En muchos sitios, una máquina servidora ofrece todos los servicios, conectada directamente a Internet.

Comúnmente se pone en marcha sin cortafuegos adecuado.

Algunos servicios son para uso de los empleados, por lo que sobrecargan al servidor (que no dará buen servicio a los usuarios externos).

Riesgo de seguridad.

Modificaciones directamente en el servidor en producción...



# 3. Un sitio web mal planificado

## Problemas con la disponibilidad (I)

Conforme los usuarios usan un sistema, los errores derivados de la programación van apareciendo.

Además, suelen solicitar mejoras o ampliaciones del sistema.

Arreglar esos errores puede resultar complicado: los retoques del código o quitar servicios pueden suponer nuevos errores más adelante.



# 3. Un sitio web mal planificado

## Problemas con la disponibilidad (II)

Las actualizaciones del SO, de aplicaciones o servicios pueden afectar al rendimiento global.

Los cambios van directamente al servidor en producción.

### POSIBLE SOLUCIÓN:

Comprar una segunda máquina (desarrollo)

Trabajo extra + coste de adquisición A red thumbs down icon.



# 3. Un sitio web mal planificado

## Segunda aproximación (I)

### Adquirir un segundo servidor para desarrollo.

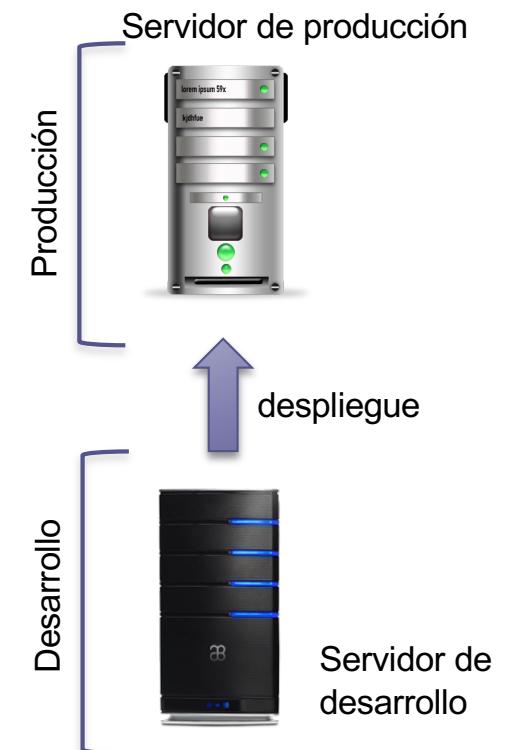


- las mejoras en software se pueden probar en la máquina de desarrollo antes de pasarlas al servidor de producción



- no se resuelven los problemas derivados de un alto tráfico

Ni siquiera actualizando el hardware del servidor de producción se podrá hacer frente a ciertos niveles de tráfico...





# 3. Un sitio web mal planificado

## Problemas con la disponibilidad (III)

Tarde o temprano el hardware falla.

### POSIBLE SOLUCIÓN:

Tener una máquina servidor replicada (*cold spare server*)



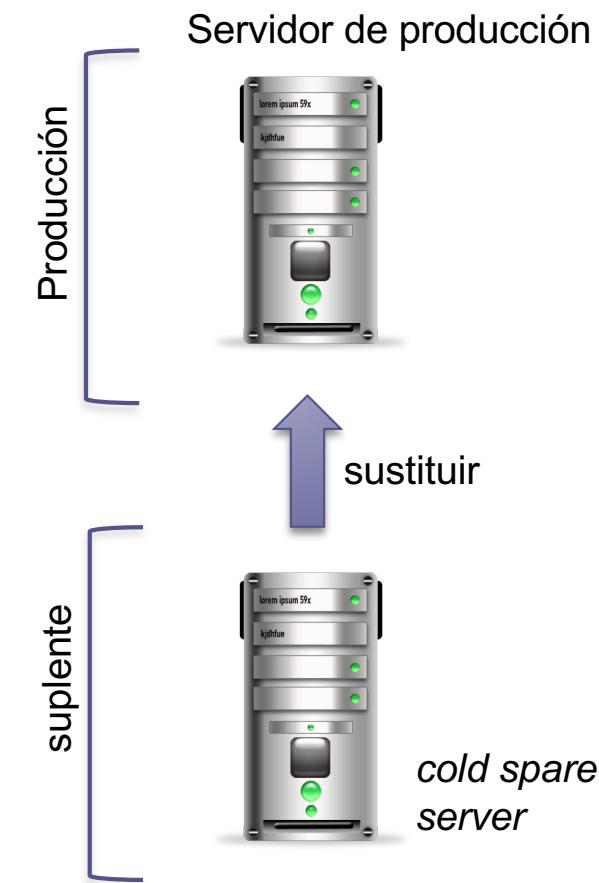
# 3. Un sitio web mal planificado

## Problemas con la disponibilidad (IV)

*cold spare server*

Hay que tenerla actualizada  
(programas y datos).

Requiere esfuerzo y  
coste de adquisición 





### 3. Un sitio web mal planificado

Segunda aproximación (II):

Adquirir un segundo servidor para sustituir el principal en caso de desastre



- en caso de desastre grave, se puede sustituir el servidor principal por el secundario y continuar trabajando
- no se resuelven los problemas derivados de un alto tráfico



Otra vez, ni siquiera actualizando el hardware del servidor de producción se podrá hacer frente a ciertos niveles de tráfico o carga...



## 3. Un sitio web mal planificado

### Problemas con la carga del sistema (I)

Un sitio con poco tráfico funcionará con una sola máquina.

Ante un incremento del número de accesos, el hardware y software comenzarán a fallar.

Los recursos se verán sobre-utilizados (uso de CPU por encima del 90%, uso del almacenamiento y de toda la memoria).

Los elementos hardware que primero suelen fallar son las fuentes de alimentación y los discos duros.

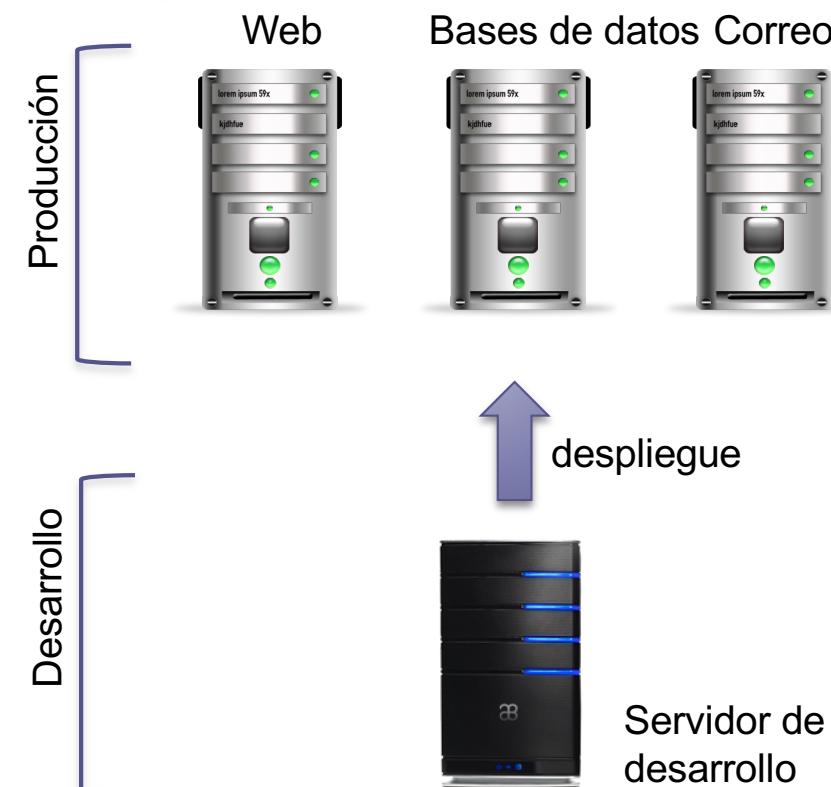


# 3. Un sitio web mal planificado

## Problemas con la carga del sistema (II)

POSIBLE SOLUCIÓN:

Configurar varias máquinas, dedicando una a cada servicio que se esté ofreciendo.





### 3. Un sitio web mal planificado

#### Problemas con la carga del sistema (III)

POSIBLE SOLUCIÓN:

Configurar varias máquinas, dedicando una a cada servicio que se esté ofreciendo.



Un fallo en una, sólo afecta a ese servicio.



Aún así, el sitio *no es escalable* (no podrá dar soporte a un número creciente de usuarios para cierto servicio).



# 3. Un sitio web mal planificado

## Problemas con la carga del sistema (IV)

La escalabilidad es la capacidad para dar soporte a un número creciente de usuarios.

Tanto el hardware y software de los servidores, pero sobre todo la estructura, organización y configuración de la red deben estar preparados para añadir recursos según sean necesarios.

# 3. Un sitio web mal planificado

## La solución final

Cuando los cambios se vuelven inmanejables, sólo queda reestructurar todo el sistema.

No merece la pena parchear el sistema a varios niveles.

Supone una inversión continua en tiempo y dinero.

Importante analizar las necesidades y hacer unas buenas especificaciones → capacity planning. Modelos analíticos

# Índice



Introducción

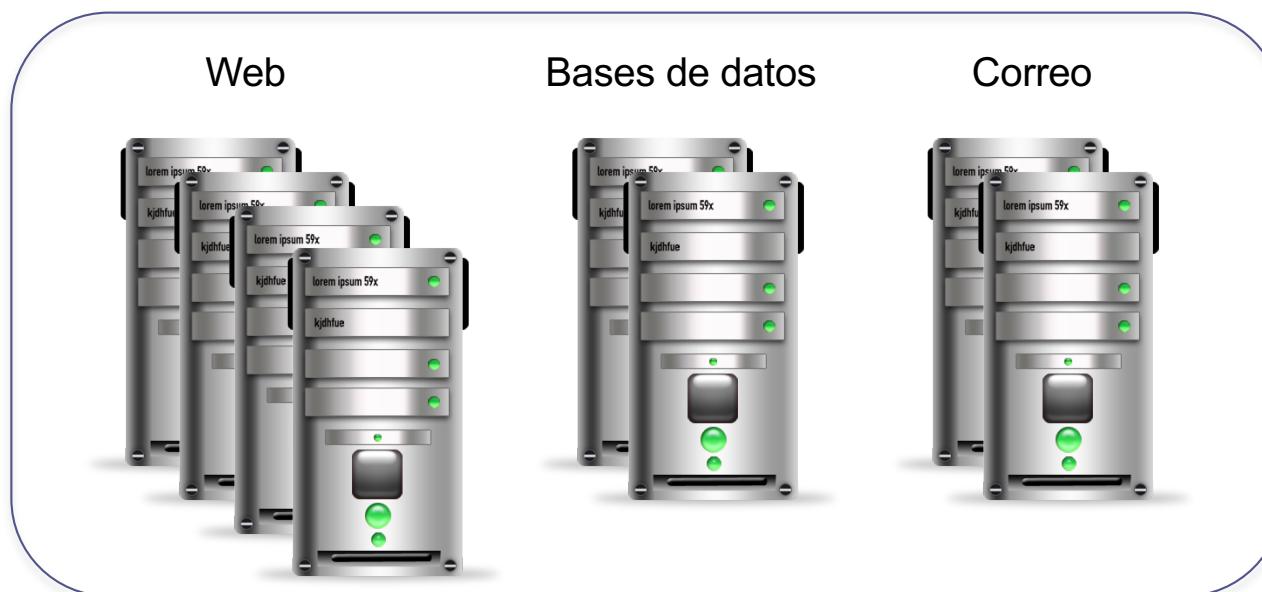
Conceptos básicos

Un sitio web mal planificado

[ Un sitio web de éxito ]

## 4. Un sitio web de éxito

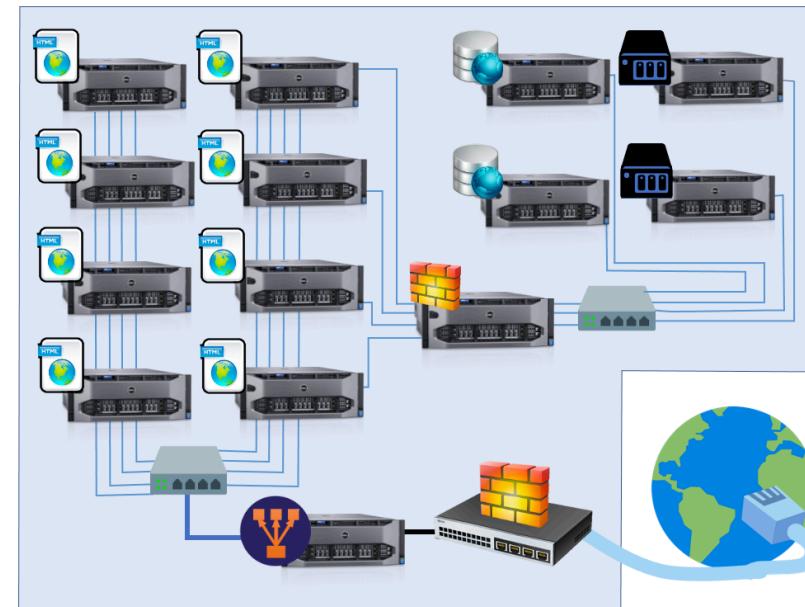
Según lo comentado, la estructura general para montar una granja web que escala correctamente sería:



Complejidad mayor que los sistemas basados en máquinas simples.

## 4. Un sitio web de éxito

Según lo comentado, la estructura general para montar una granja web que escale correctamente sería:



Complejidad mayor que los sistemas basados en máquinas simples.

# 4. Un sitio web de éxito

Tercera aproximación:

Montar una granja web

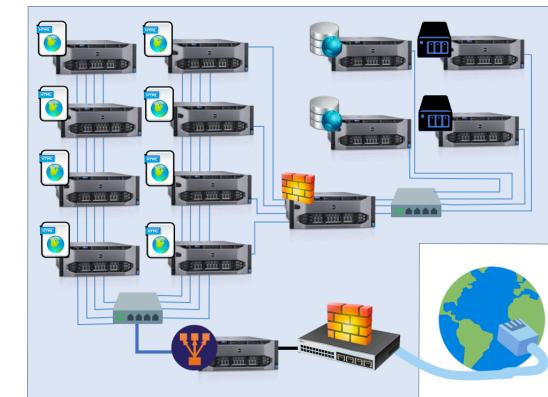


escalable y con alta disponibilidad



complejidad en la instalación y configuración

En aquellos casos en que haya que hacer frente a un alto tráfico de red y dar servicio a millones de usuarios, es la mejor solución.



# En la asignatura...

En el resto de temas estudiaremos los conceptos, herramientas y alternativas hardware y software necesarios para crear estos sistemas.

En la planificación deberemos tener en cuenta:

- Siempre habrá puntos débiles
- ¡y gente para atacarlos!
- El tipo de red es fundamental (ancho de banda)
- La seguridad es muy importante

# TEMA 2

## Alta disponibilidad y Escalabilidad

SWAP



¿Cuánto tiempo está disponible un sistema para dar respuesta a usuarios?  
¿El sistema se adapta bien a más peticiones?



José Manuel Soto Hidalgo  
Dpto. Arquitectura y Tecnología de Computadores  
Universidad de Granada

jmsoto@ugr.es

# Índice



## [ Introducción ]

Concepto de alta disponibilidad

Concepto de escalabilidad

Escalar un sitio web

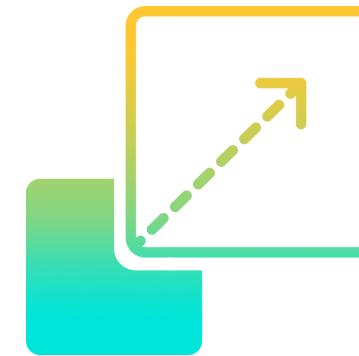
Conclusiones

# Introducción

disponibilidad



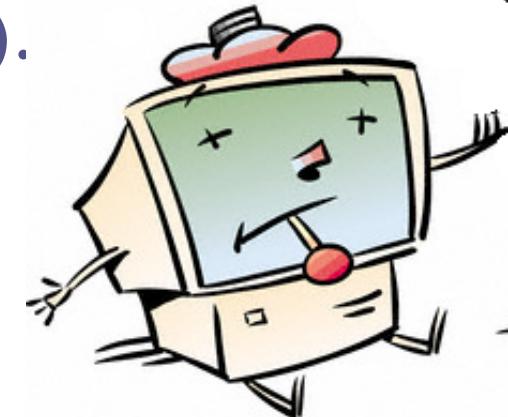
escalabilidad



Dos conceptos de los más importantes a tener presentes en el diseño de una granja web

# Introducción

El éxito de una empresa depende de que los usuarios tengan buena experiencia al visitarla (su web).



# Introducción

Nuestros servidores deben dar el mejor servicio a todos los usuarios y deben estar todo el tiempo disponible (24/7).



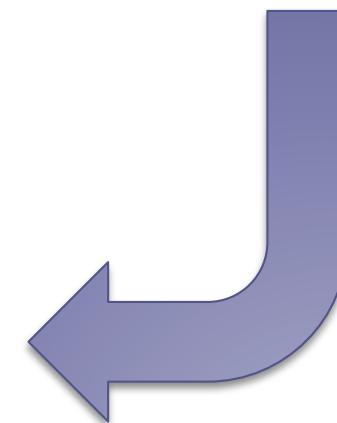
- Disponibilidad



- Escalabilidad



- Balanceo de carga



# Índice



Introducción

[ Concepto de alta disponibilidad ]

Concepto de escalabilidad

Escalar un sitio web

Conclusiones

# Alta disponibilidad

<http://www.isitdownrightnow.com/>



Is Ugr down? Check all its services

Ad closed by Google

 **Ugr.es Server Status Check**

 **Ugr.es**



No screenshot available

<b>Website Name:</b>	Ugr
<b>URL Checked:</b>	ugr.es
<b>Response Time:</b>	370.51 ms.
<b>Last Down:</b>	More than a week ago

**UP** **Ugr.es is UP and reachable by us.**  
Please check and report on local outages below ...

[Report an Issue](#)



Hipertextual   
@Hipertextual

La caída de Amazon S3 rompe medio internet



La caída de Amazon S3 rompe medio internet  
La caída de una de las zonas más populares y...  
[hipertextual.com](http://hipertextual.com)

# Alta disponibilidad

<https://outage.report/whatsapp>

[Home](#) » [WhatsApp](#) » [Outage Map](#)

## Is WhatsApp Down Right Now?

See if WhatsApp is down or having service issues today

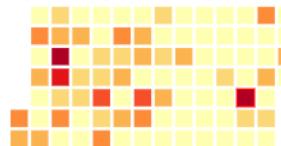


 Not Working For Me! ▾

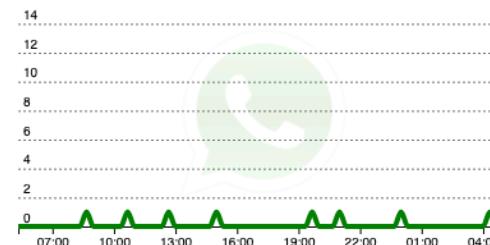
Everything is down - 50%  
Mobile app not working - 25%  
Message send problems - 25%

### Outage History

Dec Jan Feb



### Reports Dynamics EST (GMT -05:00)



Received 8 reports, originating from United States of America, Mexico, Jamaica, Portuguese Republic, Russia and 2 more countries

### WhatsApp Outage Map Live



Riga	Everything is down	36m
Castro Valley	Message send problems	4h
Flemington	Mobile app not working	7h
	Everything is down	2h



### Outage Report

@ReportOutage



WhatsApp is having issues since 04:11 PM  
EST[outage.report/whatsapp](https://outage.report/whatsapp)

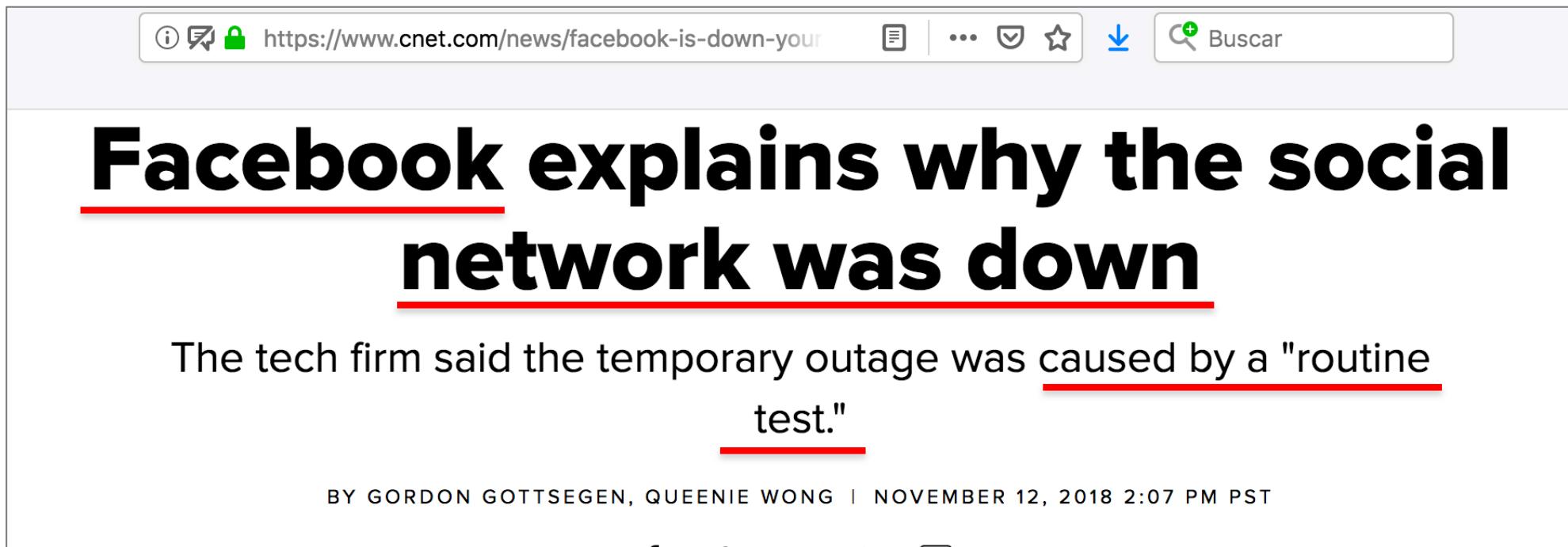
RT if you're also affected #whatsappdown

22:12 - 3 May 2017

8 60 14

# Antes de empezar el tema...

En el Tema 1 vimos la parte relativa a la DISPONIBILIDAD



The screenshot shows a web browser window with the following details:

- Address bar: https://www.cnet.com/news/facebook-is-down-your
- Toolbar icons: refresh, search, etc.
- Page content:
  - Facebook explains why the social network was down**
  - The text "The tech firm said the temporary outage was caused by a "routine test." is displayed.
  - Byline: BY GORDON GOTTSSEGGEN, QUEENIE WONG | NOVEMBER 12, 2018 2:07 PM PST

[xataka.com](https://www.xataka.com/tecnologia/la-misma-facebook-inform%C3%B3-hace-una-hora-a-trav%C3%A9s-de-twitter-que-su-familia-de-aplicaciones-facebook-whatsapp-instagram-y-messenger-estaba-presentando-problemas-y-que-ya-estaban-trabajando-en-resolverlo)

La misma Facebook informó hace una hora a través de Twitter que su familia de aplicaciones (Facebook, WhatsApp, Instagram y Messenger) estaba presentando problemas y que ya estaban trabajando en resolverlo.

 **Facebook**   
@facebook

We're aware that some people are currently having trouble accessing the Facebook family of apps. We're working to resolve the issue as soon as possible.

18:49 - 13 mar. 2019

46,2 mil likes

30,7 mil personas están hablando de esto

Hasta este momento, Facebook, WhatsApp, Instagram y Messenger están caídos en gran parte del mundo, por lo que se pensó que se trataría de un ataque DDoS, pero Facebook ya ha confirmado que no están bajo ningún tipo de ataque,

<https://downdetector.ca/status/facebook>

Más visitados

downdetector.ca 

Home Top 10 Companies Problems Pro Services About us

Home / Companies / Facebook

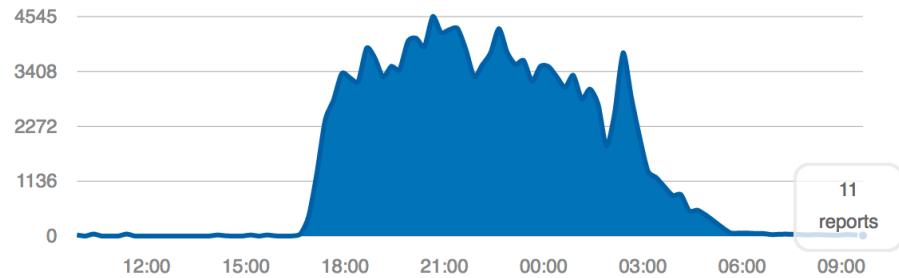
## Facebook

Facebook is a social network where member share messages and status updates with online friends. The network also offers a platform for third party developers.



Possible problems at Facebook

Facebook problems last 24 hours



Live Outage Map »

I have a problem with Facebook Check past issues

# Antes de empezar el tema...



Wardog  
@mundowdg

Tuve un jefe que cuando caía su  
único servidor (nada de  
redundancia, eso es tirar el dinero)  
siempre gritaba:

"¡A que ésto no le pasa a Facebook!  
¡Pues si Facebook puede hacerlo es  
que se puede hacer!"

Cuán atrevida es la ignorancia y qué  
puta es la informática.

# Antes de empezar el tema...

## disponibilidad



Luis del Barco  
@lbarcob

Seguir



Pavel Durov, CEO de Telegram, sacando  
pechito de la cantidad de nuevos usuarios  
que han llegado al servicio tras la caída de  
Facebook, Instagram y WhatsApp.



Fabianzoid™  
@Freakaaazoid

Seguir



Durov's Channel

31.4K

I see 3 million new users signed up for  
Telegram within the last 24 hours.

Good. We have true privacy and unlimited  
space for everyone.

Telegram: ¿Volveré a verte?

Tú: Claro, en la próxima caída de whatsapp.



# Alta disponibilidad

Mala impresión al entrar en un sitio y está caído.

Esperamos que una web esté disponible siempre.

Disponibilidad: capacidad de aceptar visitas las 24h todos los días.



SEPTIEMBRE/OCTUBRE 2014							NOVIEMBRE 2014							DICIEMBRE 2014							ENERO 2015																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

# Alta disponibilidad

Cuando un sitio no está disponible se dice que se ha caído o sufre un problema de no-disponibilidad:

- Tiempo de no-disponibilidad (downtime) programado.
- Tiempo de no-disponibilidad (downtime) no programado.

Sólo debería haber "tiempos de no-disponibilidad programados" (y lo más cortos posibles)



actualizaciones del SO, de aplicaciones o de hardware

# Alta disponibilidad

Medir la disponibilidad dando un porcentaje.

Escala “punto nueve”:

$$100 - (\text{tiempoCaido} / \text{periodoTiempo}) * 100$$

Por ejemplo:

caída de 1h en un día -> 95.83333% de disponibilidad

caída de 1h en una semana -> 99.404% de disponibilidad

Lo ideal es tener un 100% de disponibilidad.

# Alta disponibilidad

Un 100% de disponibilidad es no sufrir caídas no-programadas

Los sitios web se conforman con alcanzar un 99.9% ó 99.99%

Disponibilidad (%)	Periodo de un año
90%	36.5 días
95%	18.25 días
98%	7.3 días
99%	3.65 días
99.9%	8.76 horas
<u>99.99%</u>	<u>52.56 minutos</u>
99.999%	315 segundos
99.9999%	31.5 segundos

# Alta disponibilidad

<http://www.edgeblog.net/2007/in-search-of-five-9s/>

Otra forma de escribir la ecuación:

$$\text{Availability} = \text{Uptime} / (\text{Uptime} + \text{Downtime})$$

Horas en un año (periodo) = 8760

Horas caído (tiempo caído)= 1830

Tiempo en activo =  $8760 - 1830 = 6930$

Disponibilidad =  $6930 / 8760 = 0.791$

Disponibilidad =  $100 - (1830 / 8760) * 100 = 79.1$

# Alta disponibilidad

<http://www.edgeblog.net/2007/in-search-of-five-9s/>

¿Cómo podemos calcular la disponibilidad de un sistema?

$$AS = Ac_1 * Ac_2 * Ac_3 * \dots * Ac_n$$

Si tenemos dos servidores (web+BD) y cada uno tiene 99%, la disponibilidad del sistema será  $99\% * 99\% = 98.01\%$

Si cada uno puede estar caído 3.65 días, podemos esperar que el sistema esté caído un total de 7.3 días en un año.

Peor caso: que esté caído 3.65 días el servidor web y justo después caiga el de BD otros 3.65 días

Disponibilidad (%)	Periodo de un año
90%	36.5 días
95%	18.25 días
98%	7.3 días
99%	3.65 días
99.9%	8.76 horas
99.99%	52.56 minutos
99.999%	315 segundos
99.9999%	31.5 segundos

# Alta disponibilidad

Pero los sistemas reales son más complejos...

Hay muchos más elementos y algunos redundantes.

Necesitamos fórmulas algo más complejas:

(1) Para un sistema **s** con **n** componentes, la disponibilidad del sistema se calcula como:

$$A_s = A_{c1} * A_{c2} * A_{c3} * \dots * A_{cn}$$

<u>Component</u>	<u>Availability</u>
Web	85%
Application	90%
Database	99.9%
DNS	98%
Firewall	85%
Switch	99%
Data Center	99.99%
ISP	95%

Ejemplo de sitio de comercio electrónico con varios puntos de fallo

# Alta disponibilidad

En ese ejemplo, si cualquier componente falla, supondremos que el sistema falla.

La disponibilidad será:

$$85\% * 90\% * 99.9\% * 98\% * 85\% * 99\% * 99.99\% * 95\% = \underline{\underline{59.87\%}}$$

¿Parece baja? Al usuario le importa que el sistema proporcione el servicio. Si está caído, le dará igual que sea por el cortafuegos o por fallo de una aplicación web.

¿cómo mejorar la disponibilidad de este sistema?

# Alta disponibilidad

¿y si mejoramos la disponibilidad del servidor web y cortafuegos?

(2) Si el sistema tiene un componente replicado, la disponibilidad de esa parte del sistema completo será:

Component	Availability
Web	85%
Application	90%
Database	99.9%
DNS	98%
Firewall	85%
Switch	99%
Data Center	99.99%
ISP	95%

$$A_{NuevoC} = Ac1 + ( (1 - Ac1) * Ac2 )$$

En el ejemplo, añadir un segundo servidor web hará:

$$\text{disponibilidad\_web2} = 0,85 + (1-0,85)*0,85 = 0,9775$$

$$\text{disponibilidad\_web2} = 85\% + ((100\%-85\%) * 85\%) = 97.75\%$$

# Alta disponibilidad

Antes teníamos un 59.87% para todo el sistema.

¿Qué disponibilidad tendremos si replicamos el servidor web y el cortafuegos?

Cada uno de esos componentes (servidor web y cortafuegos) tendrán ahora 97.75%

Y el sistema:

$$97.75\% * 90\% * 99.9\% * 98\% * \underline{97.75\% * 99\% * 99.99\% * 95\%} = 79.10\%$$

Hemos mejorado en 19.23%

Pasaríamos de unas 3500 horas de no-disponibilidad al año a unas 1830 horas de no-disponibilidad al año.

# Alta disponibilidad

Si replicáramos cada elemento de red, servidores e ISP,  
dejando un solo centro de datos:

$$97.75\% * 99\% * 99.9999\% * 99.96\% * 97.75\% * 99.99\% * 99.99\% * 99.75\% = \underline{94.3\%}$$

Mejorado en 34.43%

Component	Availability
Web	97.75%
Application	99%
Database	99.9999%
DNS	99.96%
Firewall	97.75%
Switch	99.99%
Data Center	99.99%
ISP	99.75%

Pasaríamos de unas 3500 horas de no-disponibilidad al año  
a unas 500 horas de no-disponibilidad al año.

# Alta disponibilidad

Si generalizamos la última ecuación para cuando replicamos dos componentes:

$$A_{\text{nuevo}} = A_{C_{n-1}} + ( (1 - A_{C_{n-1}}) * A_{C_n} )$$

Así, si hemos añadido un tercer servidor web:

$$\text{disponibilidad\_web3} = 97.75\% + (1-97.75\%) * 85\% = 99.6625\%$$

Y si añadimos un cuarto servidor web:

$$\text{disponibilidad\_web4} = 99.6625\% + (1-99.6625\%) * 85\% = 99.949\%$$

# Alta disponibilidad

## Ejercicio T2. 1:

*Calcular la disponibilidad del sistema si tenemos dos réplicas de cada elemento (en total 3 elementos en cada subsistema).*

Disponibilidades iniciales

Component	Availability
Web	85%
Application	90%
Database	99.9%
DNS	98%
Firewall	85%
Switch	99%
Data Center	99.99%
ISP	95%

Con 2 elementos en cada subsistema

Component	Availability
Web	97.75%
Application	99%
Database	99.9999%
DNS	99.96%
Firewall	97.75%
Switch	99.99%
Data Center	99.99%
ISP	99.75%

# Alta disponibilidad

¿Cómo se consigue mejorar la disponibilidad?

El uso de subsistemas redundantes y monitorizarlos mejora la disponibilidad del sistema global.

Surgen conceptos derivados:

-  • disponibilidad de red
-  • disponibilidad de servidor
-  • disponibilidad de aplicación

Si la disponibilidad de red es baja, quizás haya que mejorar el ancho de banda, y no tenga sentido centrar esfuerzos en mejorar las aplicaciones.



# Alta disponibilidad

## Disponibilidad de la red

El diseño debe tener redundancia a todos los niveles:

- Conexión a Internet
- Routers/cortafuegos/balanceadores
- Servidores

Si hay que recortar costes, algún elemento puede ser único:  
p.ej. el router, si el proveedor se compromete a  
reemplazarlo en pocas horas.



# Alta disponibilidad

## Disponibilidad del servidor

Casi cualquier elemento hardware del servidor puede fallar:

- CPU, memoria, discos, placas, etc.

Existen productos en el mercado con cualquier elemento duplicado.





# Alta disponibilidad

## Disponibilidad del servidor (ejemplo)

Sistemas redundantes en el Curiosity (doble placa):

<https://milesdemillones.com/2013/03/01/curiosity-en-modo-seguro-por-un-problema-con-el-ordenador-de-a-bordo/>

“El día 27 de febrero de 2013 fue incapaz de guardar los datos adquiridos en la memoria flash [...]”

El fallo ocurrió mientras estaba usando la computadora A, por lo que desde control se ordenó pasar a usar el ordenador B para asegurar el correcto funcionamiento de los sistemas básicos”





# Alta disponibilidad

## Disponibilidad del servidor

Puesto que casi cualquier elemento hardware del servidor puede fallar...

Se pueden configurar los servidores con redundancia mediante el software.

Mejora de la escalabilidad.





# Alta disponibilidad

## Disponibilidad del servidor

Monitorizar la disponibilidad con las herramientas del SO.

El desarrollador Emerson ofrece herramientas para monitorizar hardware y software.

The screenshot shows the Emerson website's navigation bar with links for AUTOMATION SOLUTIONS and COMMERCIAL & RESIDENTIAL SOLUTIONS, along with a search bar and a shopping cart icon. Below the navigation, a breadcrumb trail indicates the current page: Home / DeltaV™ OPC Data Access Server Redundancy. On the left, there are tabs for IMAGES (1) and VIDEOS (0). A thumbnail image of a computer monitor displaying a network diagram is shown under the IMAGES tab. To the right, the main content area is titled "DeltaV™ OPC Data Access Server Redundancy". The text describes how the DeltaV™ OPC Data Access Server provides a fast and efficient means for transferring data between the DeltaV system and OPC Data Access client applications, using redundant OPC servers to protect against hardware and software failures. At the bottom, there are two green call-to-action buttons: "CONTACT US >" and "LEARN ABOUT >".

EMERSON

AUTOMATION SOLUTIONS

COMMERCIAL & RESIDENTIAL SOLUTIONS

Search

Home / DeltaV™ OPC Data Access Server Redundancy

IMAGES (1)

VIDEOS (0)

DeltaV™ OPC Data Access Server Redundancy

The DeltaV™ OPC Data Access Server provides a fast and efficient means for transferring data between the DeltaV system and OPC Data Access client applications. With redundant OPC servers, you don't have to worry about a failure of an OPC server or the Application Station interrupting your data transfer and causing costly downtime. Using redundant OPC servers you are automatically protected against single point OPC server hardware and software failures.

CONTACT US >

LEARN ABOUT >



# Alta disponibilidad

## Disponibilidad de las aplicaciones

Complicado medir las prestaciones de las aplicaciones.

El funcionamiento de unos módulos afectan al de otras aplicaciones (dependencias).

El desarrollador del SO suele facilitar herramientas para monitorizar también las aplicaciones en ejecución.



# Alta disponibilidad

## Disponibilidad de las aplicaciones

Dependencia entre aplicaciones o módulos de aplicación:

- Si un módulo falla, el proceso no se completa, y la experiencia del usuario es mala.

Desarrollar aplicaciones robustas -> hacerlas redundantes.

Si una parte del proceso falla, que haya una alternativa para completarlo.

# Índice



Introducción

Concepto de alta disponibilidad

[ Concepto de escalabilidad ]

Escalar un sitio web

Conclusiones

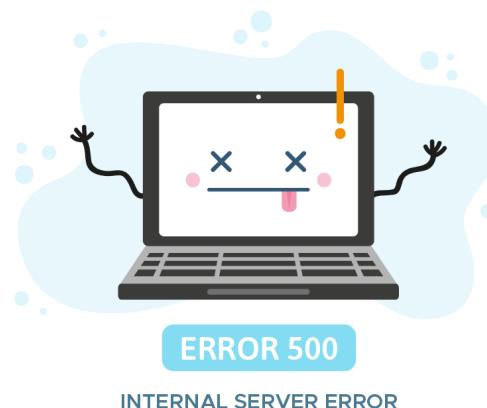


# Escalabilidad

Cuando una persona sufre estrés, su capacidad para afrontar tareas se ve mermada.



Cuando un sistema experimenta estrés, su capacidad para dar servicio también se ve afectada.





# Escalabilidad

Incremento del nivel de estrés:

- Cambios en las aplicaciones
- Fallos o caídas de algunas partes del sistema
- Incremento del número de máquinas
- Incremento repentino del número de usuarios del sitio
- Etc.

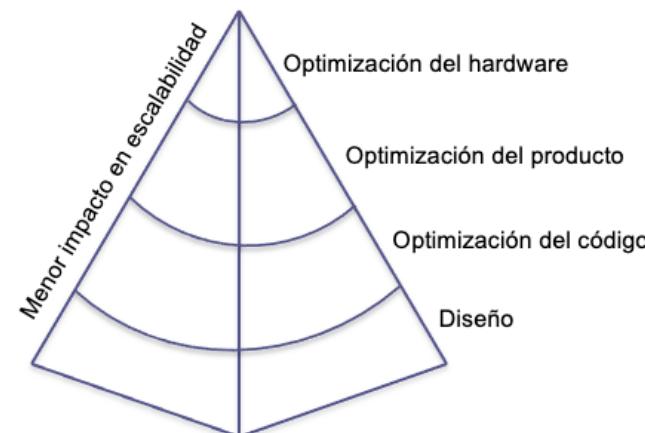
La *escalabilidad* se refiere a la capacidad de un sistema de manejar la carga, y el esfuerzo para adaptarse al nuevo nivel de carga.



# Escalabilidad

## Definición

- capacidad de un sistema de manejar la carga, y el esfuerzo para adaptarse al nuevo nivel de carga
- capacidad de adaptación y respuesta de un sistema con respecto al rendimiento del mismo a medida que aumentan de forma significativa el número de usuarios del mismo.





# Escalabilidad

Si un sitio gana popularidad, o si llega una fecha señalada, puede incrementarse su carga.

Para manejar esa carga, las empresas tienen más servidores de los necesarios normalmente.

Decidir cómo añadir más recursos al sistema web es crucial en el diseño inicial y en el mantenimiento.

En ocasiones, si la CPU del servidor está al 95% todo el tiempo, cambiándola puede ser suficiente para cierto nivel de carga. Pero si más adelante hay más carga, será insuficiente.



# Escalabilidad

Dos tipos de escalado:

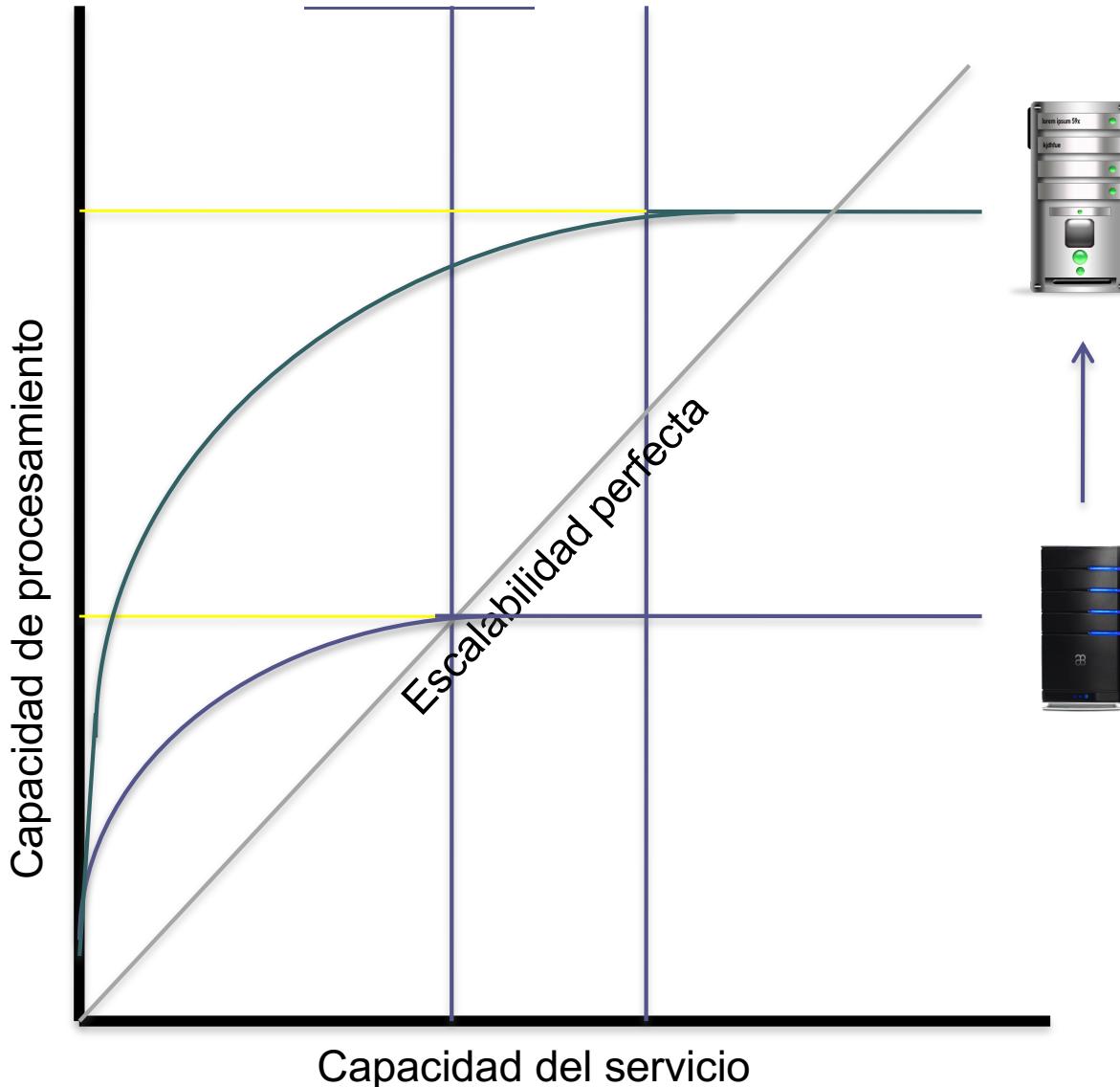
- Ampliación *vertical*:  
incrementar la RAM, CPU, disco de un servidor.
- Ampliación *horizontal*:  
añadir máquinas a algún subsistema (servidores web, servidores de datos, etc).

En ocasiones una ampliación vertical puede ser suficiente.



# Escalabilidad

## Escalado vertical:



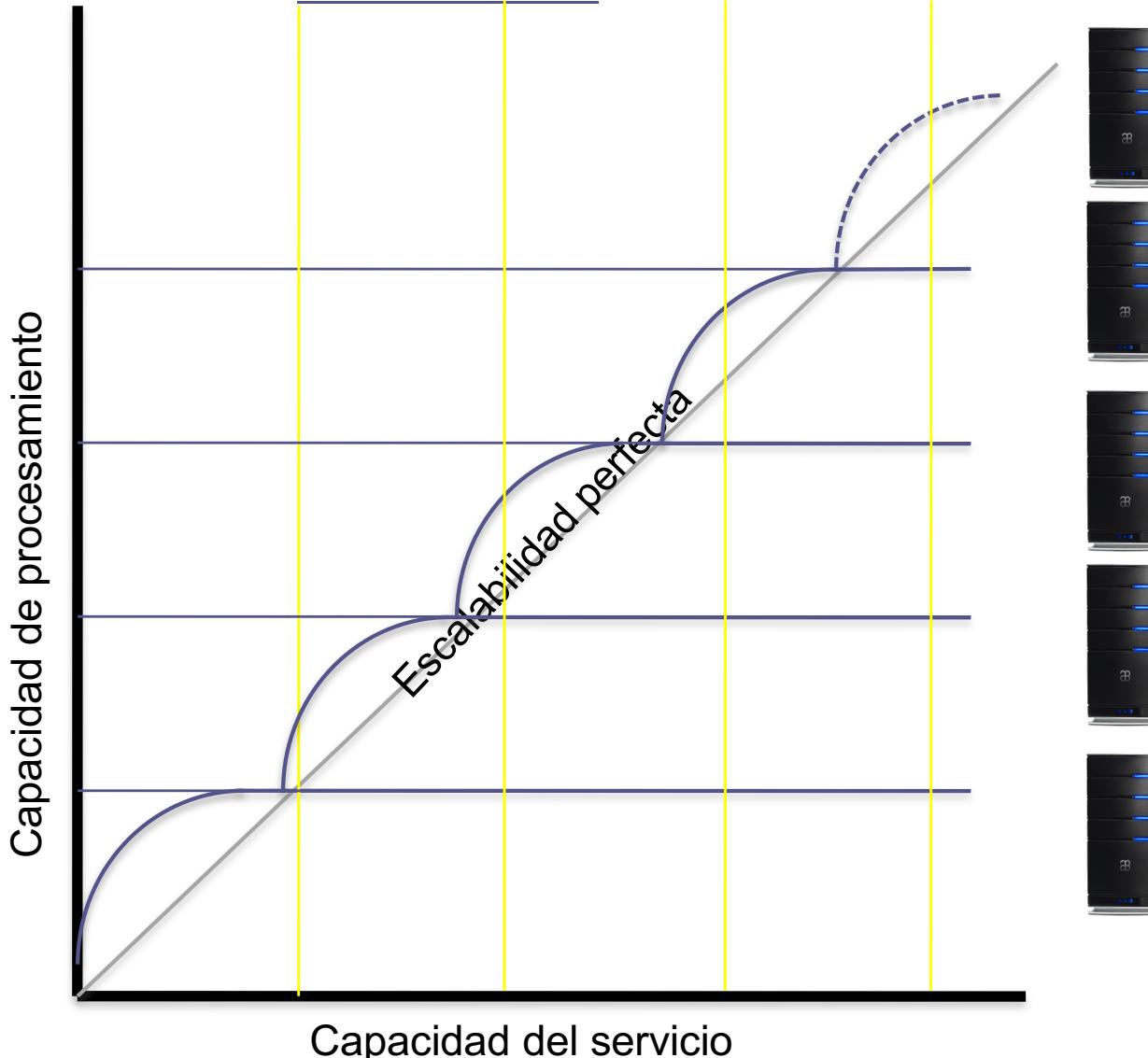
Si ponemos una máquina más potente, podrá dar servicio hasta aquí, pero a partir de ese nivel ya no podrá atender más carga

Ésta máquina sólo puede dar servicio hasta este punto.



# Escalabilidad

## Escalado horizontal:



Cada máquina incrementa un poco la escalabilidad.



# Escalabilidad

¿Cómo anализar la sobrecarga?

- Si la CPU está cerca del 100% todo el rato y el resto de subsistemas no está sobrecargado, sustituir por una CPU más potente.
- Si el uso de RAM es muy alto, veremos un uso alto de disco (por swapping). Incrementando la cantidad de RAM mejoraremos el rendimiento.
- Un ancho de banda insuficiente afectará al rendimiento. Contratando una mejor conexión será suficiente.



# Escalabilidad

*Transformar el servidor web en una granja web*

- Proceso complejo.
- Preparar aplicaciones para distribuir la carga.
- Configurar la red para soportar tráfico creciente.
- Configuración del balanceo de carga para formar un cluster web para cada servicio.

Una granja web puede tener varios clusters web.

# Índice



Introducción

Concepto de alta disponibilidad

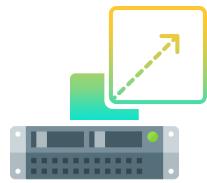
Concepto de escalabilidad

[ Escalar un sitio web ]

Conclusiones

# Escalar un sitio web

Tenemos que configurar tres niveles:



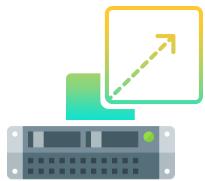
- máquinas como servidores web



- aplicaciones



- almacenamiento

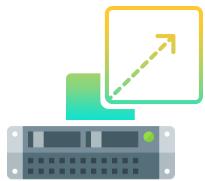


# Escalar un sitio web

El nivel web se puede configurar balanceando la carga:

- uso de una máquina con software específico

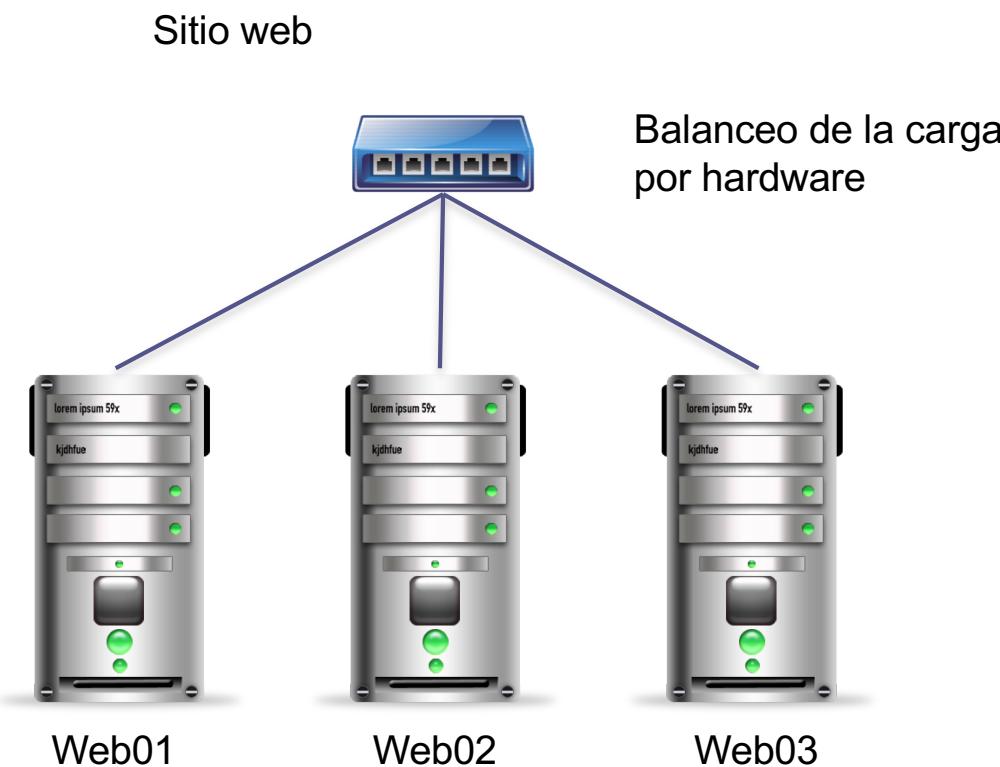


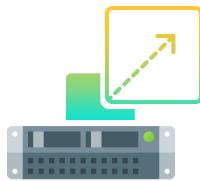


# Escalar un sitio web

También se puede usar un *balanceador hardware*:

- Local Director (Cisco)
- ServerIron (Foundry)
- BigIP (F5)





# Escalar un sitio web

El balanceador pasa peticiones a los servidores según el tráfico de la red.

Hay varios algoritmos para decidir qué máquina final servirá cada petición:

- Por turnos (round-robin)
- Según el menor número de conexiones
- Por ponderación
- Por prioridad
- Según el tiempo de respuesta



# Escalar un sitio web

Escalar el nivel de aplicaciones requiere diseñar el software pensando en que se ejecute en varios servidores:

- Paralelismo
- Transparencia de ubicación: no debe haber dependencia de una máquina concreta para ejecutarse la aplicación.

Es importante diseñar las aplicaciones desde el principio para que se ejecuten en varios servidores.

Adaptar posteriormente una aplicación dependiente de cierto servidor puede ser costoso.



# Escalar un sitio web

Escalar el nivel de almacenamiento es complejo y depende del tipo de servicios a ofrecer:

- LDAP: Protocolo Ligero de Acceso a Directorios
- NFS: Sistema de archivos de red
- Bases de datos

Cada uno de estos mecanismos suele requerir mecanismos y configuraciones diferentes.

# Índice



Introducción

Concepto de alta disponibilidad

Concepto de escalabilidad

Escalar un sitio web

[ Conclusiones ]

# Conclusiones

**Conceptos clave:** escalabilidad y alta disponibilidad.

**Monitorización** para detectar problemas y determinar posibles mejoras del sitio web.

La escalabilidad se suele implementar **replicando servidores** para las mismas tareas.

Conseguir disponibilidad y escalabilidad mediante **balanceo de carga**.

# TEMA 3

## La red de una granja web

SWAP



¿Qué configuración de red es más  
adecuada para la granja web?  
¿Será segura?



José Manuel Soto Hidalgo  
Dpto. Arquitectura y Tecnología de Computadores  
Universidad de Granada

jmsoto@ugr.es

# Índice



## [1.Introducción ]

- 2.Configurar la red del sistema web
- 3.El eje principal de la red del sistema
- 4.Configurar una zona segura
- 5.Conectar servidores al front-rail
- 6.Conectar servidores al back-rail
- 7.Resumen de configuraciones
- 8.Conectar la granja web a Internet
- 9.Conectar la granja web a redes seguras
- 10.Resumen y conclusiones

# 1. Introducción

La construcción de una red segura y escalable es fundamental para cualquier servidor.

Si la red no está bien estructurada, los servidores no pueden servir la información.

El administrador/diseñador del sistema debe analizar las opciones de conexión a Internet y diseñar la estructura de red.

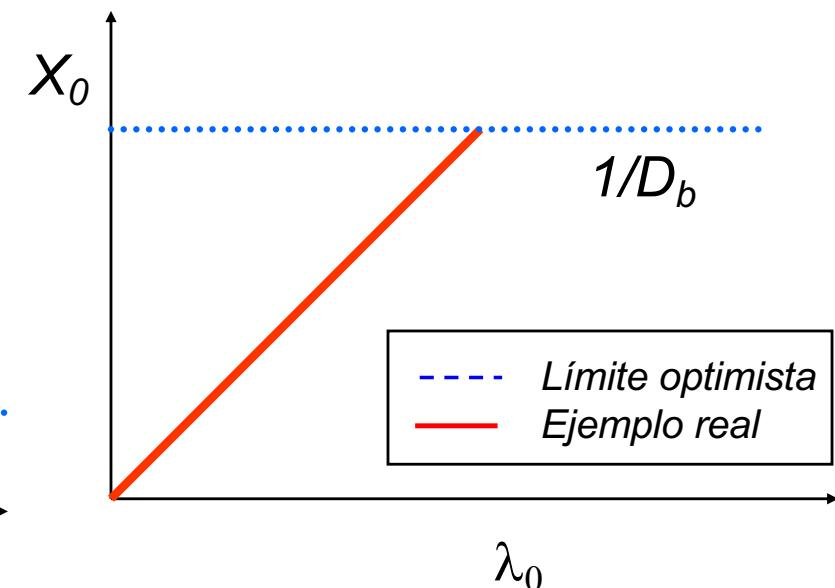
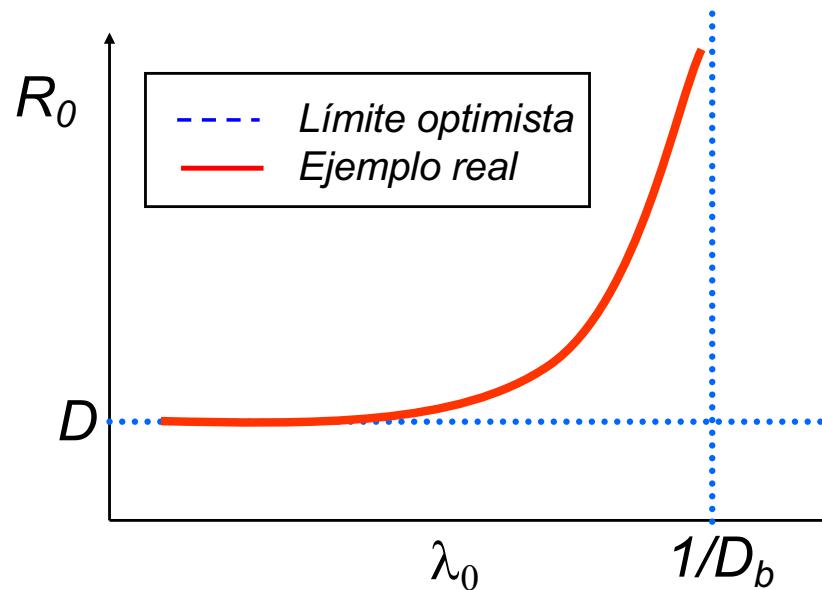
Debe separar las subredes corporativas y también conectar a redes privadas de proveedores.

# 1. Introducción

Hay que decidir el ancho de banda necesario a contratar.

Todas estas decisiones de diseño implican un estudio del hardware y aplicaciones software disponibles:

- switch, hub, router, balanceador, etc.
- sistema operativo, monitorización, balanceo, etc.
- -> Carga del sistema ----  $D = S \times V$



# Índice



1. Introducción

[ 2. Configurar la red del sistema web ]

3. El eje principal de la red del sistema
4. Configurar una zona segura
5. Conectar servidores al front-rail
6. Conectar servidores al back-rail
7. Resumen de configuraciones
8. Conectar la granja web a Internet
9. Conectar la granja web a redes seguras
10. Resumen y conclusiones

## 2. Configurar la red del sistema web

La configuración de la red requiere:

- Elegir el modelo de red más adecuado
- Elegir el hardware (estándar)
- Estructurar la red aislando subredes
- Definir los puntos de entrada a las diferentes subredes

## 2. Configurar la red del sistema web

Conceptos:

- Eje principal (backbone)
- Zona segura (DMZ)
- Front-rail / back-rail
- Redes seguras externas

# Índice



1. Introducción
2. Configurar la red del sistema web
- 3. El eje principal de la red del sistema**
4. Configurar una zona segura
5. Conectar servidores al front-rail
6. Conectar servidores al back-rail
7. Resumen de configuraciones
8. Conectar la granja web a Internet
9. Conectar la granja web a redes seguras
10. Resumen y conclusiones

# 3. El eje principal de la red del sistema

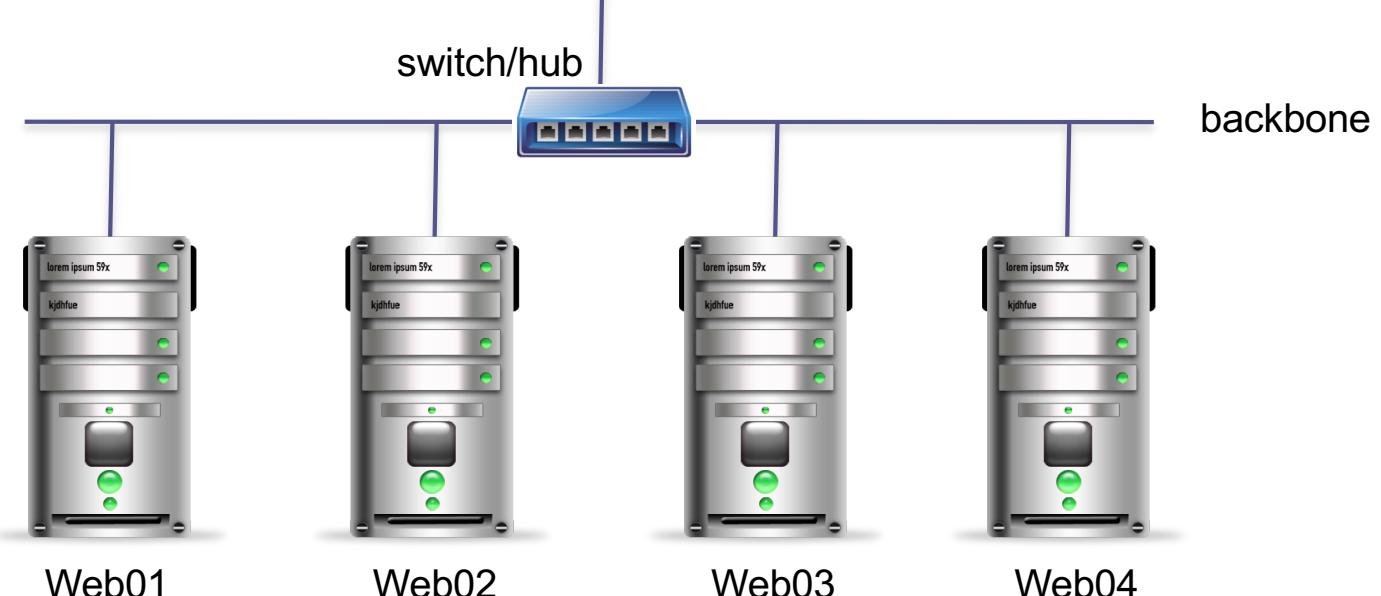
***Backbone:*** eje principal de enlace entre máquinas.

Se puede formar con:

- Switch
- Router
- Hub

<http://computerhoy.com/noticias/internet/cuales-son-diferencias-hub-switch-router-43325>

Gestiona las comunicaciones entre servidores y redes:



# 3. El eje principal de la red del sistema

**hub** (<http://es.wikipedia.org/wiki/Concentrador>)

- Dispositivo sencillo; recibe datos procedentes de un ordenador para transmitirlo a todos los demás que estén conectados.

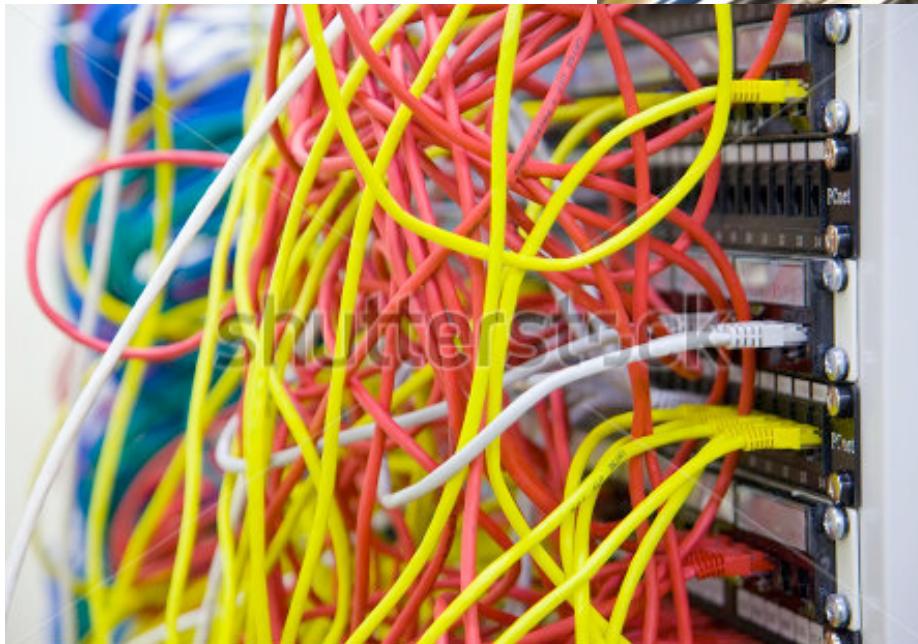
**switch** ([http://es.wikipedia.org/wiki/Comutador\\_\(dispositivo\\_de\\_red\)](http://es.wikipedia.org/wiki/Comutador_(dispositivo_de_red)))

- Además de la funcionalidad del hub, hace que la información proveniente del ordenador de origen se envíe al de destino.

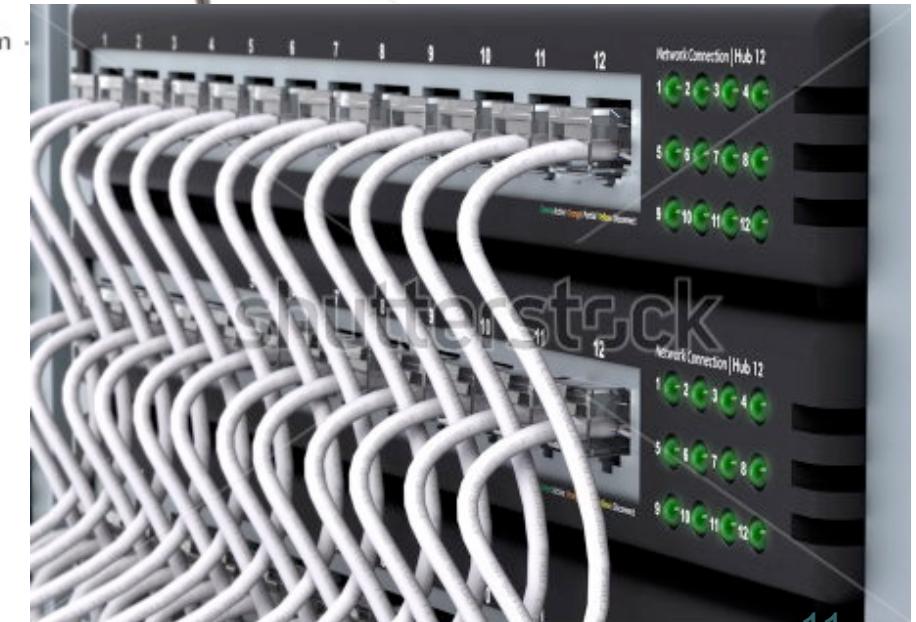
**router** (<http://es.wikipedia.org/wiki/Router>)

- Además de la funcionalidad del switch, interconectan varias redes y tienen la capacidad de escoger la mejor ruta para que los paquetes de datos lleguen a su destino.

# 3. El eje principal de la red del sistema

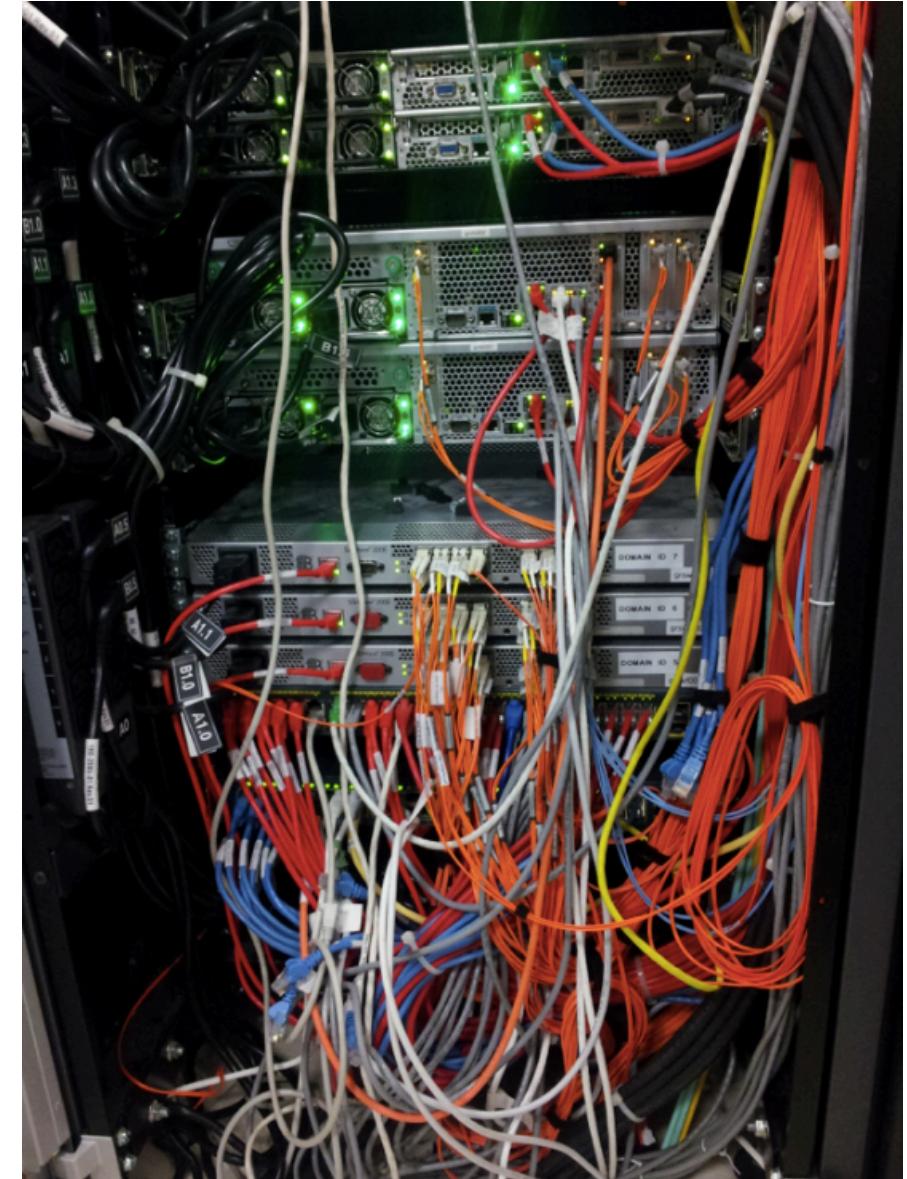
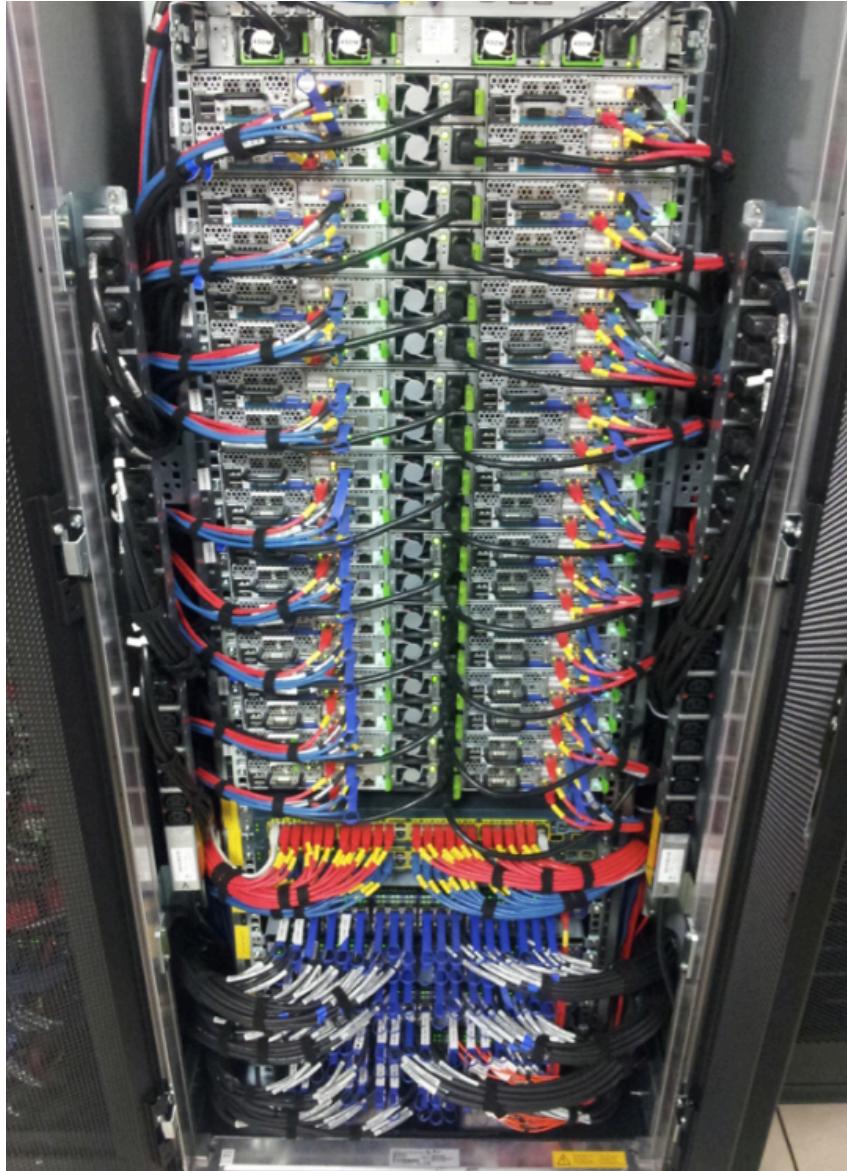


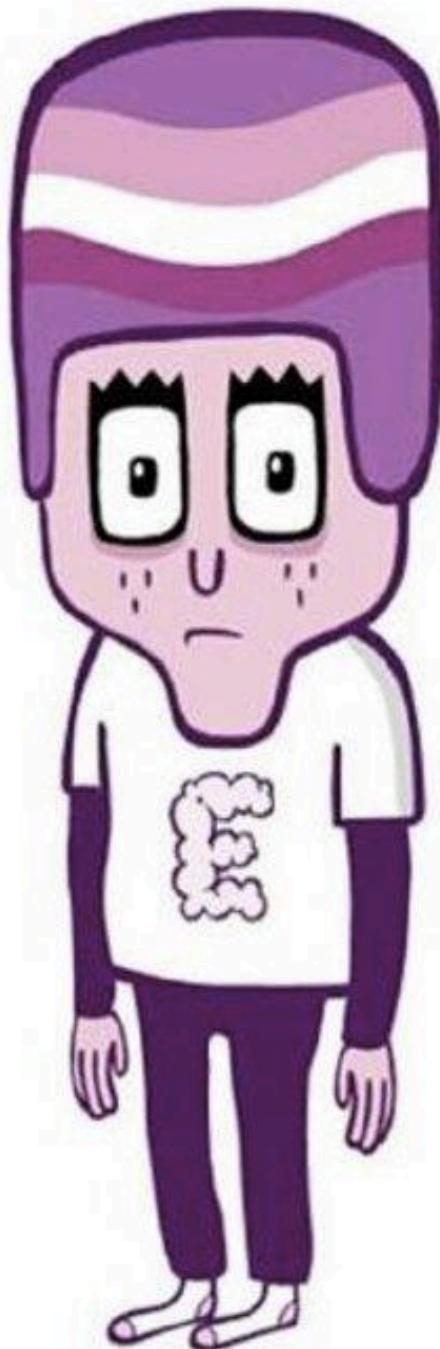
www.shutterstock.com · 70112269



www.shutterstock.com · 62956390

## 3. Los cables...



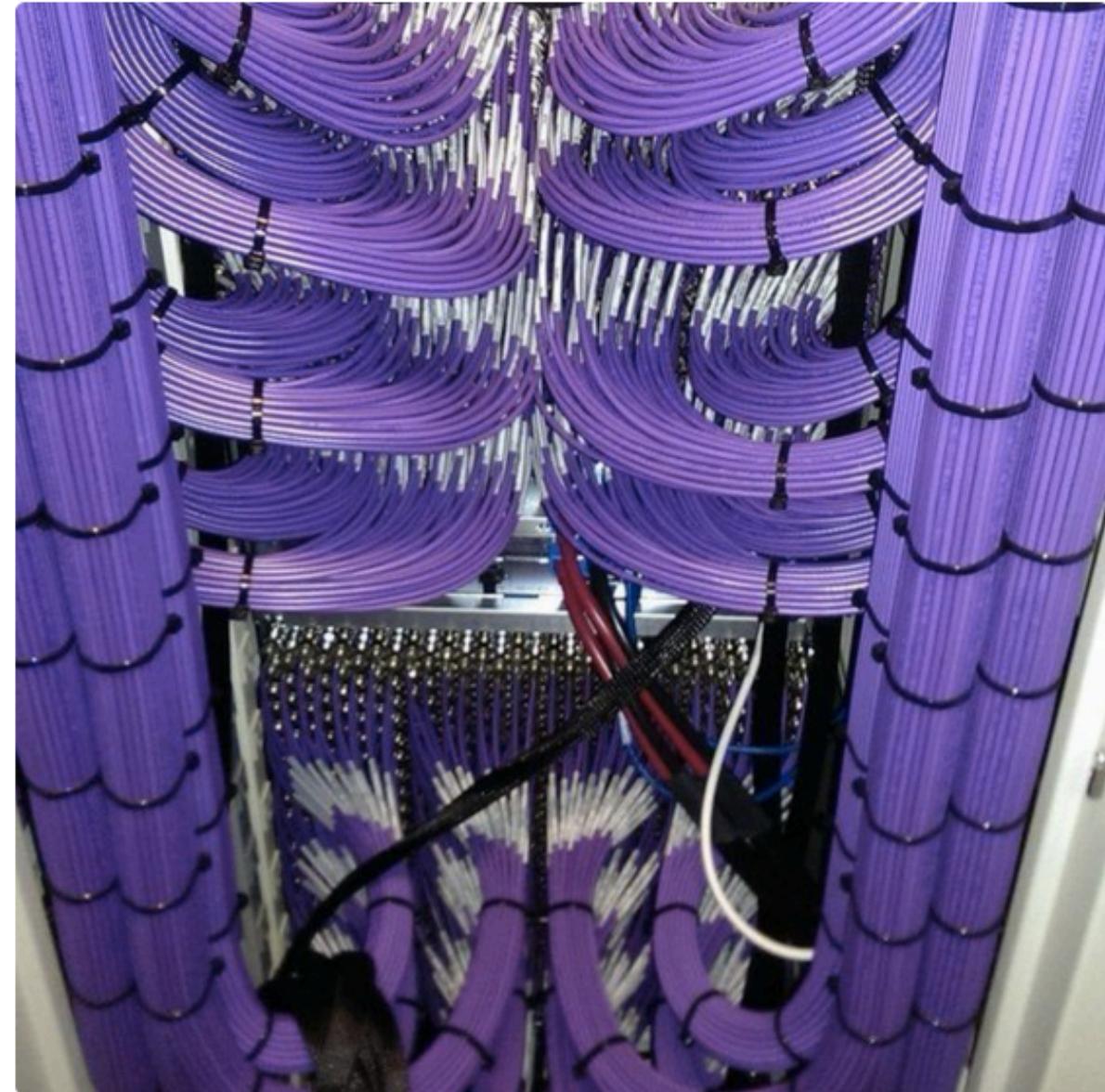


**EnjutoMojamuto**  
@enjutomojamuto



Seguir

Muchos no lo entenderéis pero esto es  
porno para informáticos



# Índice

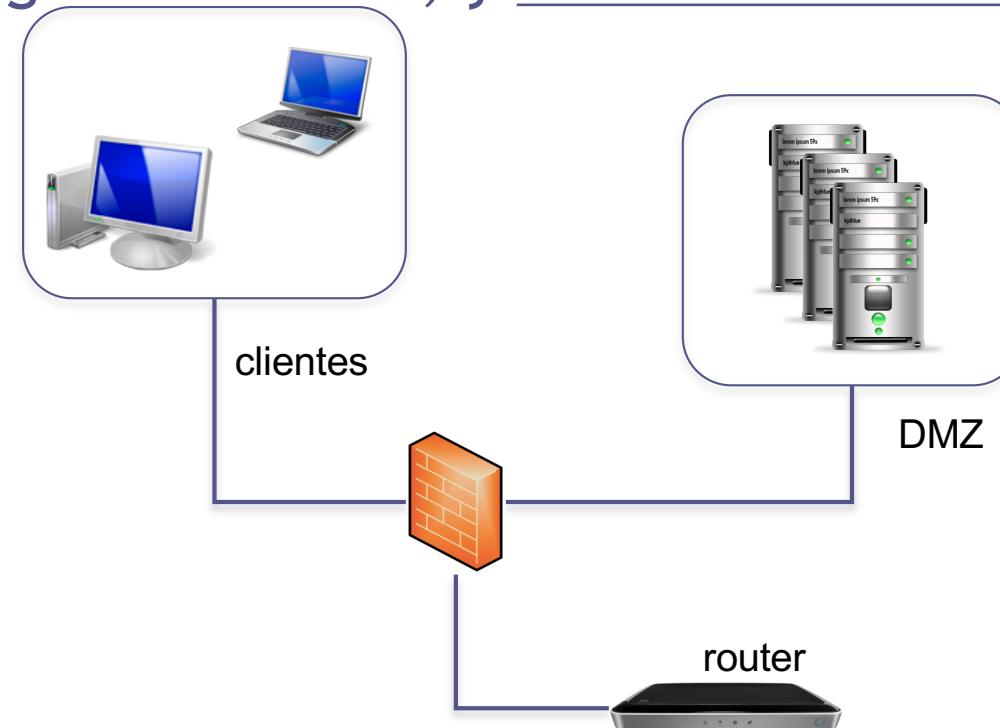


1. Introducción
2. Configurar la red del sistema web
3. El eje principal de la red del sistema
4. Configurar una zona segura - DMZ
5. Conectar servidores al front-rail
6. Conectar servidores al back-rail
7. Resumen de configuraciones
8. Conectar la granja web a Internet
9. Conectar la granja web a redes seguras
10. Resumen y conclusiones

# 4. Configurar una zona segura

Zona desmilitarizada o *DMZ (demilitarized zone)*.

Área restringida o aislada, y totalmente controlada.



[http://es.wikipedia.org/wiki/Zona\\_desmilitarizada\\_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Zona_desmilitarizada_(inform%C3%A1tica))

## 4. Configurar una zona segura

Quedan controlados los servicios y aplicaciones ofrecidos a otras redes externas al DMZ.

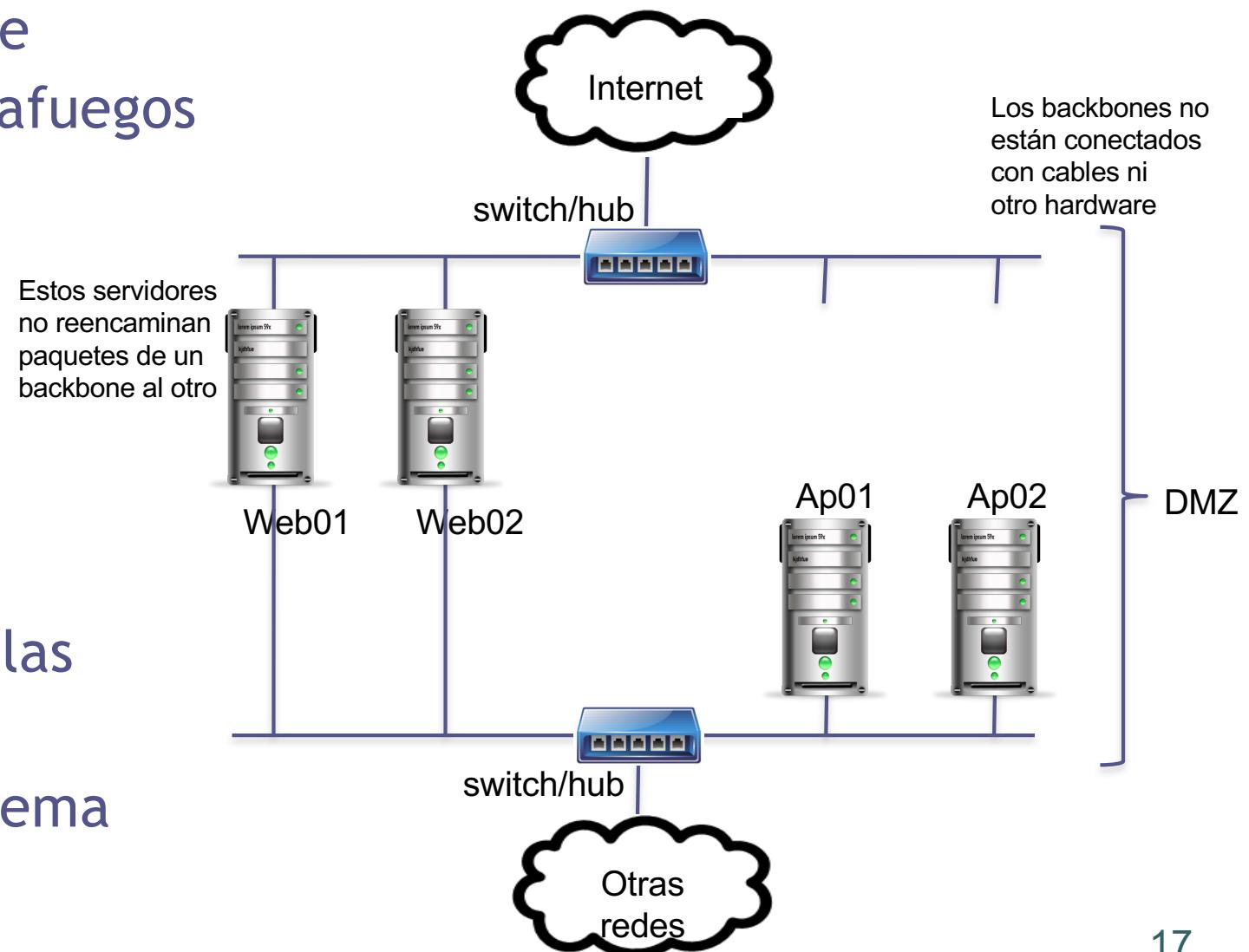
Los servicios de la granja web se ofrecen de forma estándar mediante dirección IP y puerto.

Los cortafuegos, routers y balanceadores de carga restringen el tráfico de entrada o salida.

# 4. Configurar una zona segura

La comunicación entre backbones se hace mediante un cortafuegos o configurando servidores con doble tarjeta de red.

La separación de las redes refuerza la seguridad del sistema



# 4. Configurar una zona segura

Existen varias alternativas para conectar la granja web a otras redes:

1. Configuración sin DMZ
2. Configuración de DMZ simple
3. Configuración de DMZ tradicional
4. Configuración de DMZ doble

# 4. Configurar una zona segura

## *1. Configuración sin DMZ*

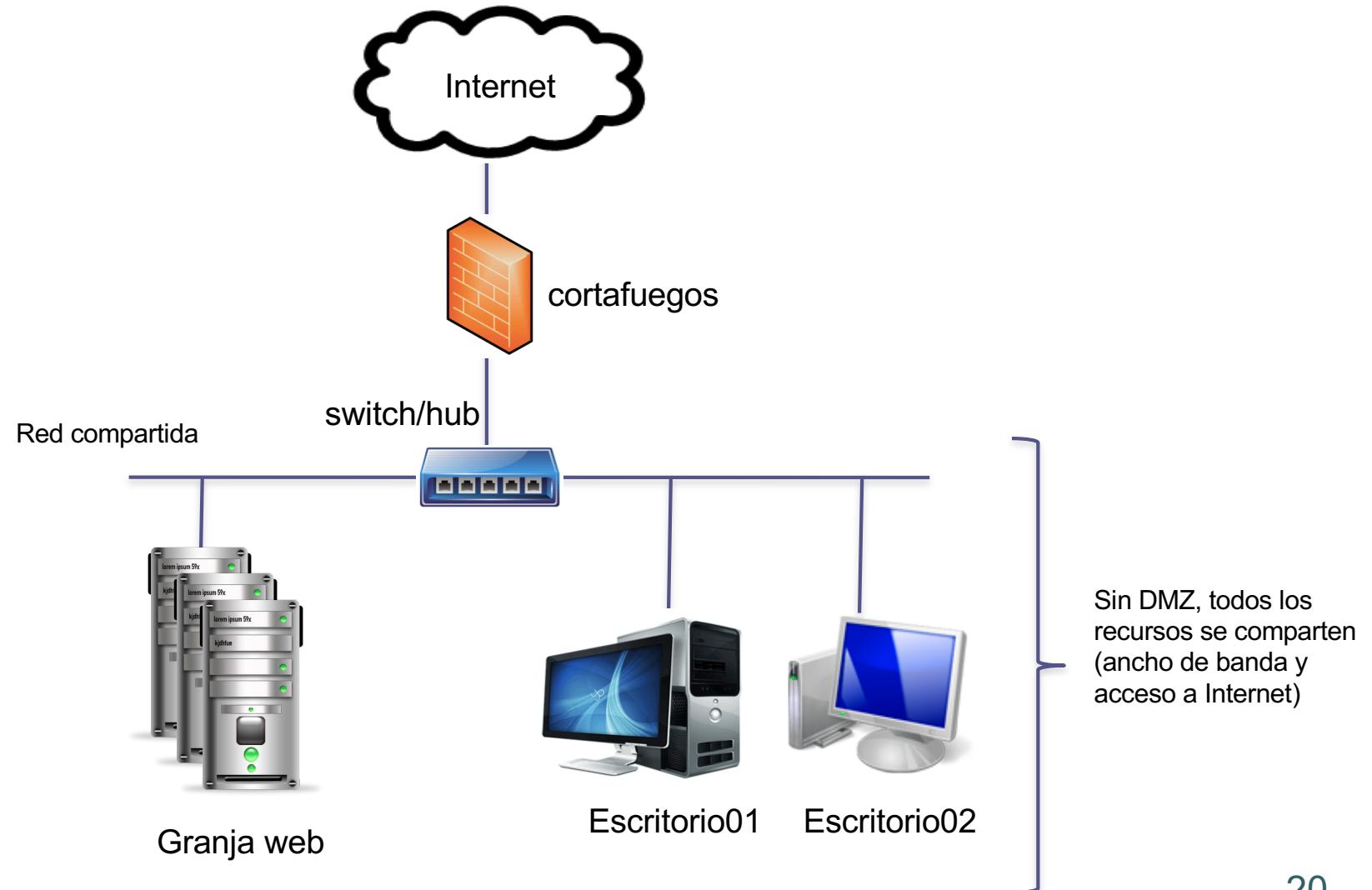
Tanto los **servidores** de la granja web como otras máquinas están conectadas a la **misma subred**.

Se **comparten recursos** (incluso salida a Internet).

Sólo tiene sentido en empresas muy pequeñas donde no hay problemas de prestaciones.

# 4. Configurar una zona segura

## 1. Configuración sin DMZ



# 4. Configurar una zona segura

## *1. Configuración sin DMZ*



### Problemas:

- Compartición del ancho de banda (servidores y máquinas de escritorio).
- Asegurar los servidores es más complicado.
- Si uno de los servidores se ve comprometido, el resto de recursos puede ser atacado.
- Las máquinas de escritorio suponen un problema de seguridad.

# 4. Configurar una zona segura

## 2. *Configuración de DMZ simple*

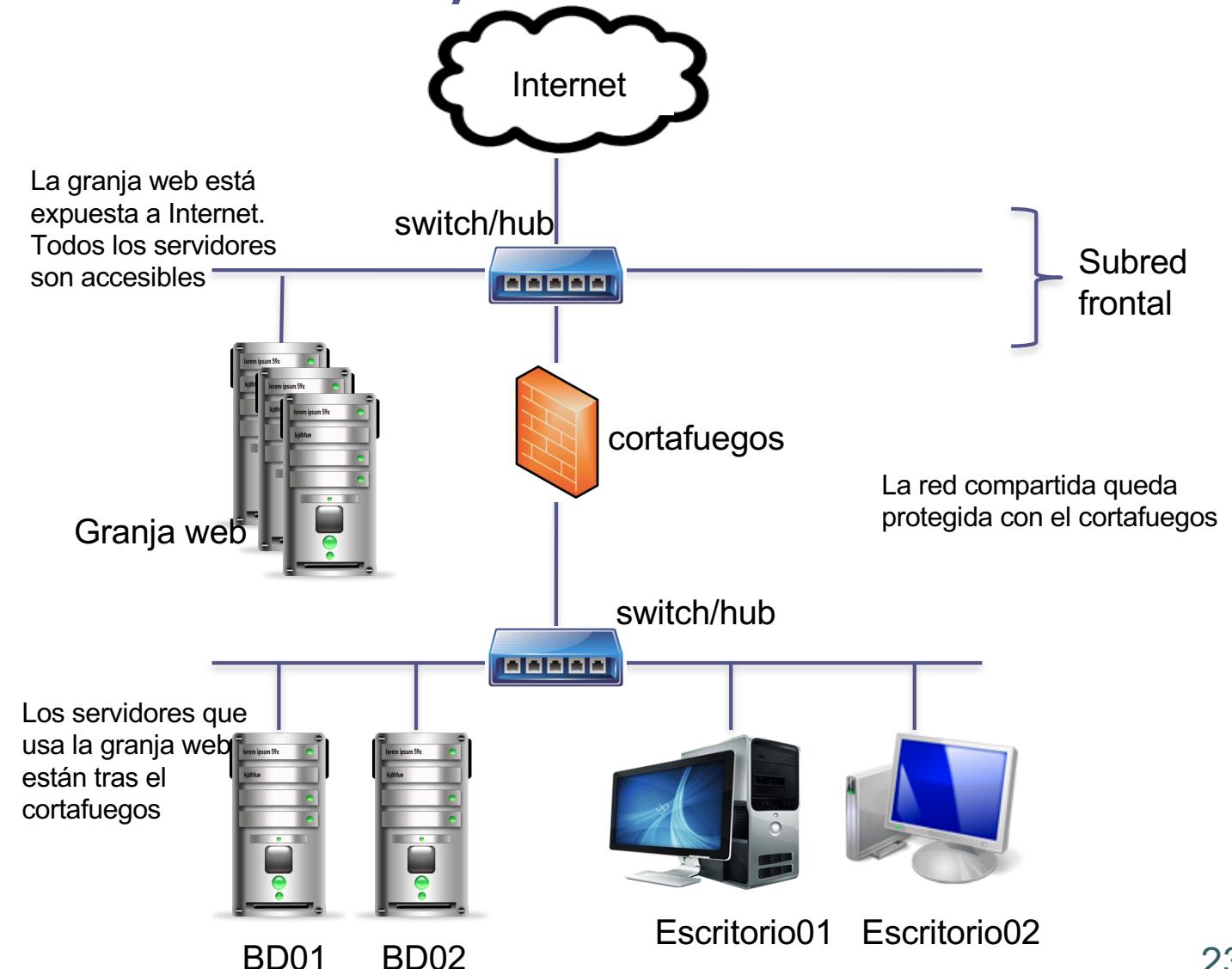
Los **servidores** expuestos deben **aislarse** con un cortafuegos.

Servidores web en una red y servidores de bases de datos o disco en otra, protegida por un cortafuegos

→ Así se protegen los servidores de bases de datos o disco y las máquinas de escritorio.

# 4. Configurar una zona segura

## 2. Configuración de DMZ simple



# 4. Configurar una zona segura

## 2. *Configuración de DMZ simple*



### Problemas:

- Los servidores web están conectados directamente a Internet.
- El cortafuegos puede ser un cuello de botella.
- Los servidores y máquinas tras el cortafuegos aún comparten ancho de banda.
- Las máquinas de escritorio aún están en la misma red que las BD.

# 4. Configurar una zona segura

## *3. Configuración de DMZ tradicional*

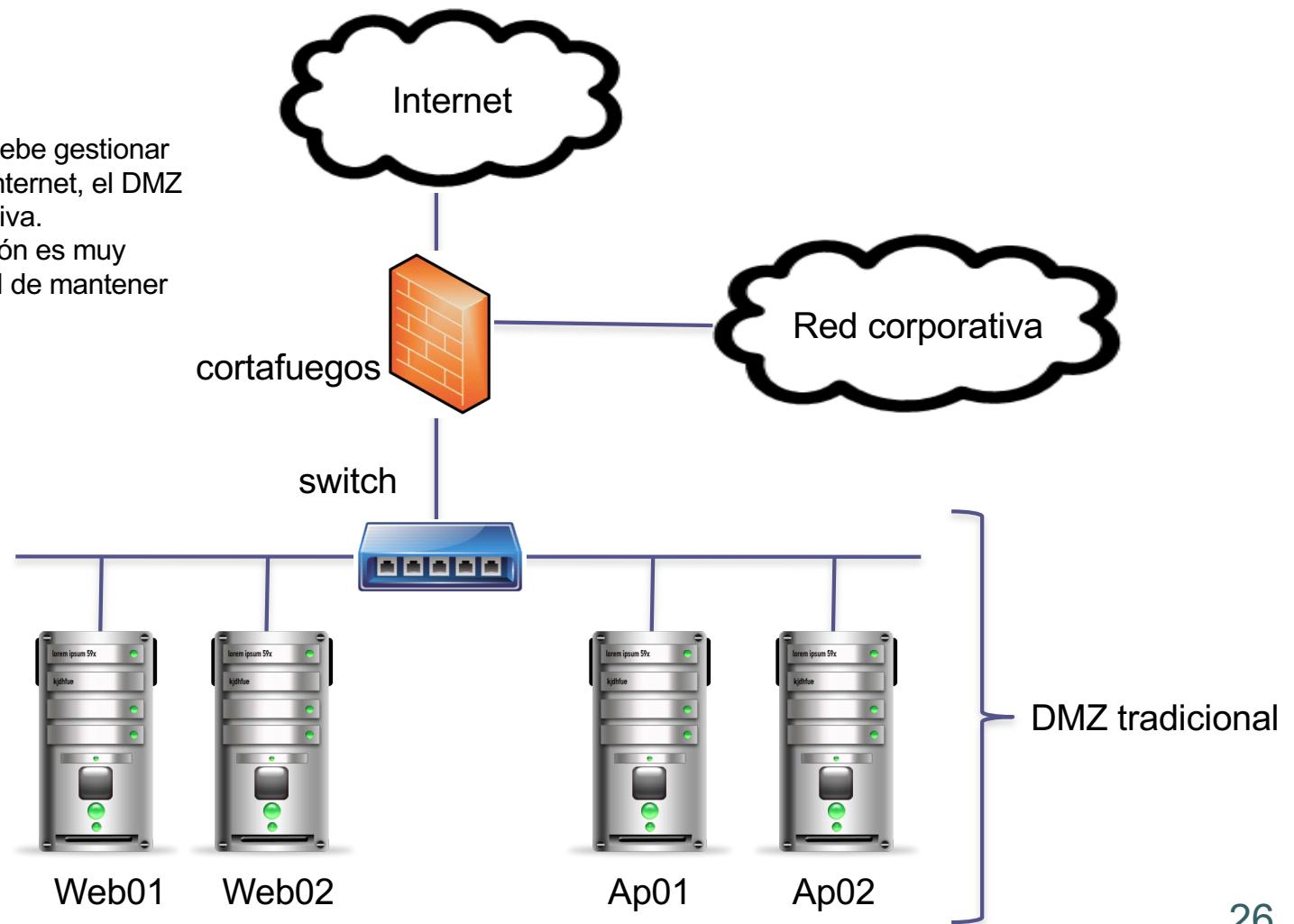
La idea es resolver los problemas de la configuración anterior:

- Evitar ancho de banda de la red corporativa compartido
- Evitar inseguridad de los servidores expuestos

# 4. Configurar una zona segura

## 3. Configuración de DMZ tradicional

El cortafuegos debe gestionar el tráfico entre Internet, el DMZ y la red corporativa.  
Esta configuración es muy compleja y difícil de mantener



# 4. Configurar una zona segura

## *3. Configuración de DMZ tradicional*



### Problemas:

- Dificultad para configurar correctamente el cortafuegos (controlar distintos tipos de redes).
- El cortafuegos es un posible cuello de botella.
- Si la configuración del cortafuegos tiene errores, entonces ¡todo queda expuesto!

# 4. Configurar una zona segura

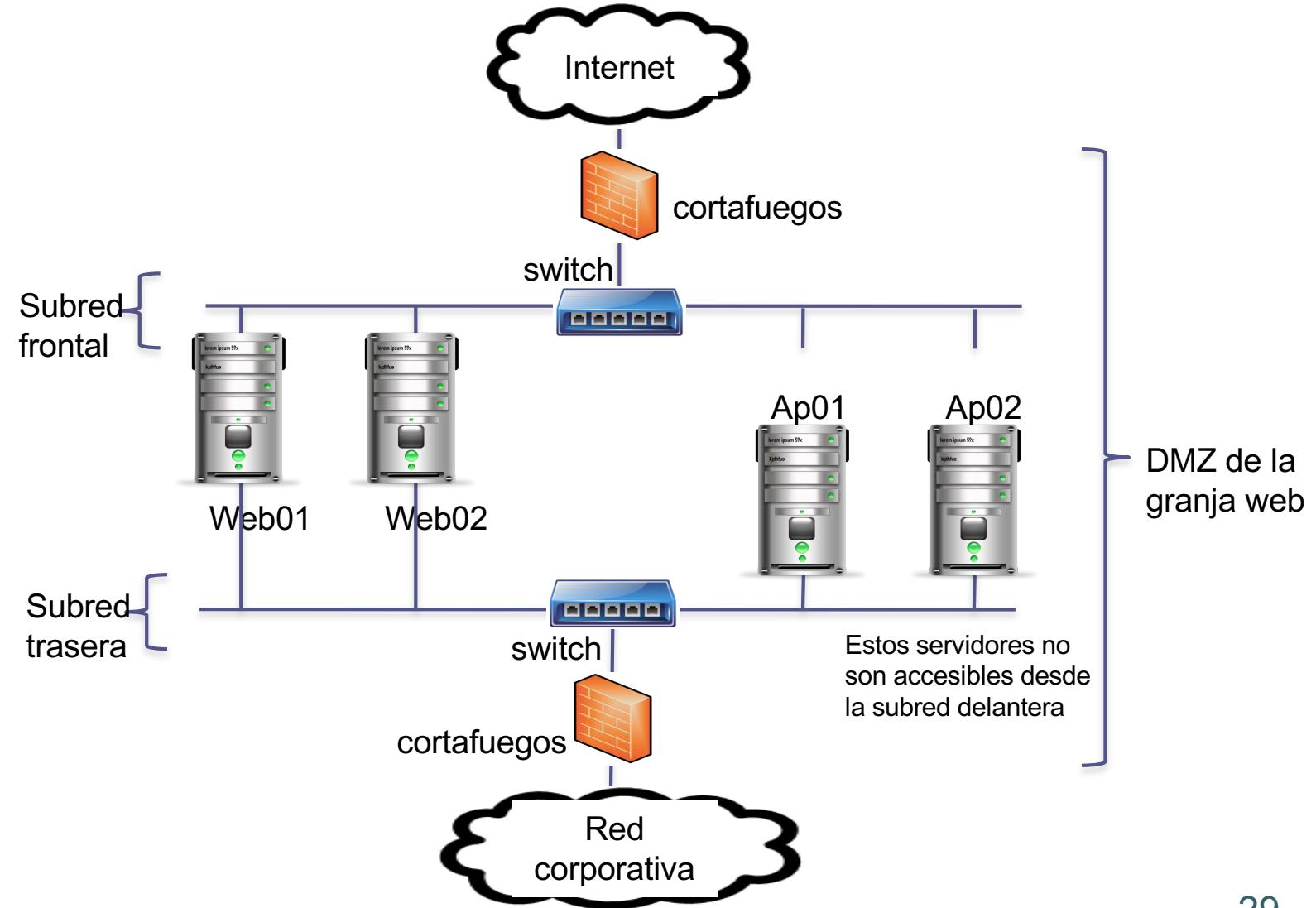
## *4. Configuración de DMZ doble*

- Configuración ideal para una granja web.
- Se basa en **aislar todos los servidores con varios cortafuegos**.



# 4. Configurar una zona segura

## 4. Configuración de DMZ doble



# 4. Configurar una zona segura

## *4. Configuración de DMZ doble*



Es la configuración más segura:

- El DMZ tiene un **front-rail** y un **back-rail**.
- El delantero es un segmento de red conectado a Internet.
- Los servidores quedan protegidos con el cortafuegos.
- El trasero está conectado a la subred interna (segura), y protegido con otro cortafuegos.

# Índice



1. Introducción
2. Configurar la red del sistema web
3. El eje principal de la red del sistema
4. Configurar una zona segura
5. Conectar servidores al front-rail
6. Conectar servidores al back-rail
7. Resumen de configuraciones
8. Conectar la granja web a Internet
9. Conectar la granja web a redes seguras
10. Resumen y conclusiones

# 5. Conectar servidores al front-rail

Los servidores conectados al front-rail deben dar servicios a clientes a través de Internet:

- HTTP, SMTP, POP3, FTP, etc.

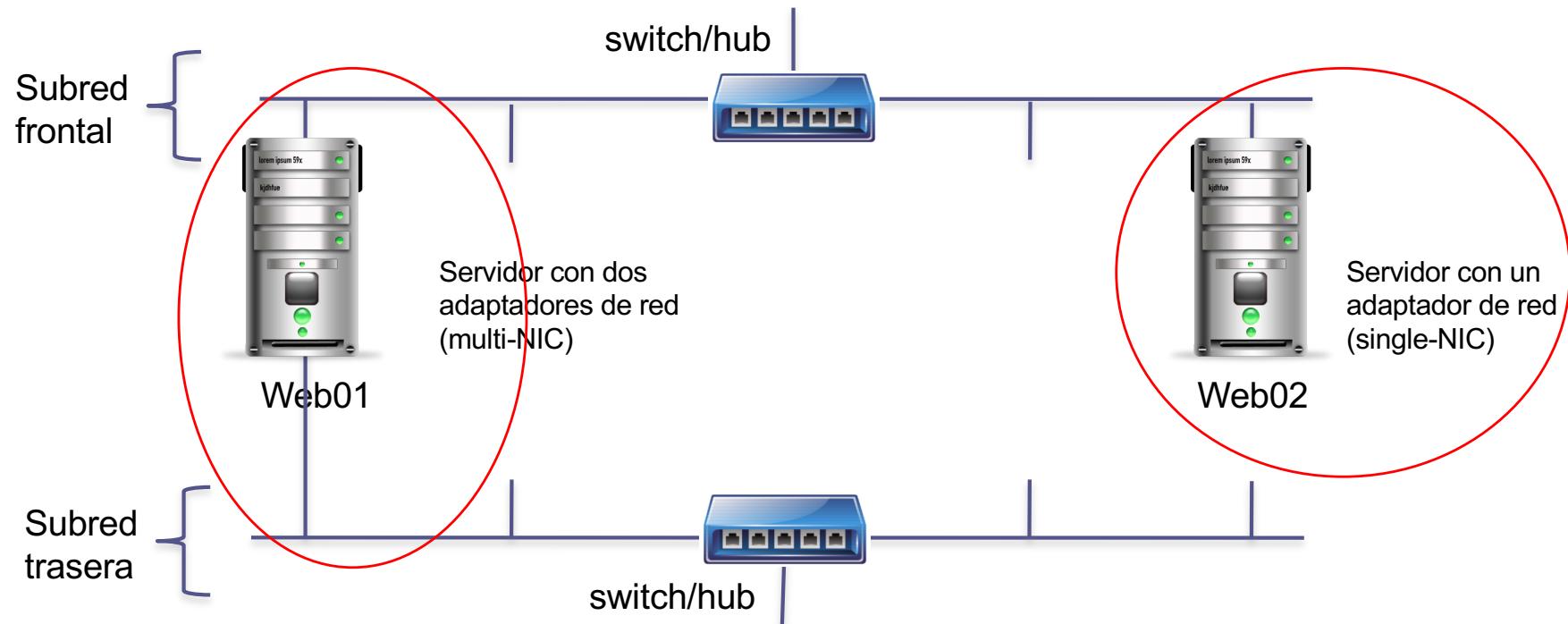
Otros servicios no se ofrecen por el front-rail:

- Bases de datos, terminal

Hay dos tipos de servidores:

- *Single-NIC*: conectado sólo a la subred frontal; aislado de la trasera
- *Multi-NIC*: conectado a la frontal y la trasera

# 5. Conectar servidores al front-rail



## 5. Conectar servidores al front-rail

Un servidor multi-NIC puede acceder a la subred trasera para consumir un recurso.

Su configuración requiere de reglas específicas en la tabla de enrutamiento para encaminar el tráfico hacia la subred trasera.

Hay que ser cuidadosos al establecer las reglas para no dejar caminos que comprometan la seguridad.

# Índice



1. Introducción
2. Configurar la red del sistema web
3. El eje principal de la red del sistema
4. Configurar una zona segura
5. Conectar servidores al front-rail
6. Conectar servidores al back-rail
7. Resumen de configuraciones
8. Conectar la granja web a Internet
9. Conectar la granja web a redes seguras
10. Resumen y conclusiones

## 6. Conectar servidores al back-rail

Los servidores conectados a la subred trasera son accesibles

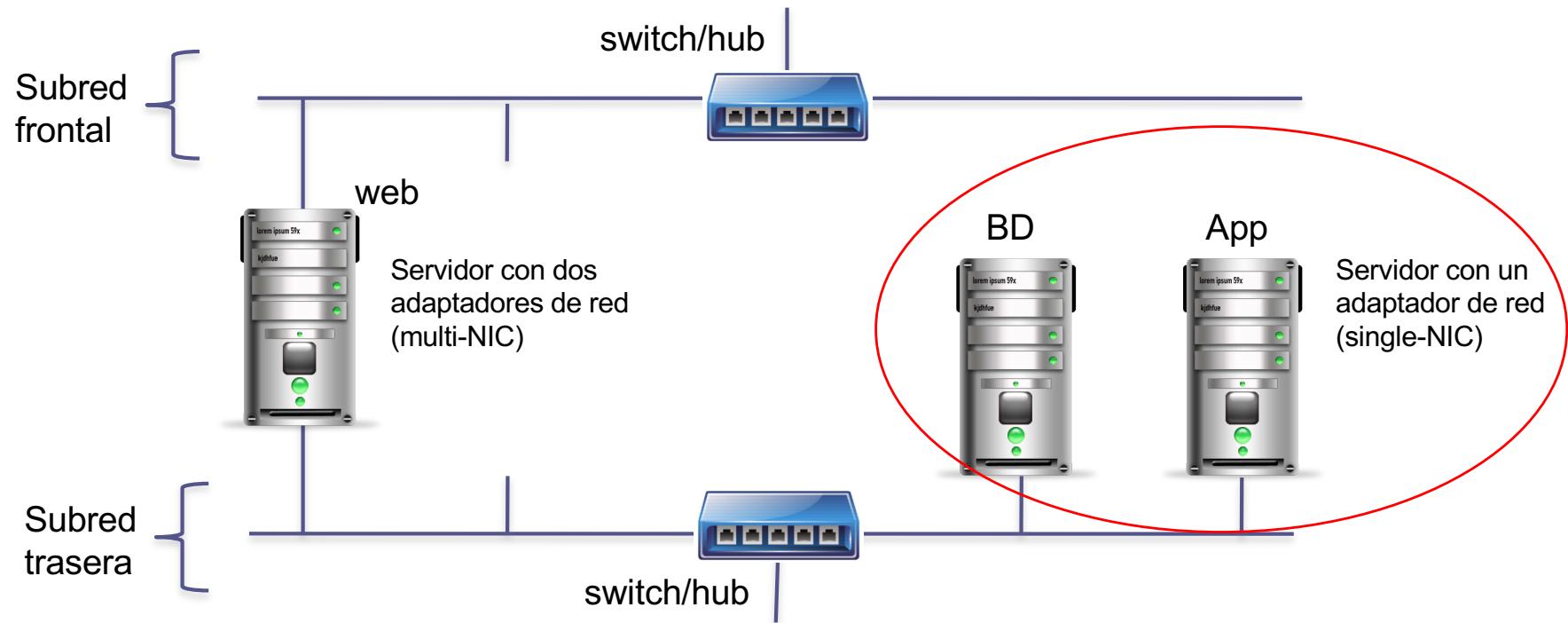
- desde subredes seguras y controladas
- o bien desde servidores con multi-NIC

La subred trasera no debe conectarse directamente a Internet.

Se pueden conectar servidores single-NIC para servir aplicaciones, BD o disco.

El cortafuegos protege los servidores. Sus reglas deben dejar acceso a ciertas aplicaciones y servicios según tipo de usuario.

# 6. Conectar servidores al back-rail



# Índice



1. Introducción
2. Configurar la red del sistema web
3. El eje principal de la red del sistema
4. Configurar una zona segura
5. Conectar servidores al front-rail
6. Conectar servidores al back-rail
- [ 7. Resumen de configuraciones ]**
8. Conectar la granja web a Internet
9. Conectar la granja web a redes seguras
10. Resumen y conclusiones

# 7. Resumen de configuraciones

Resumen de las configuraciones de los servidores y a qué subred pueden acceder:

## (1) Doble conexión al front-rail y back-rail:

- Requiere doble tarjeta de red
- Adecuado para acceder a Internet y servidores internos
- Configuración para servidores HTTP, SMTP, POP3, FTP, etc
- Ofrecen servicios hacia Internet y a las subredes seguras

# 7. Resumen de configuraciones

Resumen de las configuraciones de los servidores y a qué subred pueden acceder:

## (2) Conexión sólo al front-rail:

- Requiere sólo una tarjeta de red
- Adecuado para acceder sólo a Internet
- Los servicios ofrecidos quedan aislados
- Configuración para servidores HTTP, SMTP, POP3, FTP, etc
- Ofrecen servicios hacia Internet

# 7. Resumen de configuraciones

Resumen de las configuraciones de los servidores y a qué subred pueden acceder:

## (3) Conexión sólo al back-rail:

- Requiere sólo una tarjeta de red
- Para servidores que no necesitan acceso a Internet
- Servicios ofrecidos a las redes corporativas/seguras
- Configuración para servidores de BD o aplicaciones

# Índice



1. Introducción
2. Configurar la red del sistema web
3. El eje principal de la red del sistema
4. Configurar una zona segura
5. Conectar servidores al front-rail
6. Conectar servidores al back-rail
7. Resumen de configuraciones
- 8. Conectar la granja web a Internet**
9. Conectar la granja web a redes seguras
10. Resumen y conclusiones

# 8. Conectar la granja web a Internet

La conexión a Internet depende de varios factores para asegurar la calidad del servicio y la seguridad:

- 1. Calidad del servicio y ancho de banda**
- 2. Filtrado y bloqueo de paquetes**
- 3. Network address translation (NAT)**

# 8. Conectar la granja web a Internet

## *1. Calidad de servicio y ancho de banda*

La **calidad del servicio** está directamente relacionada con el ancho de banda para salir a Internet.

**Ancho de banda:** cantidad de información que puede fluir por una conexión de red en un período determinado.  
Se mide en bits por segundo (kbps, Mbps, Gbps, Tbps).

Es adecuado definir qué porcentaje del ancho de banda se reserva para cada tipo de tráfico (HTTP, SSL, FTP...)

Los routers actuales permiten establecer esos parámetros.

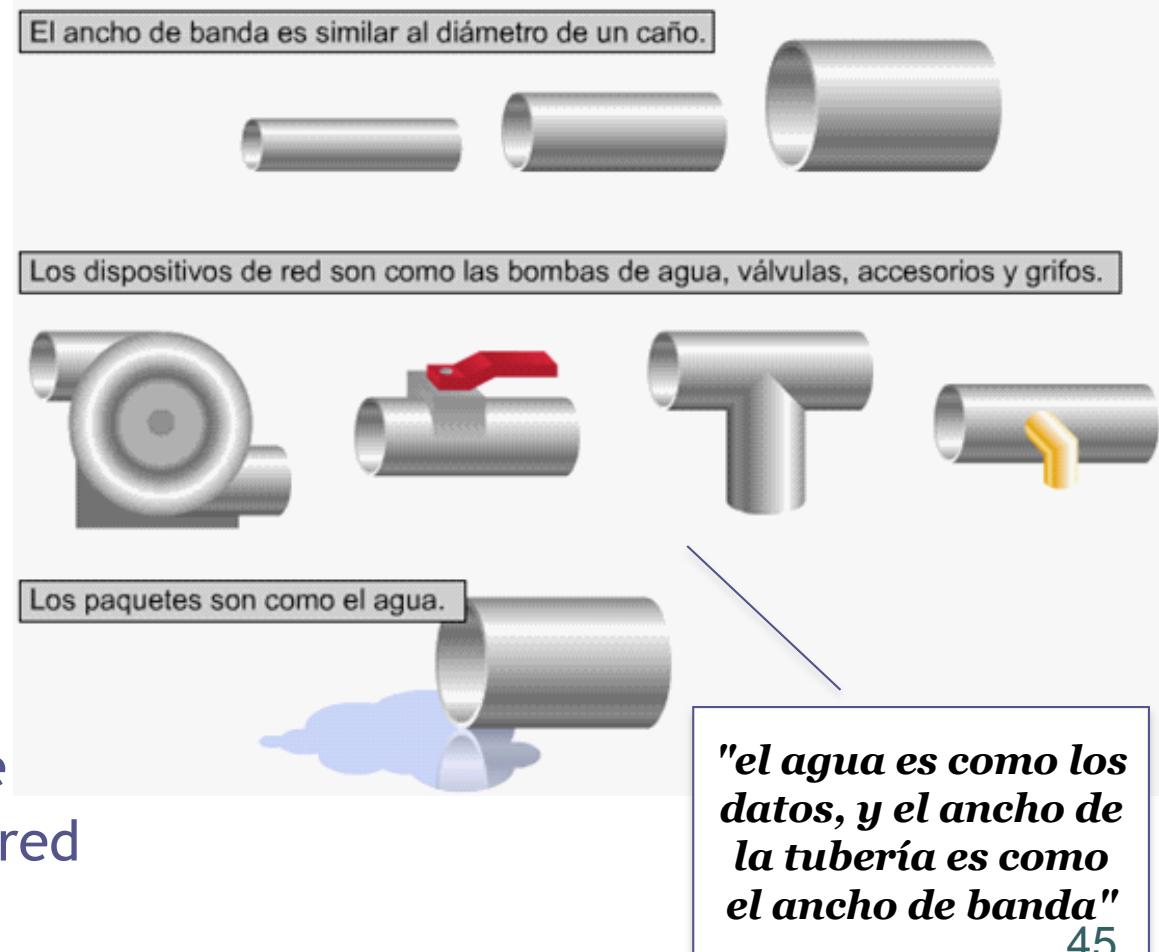
# 8. Conectar la granja web a Internet

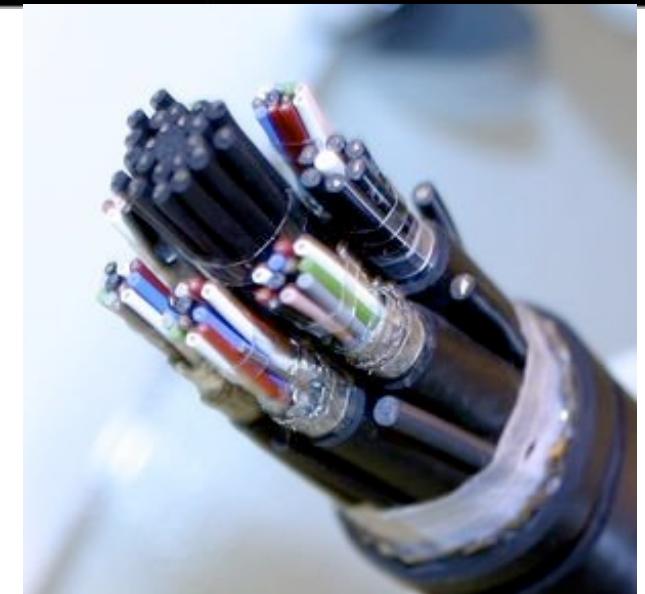
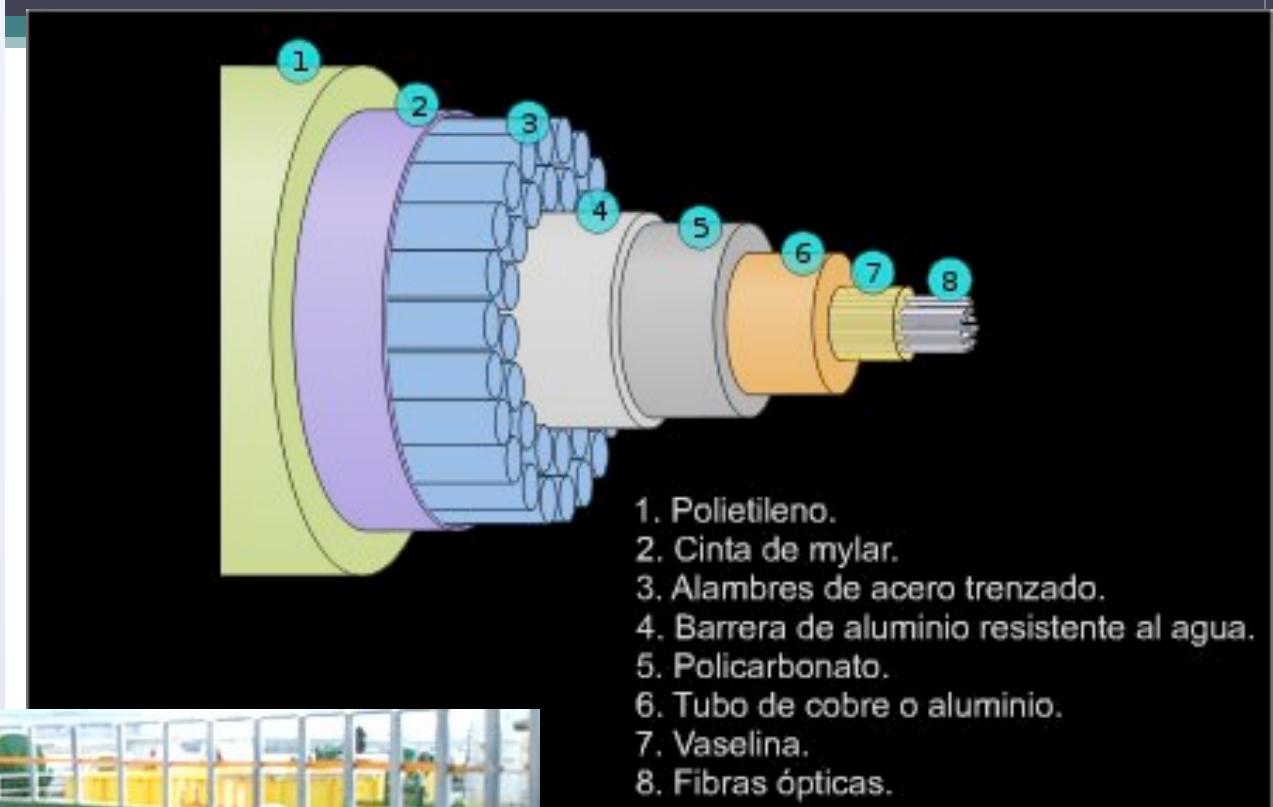
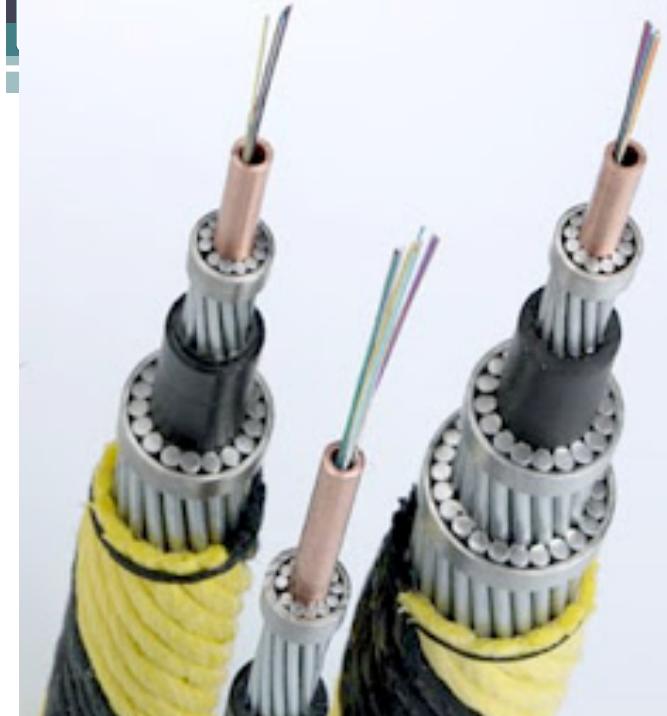
## 1. *Calidad de servicio y ancho de banda*

<http://www.monografias.com/trabajos30/redes-de-datos/redes-de-datos.shtml>

Se suele expresar en Kbps, Mbsp, Gbps y Tbps.

Queda determinado por los métodos de señalización, las tarjetas de red y los demás equipos de red



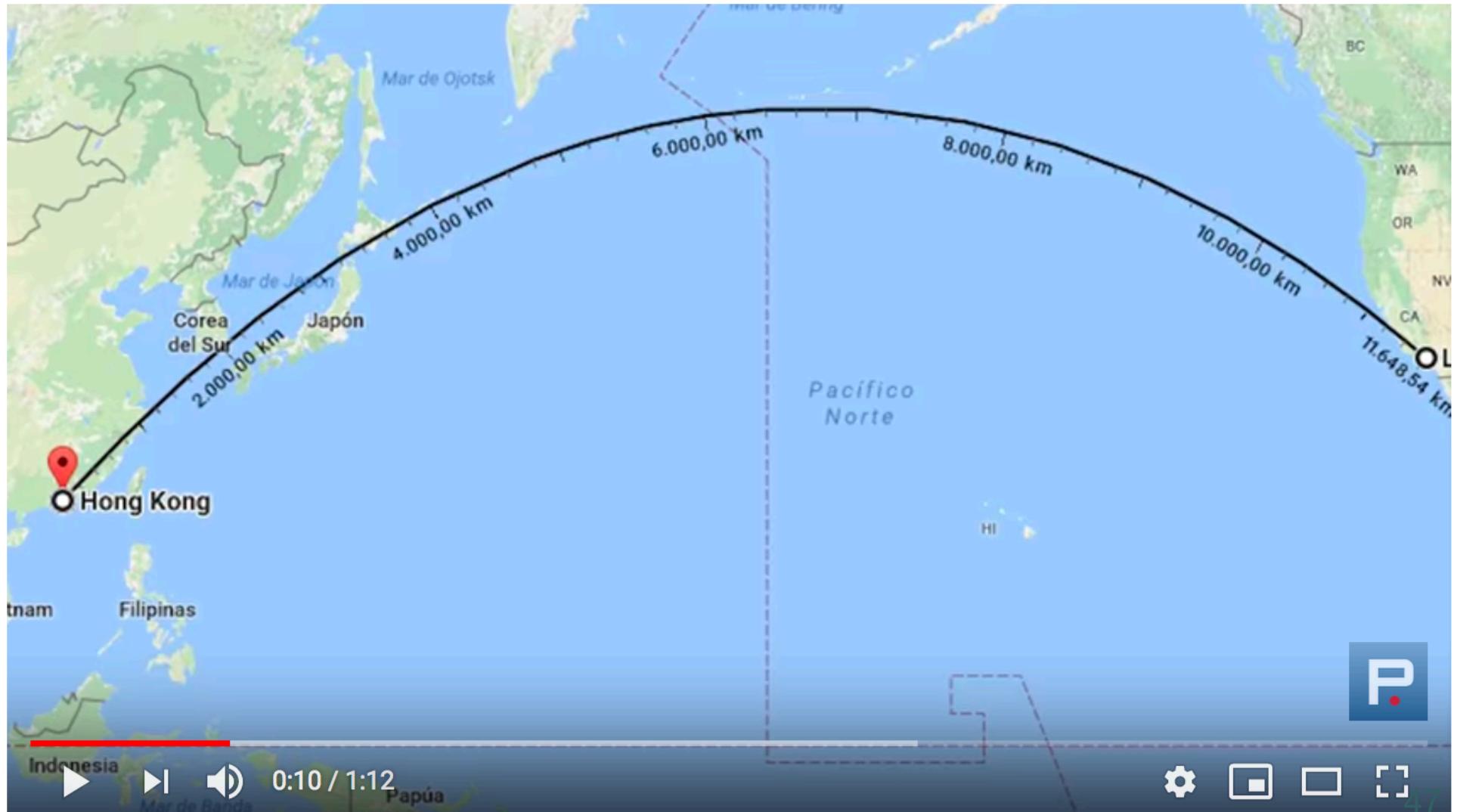


<http://almadeherrero.blogspot.com.es/2008/11/cables-submarinos.html>

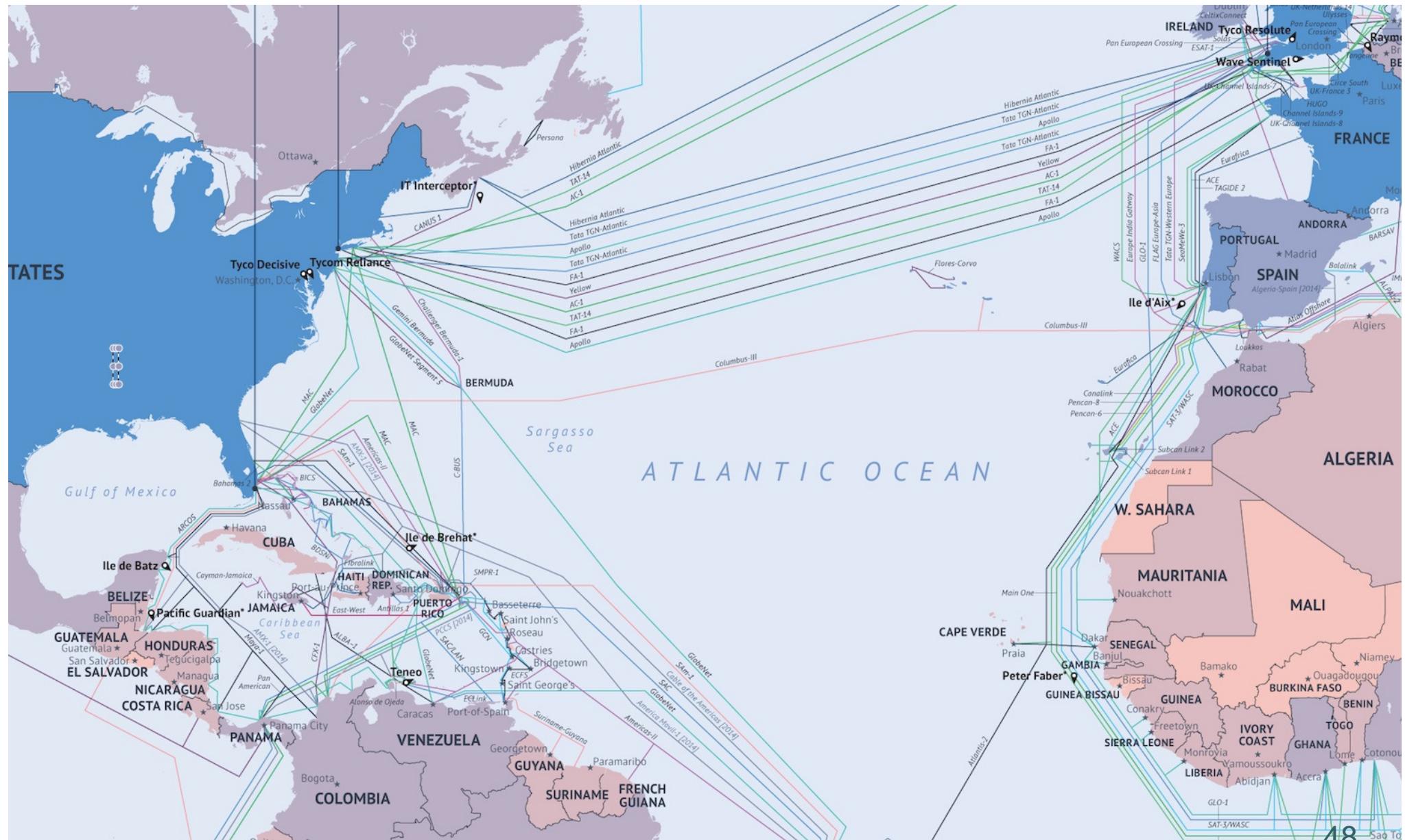
<http://enbytes.com/site/2012/12/06/alcatel-lucent-tiende-cable-submarino-para-consorcio-de-operadores/>

# Cables submarinos

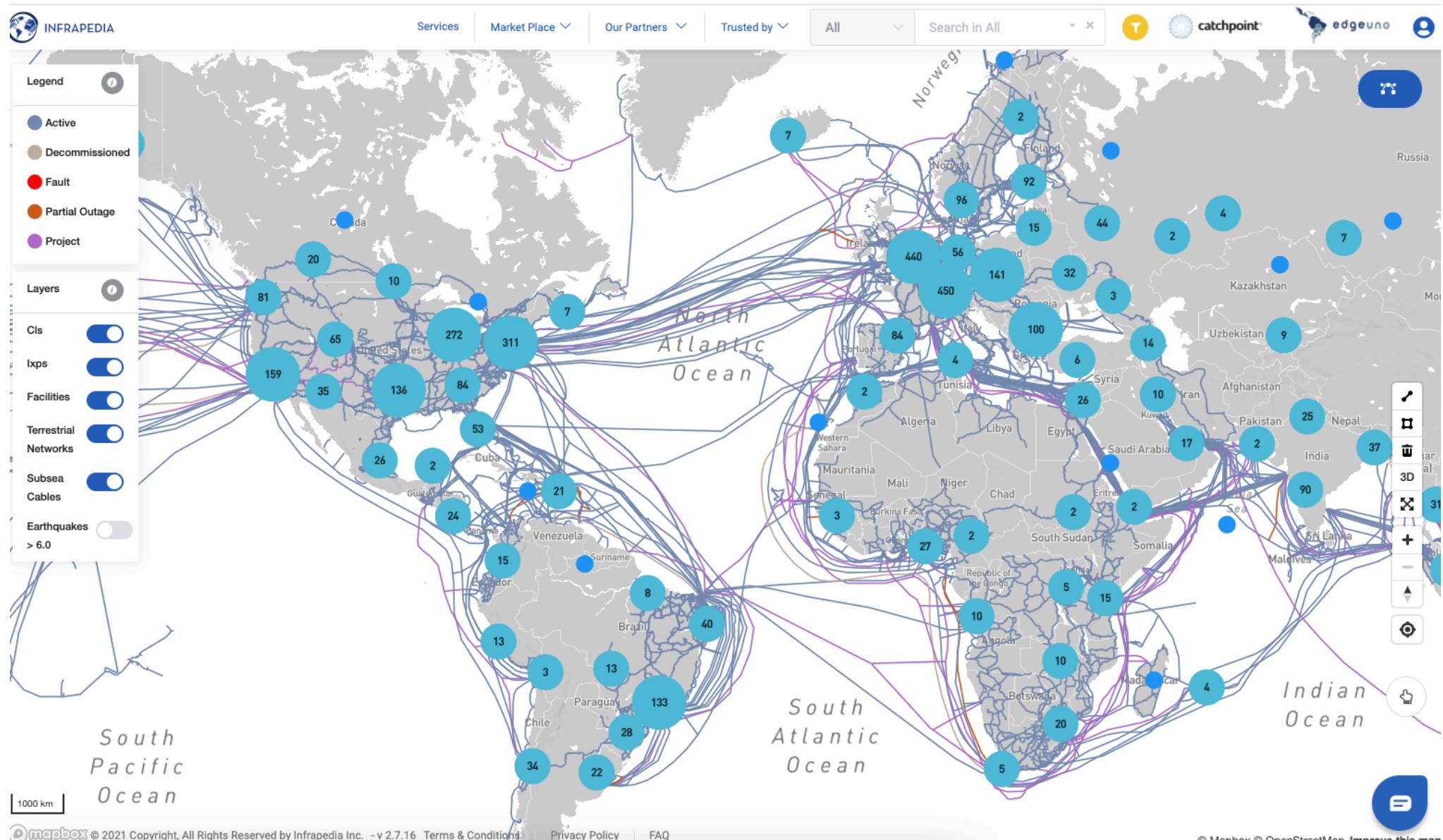
<https://www.youtube.com/watch?v=LMfSIoN7Qf8>



<http://alt1040.com/2014/03/mapa-cables-submarinos-2014>



# <https://www.infrapedia.com/>





MUNDOREAL™

## La Guerra Fría en 2015: del temor a la bomba atómica al temor a la motosierra

POR **NACHO PALOU** — 26 DE OCTUBRE DE 2015

En *The New York Times*, **Russian Ships Near Data Cables Are Too Close for U.S. Comfort**,



Que haya submarinos y buques situados cerca de los cables submarinos de telecomunicaciones e Internet supone una preocupación para militares y agentes de inteligencia

Líder mundial en español | Miércoles 06/04/2011. Actualizado 22:37h.

SUCESO | Buscaba chatarra

## Detenida una mujer de 75 años tras dejar a Armenia sin conexión a Internet

Afp | Tbilisi

Actualizado miércoles 06/04/2011 22:37 horas

[a-](#) [a+](#)

Una georgiana de 75 años de edad ha sido detenida por cortar la conexión de Internet de la totalidad de Armenia.

El pasado 28 de marzo la jubilada descubrió el cable de fibra óptica que suministra la conexión web entre Georgia y Armenia mientras buscaba chatarra. Reconociendo el valor del cobre contenido en el cable, la anciana decidió cortar y robarlo, y al hacerlo interrumpió el servicio de miles de usuarios en el país vecino.

# Cables submarinos

<https://computerhoy.com/noticias/internet/cables-submarinos-google-atacados-tiburones-17231>



# 8. Conectar la granja web a Internet

La conexión a Internet depende de varios factores para asegurar la calidad del servicio y la seguridad:

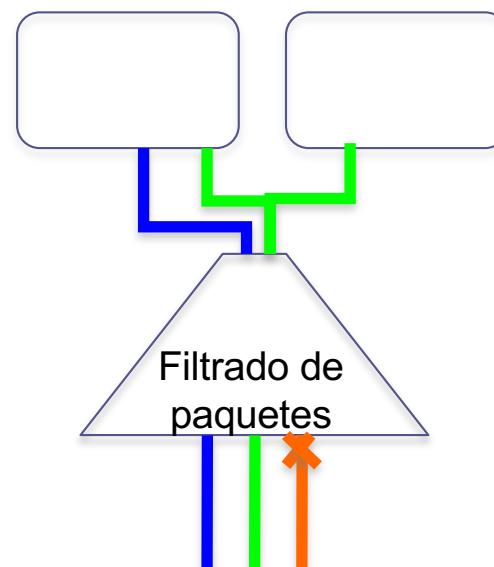
1. Calidad del servicio y ancho de banda
2. **Filtrado y bloqueo de paquetes**
3. Network address translation (NAT)

# 8. Conectar la granja web a Internet

## 2. *Filtrado y bloqueo de paquetes*

Conviene establecer filtros de forma que sólo le llegue a una máquina el tráfico que debe llegarle.

Otros tipos de tráfico se bloquearán para que no le lleguen.  
Aunque los ignorase, los paquetes sobrecargan la red.

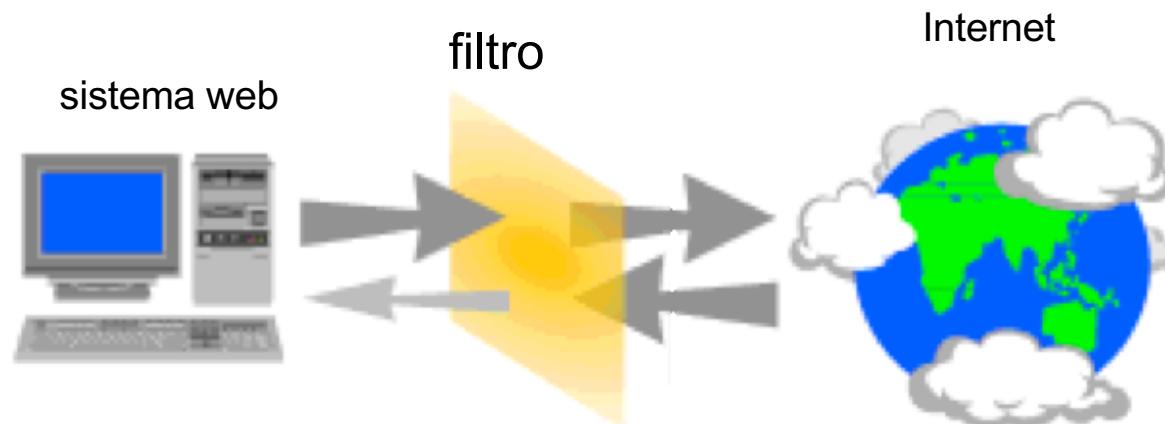


# 8. Conectar la granja web a Internet

## 2. *Filtrado y bloqueo de paquetes*

Los paquetes contienen información de IP origen, IP destino y puerto (servicio), por lo que esta información se usará para el filtrado.

Se pueden usar *cortafuegos*, routers o concentradores.

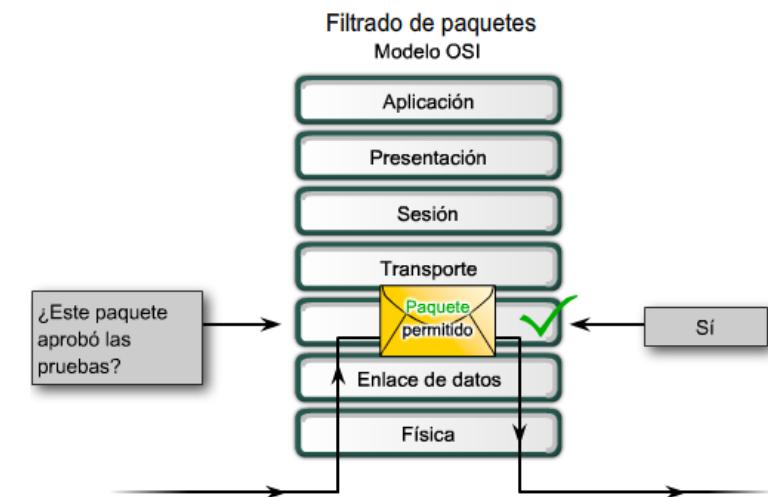


# 8. Conectar la granja web a Internet

## 2. *Filtrado y bloqueo de paquetes*

**Routers.** Un router de filtrado de paquetes utiliza reglas para determinar la autorización o denegación del tráfico según:

- Dirección IP de origen
- Dirección IP de destino
- Tipo de mensaje ICMP
- Puerto TCP/UDP de origen
- Puerto TCP/UDP de destino



“Una ACL es una lista secuencial de sentencias de permiso o denegación que se aplican a direcciones IP o protocolos de capa superior”

# 8. Conectar la granja web a Internet

## *2. Filtrado y bloqueo de paquetes*

### Routers. ¿Cómo funcionan las ACL?

Las ACL no actúan sobre paquetes que se originan en el mismo router.

Las ACL se configuran para ser aplicadas al tráfico entrante o saliente.

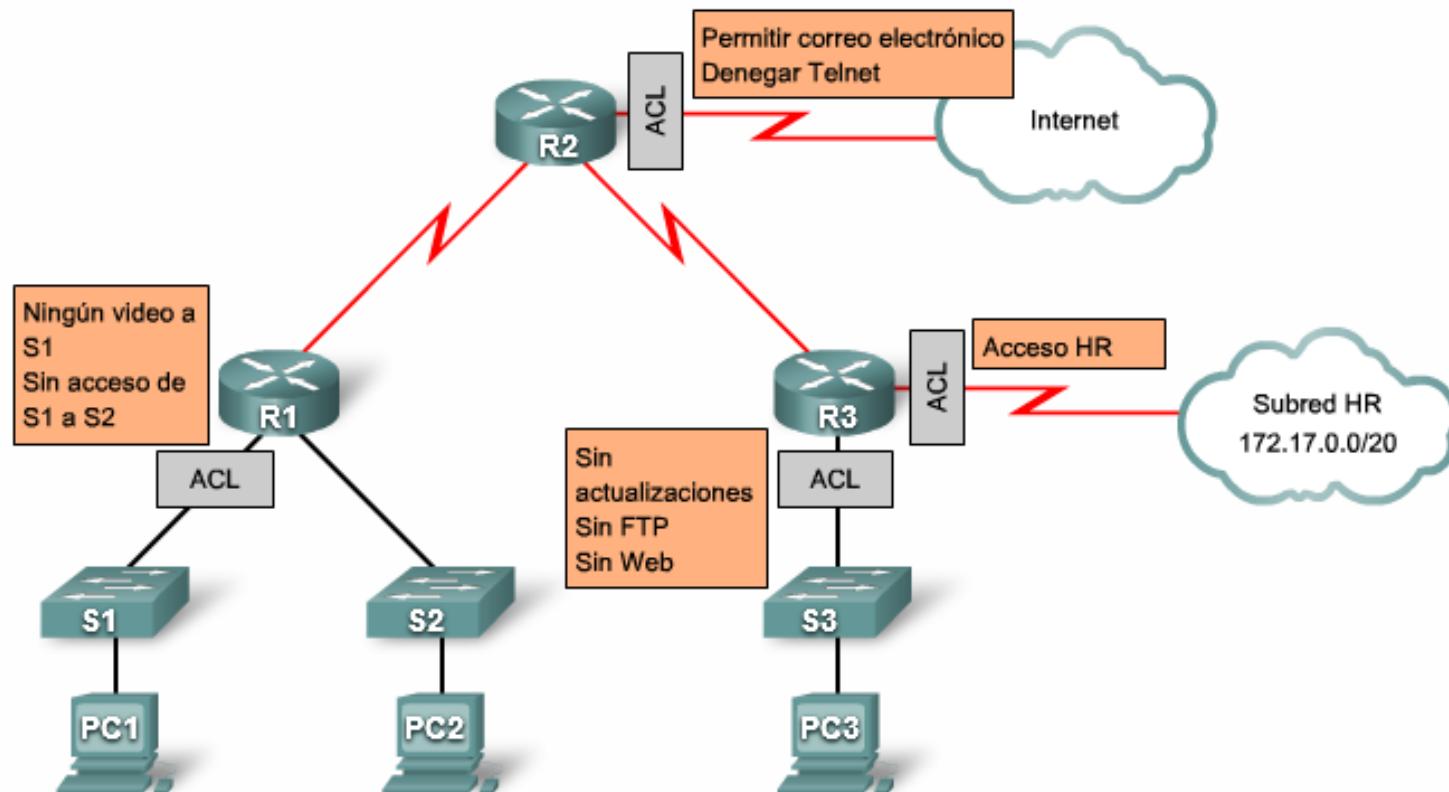
Las sentencias de la ACL operan en orden secuencial.

Una sentencia implícita final cubre todos los paquetes para los cuales las condiciones no resultan verdaderas (implicit deny any statement/deny all traffic).

# 8. Conectar la granja web a Internet

## 2. Filtrado y bloqueo de paquetes

Routers. ¿Cómo funcionan las ACL?



## 8. Conectar la granja web a Internet

La conexión a Internet depende de varios factores para asegurar la calidad del servicio y la seguridad:

1. Calidad del servicio y ancho de banda
2. Filtrado y bloqueo de paquetes
3. Network address translation (NAT)

# 8. Conectar la granja web a Internet

## *3. NAT: Network Address Translation*

Con NAT mapeamos una dirección pública a una dirección privada de una de las máquinas servidoras internas.

Mejora la seguridad: se ocultan las verdaderas IP de los servidores últimos (back-end).

Esto lo pueden hacer los routers, cortafuegos y balanceadores de carga.

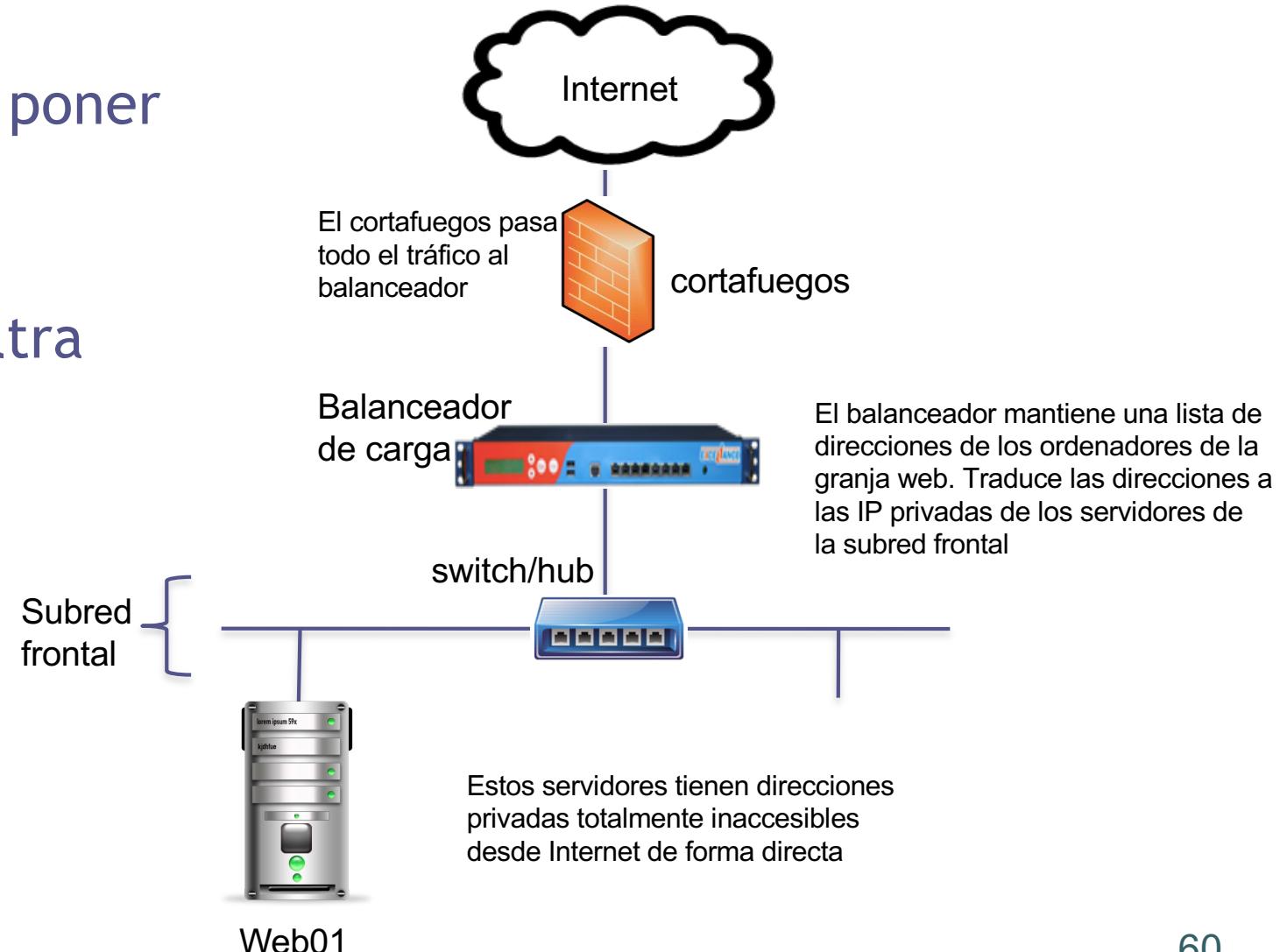
# 8. Conectar la granja web a Internet

## 3. NAT: Network Address Translation

Incluso podemos poner varios niveles:

El cortafuegos filtra los paquetes.

El balanceador distribuye.



# Índice



1. Introducción
2. Configurar la red del sistema web
3. El eje principal de la red del sistema
4. Configurar una zona segura
5. Conectar servidores al front-rail
6. Conectar servidores al back-rail
7. Resumen de configuraciones
8. Conectar la granja web a Internet
9. Conectar la granja web a redes seguras
10. Resumen y conclusiones

## 9. Conectar la granja web a redes seguras

Algunas organizaciones necesitan los servicios de otras empresas (bancos, p.ej).

Para ello se conectan a redes seguras de esas empresas.

La conexión a redes aseguradas es similar a la conexión a Internet, pero con menos riesgos.

Hay que poner un **mecanismo de filtrado y bloqueo de paquetes** para evitar posibles ataques desde las máquinas de esas redes.

## 9. Conectar la granja web a redes seguras

Hay que tener en cuenta las necesidades de los usuarios en relación a los servicios que queremos obtener de la empresa.

Podemos realizar la conexión mediante cortafuegos o mediante protocolos seguros (SSL).

### Por ejemplo:

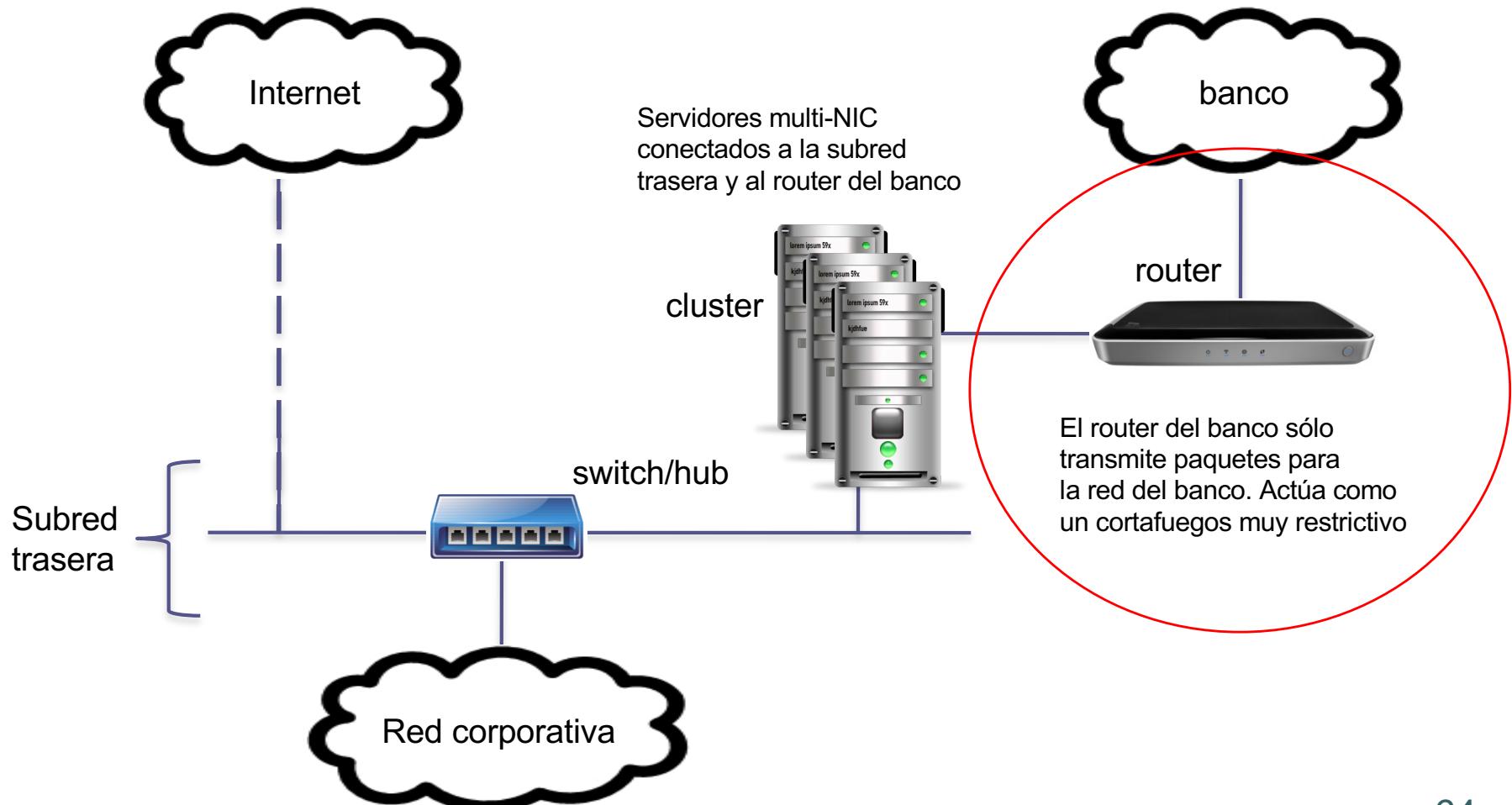
Queremos usar los servicios de un banco para cobrar con tarjeta de crédito (operación de riesgo).

Mediante conexión segura controlada por el banco. 

# 9. Conectar la granja web a redes seguras

Ejemplo:

El banco nos instala un router para conectarnos a su red:



# 9. Conectar la granja web a redes seguras

## Ejemplo (cont.):

Instalar una interfaz de red dedicada y conectada a ese router en los servidores que vayan a consumir ese tipo de servicio.

p.ej. configurar un servidor en el back-rail como pasarela para las operaciones con tarjeta de crédito.

# Índice



1. Introducción
2. Configurar la red del sistema web
3. El eje principal de la red del sistema
4. Configurar una zona segura
5. Conectar servidores al front-rail
6. Conectar servidores al back-rail
7. Resumen de configuraciones
8. Conectar la granja web a Internet
9. Conectar la granja web a redes seguras

[10. Resumen y conclusiones]

# 10. Conclusiones (I)

La configuración de la red de la granja web se puede hacer de varias formas.

La más segura es con el doble DMZ (subred frontal + subred trasera).

Los usuarios acceden desde Internet a los servidores conectados en la subred frontal.

Las máquinas en la subred trasera dan servicios a los usuarios en la red corporativa y a los servidores de la subred frontal.

# 10. Conclusiones (II)

En la calidad del servicio influye:

- el ancho de banda de conexión a Internet
- el filtrado y bloqueo de paquetes
- el balanceo de la carga entre los servidores

Se pueden obtener servicios externos conectando a redes seguras (p.ej. un banco).

# TEMA 4

## Balanceo de Carga

SWAP



¿Qué podemos hacer para distribuir la carga en la granja web?  
¿Qué opción es más adecuada?



José Manuel Soto Hidalgo  
Dpto. Arquitectura y Tecnología de Computadores  
Universidad de Granada

jmsoto@ugr.es

# Índice



- [ 1. Introducción ]
- 2. Funcionamiento básico de un servidor
- 3. Conceptos del balanceo de carga
- 4. Otras tecnologías
- 5. Estructura de la red
  
- 6. Algoritmos de balanceo de carga
- 7. Balanceo de carga global
- 8. Ejemplo 1
- 9. Ejemplo 2
- 10. Ejemplo 3
- 11. Futuro de las tecnologías de balanceo
- 12. Resumen y conclusiones

# Introducción

Inicialmente se usaban grandes mainframes como servidores

Caros de adquirir y mantener.



Hardware y software propietario.



Memoria y disco compartidos.

Las desventajas llevaron al desarrollo del balanceo de carga.

¡La unión hace la fuerza!

# Introducción

-  Varias máquinas trabajando en paralelo es mejor que una sola máquina muy grande (y cara).
-  Si una máquina del grupo se rompe, se puede sustituir fácilmente.
-  Crear una granja web es más barato que adquirir un mainframe de la misma capacidad.

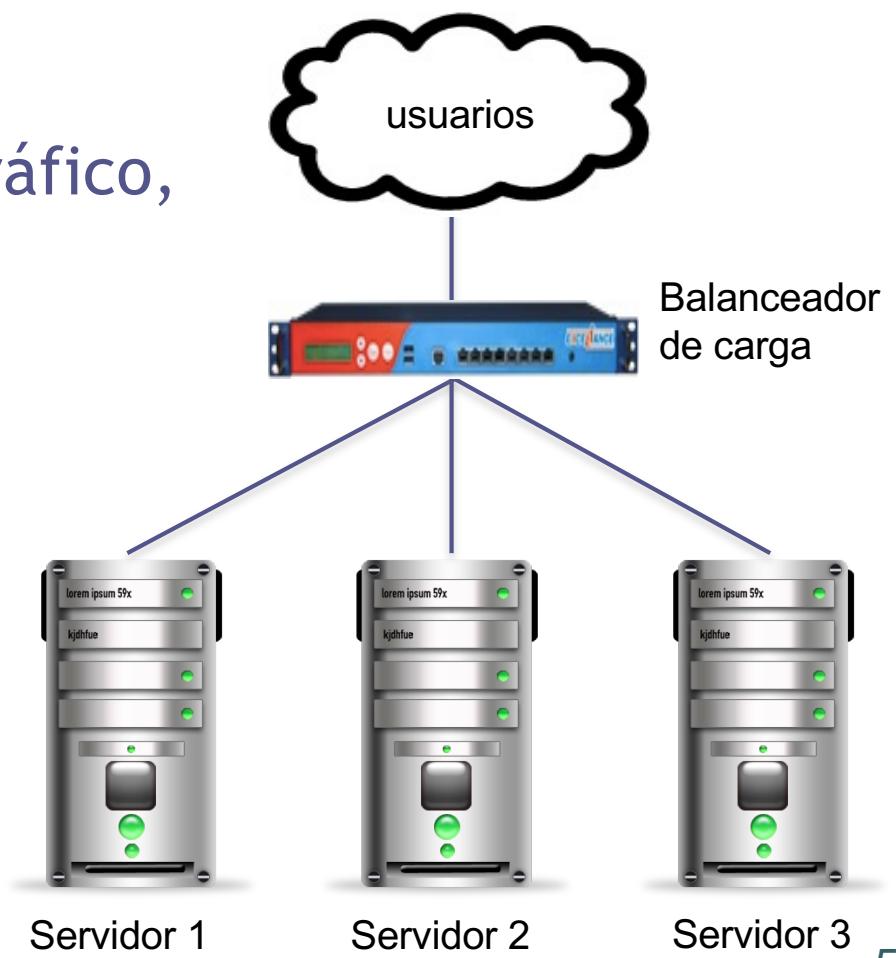
Las tareas se pueden repartir entre grupos de servidores.

¿Cómo repartir las tareas entre servidores?

# Introducción

El balanceo de carga distribuye el tráfico entre varios servidores.

Un dispositivo intercepta el tráfico,  
lo analiza y lo distribuye:



# Introducción

Existen diversos dispositivos hardware y software que pueden hacer funciones de balanceo:

- p.ej. los routers derivan los paquetes por diferentes caminos

Por otro lado, el balanceador realiza tareas adicionales:

- comprobar la disponibilidad y estado de los servidores
- proteger de diversos ataques
- derivar en función del tipo de tráfico

# Índice



1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. Estructura de la red
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
11. Futuro de las tecnologías de balanceo
12. Resumen y conclusiones

# Funcionamiento básico de un servidor

Proceso tras teclear una URL en el navegador:

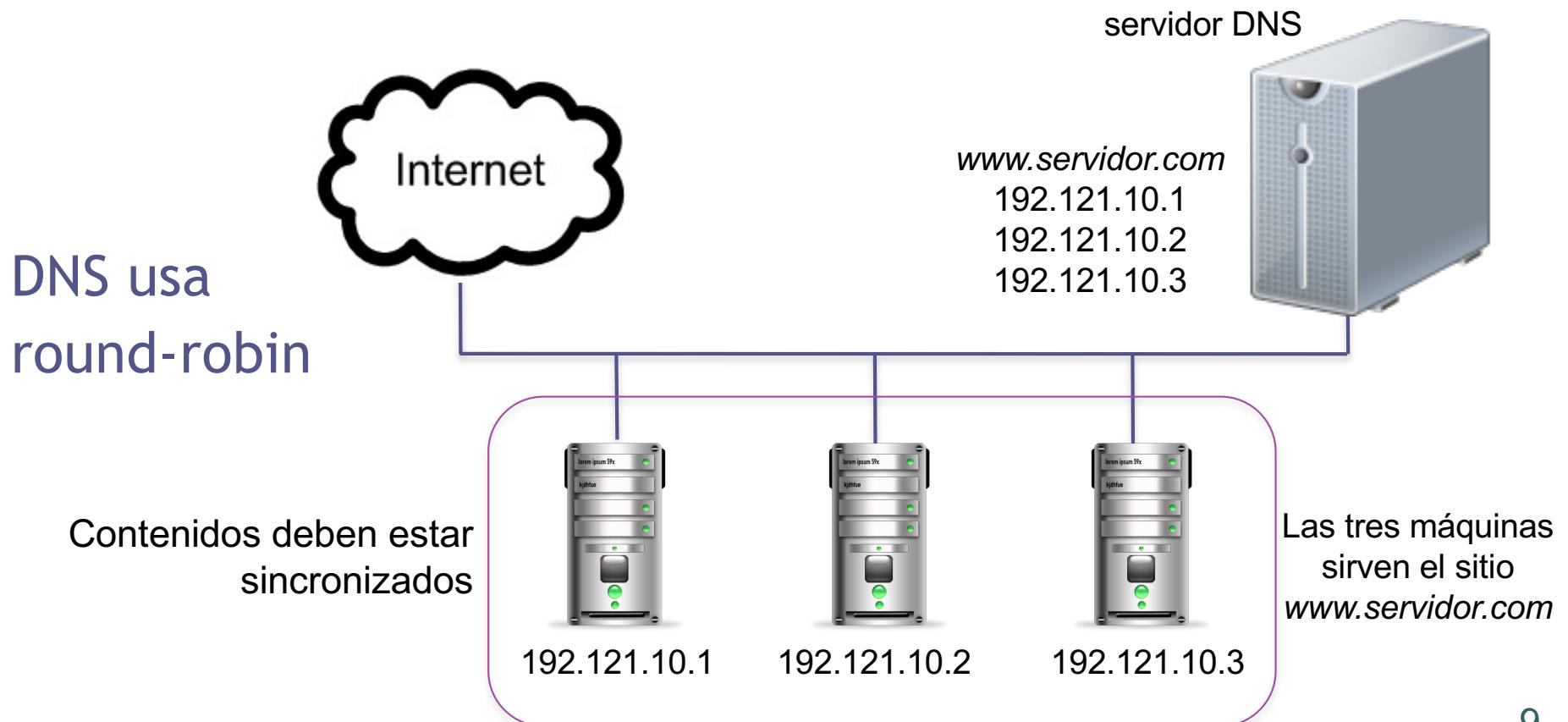
1. El navegador resuelve el nombre del sitio web (DNS).
2. Ya con la IP, el navegador contacta por TCP.
3. Justo a continuación, hace la petición HTTP.
4. El servidor devuelve la página HTML completa.
5. El navegador muestra la información.

¿Y si queremos que varias máquinas sirvan nuestro contenido web? (para disponer de más prestaciones)

# Funcionamiento básico de un servidor

## (1) Balanceo mediante DNS

Configurar el DNS asignando tres IP diferentes al mismo sitio  
(misma dirección)



# Funcionamiento básico de un servidor

## (1) Balanceo mediante DNS

Es una primera aproximación un poco rudimentaria. 

Es muy sencilla de implementar. 

DNS no tiene información sobre la disponibilidad de cada máquina servidora final. 

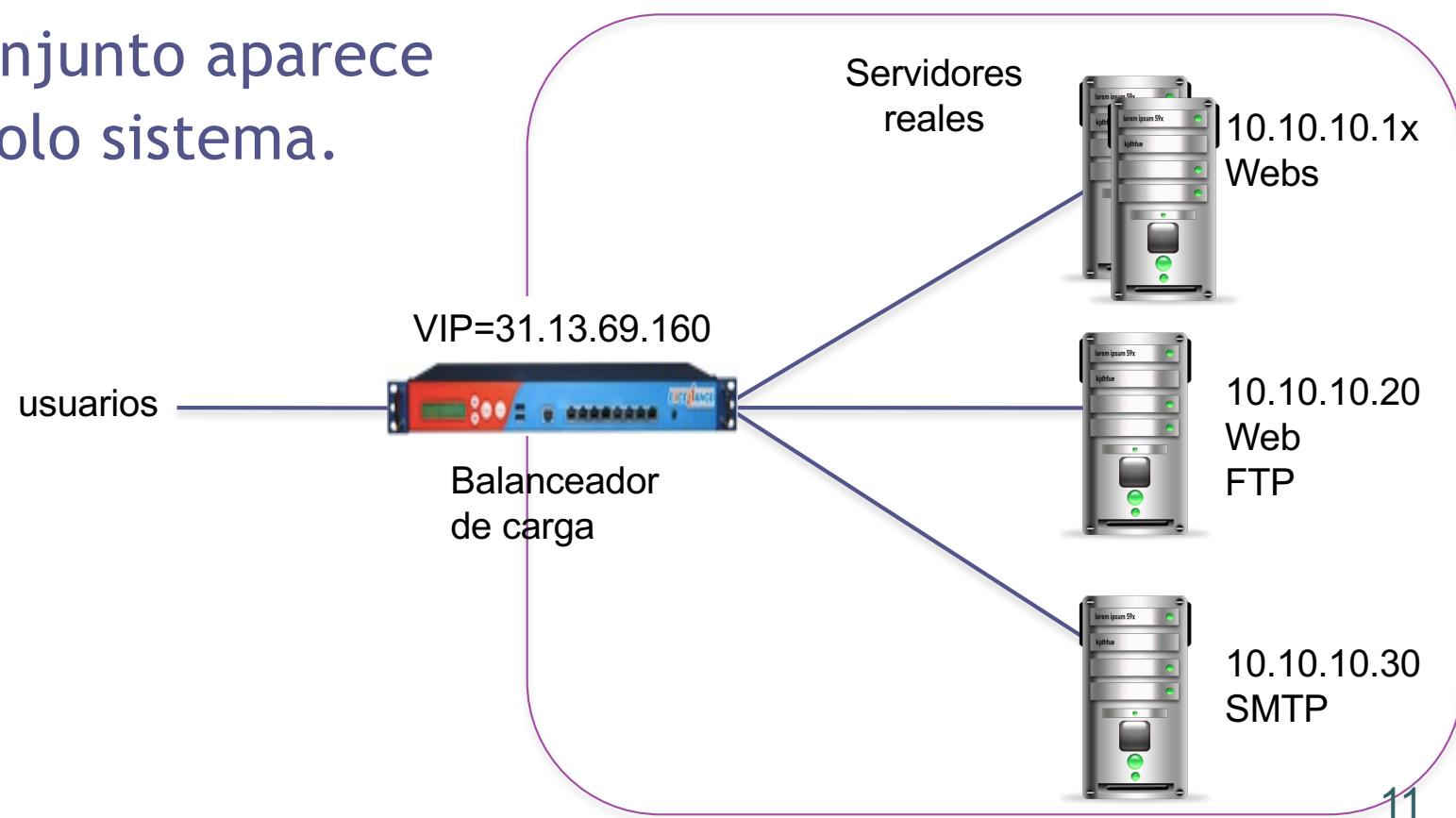
¡Los DNS se inventaron para resolver (traducir) nombres a IP!

# Funcionamiento básico de un servidor

## (2) Granja web con balanceo de carga

El balanceador se configura como punto de entrada para los usuarios.

Todo el conjunto aparece como un solo sistema.



# Funcionamiento básico de un servidor

## (2) Granja web con balanceo de carga

El usuario recibe del DNS la IP pública del balanceador (VIP), que se ocupa en distribuir el tráfico.



### Ventajas:

- Mejora la escalabilidad
- Mejora la disponibilidad
- Facilita el mantenimiento (arreglar o añadir máquinas sobre la marcha)
- Mejora la seguridad (servidores finales están protegidos)

# Índice



1. Introducción
2. Funcionamiento básico de un servidor
- [ 3. Conceptos del balanceo de carga ]**
4. Otras tecnologías
5. Estructura de la red
  
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
11. Futuro de las tecnologías de balanceo
12. Resumen y conclusiones

# Conceptos del balanceo de carga

Tener en cuenta algunos conceptos y términos:

- Modelo OSI
- Balanceador de carga
- VIP
- Servidor
- Grupo o cluster
- Niveles de acceso de usuario
- Redundancia
- Persistencia
- Disponibilidad de servicio
- Algoritmos de balanceo de carga
- Centro de datos
- NAT: network address translation

# Conceptos del balanceo de carga

## Modelo OSI

Los concentradores y routers trabajan con IP y MAC.  
Reenvían tráfico a servidores.

No pueden reenviar tráfico concreto a un servidor y para una aplicación concreta (no tienen en cuenta puertos).

No pueden comprobar si una máquina está disponible.

Se basan en el modelo OSI:

# Conceptos del balanceo de carga

## Modelo OSI

Los protocolos IP, TCP, UDP, HTTP se mapean en las 7 capas.

Capa OSI	Función	Unidades	Ejemplo	Relación con el balanceo de carga
Capa 1	Física	Bits (unos y ceros)	Cable, fibra SX	Cable usado para conectar los switches y hubs
Capa 2	Enlace	Tramas Ethernet	Switches, hubs	Dispositivos que gestionan el tráfico
Capa 3	Red	Direcciones IP	Routers	Dispositivos con características de router
Capa 4	Transporte	TCP, UDP, ICMP	Puerto TCP 80 para http, puerto UDP 161 para SNMP	Nivel típico del balanceo. Principalmente se trabaja con IP y puertos
Capas 5-7	Sesión, presentación y aplicación	URL, cookie, etc	<a href="http://www.servidor.dom">http://www.servidor.dom</a> cookie	Informaciones incluidas en las URL o en las cookies

# Conceptos del balanceo de carga

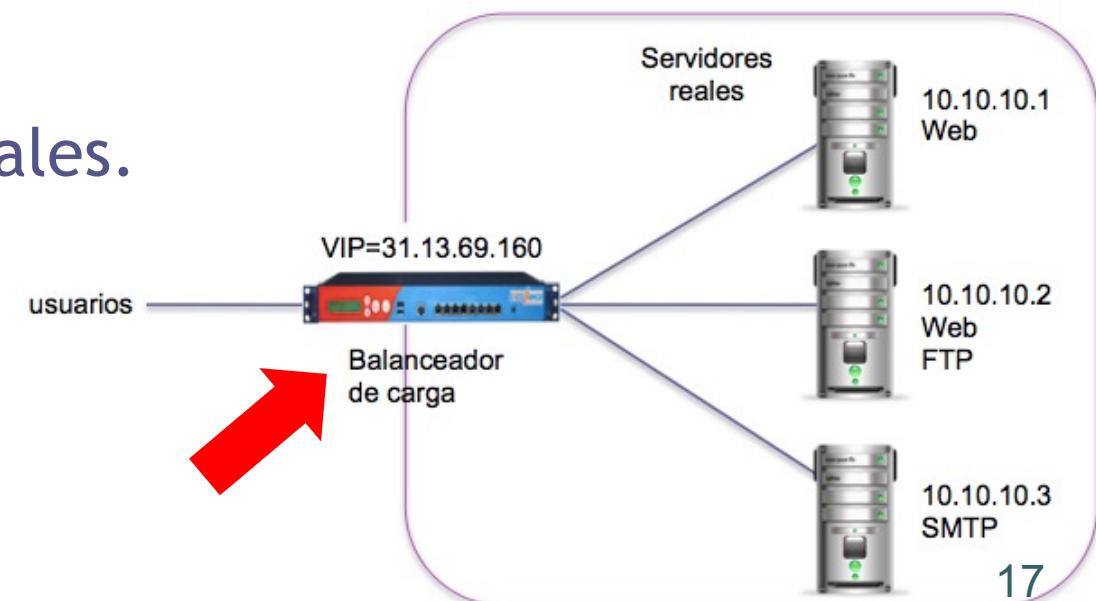
## Balanceador de carga

Es un puente entre la red y los servidores finales.

Puede manejar los protocolos de alto nivel.

Puede manejar los protocolos de nivel de red.

Protege los servidores finales.



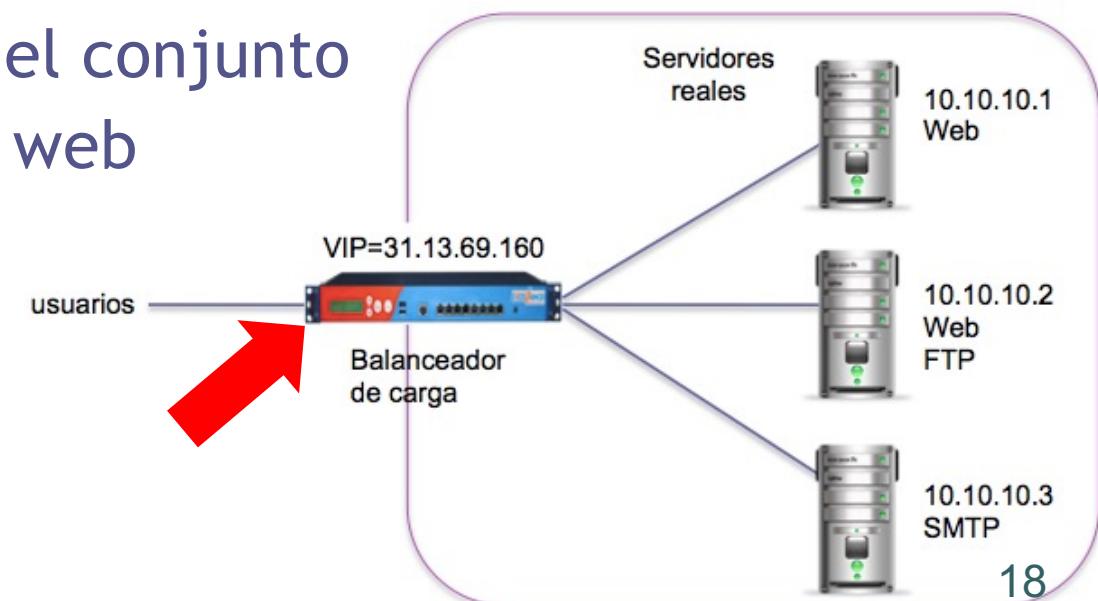
# Conceptos del balanceo de carga

## Virtual IP (VIP)

Dirección pública a la que acceden los usuarios cuando piden un servicio al sistema.

Suele ser la IP del balanceador de carga.

Es la forma de hacer que el conjunto de máquinas de la granja web aparezca como una sola.



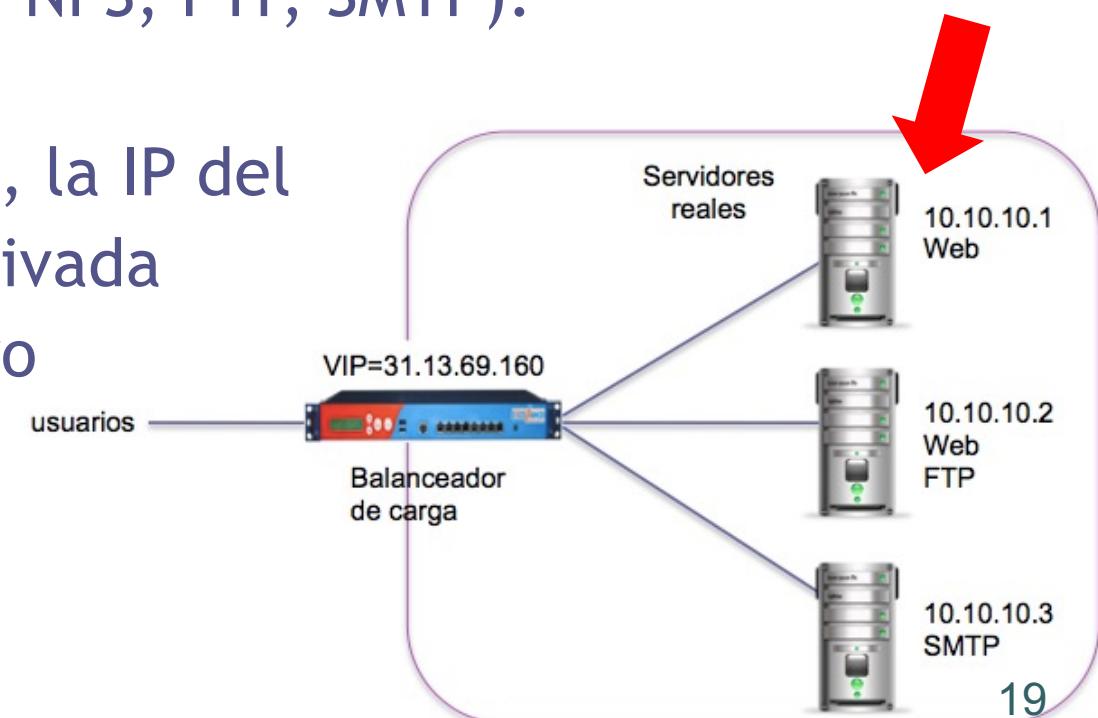
# Conceptos del balanceo de carga

## Servidor

Dispositivo que ejecuta un servicio.

En nuestro caso nos referiremos a servidores HTTP o servicios asociados (BD, NFS, FTP, SMTP).

Según la topología de red, la IP del servidor será pública o privada (accesible a través de otro dispositivo de red que haga NAT).

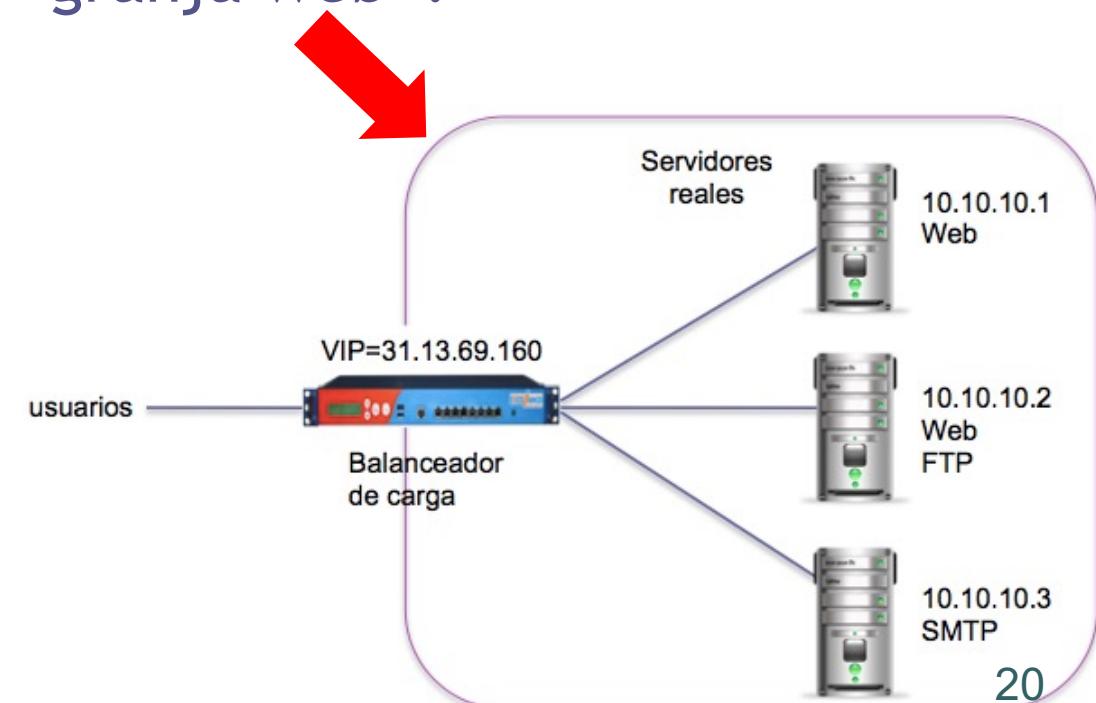


# Conceptos del balanceo de carga

## Grupo o cluster

Conjunto de servidores con un平衡eador al frente para repartir la carga.

Lo que estamos llamando “granja web”.



# Conceptos del balanceo de carga

## Niveles de acceso de usuario

Nos referimos a los permisos de control de los usuarios.

Ej: sólo lectura, superusuario

Entre las políticas de acceso y seguridad se suele definir un tipo de usuario “administrador-web”, con más privilegios que un usuario normal, pero menos que “root”, y que se utiliza para administrar las configuraciones de los servidores que tienen que ver con la web.

# Conceptos del balanceo de carga

## Redundancia

Si un dispositivo falla, otro pasa a hacer su función.

Se trata de evitar la degradación del servicio.

Se suelen poner dos dispositivos idénticos que monitoricen el estado del otro:

- relación maestro-maestro
- relación maestro-esclavo

Protocolo VRRP (Virtual Router Redundancy Protocol).

# Conceptos del balanceo de carga

## Persistencia (sesiones de navegación)

Se refiere a mantener el tráfico de un usuario dirigido al servidor que comenzó a atenderlo.

El balanceador elige a qué servidor final dirige el tráfico, pero a partir de ahí, seguirá dirigiéndolo mientras dure la sesión de navegación.

Es importante en aplicaciones web que almacenan el estado del usuario mientras navega (ej. tienda web).

Otra opción es usar una memoria compartida-distribuida entre los servidores (memcache).

# Conceptos del balanceo de carga

## Comprobación de disponibilidad de servicio

El balanceador debe comprobar regularmente si un servidor está activo o caído para reenviarle más tráfico.

Se puede implementar con ICMP (ping) a los servidores finales.

También comprobando contenido: accede a un servicio web específico y se espera una respuesta concreta por parte del servidor.

# Conceptos del balanceo de carga

## Algoritmos de balanceo de carga

Hay varios algoritmos o estrategias de distribución del tráfico entre el grupo de servidores:

- por turnos, prioridad, ponderación, número de conexiones, etc.

El balanceador los implementa y los aplica según la configuración que hagamos.

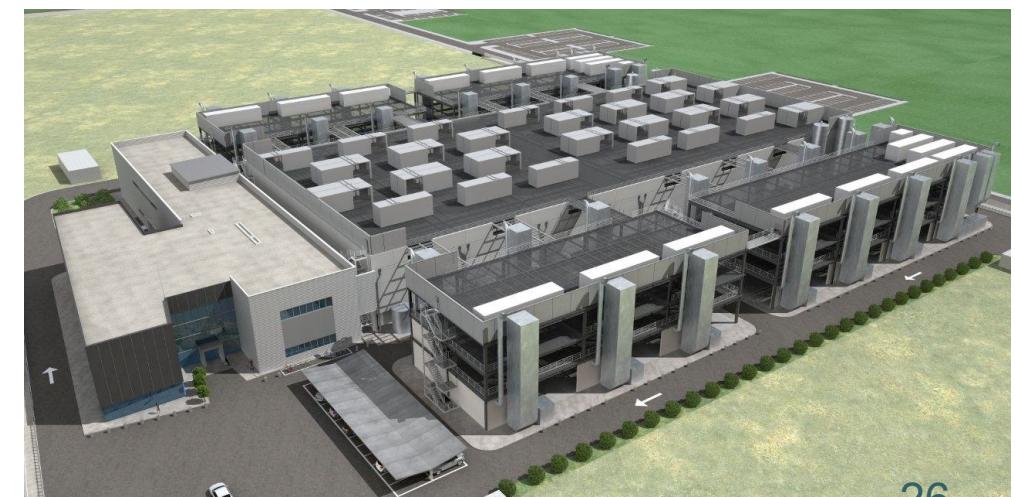
# Conceptos del balanceo de carga

## Centro de datos

Los equipos hardware están hospedados en un edificio con alta seguridad, refrigeración controlada, alimentación continua y sistemas antiincendios específicos.

Se ocupa de la seguridad, alimentación, refrigeración y conexión.

Ahorro en mantenimiento.



# Conceptos del balanceo de carga

## NAT: Network Address Translation

El balanceador de carga hace traducción de direcciones.

Toma los paquetes de entrada y cambia la IP de destino (la VIP por la IP privada de un servidor real final).

Cuando el servidor sirve contenido, el balanceador hace el cambio de nuevo (pone la VIP en lugar de la IP final).

Opción más eficiente =>

el **servidor final** ya genera los paquetes con la **VIP como origen** (el balanceador se evita un trabajo).

# Conceptos del balanceo de carga



el cliente hace una petición enviando un paquete:

$SRC=56.56.56.56$  **DST=88.88.88.88**

Traducción =>  $SRC=88.88.88.88$   $DST=172.16.0.5$

el servidor procesa la petición y devuelve la respuesta en un paquete:

**SRC=172.16.0.5**  $DST=88.88.88.88$

Traducción =>  $SRC=88.88.88.88$   $DST=56.56.56.56$

Opción más eficiente =>

el servidor final ya genera los paquetes con la VIP como origen (el balanceador se evita un trabajo).

# Índice



1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. Estructura de la red
  
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
11. Futuro de las tecnologías de balanceo
12. Resumen y conclusiones

# Otras tecnologías similares

El balanceo de carga consiste en interceptar el tráfico para redirigirlo a varios servidores.

Hay otras tecnologías similares:



- Balanceo de carga en cortafuegos



- Balanceo de carga global



- Clustering



# Otras tecnologías similares

## Balanceo de carga en cortafuegos

Se trata de estructurar varios cortafuegos de forma balanceada.

Normalmente, un cortafuegos es una máquina más.

La CPU tiene una limitación: más de 80Mbps

En algunos casos puede no ser suficiente.



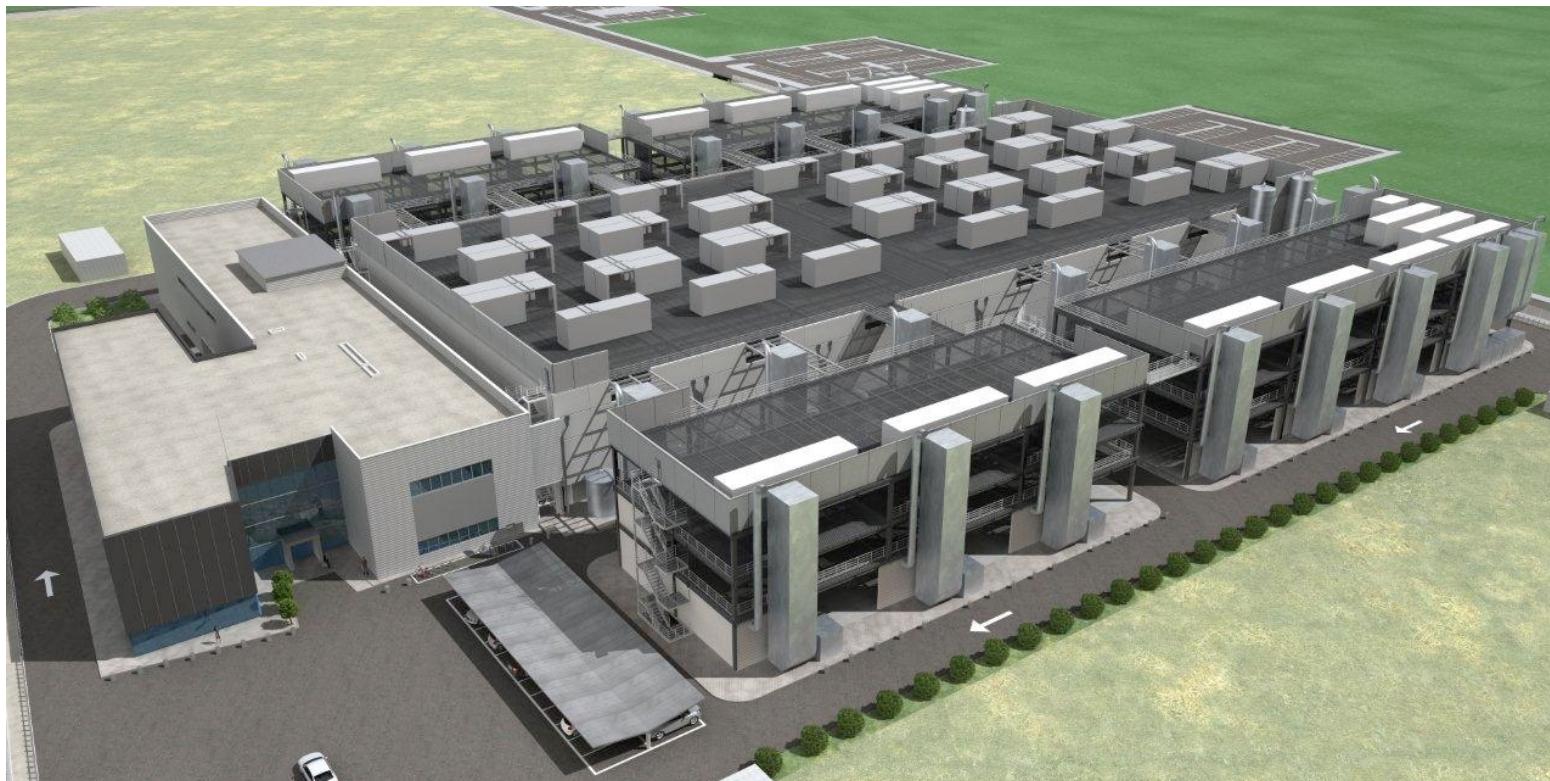
Configurar varias máquinas como cortafuegos balanceados.



# Otras tecnologías similares

## Balanceo de carga global (I)

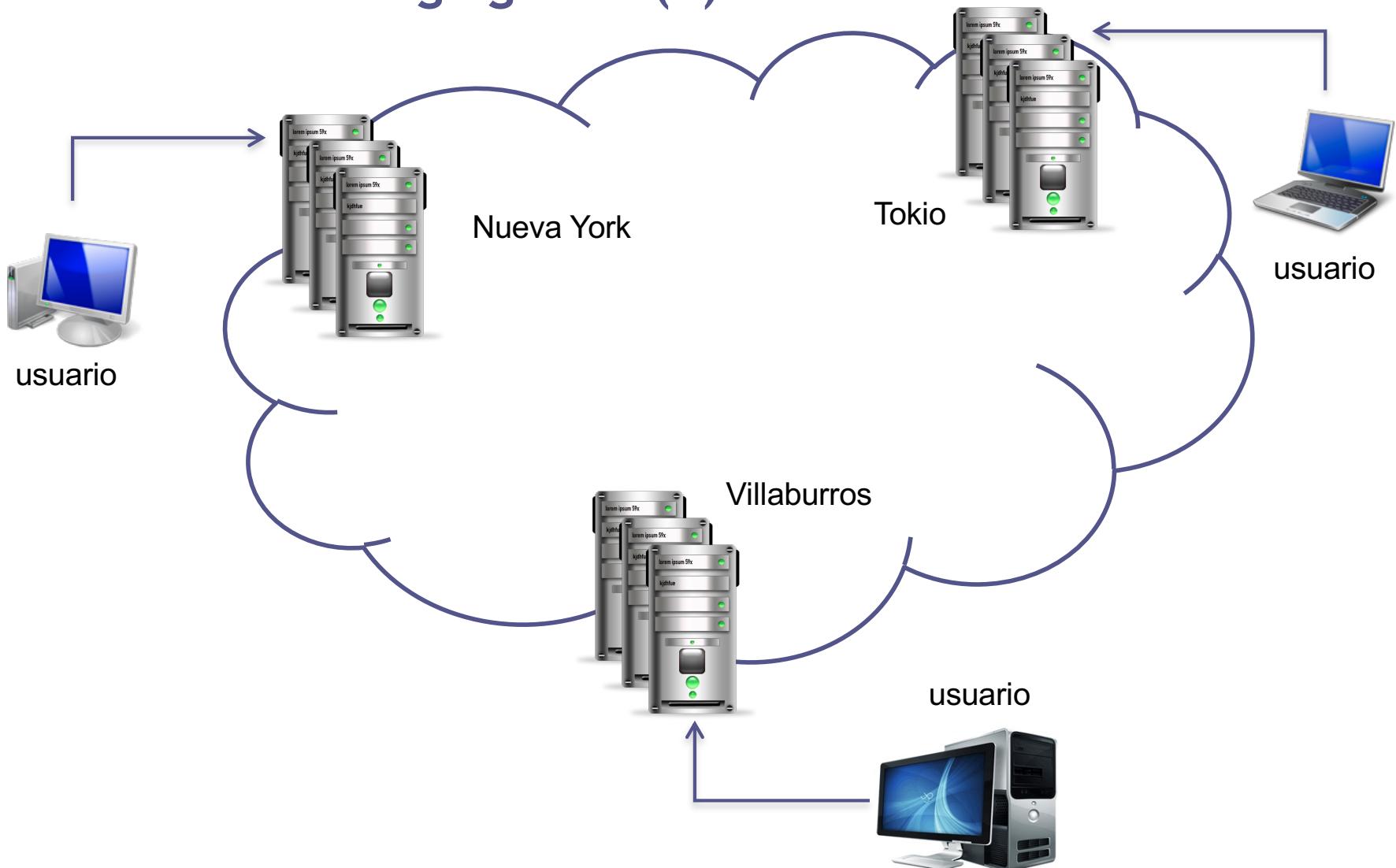
Realizar balanceo de carga, similar al estudiado, pero entre centros de datos.





# Otras tecnologías similares

## Balanceo de carga global (II)





# Otras tecnologías similares

## Balanceo de carga global (III)

Balanceo de carga entre centros de datos.

El tráfico de cierto usuario va (generalmente) desde su navegador cliente hasta el centro de datos más cercano geográficamente.

Una vez que las peticiones llegan a cierto centro de datos, se balancea también para distribuir entre el grupo de servidores.



# Otras tecnologías similares

## Balanceo de carga global (IV)

Se evita un retraso por el viaje de los paquetes a un centro situado a miles de kilómetros.

Redundancia y balanceo: si un centro falla (corte de luz, maremoto, etc) el tráfico se redirige automáticamente a otro centro disponible.

Se puede implementar con los DNS o con BGP (*border gateway protocol*).



# Otras tecnologías similares

## Clustering

Solución basada en la disponibilidad y escalabilidad.

Técnica orientada a disponer de un cluster de máquinas que aceptarán grandes tareas computacionales.

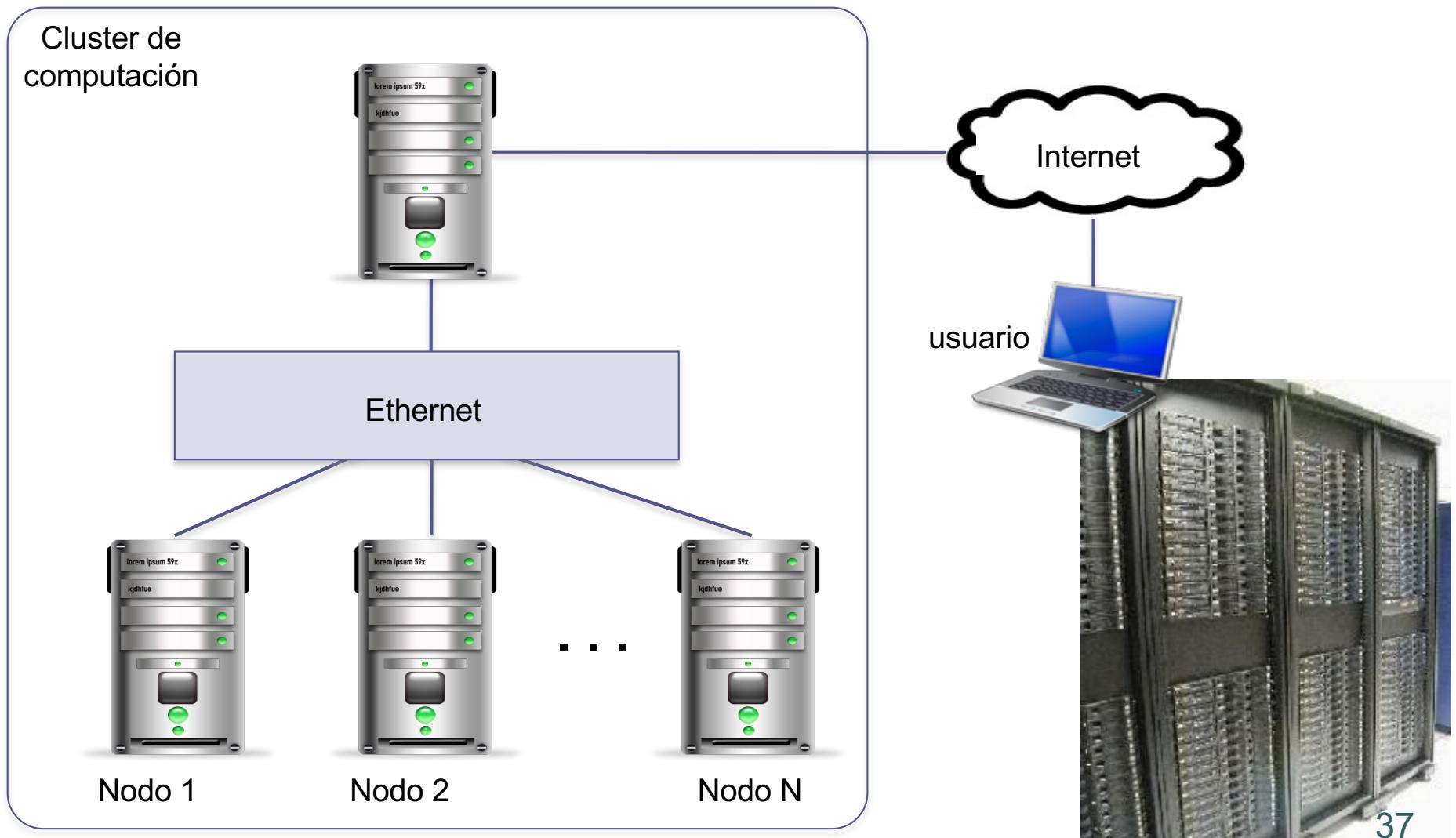
Ejecución paralela repartiendo el trabajo entre varias máquinas.

*Se accede por la máquina principal para lanzar trabajos (carga), y las máquinas del cluster trabajan juntas para dar el servicio (en este caso, de computación en lugar de servir peticiones)*



# Otras tecnologías similares

## Clustering



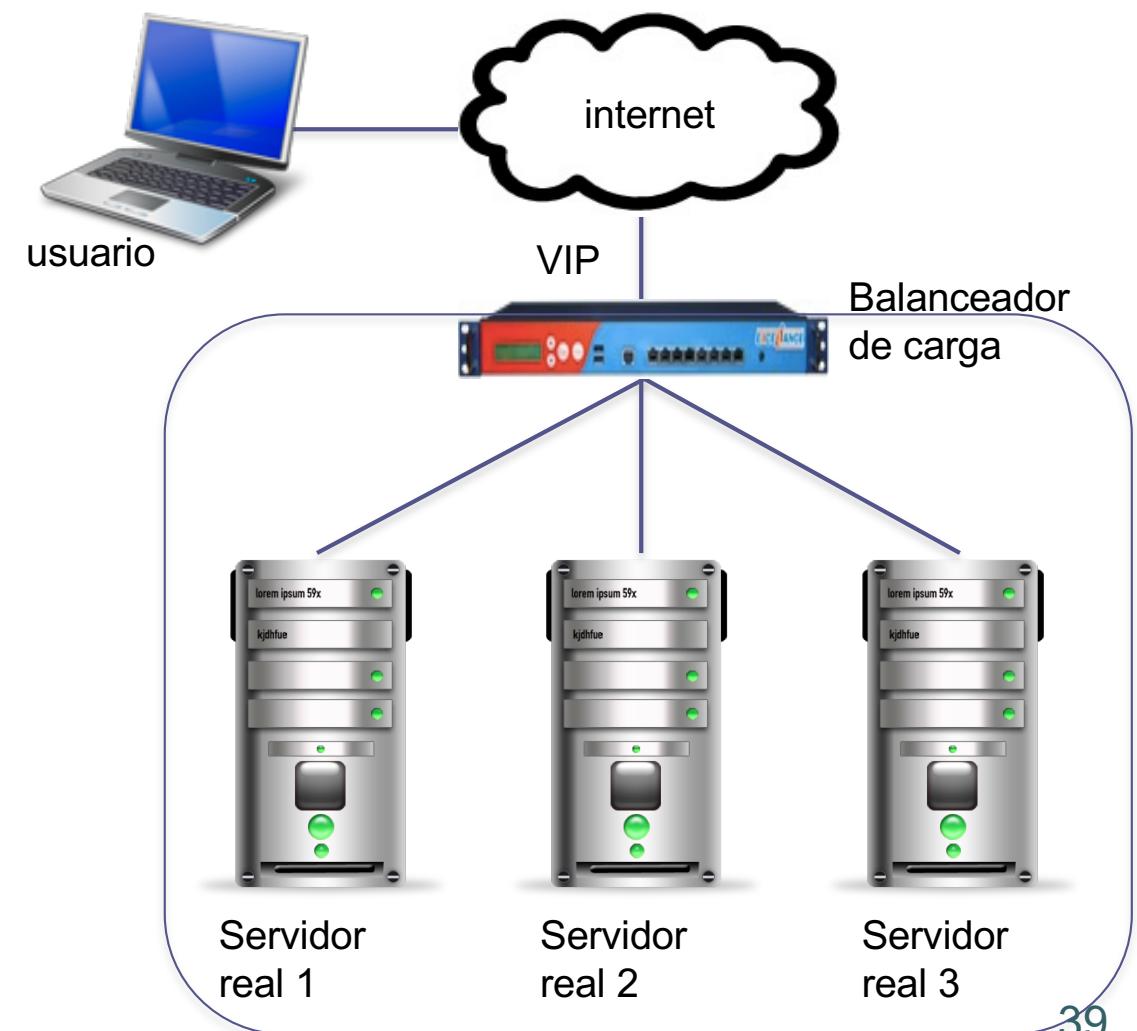
# Índice



1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. **Estructura de la red**
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
11. Futuro de las tecnologías de balanceo
12. Resumen y conclusiones

# Estructura de la red

Una instalación basada en el uso de balanceador queda representada en el siguiente esquema:



# Estructura de la red

El tráfico generado por el usuario va a través del balanceador hasta la máquina servidora final.

La respuesta va hasta el cliente a través del balanceador.

La configuración se basará en (1) servidor con software específico, o bien (2) concentrador o switch con funciones de balanceo.

Cada opción tiene ventajas e inconvenientes...



# Estructura de la red

## (1) Configurar un servidor como balanceador

Existe software específico para configurar un PC como balanceador. Casi cualquier SO servirá.

Usa algoritmos de reparto de carga conocidos que podemos configurar.

# Estructura de la red

## (1) Configurar un servidor como balanceador

Opciones de software libre:

- HAProxy
- nginx
- Apache
- Pound

Opciones de software propietario:

- Local Director (**cisco**):  
[https://www.cisco.com/web/offer/localdirector/docs/lodir\\_rg.htm](https://www.cisco.com/web/offer/localdirector/docs/lodir_rg.htm)
- BIG-IP (**F5**): <http://www.f5.com/products/big-ip/>
- NLB (**Microsoft**): <http://support.microsoft.com/kb/323437>

# Estructura de la red

## (1) Configurar un servidor como balanceador

Más opciones de software libre:

<https://geekflare.com/open-source-load-balancer/>

The screenshot shows a web browser window with the URL <https://geekflare.com/open-source-load-balancer/> in the address bar. The main content area displays the heading "10 Open Source Load Balancer for HA and Improved Performance". Below the heading is a green rectangular graphic containing a diagram of a load balancer. The diagram shows a top-level box with three dots above it, connected by arrows to three blue square boxes below, representing a load balancer distributing traffic to multiple servers.

- [Seesaw](#)
- [LoadMaster by KEMP](#)
- [HAProxy](#)
- [ZEVENET](#)
- [Neutrino](#)
- [Balance](#)
- [Pen](#)
- [Nginx](#)
- [Traefik](#)
- [Gobetween](#)

# Un ejemplo: Balanceo con gobetween

- Disponible para varias plataformas.
- Balanceo de carga, balanceo usando Docker/Swarm, ElasticSearch balancing.

<http://gobetween.io/downloads.html>

The screenshot shows a web browser window with the URL [gobetween.io/downloads.html](http://gobetween.io/downloads.html) in the address bar. The page itself has a blue header with the gobetween logo and navigation links for Documentation. Below the header, there are download links for Linux, Windows, Darwin, and Docker Image, each accompanied by a small icon and a red cartoon character.

Platform	File Type	Architecture
Linux	(tar.gz)	x86   x64
Windows	(zip)	x86   x64
Darwin	(zip)	x86   x64
Docker Image	yyyar/gobetween	

# Un ejemplo: Balanceo con gobetween

The screenshot shows a web browser displaying the [gobetween documentation](http://gobetween.io/documentation.html#Static-balancing). The title of the page is "Simple load balancing". The diagram illustrates a flow from "Internet" to a "gobetween" proxy, which then connects to three "backend" servers: "backend-1", "backend-3", and "backend-3" (repeated). The sidebar on the left lists navigation links: Introduction, Installation, Configuration, Protocols, Balancing, and Downloads.

Simple load balancing

```
graph LR; Internet((Internet)) --> gobetween[gobetween]; gobetween --> backend1[backend-1]; gobetween --> backend3_1[backend-3]; gobetween --> backend3_2[backend-3]
```

Introduction

Installation

Configuration

Protocols

Balancing

Downloads

<http://gobetween.io/downloads.html>

# Estructura de la red

## (2) Usar un dispositivo balanceador específico:

Dispositivos tipo “caja negra” que incluyen hardware y software para el balanceo.

Un primer tipo usa **procesadores específicos (ASIC, *application specific integrated circuit*)** para realizar las tareas de modificación de paquetes.



# Estructura de la red

## (2) Usar un dispositivo balanceador específico:

Procesadores extremadamente rápidos haciendo esas modificaciones (aunque no pueden realizar otras tareas que sí pueden hacer las CPUs de propósito general).



# Estructura de la red

(2) Usar un dispositivo balanceador específico:



MODEL COMPARISON	MODEL 240	MODEL 340	MODEL 440	MODEL 640
<b>CAPACITY*</b>	<b>\$1,424</b>	<b>\$1,899</b>	<b>\$3,799</b>	<b>\$8,549</b>
Maximum Throughput	95 Mbps	950 Mbps	1 Gbps	4 Gbps
Real Server Support	10	35	50	250
SSL Offloading/Acceleration		500 TPS	4,000 TPS	15,000 TPS
<b>HARDWARE</b>				
Rackmount Chassis	1U Mini	1U Mini	1U Mini	1U Fullsize
Dimensions (in)	16.8 x 1.7 x 9	16.8 x 1.7 x 14	16.8 x 1.7 x 14	16.8 x 1.7 x 22.6
Dimensions (cm)	42.7 x 4.3 x 23	42.7 x 4.3 x 35.6	42.7 x 4.3 x 35.6	42.7 x 4.3 x 57.4
Weight (lbs/kg)	8 / 3.6	12 / 5.4	12 / 5.4	26 / 11.8
Ethernet	2 x 10/100	2 x Gigabit	2 x Gigabit	12 x Gigabit
AC Input Current (Amps)	1.0	1.2	1.4	1.8
ECC Memory				✓
<b>FEATURES</b>				
Layer 4 Load Balancing	✓	✓	✓	✓
Direct Server Return	✓	✓	✓	✓
Intrusion Prevention	✓	✓	✓	✓
High Availability		✓	✓	✓
VLAN		✓	✓	✓
Layer 7 Load Balancing		✓	✓	✓
SIP Call ID Persistence		✓	✓	✓
Cookie Persistence		✓	✓	✓
SSL Offloading		✓	✓	✓
Content Routing		✓	✓	✓
SNMP		✓	✓	✓
Programming Interface/API			✓	✓
Global Server Load Balancing			✓	✓
HTTP Compression			✓	✓
Content Caching			✓	✓
Link Bonding (LACP)				✓

# Estructura de la red

## (2) Usar un dispositivo balanceador específico:

Dispositivos que realmente son servidores con un SO comercial muy optimizado para realizar estas tareas.

Productos de Cisco Systems, Barracuda, Foundry Networks, Nortel Networks, F5 Networks y Radware.

Los平衡adores tipo caja-negra específicos resultan más rápidos que los basados en una máquina con software específico.

# Balanceo con dispositivos hardware

Los balanceadores hardware están basados en un núcleo Linux/Unix que ejecuta los algoritmos estudiados.

Fabricantes:

- Cisco Systems: <http://www.cisco.com/web/ES/index.html>
- Foundry Networks: <http://www.brocade.com/index.page>
- Nortel Networks: <http://www.nortel-us.com/>
- F5 Networks: <http://www.f5.com/>
- Radware: <http://www.radware.com/>

# Balanceo con dispositivos hardware

La funcionalidad descrita hasta ahora está soportada por los balanceadores hardware.

Algunas funciones no se implementan en la mayoría del software de balanceo.

Los balanceadores hardware son más eficientes que los sistemas basados en un ordenador con software específico.

# Balanceo con dispositivos hardware

La configuración y control de su funcionalidad se puede hacer con interfaz de ventanas o por línea de comandos.

<https://www.zevenet.com/zen-load-balancer/>



The screenshot displays the Zen Load Balancer web interface. On the left, there's a large logo and the text "Zen Load Balancer". The main area has two main sections: "Manage::Farms::" and "Settings::Interfaces".

**Manage::Farms::** shows a table with two entries:

Name	IP	Cluster
wwwsrv0A	192.168.0.101	Cluster 1
IP		
wwwsrv0B	192.168.0.101	Cluster 2

Below this table, there are status messages: "Zen latency is UP on zenlb1.192.168.0.102 is active", "Zen is running on zenlb1", and "Global status: GREEN".

**Settings::Interfaces** shows a table with three entries:

Name	Addr	HWaddr	Netmask	Gateway	Status	Actions
eth0	192.168.0.101	56:54:00:9a:b1:a1	255.255.255.0		GREEN	
eth0.1	192.168.0.102	56:54:00:9a:b1:a2	255.255.255.0		GREEN	
eth1		56:54:00:76:b0:81			RED	

**Edit real IP servers configuration**

Server	Address	Port	Timeout	Weight	Actions
0	192.168.0.100	80	-	5	
1	192.168.0.101	80	-	3	

**Cancel**

**53**

# Balanceo con dispositivos hardware



Zen Load Balancer



Products ▾ Services ▾ Alliances ▾

ZVA64 EE 3110 Virtual Appliance

ZNA64 EE 3300 HW Appliance

**ZNA64**  
**Enterprise Edition**  
**3300 Appliance**



- ✓ Intel Multi-core New CPU's Generation.
- ✓ Powered by Zen Load Balancer Enterprise Edition.
- ✓ 64 bits support.
- ✓ Light weight distribution.
- ✓ Ready for fast deployment without installation.
- ✓ Full warranty.



# Balanceo con dispositivos hardware



Zen Load Balancer



Products ▾ Services ▾ Alliances ▾

ZVA64 EE 3110 Virtual Appliance

ZNA64 EE 3300 HW Appliance

**ZVA64**  
**Enterprise Edition**  
**3110 Virtual Appliance**



- ✓ VMware ESXi 5.5/5.1/5.0 host platform support.
- ✓ Hyper-V 2012/2012 R2 host platform support.
- ✓ KVM 3.2.0 host platform support
- ✓ Zen Load Balancer Enterprise Edition.



# Balanceo con dispositivos hardware

The screenshot shows two versions of the Zevenet website side-by-side.

**Left Side (Original Zen Load Balancer page):**

- Header: "Zen Load Balancer is now ZEVENET | Load Balancing made easy"
- Text: "Zevenet is easier"
- Logo: "ZEVENET" with a stylized green square icon.
- Navigation: PRODUCTS, SERVICES, PARTNERS, ABOUT US, TRY ZEVENET
- Search: Buscar

**Right Side (Updated Zevenet page):**

- Header: "ZEVENET Community Edition" and "Open Source load balancer"
- Image: GitHub logo and Symfony logo.
- Background: A network graph with green and grey nodes and connecting lines.
- Text: "ZEVENET Community Edition" and "Open Source load balancer".
- Navigation: PRODUCTS (underlined), SERVICES, PARTNERS, ABOUT US, TRY ZEVENET
- Buttons: "DOWNLOAD SOURCE CODE" and "DOWNLOAD ISO IMAGE" (the "ISO IMAGE" button is highlighted with a red border).
- Search: Buscar

<https://www.zevenet.com/products/community/#repository>

# Balanceo con dispositivos hardware

## Funcionalidad (I):

- 1) Los balanceadores no solo **organizan** la red y **reparten** tráfico entre los servidores de la granja, sino que **monitorizan** la disponibilidad y fallos en aplicaciones.
- 2) Algunos pueden balancear la carga de dispositivos como cortafuegos o concentradores.
- 3) Los balanceadores hardware tienen varios niveles de **redundancia**: configurar varios en paralelo, componentes internos redundantes (cpu, ram, alimentación).

# Balanceo con dispositivos hardware

## Funcionalidad (II):

- 4) Podemos hacer configuraciones por parejas, tipo activo/pasivo o activo/activo. Si uno cae, el otro asume su carga.
- 5) Los balanceadores hardware mantienen y aseguran la **sesión de navegación** de los usuarios. Así se hace compatible el uso de cookies.
- 6) Si un grupo de servidores o un balanceador falla, el suplente puede copiar la información de sesión a otro grupo de servidores para mantener la navegación.

# Balanceo con dispositivos hardware

## Funcionalidad (III):

- 7) Los balanceadores permiten la entrada/salida de servidores de forma automática. Los monitoriza, y si uno falla, deja de enviarle tráfico, hasta que vuelva.
- 8) Hacen traducción NAT, y analizan las cabeceras TCP/IP para derivar tráfico al servidor más adecuado, en función del servicio solicitado.
- 9) Se pueden configurar para evitar algunos ataques del tipo TCP SYN, DoS, ping of death, IP spoofing, etc.

# Balanceo con dispositivos hardware

## Funcionalidad (IV):

10) Los balanceadores hardware permiten servir el contenido estático directamente: **caching service**.

Así quitamos trabajo a los servidores finales.

Analizan la petición HTTP y lo sirven de la forma más eficiente:

- o bien enviándolo a un grupo de servidores que servirán contenido estático
- o bien determinan el contenido estático más demandado y lo sirven directamente, *cacheándolo* en su memoria RAM.

# Índice



1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. Estructura de la red
- [ 6. Algoritmos de balanceo de carga ]**
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
11. Futuro de las tecnologías de balanceo
12. Resumen y conclusiones

# Algoritmos de balanceo de carga

Los dispositivos y el software de balanceo ofrecen métodos y opciones para repartir carga.

Algunos métodos son estáticos, pero otros tienen en cuenta el estado de las máquinas servidoras.

- Si las máquinas van a ser similares y vamos a servir contenido estático => algoritmos estáticos
- Si las máquinas son heterogéneas y vamos a servir contenido dinámico => algoritmos basados en ponderación.



Probar varios métodos para decidir.

# Algoritmos de balanceo de carga

Los algoritmos más comunes son:

1. balanceo basado en turnos (round-robin)
2. balanceo basado en el menor número de conexiones
3. balanceo basado en ponderación
4. balanceo basado en prioridad
5. balanceo basado en tiempo de respuesta
6. combinación de los algoritmos de tiempo de respuesta y menor número de conexiones

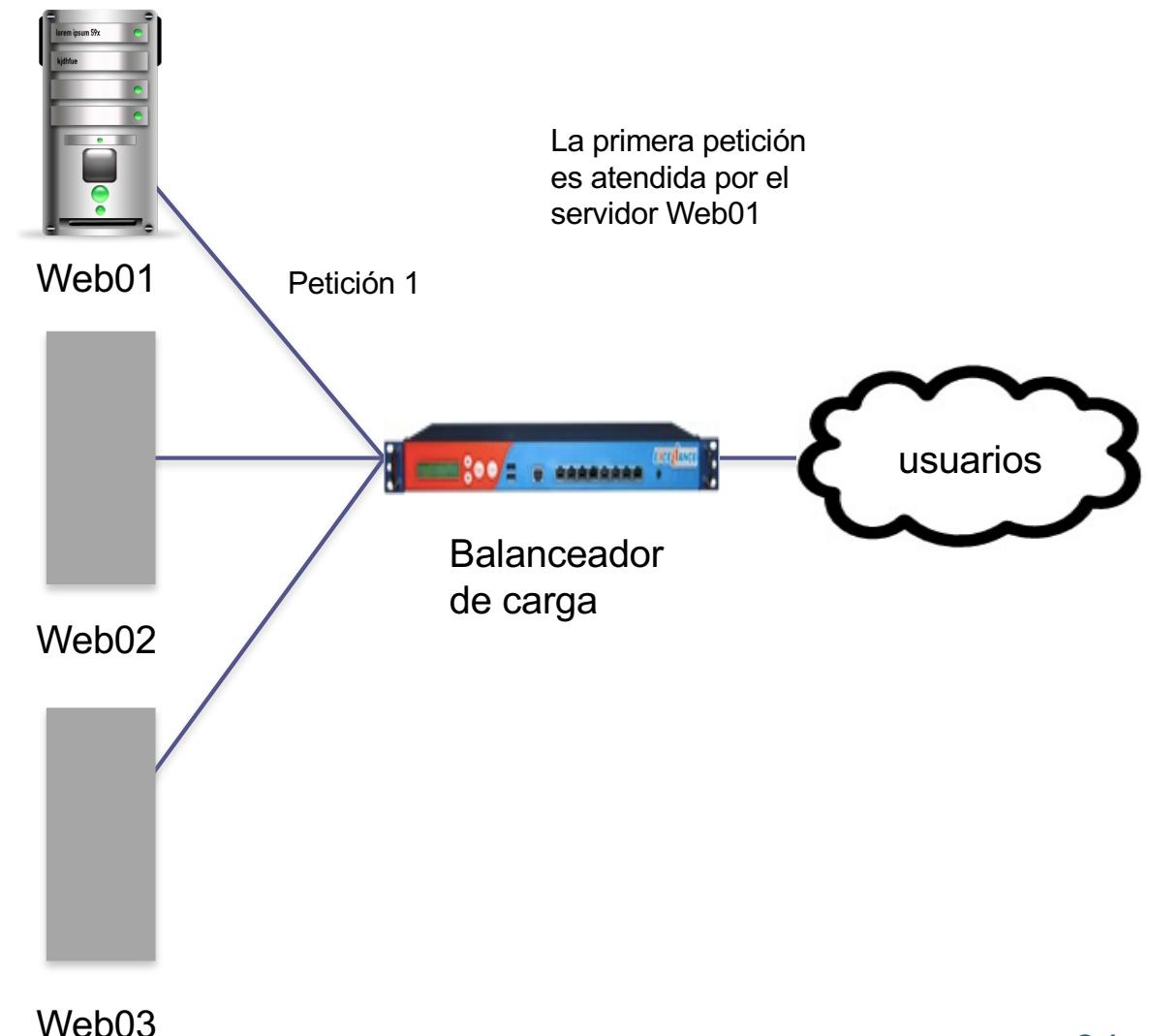


# Algoritmos de balanceo de carga

## 1. Algoritmo de balanceo basado en turnos (round-robin)

Imaginemos que están en una fila.

El primero sirve la petición y pasa al final.



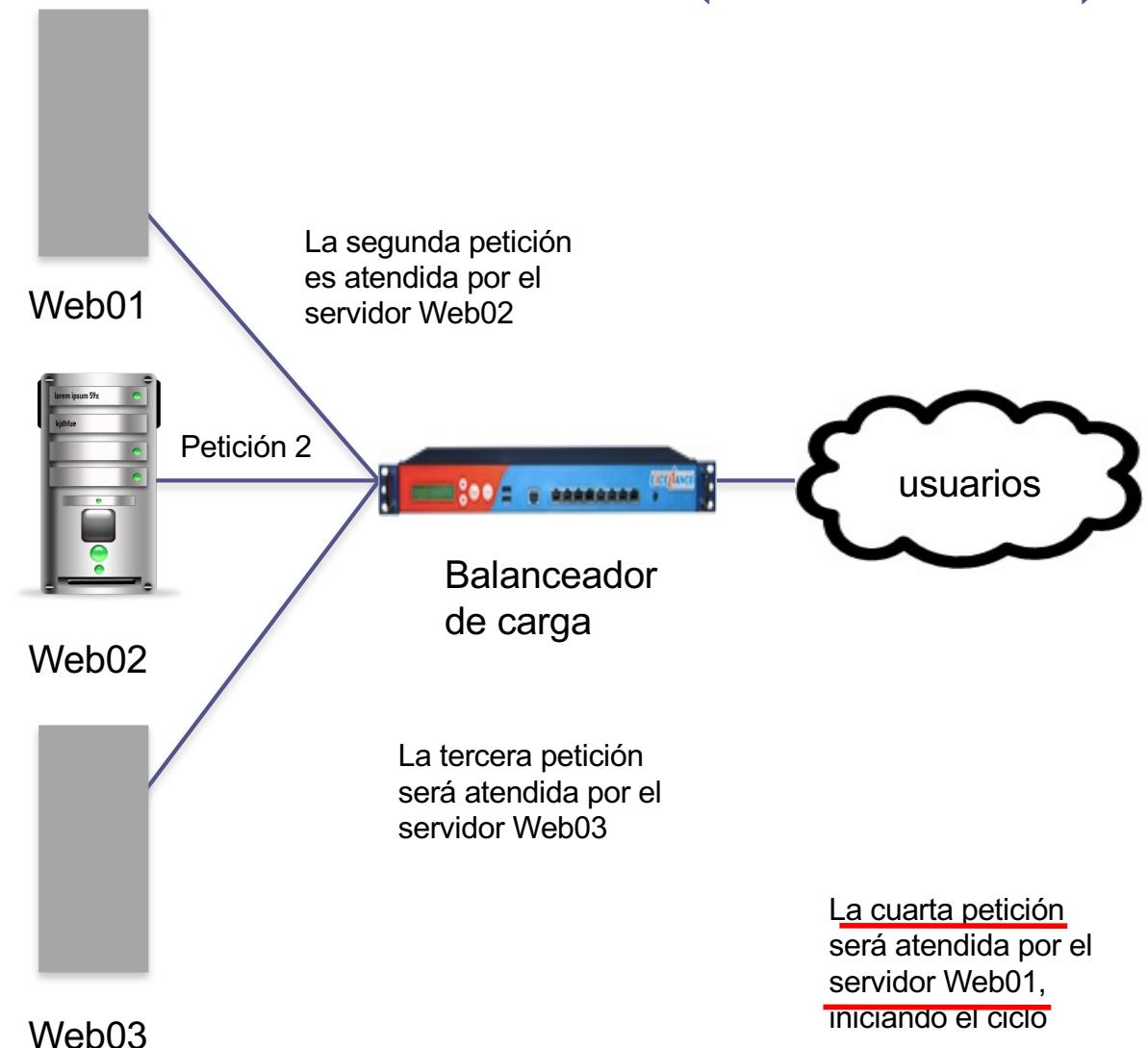


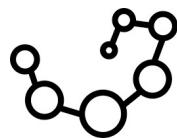
# Algoritmos de balanceo de carga

## 1. Algoritmo de balanceo basado en turnos (round-robin)

El segundo en servir  
es el servidor 2.

etc...





# Algoritmos de balanceo de carga

## 1. Algoritmo de balanceo basado en turnos (round-robin)

Este algoritmo supone a todas las máquinas con la misma potencia.

Es adecuado

si todos tienen potencia similar

o bien

si vamos a servir aplicaciones o servicios sencillos.

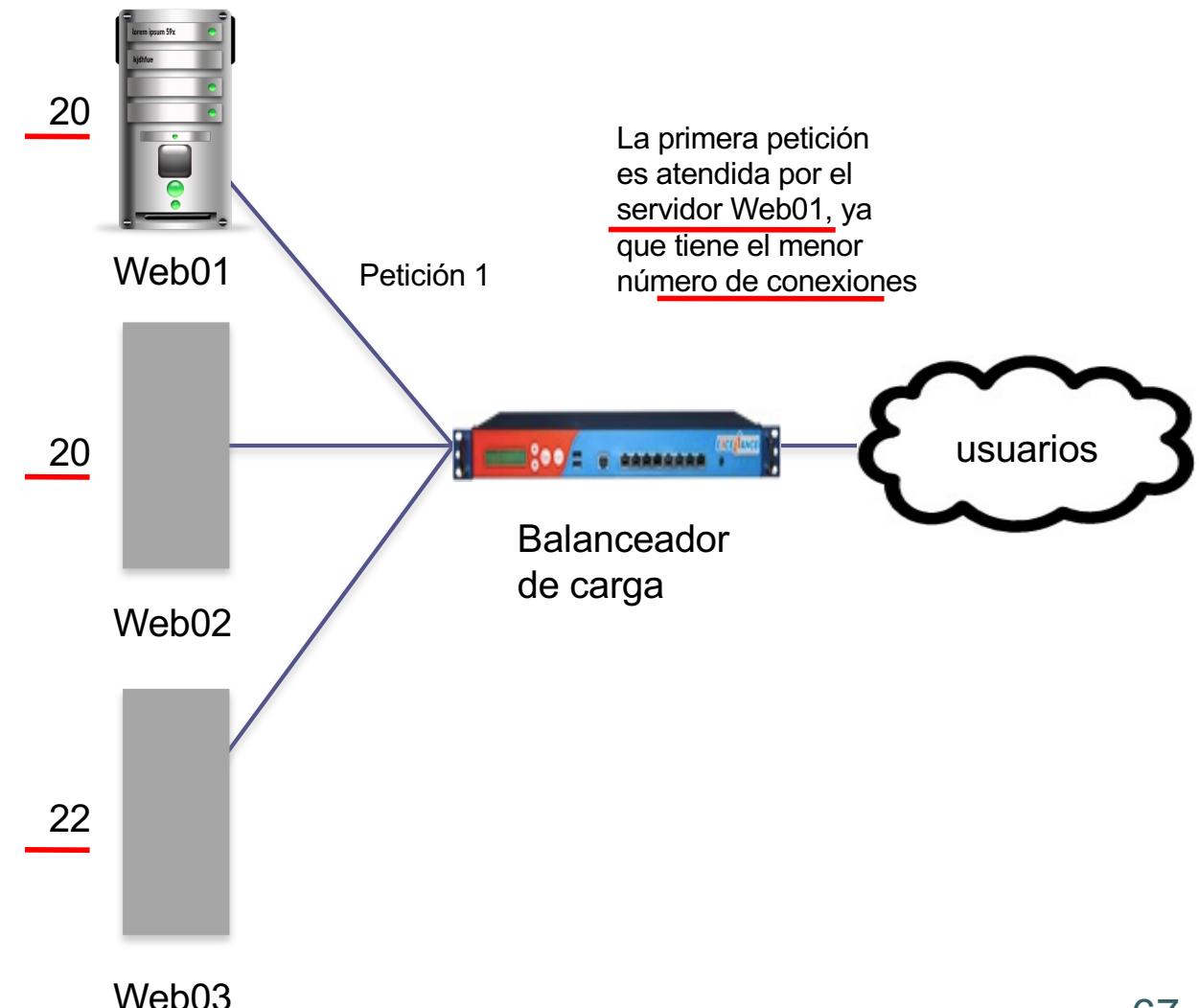


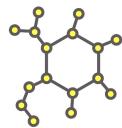
# Algoritmos de balanceo de carga

## 2. Algoritmo de balanceo basado en el menor número de conexiones

El balanceador lleva la cuenta del número de conexiones a cada servidor.

Así repartiremos el trabajo según la utilización de cada una.



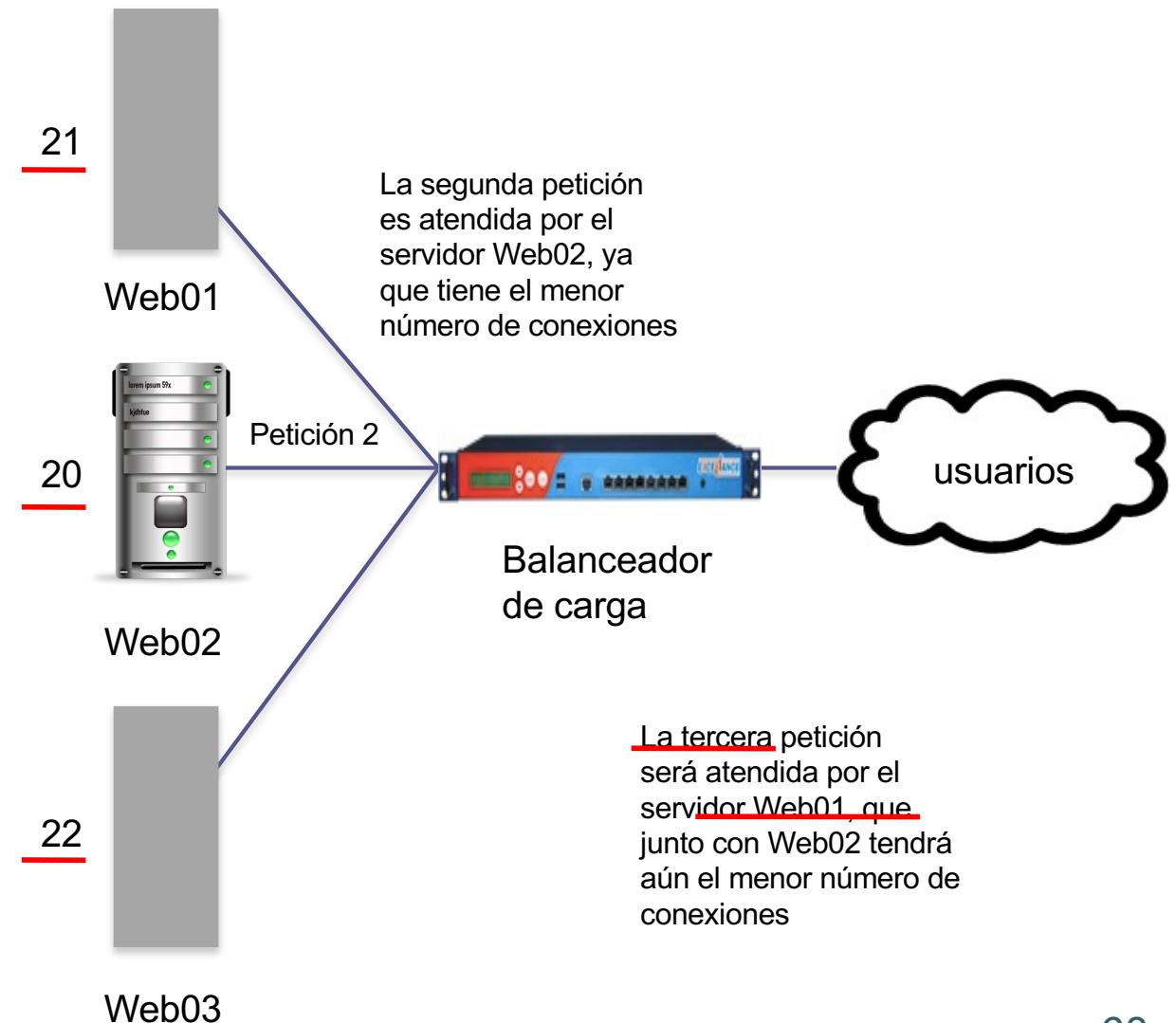


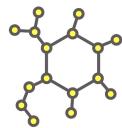
# Algoritmos de balanceo de carga

## 2. Algoritmo de balanceo basado en el menor número de conexiones

Mientras que los servidores 1 y 2 no igualen al 3 en núm. conexiones, seguirán recibiendo.

Sólo entonces se le pasa trabajo al 3.





# Algoritmos de balanceo de carga

## 2. Algoritmo de balanceo basado en el menor número de conexiones

Con este algoritmo se consigue una distribución del trabajo muy adecuada entre máquinas similares.

Se evita la sobrecarga de las que puedan tener más trabajo.

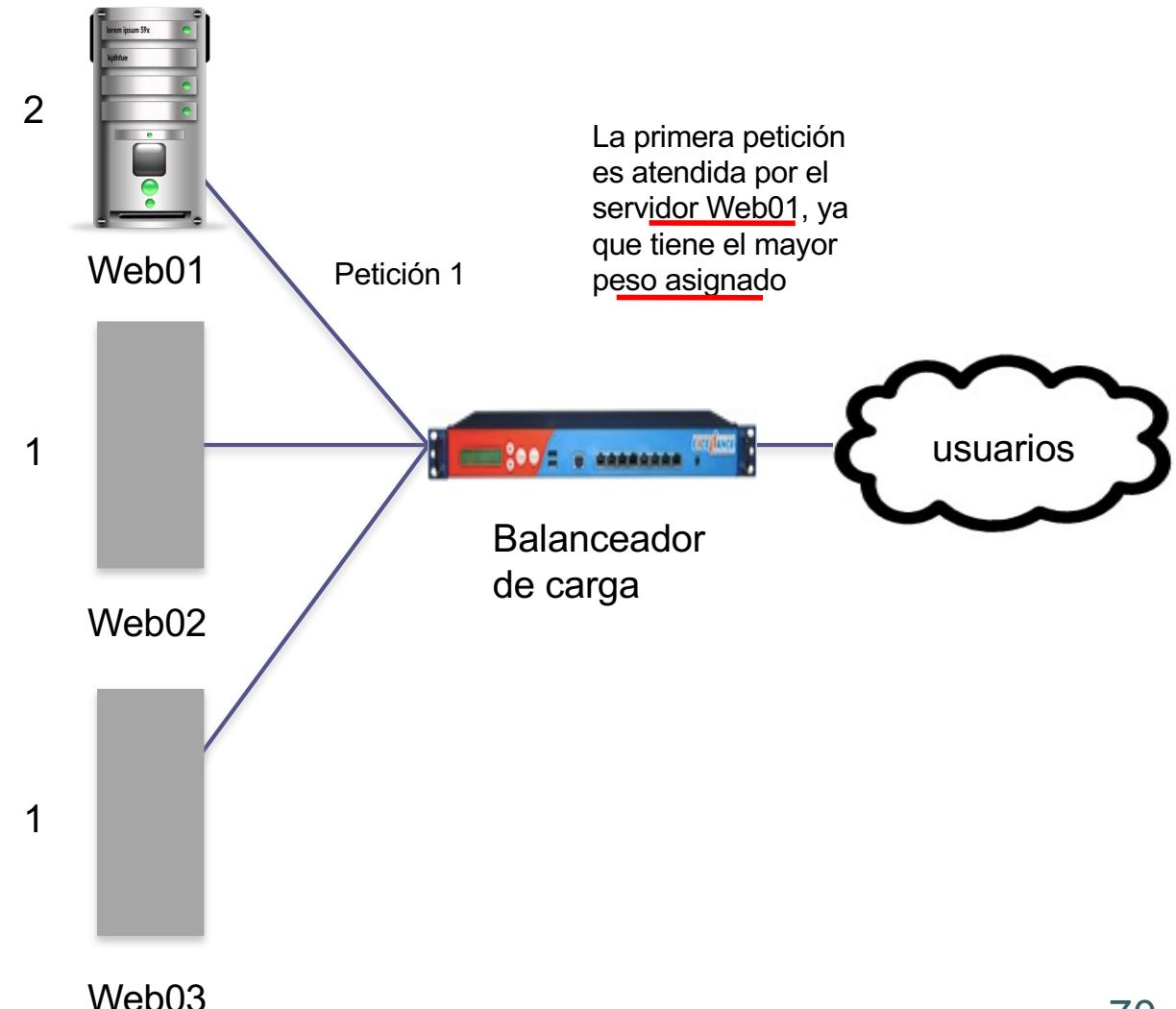


# Algoritmos de balanceo de carga

## 3. Algoritmo de balanceo basado en ponderación

Podemos asignar un peso a cada máquina.

Así controlamos el % de conexiones que le pasamos a cada una.



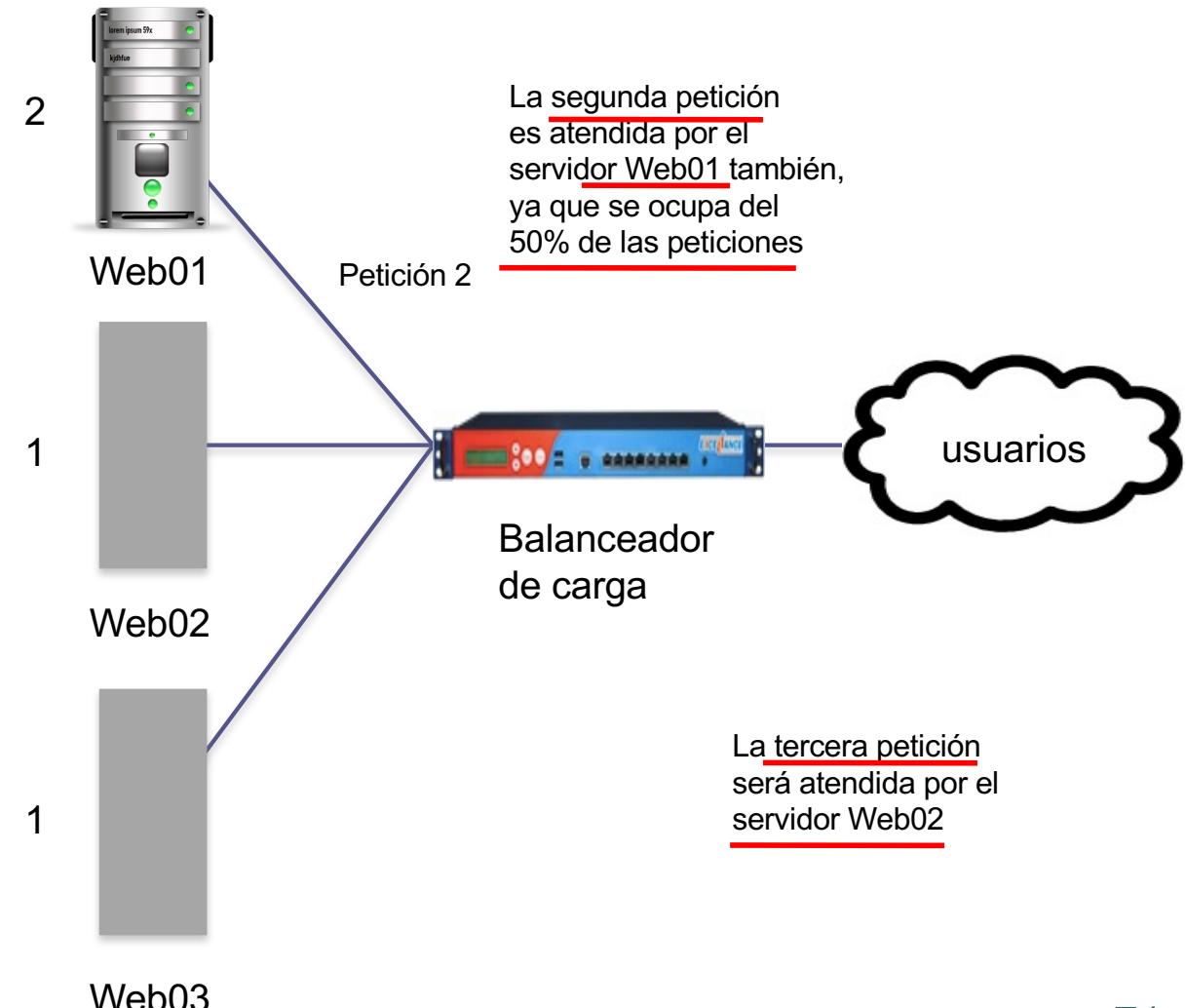


# Algoritmos de balanceo de carga

## 3. Algoritmo de balanceo basado en ponderación

Si en una granja de 4 les damos 25% => todas trabajan igual.

Si damos 50%, 20%, 15%, 15% => la primera se ocupa de la mitad del trabajo y las otras hacen menos.





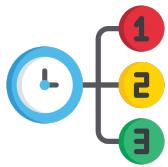
# Algoritmos de balanceo de carga

## 3. Algoritmo de balanceo basado en ponderación

Se puede probar este algoritmo **cuando los dos anteriores no den buenos resultados.**

Puede resultar adecuado

- cuando el número de conexiones no es un buen indicador
- el trabajo por turnos hace que se pierda tiempo en las máquinas más lentas



# Algoritmos de balanceo de carga

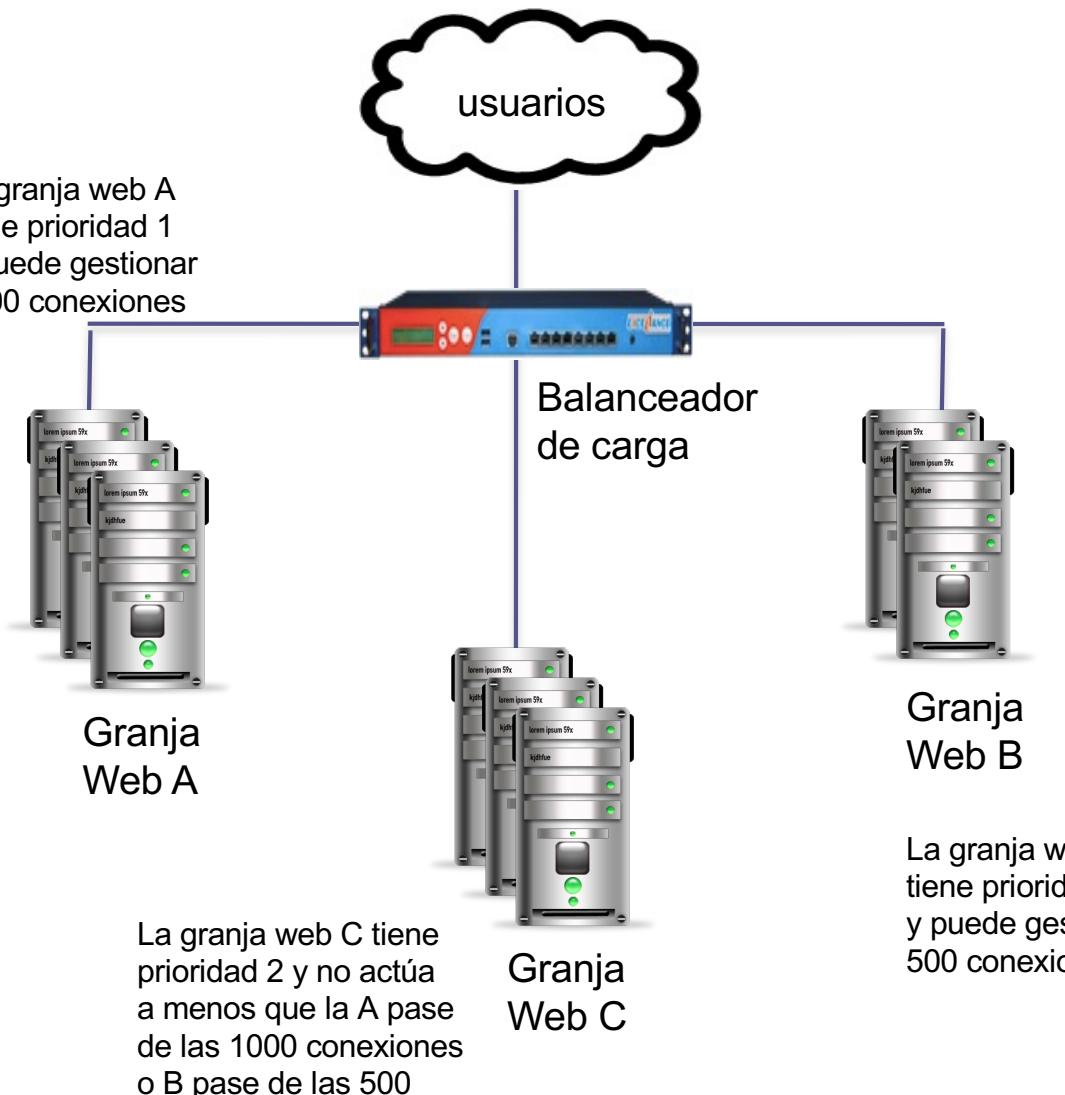
## 4. Algoritmo de balanceo basado en prioridad

Similar a la ponderación pero con grupos.

Cada grupo tiene una prioridad y máximo de conexiones que puede atender.

Hay reparto por turnos dentro del mismo grupo.

La granja web A tiene prioridad 1 y puede gestionar 1000 conexiones





# Algoritmos de balanceo de carga

## 4. Algoritmo de balanceo basado en prioridad

Dentro del mismo grupo se hace reparto por turnos.

Si un grupo recibe más peticiones de las que tiene establecidas entonces el siguiente grupo con más prioridad pasa a recibir.



# Algoritmos de balanceo de carga

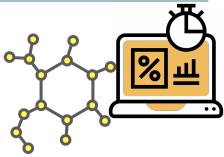
## 5. Algoritmo de balanceo basado en tiempo de respuesta

Método dinámico presente en todos los sistemas de balanceo de carga.

Si una petición tarda más en una máquina concreta, es que está más cargada que el resto.

Los平衡adores pueden ir calculando esos tiempos para decidir a qué máquina le pasa la siguiente petición.

Algunos además pueden calcular y tener en cuenta el uso de CPU y memoria de cada máquina.



# Algoritmos de balanceo de carga

## 6. Combinación de los algoritmos de tiempo de respuesta y menor número de conexiones

Algunos dispositivos平衡adores pueden usar combinaciones de métodos.

Por ejemplo:

tener en cuenta el tiempo de respuesta y el menor número de conexiones para elegir la máquina a la que enviar la siguiente petición.

# Índice



1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. Estructura de la red
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
11. Futuro de las tecnologías de balanceo
12. Resumen y conclusiones

# Balanceo de carga global

Balanceo de carga global (GSLB, *global server load balancing*).

GSLB parte de la idea del balanceo basado en DNS.

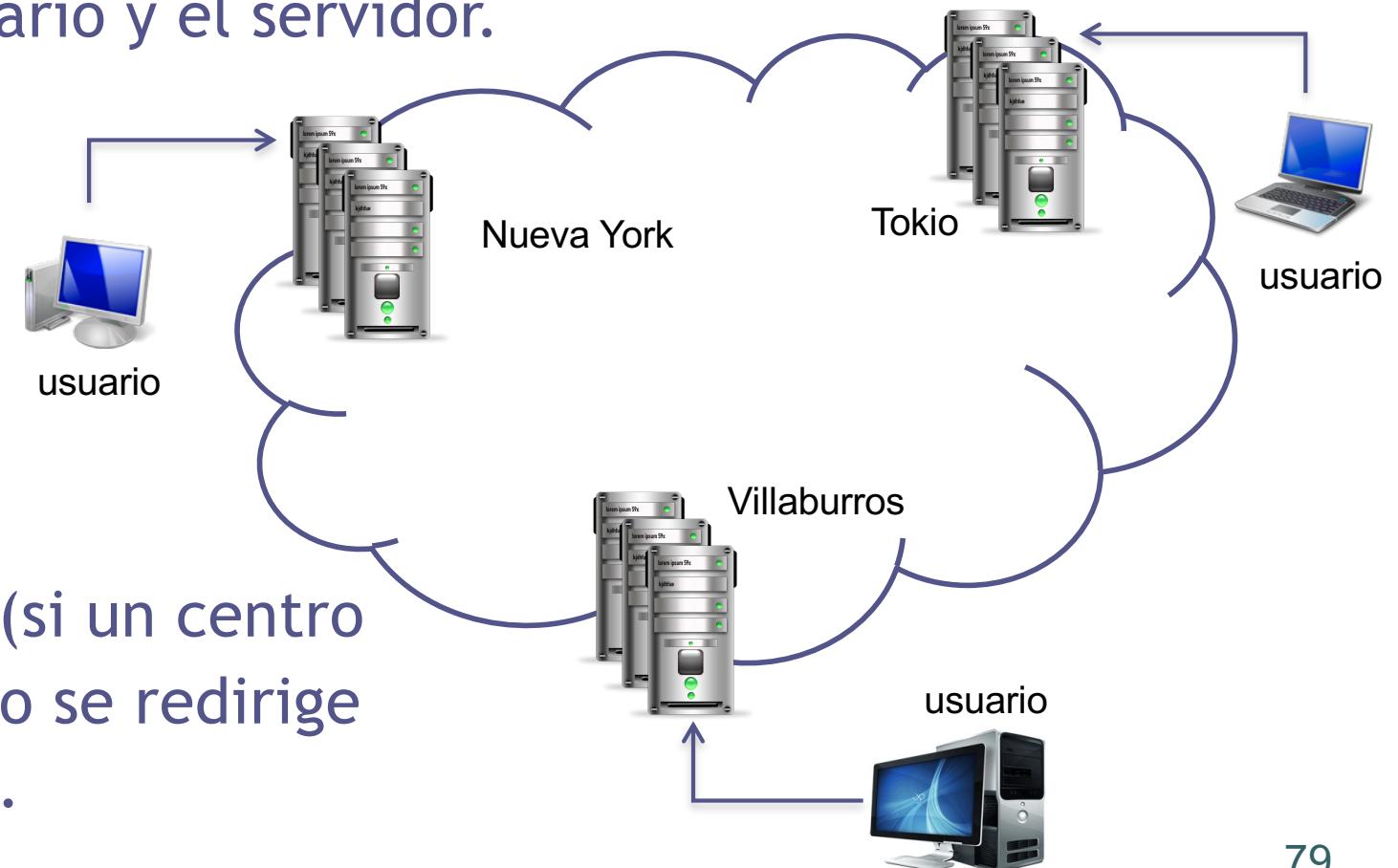
Mejorar el sistema con los conceptos de “alta disponibilidad” y “tiempo de respuesta”.

Queremos evitar una caída total del sistema por un problema en el centro de datos (corte de luz, red, o desastre natural).

# Balanceo de carga global

Distribuir la carga entre varios centros.

Evitar retrasos en las comunicaciones por las distancias entre el usuario y el servidor.



# Balanceo de carga global

Posibles implementaciones:

- Uso del DNS
- Redirección HTTP
- GSLB basado en DNS
- GSLB usando protocolos de enrutamiento



# Balanceo de carga global

## Primera aproximación: uso del DNS

DNS puede usarse para hacer balanceo de carga (turnos).

El mismo funcionamiento podemos aprovecharlo para configurar a nivel de DNS las IP de los平衡adores de varios centros (granjas web).



Así repartiremos la carga entre los centros.

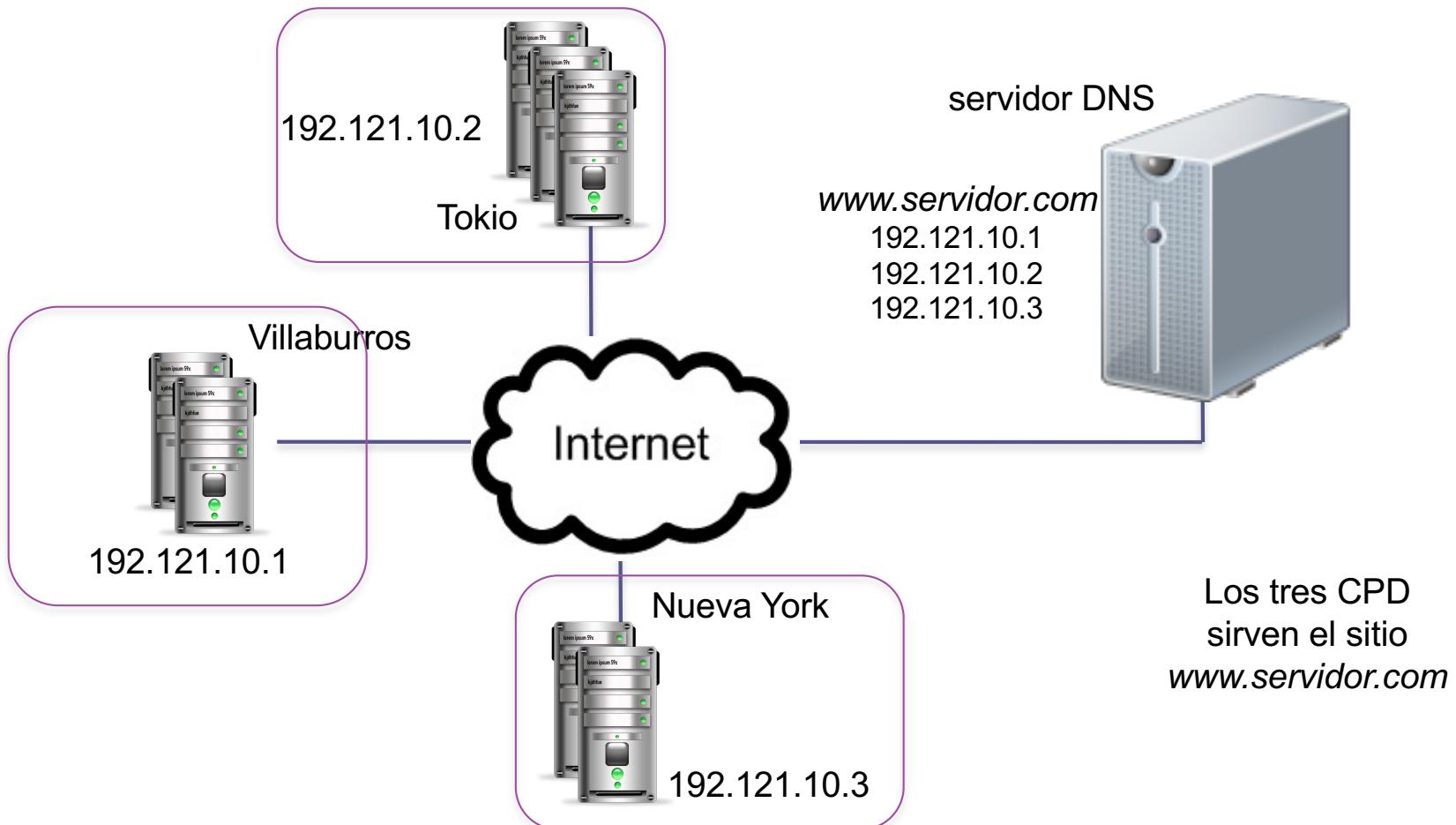


Los DNS no pueden saber si uno de los centros está caído ni reencaminar el tráfico al centro más cercano.

# Balanceo de carga global

## Primera aproximación: uso del DNS

DNS puede usarse para hacer balanceo de carga (turnos).



# Balanceo de carga global

Primera aproximación: uso del DNS



Así repartiremos la carga entre los centros.



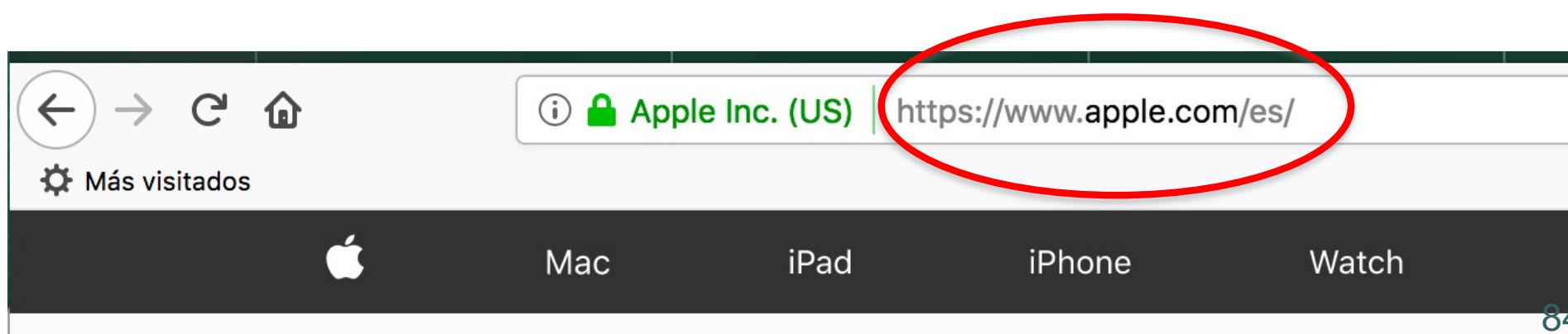
Los DNS no pueden saber si uno de los centros está caído ni reencaminar el tráfico al centro más cercano.

# Balanceo de carga global

## Redirección HTTP

El protocolo HTTP tiene un mecanismo para hacer redirección a otra URL.

Así, si un usuario está en España y accede a la IP de una web, el servidor puede ver la IP del cliente, determinar dónde está exactamente, y redirigirlo a una parte de la web en español o a un servidor en España.



# Balanceo de carga global

## Redirección HTTP

-  Esta técnica no necesita hacer cambios en los DNS. Sólo programación web muy sencilla.
  
-  Necesita dos accesos (latencia). Cuello de botella en la IP.

# Balanceo de carga global

## Redirección HTTP (con HTML, JS o PHP)

<http://es.kioskea.net/faq/537-php-redireccionar-a-otra-pagina-web>  
[http://en.wikipedia.org/wiki/Meta\\_refresh](http://en.wikipedia.org/wiki/Meta_refresh)

```
<?php
    $idioma0 = explode(";", $_SERVER['HTTP_ACCEPT_LANGUAGE']);
    $idioma1 = explode(",", $idioma0[0]);
    $idioma2 = explode("-", $idioma1[0]);
    $id = $idioma2[0];
    switch ($id)
    {
        case 'en':
            header("Location: /en/");
            break;
        case 'es':
            header("Location: /es/");
            break;
    }
?>
```

```
<html>
<head>
<meta http-equiv="Refresh"
      content="5;url=http://servidor.com">
</head>
<body>
</body>
</html>
```

```
<?php header ("Location: http://servidor.com"); ?>
```

```
<body>
<script type="text/javascript">
    window.location="http://servidor.com";
</script>
</body>
```

# Balanceo de carga global

Redirección básica con HTML:

[http://en.wikipedia.org/wiki/Meta\\_refresh](http://en.wikipedia.org/wiki/Meta_refresh)

```
<html>
<head>

<meta http-equiv="Refresh"
      content="5;url=http://servidor.com">

</head>
<body>
</body>
</html>
```

# Balanceo de carga global

## Redirección con PHP (I):

<http://es.kioskea.net/faq/537-php-redireccionar-a-otra-pagina-web>  
<http://stackoverflow.com/questions/6038236/http-accept-language>

```
<?php

$idioma0 = explode(";", $_SERVER['HTTP_ACCEPT_LANGUAGE']);
$idioma1 = explode(", ", $idioma0['0']);
$idioma2 = explode("-", $idioma1['0']);
$id = $idioma2['0'];

switch ($id) {
    case 'en':
        header("Location: /en/");
        break;
    case 'es':
        header("Location: /es/");
        break;
}
?>
```

# Balanceo de carga global

## Redirección con PHP (II):

<http://www.thefutureoftheweb.com/blog/use-accept-language-header>

### Cabecera “Accept-Language”:

en-ca,en;q=0.8,en-us;q=0.6,de-de;q=0.4,de;q=0.2

```
<?php
$langs = array();
if (isset($_SERVER['HTTP_ACCEPT_LANGUAGE'])) {
    preg_match_all('/(([a-z]{1,8})(-[a-z]{1,8})?)\s*(;|\s*q\s*)=\s*(1|0\.[0-9]+))?\s*/i',
        $_SERVER['HTTP_ACCEPT_LANGUAGE'], $lang_parse);

    if (count($lang_parse[1])) {
        $langs = array_combine($lang_parse[1], $lang_parse[4]);

        foreach ($langs as $lang => $val) {
            if ($val === '') $langs[$lang] = 1;
        }
        arsort($langs, SORT_NUMERIC);
    }
}
?>
```

# Balanceo de carga global

## Redirección con JavaScript (I):

<http://blog.joason.com/2013/08/detectar-idioma-del-navegador-y.html>

```
<script>

if (navigator.appName == 'Netscape')
    var language = navigator.language;
else
    var language = navigator.browserLanguage;

if (language.indexOf('en') > -1){
    window.location = 'http://www.google.com';
}else if (language.indexOf('es') > -1){
    window.location = 'http://www.google.es';
}else if (language.indexOf('fr') > -1){
    window.location = 'http://www.google.fr';
}else if (language.indexOf('pt') > -1){
    window.location = 'http://www.google.pt';
}else{
    window.location = 'http://www.ugr.es';
}
</script>
```

# Balanceo de carga global

## Redirección con JavaScript (II):

<http://www.forosdelweb.com/f13/redireccion-segun-idioma-del-navegador-981111/>

```
<script>
var langcodes=new Array("es", "en", "default")
var langredirects=new Array("http://www.google.es", "http://www.google.com", "http://www.ugr.es")

var languageinfo = navigator.language ? navigator.language : navigator.userLanguage
var gotodefault=1

function redirectpage(dest){
    if (window.location.replace)
        window.location.replace(dest);
    else
        window.location=dest;
}

for (i=0;i<langcodes.length-1;i++) {
    if (languageinfo.substr(0,2)==langcodes[i]) {
        redirectpage(langredirects[i]);
        gotodefault=0;
        break;
    }
}
if (gotodefault)
    redirectpage(langredirects[langcodes.length-1]);
</script>
```

# Balanceo de carga global

## GSLB basado en DNS (I)

Balanceo de carga a nivel del DNS para ayudar a seleccionar la IP del mejor sitio.

DNS consiste en un conjunto jerárquico de servidores DNS. Cada dominio o subdominio tiene una o más zonas de autoridad (authoritative DNS).

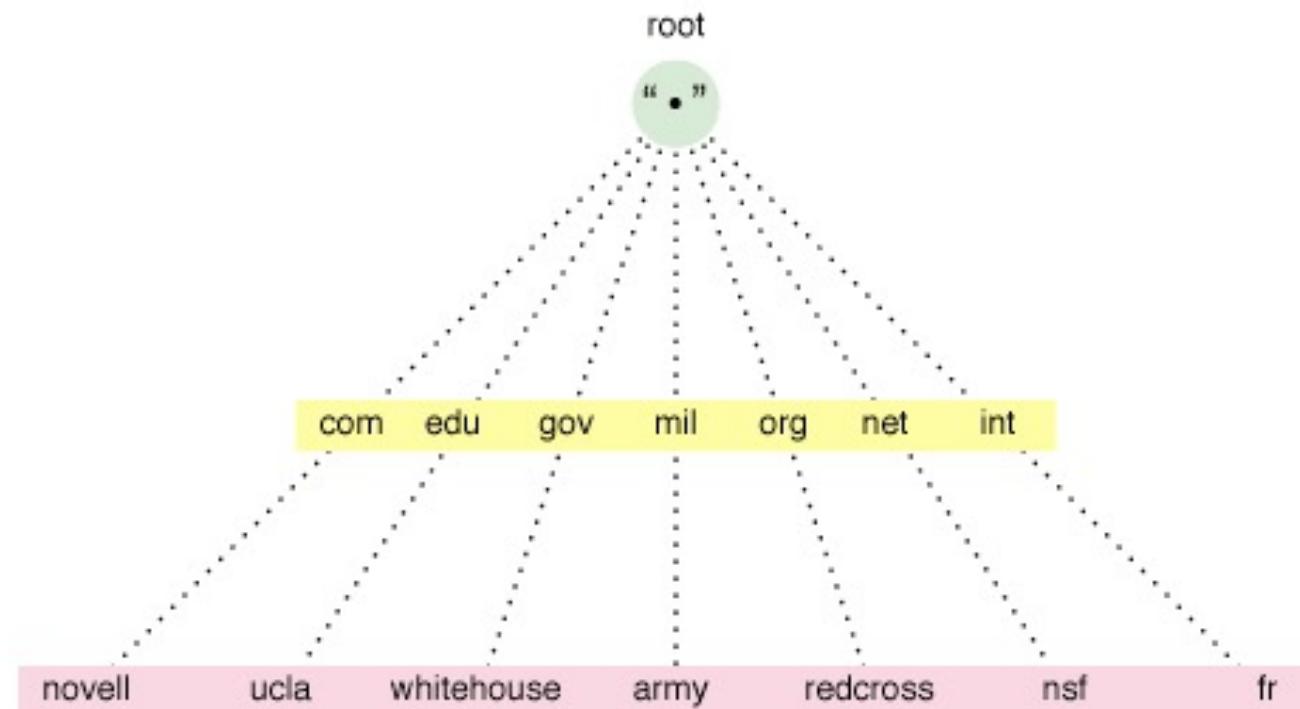
Publican la información acerca del dominio y los nombres de servicios de cualquier dominio incluido.

Al inicio de esa jerarquía se encuentran los servidores raíz.

# Balanceo de carga global

## GSLB basado en DNS (I)

DNS consiste en un conjunto jerárquico de servidores DNS. Cada dominio o subdominio tiene una o más zonas de autoridad (authoritative DNS).



# Balanceo de carga global

## GSLB basado en DNS (II)

Una opción es poner un **balanceador** como servidor DNS a nivel de zona de autoridad.

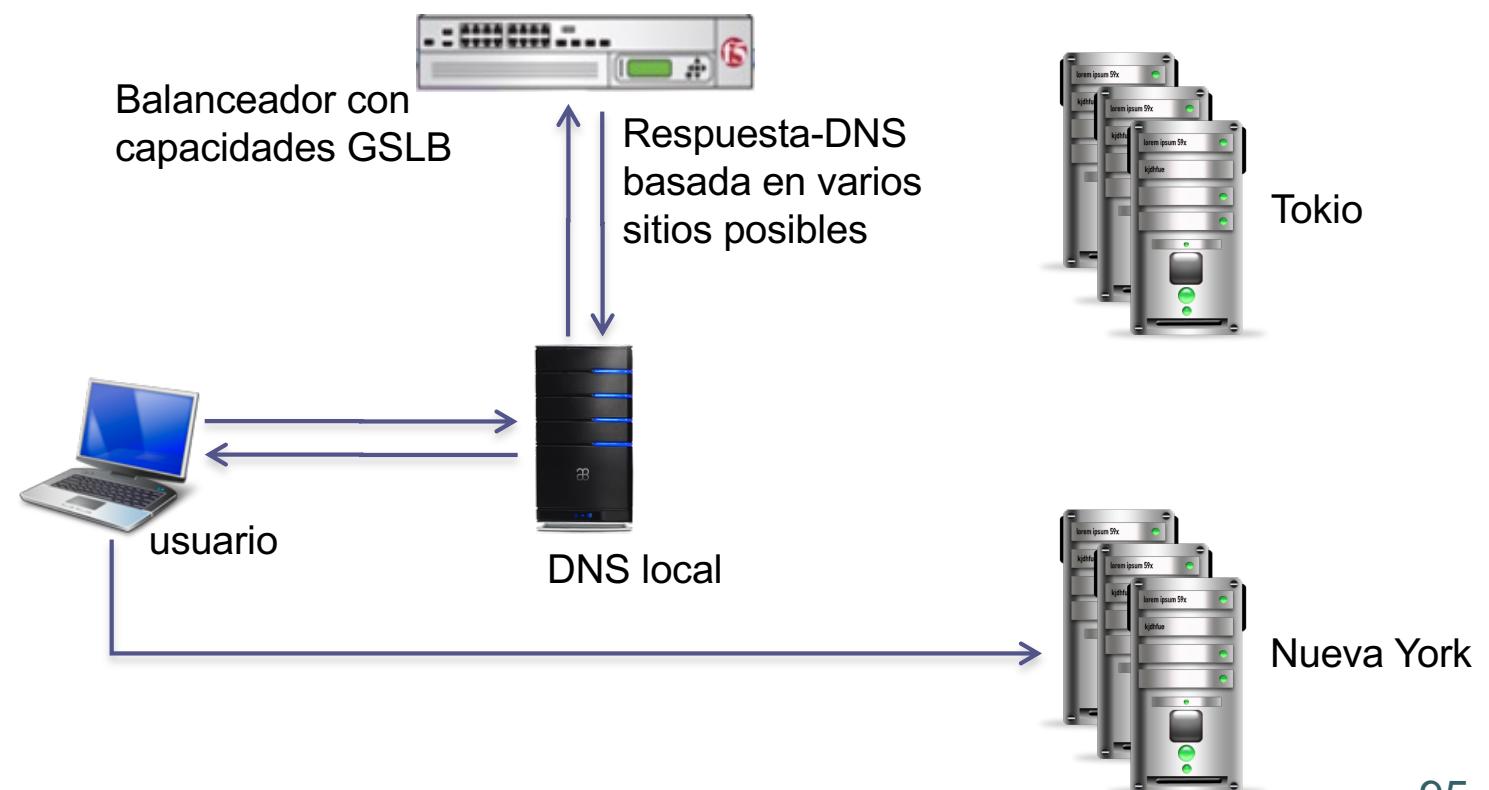
La VIP del balanceador será a la que se enviarán las peticiones al servicio DNS.

Forma en que casi todos los productos GSLB funcionan actualmente.

# Balanceo de carga global

## GSLB basado en DNS (III)

El dispositivo GSLB actúa al nivel *authoritative DNS*.



# Balanceo de carga global

## GSLB basado en DNS (IV)

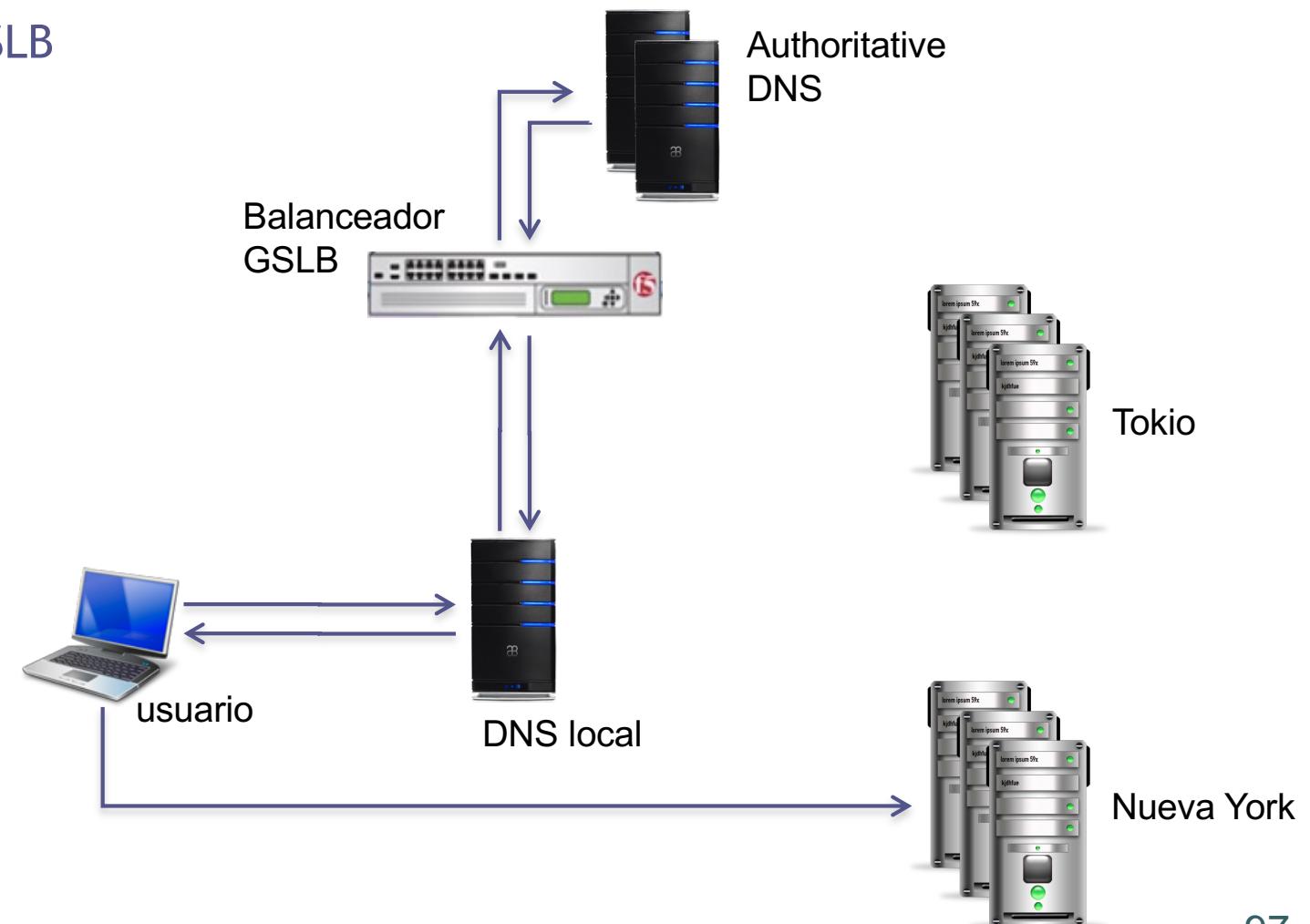
Otra tecnología con la que podemos hacer la integración es configurar un balanceador como proxy.

La idea es poner un **balanceador delante** del "authoritative DNS", de forma que algunas peticiones se pasarán al DNS directamente (y reenvía las respuestas también directamente).

# Balanceo de carga global

## GSLB basado en DNS (V)

El balanceador GSLB  
deja pasar las  
respuestas para  
los sitios que no  
requieren GSLB  
y hace tareas  
de DNS para  
los sitios que  
sí lo requieren



# Balanceo de carga global

## GSLB basado en DNS (VI)

El balanceador puede recoger la respuesta y si hay más de un sitio disponible, la modifica para ofrecer el mejor sitio al cliente.

El balanceador debe hacer ese trabajo extra para modificar la respuesta del DNS para aquellos nombres de dominio que requieren GSLB.

El balanceador no tendrá implementada toda la funcionalidad de un DNS, sino solo la necesaria para mejorar la respuesta del DNS.

# Balanceo de carga global

## GSLB basado en DNS (VII)

Queda determinar el mejor sitio a proponer como respuesta, ya que es algo crítico y que debe decidir el balanceador GSLB a partir sólo de la información intercambiada con el DNS local.

El **GSLB** está monitorizando continuamente el estado de disponibilidad de los diferentes sitios. Así, reenviará tráfico sólo a los sitios activos.

Esta tarea la puede realizar enviando una petición HTTP a cierta URL predefinida, para comprobar el código de retorno devuelto.

# Balanceo de carga global

## GSLB basado en DNS (VIII)

Junto con esa petición HTTP, el balanceador GSLB puede **medir el tiempo de respuesta** que experimenta él.

Cada sitio podrá aceptar un número de conexiones concurrentes diferente. Esta capacidad de aceptar tráfico puede afectar al tiempo de respuesta, aunque no es determinante.

Conviene que el **GSLB conozca la capacidad teórica** de conexiones concurrentes de cada sitio para reenviar tráfico hacia ese sitio. En principio esto mejorará la navegación de los usuarios, pero no asegura nada.

# Balanceo de carga global

## GSLB basado en DNS (IX)

Otra forma de determinar el mejor sitio al que reenviar el tráfico de cierto usuario es **usar información geográfica**.

El GSLB no conoce la IP del usuario, pero sí conoce la del DNS local al usuario (suele ser un DNS cercano al mismo).

La información geográfica **no es el mejor criterio** ya que un sitio más cercano puede estar colapsado por las conexiones.

# Balanceo de carga global

## GSLB basado en DNS (X)

<http://www.ripe.net> ofrece información sobre los bloques de IPs asignadas a la región que incluye Europa y África.

De la misma forma existen webs para obtener información sobre las regiones Asia-Pacífico y América.

# Balanceo de carga global

<https://www.ripe.net/manage-ips-and-asns/dns/reverse-dns>

The screenshot shows a web-based interface for managing IP ranges and Autonomous System Numbers (ASNs). At the top, it displays the RIPE NCC logo and the text "Reverse DNS (150.214.205.46)". Below this, there is a button labeled "Reload this widget by ". A message indicates "1 reverse delegation object related to this prefix found in the RIPE Database". A "Show more fields" link is visible. A detailed table lists the following delegation information:

	domain
descr	214.150.in-addr.arpa
descr	Centro Informatico Cientifico de Andalucia
descr	Av. de Reina Mercedes SN
descr	Sevilla, 41012
nserver	dns1.cica.es
nserver	dns2.cica.es
nserver	sun.rediris.es
nserver	chico.rediris.es
nserver	ns.ripe.net

At the bottom, a note states "Last updated 6 years ago". The footer of the page includes the text "Showing results for 150.214.205.46 as of 2019-04-05 10:33:00 UTC".

# Balanceo de carga global

## GSLB basado en DNS (productos)

El 3DNS de F5 Networks implementa estas ideas:

[http://support.f5.com/content/kb/en-us/archived\\_products/3-dns/manuals/product/3dns4\\_5ref\\_jcr\\_content/pdfAttach/download/file.res/3-DNS\\_Reference\\_Guide,\\_version\\_4.5.pdf](http://support.f5.com/content/kb/en-us/archived_products/3-dns/manuals/product/3dns4_5ref_jcr_content/pdfAttach/download/file.res/3-DNS_Reference_Guide,_version_4.5.pdf)



Otros productos:

[http://www.tractionet.com/index.php?main\\_page=index&cPath=73](http://www.tractionet.com/index.php?main_page=index&cPath=73)

# Balanceo de carga global



## GSLB basado en DNS (desventajas)

GSLB es **complicado de configurar** y poner en funcionamiento, ya que la tecnología DNS no se desarrolló para hacer balanceo de carga.

Además, el funcionamiento de las **cachés de los navegadores y sistemas operativos** hace que ciertas IP se guarden durante horas e incluso días, por lo que si las condiciones de cierto sitio cambian, el usuario seguirá enviando tráfico a dicho sitio.

**Solución:** limpiar la caché del navegador.

<http://www.tenereillo.com/GSLBPageOfShame.htm>

# Balanceo de carga global

## GSLB usando protocolos de enrutamiento (I)

Independiente de la tecnología DNS.

Utilizar a nivel de proveedor de servicio.

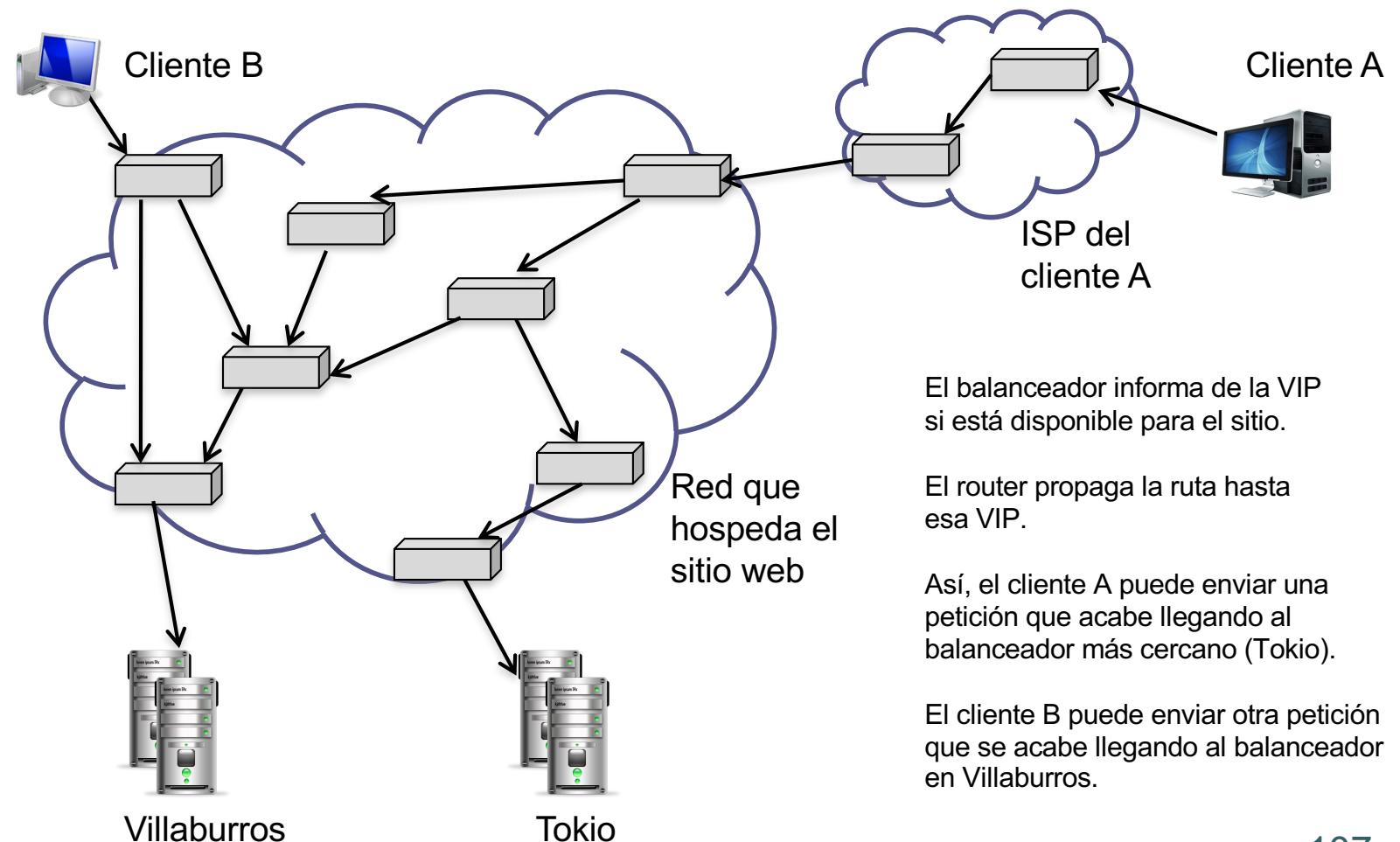
La idea es tener la **dirección del sitio web** hospedada en **diferentes DNS**.

Cuando un DNS responda al usuario con una IP para cierto dominio, esta aproximación dirigirá el tráfico al mejor sitio posible.

Balanceadores de carga en sitios diferentes (A y B), todos configurados con la misma VIP.

# Balanceo de carga global

## GSLB usando protocolos de enrutamiento (II)



# Balanceo de carga global

## GSLB usando protocolos de enrutamiento (III)

Para los routers sólo hay **caminos** para los paquetes.

A nivel de router, que existan dos sitios con la misma VIP significa que **hay dos caminos** para llegar al mismo sitio.

Cuando un usuario A teclea una URL para ir al sitio web, el DNS le facilita la IP (en este caso, la VIP).

El navegador cliente comienza la conexión enviando la petición a la VIP.

# Balanceo de carga global

## GSLB usando protocolos de enrutamiento (IV)

Esta petición se propagará por la red y llegará al router A, que mirará en su tabla de enrutamiento para decidir por qué ruta lo envía.

Encontrará dos "caminos" hasta la VIP, y en función de algún algoritmo de camino mínimo, decide hacia dónde enviar este tráfico.

Como ejemplo de protocolo de enrutamiento cabe destacar OSPF (Open Shortest Path First).

# Balanceo de carga global

## GSLB usando protocolos de enrutamiento (V)



### Possible problema:

Cuando un usuario navega, abre varias conexiones TCP. Los routers recibirán los paquetes generados y los reenviarán hacia cualquiera de los balanceadores de los sitios, en función del estado de la red.

Sin embargo para que la comunicación funcione, todos los paquetes deben ir al mismo balanceador.

Si en un momento, algún router decide cambiar la ruta hacia otro de los balanceadores, se perderá la integridad de la comunicación (las aplicaciones web necesitan mantener la persistencia en las comunicaciones).

# Balanceo de carga global

## Resumen

-  GSLB presenta ventajas: alta disponibilidad.
-  Es muy complejo de implementar y comprender.
- Requiere de conocimientos de los DNS.
-  Si la aplicación web usa BD, éstas deben estar sincronizadas entre los diversos sitios. Muy complejo.
- Para que GSLB funcione es necesario un entorno de red muy controlado, con alta coordinación entre los operadores.

# Índice



1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. Estructura de la red
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
11. Futuro de las tecnologías de balanceo
12. Resumen y conclusiones

## Ejemplo: recorrido de un paquete a través de un balanceador

Veamos cómo funcionaría el tráfico de red a través de un sistema web con un balanceador de carga.

Supongamos 4 máquinas servidoras:

- m1 y m2 sirven HTTP
- m3 sirve FTP
- m4 sirve SMTP
- b hace de balanceador

Así se desacoplan las aplicaciones o servicios de las máquinas, dando alta flexibilidad.

## Ejemplo: recorrido de un paquete a través de un balanceador

Un cliente establece una conexión TCP, envía la petición HTTP, recibe la respuesta y cierra la conexión TCP.

b recibe en el inicio de la conexión un paquete que lleva como origen la IP del cliente y como destino la VIP del sistema web. También indica el puerto del servicio.

b comprueba la disponibilidad de m1 y m2.

b selecciona a una de las dos máquinas y cambia en el paquete la VIP por la IP privada de la máquina a la que enviará ese tráfico de ese cliente.

## Ejemplo: recorrido de un paquete a través de un balanceador

La máquina servidora envía datos como respuesta.

Los paquetes pasan por el balanceador, que tiene que cambiar la IP privada de la máquina servidora por la VIP antes de enviar el paquete hacia el cliente (que espera recibir los paquetes con origen la VIP).

# Índice

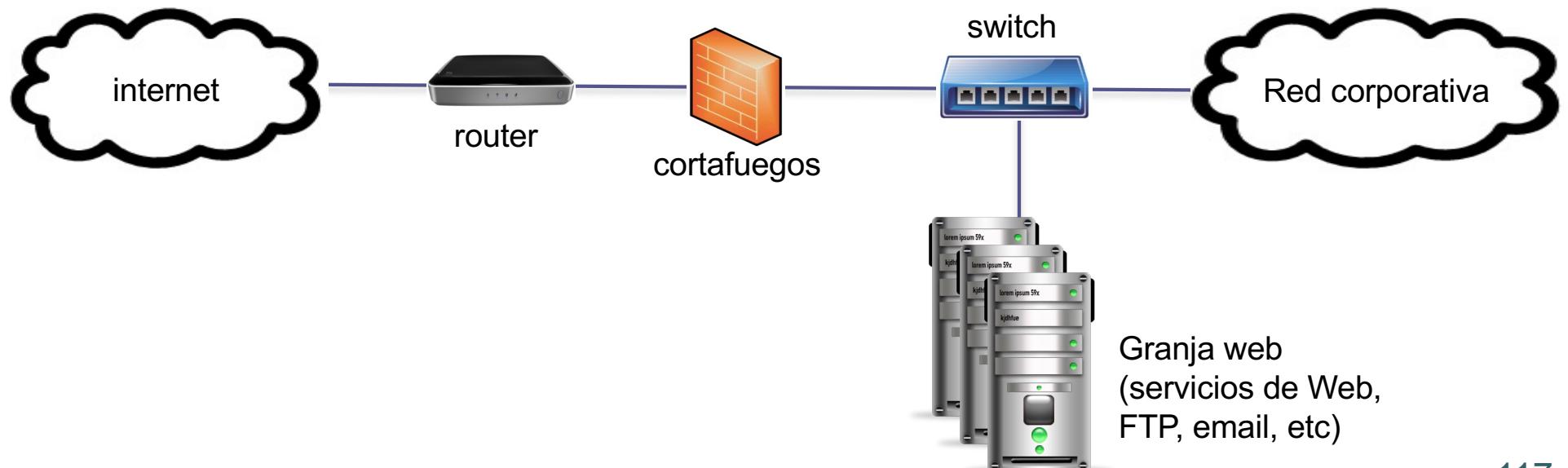


1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. Estructura de la red
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
11. Futuro de las tecnologías de balanceo
12. Resumen y conclusiones

# Ejemplo: diseño de la red de una empresa

Esquema, a alto nivel, de la red de una empresa con unas necesidades especiales en cuanto a seguridad, escalabilidad y alta disponibilidad del sitio web.

Partiremos del siguiente diseño de red:

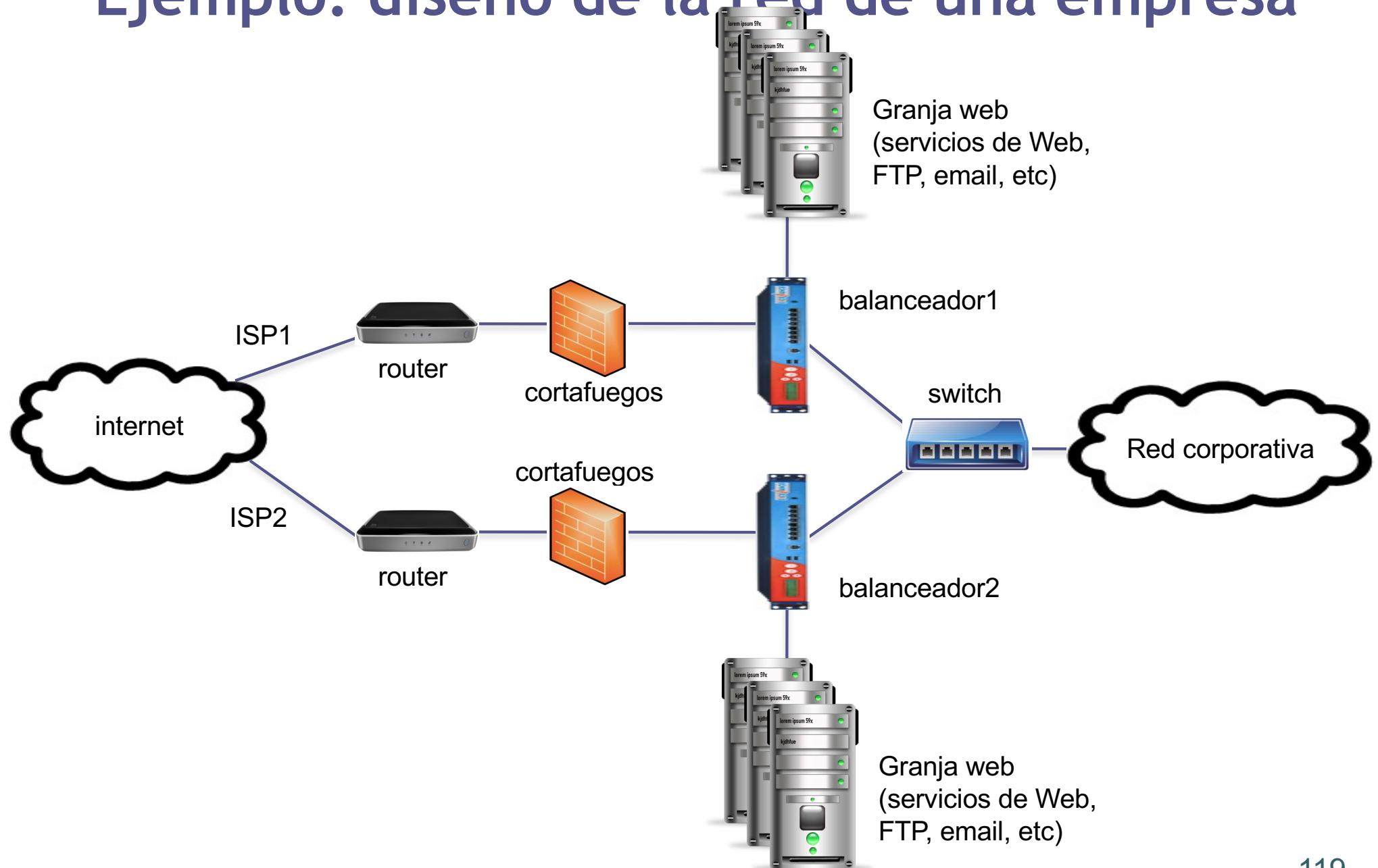


# Ejemplo: diseño de la red de una empresa

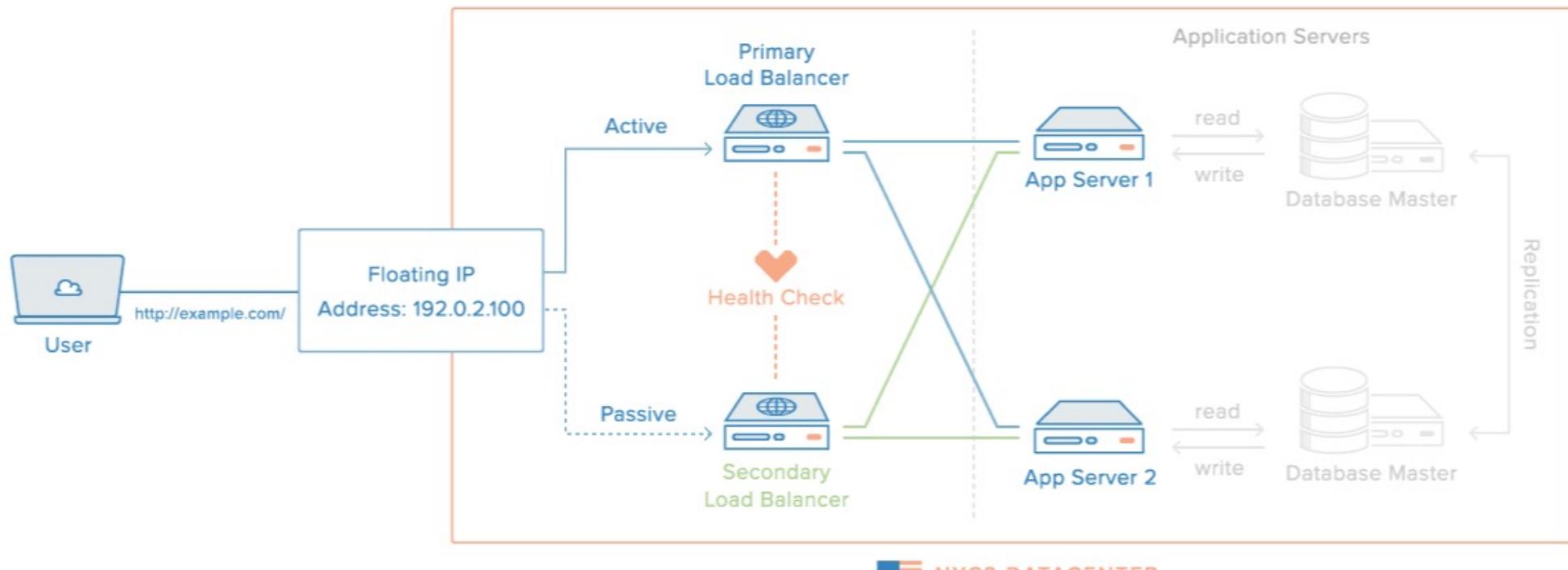
Modificar ese diseño inicial para mejorar la disponibilidad, escalabilidad y usabilidad:

- añadiremos dos routers así como dos conexiones replicadas a sendos proveedores de Internet, replicando cortafuegos
- podemos definir una zona segura de la red y mover los servidores web y FTP a esa zona DMZ
- para realizar balanceo de carga, mejorando así la escalabilidad, disponibilidad y manejabilidad
- replicar la instalación y configuración de red en varias localizaciones (países) para llevar a cabo balanceo global

# Ejemplo: diseño de la red de una empresa



# Ejemplo: diseño de la red de una empresa



- 1 Active/Passive Cluster is healthy
- 2 Primary node fails
- 3 Floating IP is assigned to Secondary node

# Índice



1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. Estructura de la red
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
- 10. Ejemplo 3**
11. Futuro de las tecnologías de balanceo
12. Resumen y conclusiones

# Ejemplo: redes de distribución de contenidos

Comentar la función de los balanceadores de carga en las redes de distribución de contenidos.

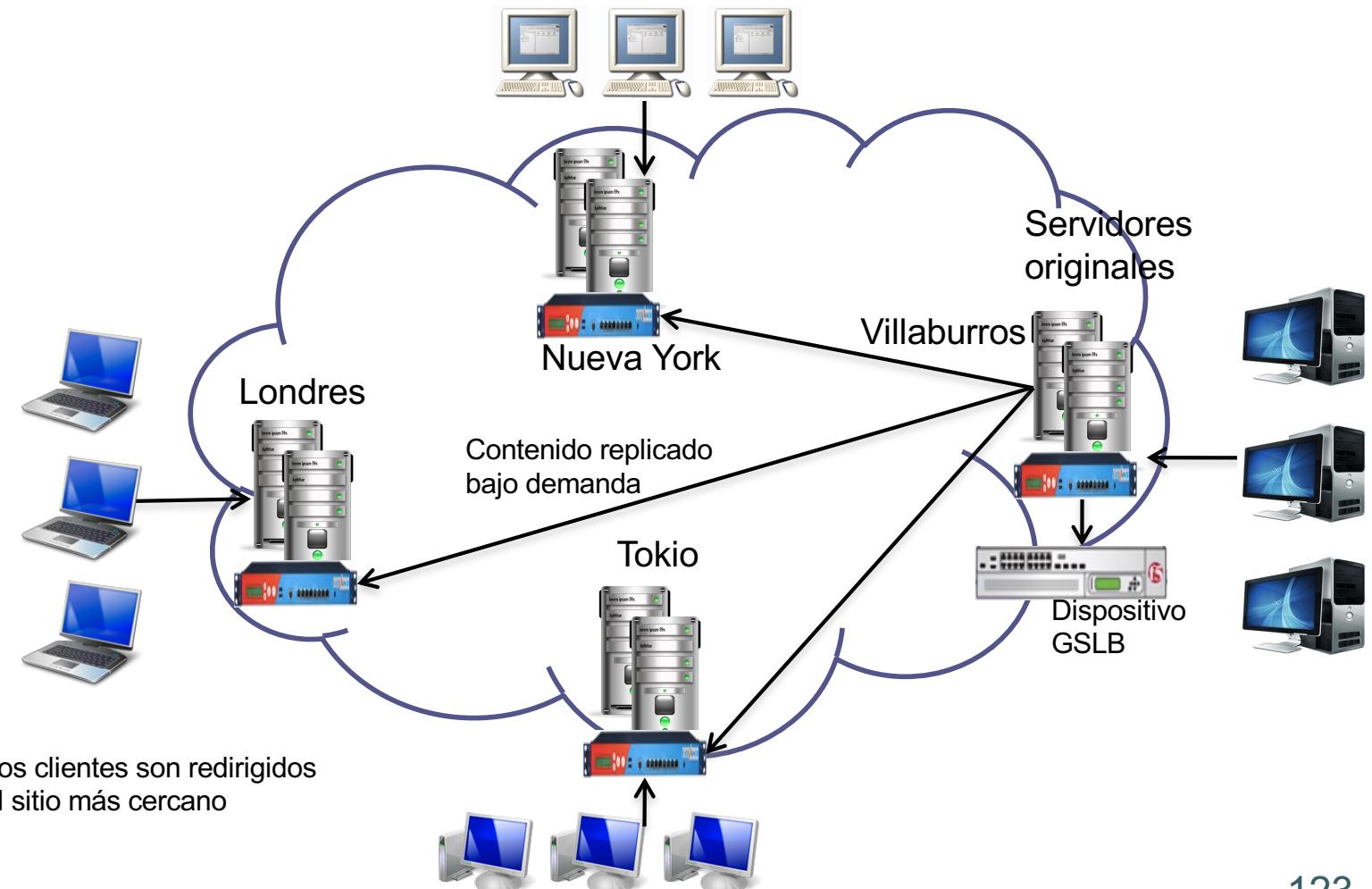
Los proveedores de contenido deben ofrecer el mejor tiempo de respuesta posible a los usuarios finales.

Localizar los centros de datos cerca de los usuarios.

Las empresas despliegan réplicas (caches) en diferentes centros de datos para servir el contenido estático.

# Ejemplo: redes de distribución de contenidos

Una posibilidad es usar GSLB:



# Ejemplo: redes de distribución de contenidos

Hay empresas que consiguen reducir los costes de este tipo de instalaciones usando los centros de datos existentes (especie de alquiler).

Todo el coste de la instalación y mantenimiento de la red en varios centros de datos por todo el mundo se lo ahorra la empresa.

Mediante GSLB se mejora la disponibilidad y el usuario experimenta menores tiempos de respuesta.

# Índice



1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. Estructura de la red
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
- 11. Futuro de las tecnologías de balanceo**
12. Resumen y conclusiones

# Futuro de las tecnologías de balanceo

Reducir precios, incrementar prestaciones y funcionalidades.

Evolución de los balanceadores: cubrir las necesidades de servidores de ficheros, base de datos y otras aplicaciones.

Mejora de las prestaciones y reducción de la latencia.

Tareas relativas a la seguridad.

Servir directamente contenido estático (espacio en RAM)

El balanceador podrá integrar hardware y software para acelerar el procesamiento de datos SSL.

# Índice



1. Introducción
2. Funcionamiento básico de un servidor
3. Conceptos del balanceo de carga
4. Otras tecnologías
5. Estructura de la red
6. Algoritmos de balanceo de carga
7. Balanceo de carga global
8. Ejemplo 1
9. Ejemplo 2
10. Ejemplo 3
11. Futuro de las tecnologías de balanceo
- [12. Resumen y conclusiones]**

# Resumen y conclusiones

El balanceo de carga ha permitido ofrecer más servicios a un número creciente de usuarios.

El balanceador reparte el tráfico web entre varios servidores, y realiza comprobaciones para asegurar la disponibilidad.

Diversos algoritmos de balanceo de carga para repartir el tráfico entre los servidores.

Aporta diversos beneficios: escalabilidad, disponibilidad, mantenimiento, seguridad, calidad de servicio.

# TEMA 5

## Asegurar la granja web

SWAP

¿Qué podemos hacer para proteger y  
asegurar la granja web?  
¿Evitará acciones maliciosas?



José Manuel Soto Hidalgo  
Dpto. Arquitectura y Tecnología de Computadores  
Universidad de Granada

jmsoto@ugr.es

# Índice



- [ 1. Introducción ]
- 2. Defensa en profundidad
- 3. Políticas de seguridad
- 4. Asegurar un servidor
- 5. Cortafuegos
- 6. Evitar ataques
- 7. Prácticas de seguridad recomendadas
- 8. Conclusiones

# Introducción

Asegurar la granja web es una tarea muy importante para cualquier sitio web.

Puede permitir además, saber quién hizo cada cosa y en qué momento.

La seguridad es fundamental para proteger los datos propiedad de la empresa y la información de los usuarios.

El fin último es evitar (o al menos dificultar en lo posible) que un hacker malicioso realice cualquier acción que afecte al sistema.

# Introducción

Se trata de **asegurar y mejorar la disponibilidad del sitio** y también de asegurarse de que las **operaciones** que se lleven a cabo en el sitio sean **seguras**.

Las políticas de seguridad y los procedimientos para implementar esas políticas son clave en el diseño de una granja web.

# Introducción

Los objetivos de seguridad deben definirse correctamente y se basan generalmente en los siguientes conceptos:

- **Confidencialidad:** las comunicaciones deben ser secretas.
- **Integridad:** los mensajes enviados deben ser exactamente los recibidos.
- **Disponibilidad:** la comunicación con cualquier aplicación o servicio de la granja web debe estar disponible en el momento en que sea requerida.

# Introducción

En este tema trataremos:

- Comprender el concepto de **defensa en profundidad** (diferentes capas de defensa).
- Establecer **políticas de seguridad**, incluyendo claves seguras, para todas las cuentas.
- Asegurar un servidor mediante la **eliminación de servicios innecesarios y vulnerabilidades**.
- **Usar un cortafuegos**: comprender el funcionamiento de los cortafuegos y los beneficios de estos.

# Índice



1. Introducción
2. Defensa en profundidad
3. Políticas de seguridad
4. Asegurar un servidor
5. Cortafuegos
6. Evitar ataques
7. Prácticas de seguridad recomendadas
8. Conclusiones

# Defensa en profundidad

Importancia de la arquitectura de seguridad.

Incluso en el mundo real, se controla el acceso a los recursos de un edificio o empresa con varias capas.

EJ: *en un banco hay varios niveles de seguridad para proteger el dinero (varios sistemas de seguridad de diferente tipo que superar para hacerse con el dinero):*

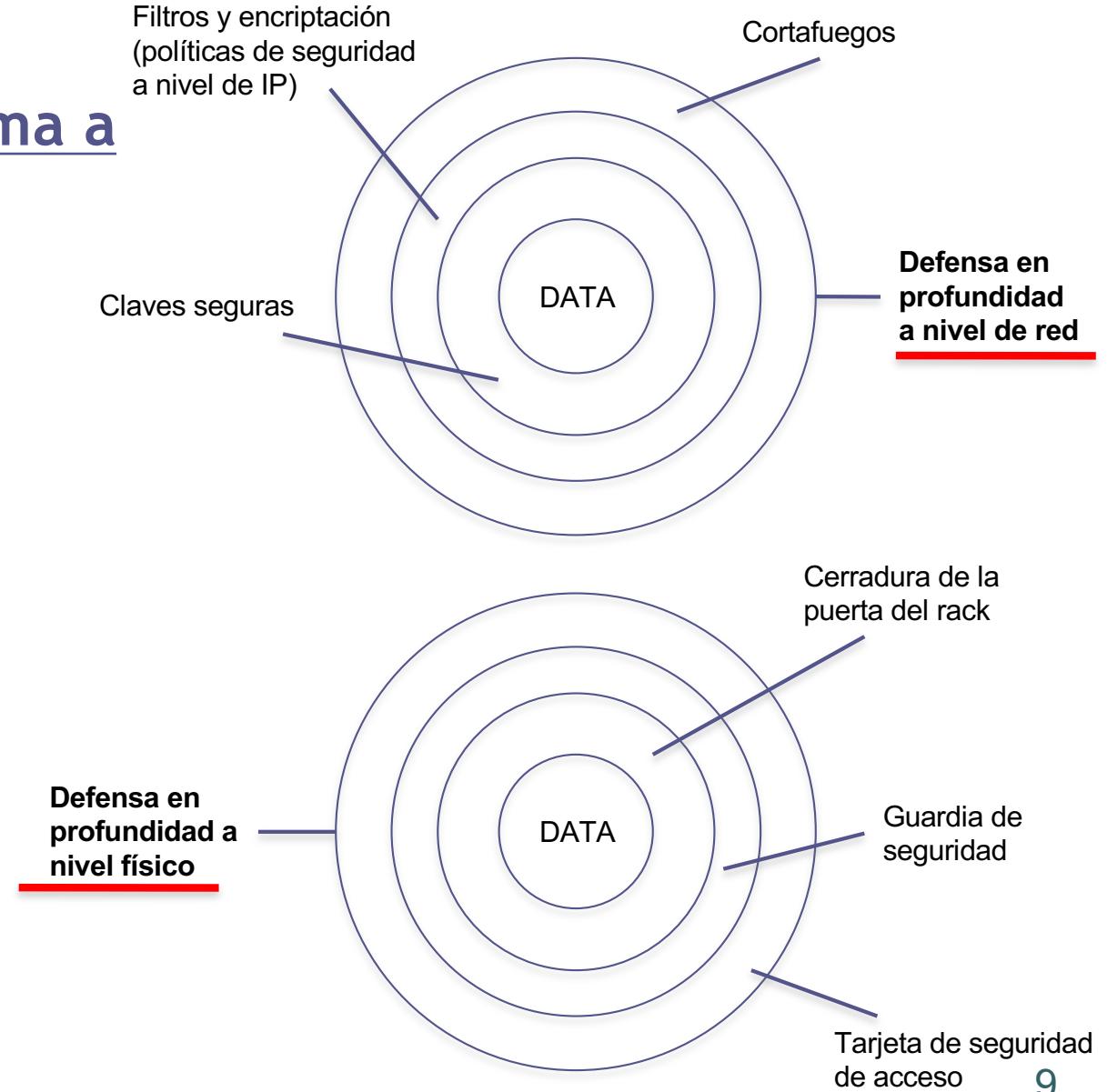
- (1) *el dinero está guardado en cajas fuertes. Para acceder a ese dinero, los clientes deben identificarse.*
- (2) *el banco utiliza video-vigilancia y mantiene registros detallados de todas las transacciones.*

¿Por qué no en una granja web?

# Defensa en profundidad

Protección del sistema a diferentes niveles.

Habrá que superar cada una de las capas independientemente para acceder a los datos



# Defensa en profundidad

¿Son necesarios tantos niveles?

**Sí**

Ningún sistema de seguridad es totalmente seguro...

La forma de complicarle la tarea a un hacker malicioso es poner más de un nivel de seguridad.

Incrementar el tiempo necesario para superar cada nivel hace que sea más probable detectar un ataque, y así evitar que las últimas defensas se vean comprometidas.

# Defensa en profundidad

Importante estar al día en cuanto a temas de seguridad en todos los frentes.

El administrador responsable de la seguridad informática debe conocer los temas relativos a la seguridad así como las vulnerabilidades a nivel de red, de cortafuegos, de sistema operativo y de las aplicaciones en el sistema web.

# Defensa en profundidad

Hay que estar pendientes a los grupos de noticias, listas de correo, blogs y foros sobre estos temas.

Cuando se identifica una vulnerabilidad, los administradores de seguridad deben tomar **medidas de prevención** ya que siempre habrá quien esté atento para aprovecharla...

Estas investigaciones y estudios sobre seguridad en ciertas organizaciones suelen **revelar los puntos débiles** de los sistemas web de otras en las que no aplican políticas de seguridad.



# Índice

1. Introducción
2. Defensa en profundidad
- 3. Políticas de seguridad**
4. Asegurar un servidor
5. Cortafuegos
6. Evitar ataques
7. Prácticas de seguridad recomendadas
8. Conclusiones

# Políticas de seguridad

Las políticas de seguridad definen cómo se les permite interaccionar a los usuarios con los servidores y el hardware del sistema web.

Todas las políticas definen:

- procedimientos de identificación y acceso
- o privilegios de uso (qué acciones puede llevar a cabo cada tipo de usuario).

# Políticas de seguridad

Los procedimientos de identificación comienzan solicitando una identificación (nombre de usuario + clave). De la validez de esta identificación dependerá que se permita o deniegue el acceso.

¿Qué se suele utilizar?

- Una clave o PIN
- Una tarjeta física que incluirá la clave
- Un escáner de retina, huella dactilar o ADN

...del menos efectivo al más efectivo.

Usar dos, especialmente si el primero es una simple clave.

# Políticas de seguridad

retina / huella / tarjeta



# Políticas de seguridad

Ejemplo: algunas empresas usan dos factores:

El primero, una **tarjeta de identificación** con la que se le permite a los empleados acceder a ciertas áreas.

El segundo suele ser una **identificación** (usuario y clave) en la red de ordenadores. Con ella podrá acceder a ciertos recursos, aunque a ciertas otras máquinas no.

En los **dominios de seguridad** los administradores definen listas de control de acceso (usuarios o grupos que pueden acceder a ciertos recursos concretos).

# Políticas de seguridad

## Formas de usar la huella dactilar...

→ C ⌂ https://www.elsiglodetorreon.com.mx/noticia/163290.asesina-a-su-marido-y-le-corta-el-dedo-para-c.html

SUCESOS | El Siglo de Torreón | lun 8 ago 2005, 11:22am | 4 de 7

## Asesina a su marido y le corta el dedo para cobrar pensión



✉ ENVIAR  
★ FAVORITO  
🖨 IMPRIMIR  
💬 COMENTAR

**Bogotá, (Notimex).- La policía colombiana capturó a una mujer que asesinó a su esposo en el año 2000 y luego le cortó y congeló el dedo índice derecho para poder estampar mensualmente su huella digital en un poder para cobrar su pensión.**

# Políticas de seguridad

## Tipos de sensores biométricos:

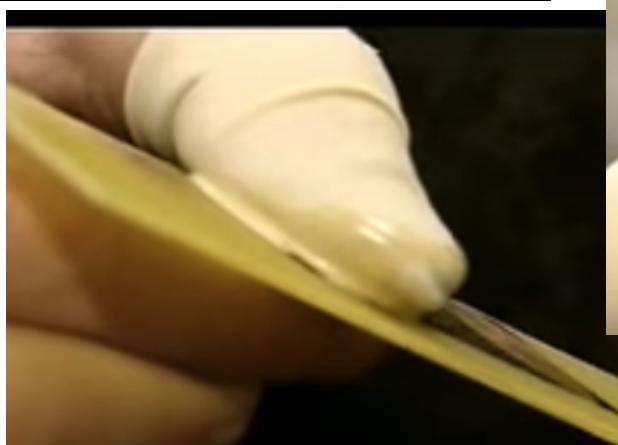
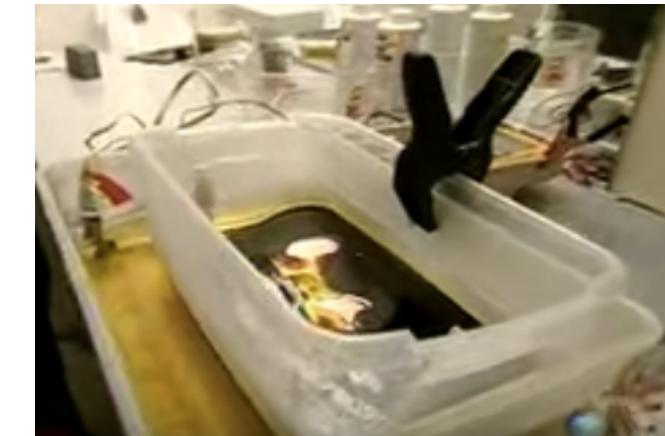
	Ojo (Iris)	Ojo (Retina)	Huellas dactilares	Vascular dedo	Vascular mano	Geometría de la mano	Escritura y firma	Voz	Cara 2D	Cara 3D
<b>Fiabilidad</b>	Muy alta	Muy Alta	Muy Alta	Muy Alta	Muy Alta	Alta	Media	Alta	Media	Alta
<b>Facilidad de uso</b>	Media	Baja	Alta	Muy Alta	Muy Alta	Alta	Alta	Alta	Alta	Alta
<b>Prevención de ataques</b>	Muy alta	Muy Alta	Alta	Muy Alta	Muy Alta	Alta	Media	Media	Media	Alta
<b>Aceptación</b>	Media	Baja	Alta	Alta	Alta	Alta	Muy Alta	Alta	Muy alta	Muy alta
<b>Estabilidad</b>	Alta	Alta	Alta	Alta	Alta	Media	Baja	Media	Media	Alta

<http://www.interempresas.net/Seguridad/Articulos/50527-Lectores-de-reconocimiento-biometrico-seguridad-y-control-de-acceso.html>

# Políticas de seguridad

¿se puede engañar a un sensor biométrico?

YouTube<sup>ES</sup>



# Políticas de seguridad

## ¿se puede engañar a un sensor biométrico?

 [clipset.20minutos.es/se-puede-enganar-al-iphone-5s-con-un-dedo-cortado/](http://clipset.20minutos.es/se-puede-enganar-al-iphone-5s-con-un-dedo-cortado/)

## ¿Se puede engañar al iPhone 5s con un dedo cortado?

Por Juan Castromil (@castromil) el 17/09/2013 | 10 comentarios

Tus dedos son tu identidad digital para el iPhone 5s, pero ¿están a salvo tus dedos?

**La respuesta corta es no.** La respuesta larga es que el sistema de reconocimiento utilizado en el iPhone 5s, desarrollado por **AuthenTec** y basado en un **sensor capacitivo RF**, utiliza tecnología capacitiva para determinar patrón eléctrico formado por las huellas dactilares de cada dedo. Esto significa que, a diferencia de los escáneres ópticos que sólo ven la imagen formada por el contorno de los surcos sub epidérmicos, el sistema de Apple localiza algunos puntos clave del 'circuito eléctrico' que forma la huella dactilar.

# Políticas de seguridad

¿se puede engañar a un sensor biométrico?

<https://www.youtube.com/watch?v=C2cVAQmcMf0>



How to make the fakefingerprints (VIRDI)  
VIRDI Biometric  
187.963 visualizaciones



Falsificación de Huellas Digitales en Control de Acceso  
Cybertronics Security  
31.561 visualizaciones



Con huellas dactilares falsas checaba por compañeros - El Universal  
Miguel Angel Sanchez Pacheco  
11.437 visualizaciones



Seguridad Fisica - Duplicacion de Huellas y Acceso  
Lockpick AR  
7.420 visualizaciones

# Políticas de seguridad

**Aplicar políticas a diferentes niveles:**

- 1. Seguridad a nivel físico:** asegurarnos de que no entren en las salas y roben las máquinas o los discos; ambiente refrigerado, cerradura de seguridad, vigilancia; contraseñas de BIOS y de consola...
- 2. Seguridad a nivel de red:** cortafuegos; subred privada, ACL.
- 3. Seguridad a nivel de administrador:** administradores por tipo de servicio.
- 4. Cuentas de servicios (o aplicaciones):** accesos controlados desde Internet (usuario “apache” o “www” + cuentas de aplicaciones).

# Políticas de seguridad

Toda organización con un gran sistema web debe tener un **equipo de ingenieros con dedicación exclusiva** a desarrollar, investigar, responder y arreglar temas de **seguridad** del sistema a todos los niveles.

**Ética profesional.** Si soy un extrabajador de una empresa, conozco la política de seguridad y podría “fácilmente” tener acceso no permitido...

# Índice



1. Introducción
2. Defensa en profundidad
3. Políticas de seguridad
- 4. Asegurar un servidor**
5. Cortafuegos
6. Evitar ataques
7. Prácticas de seguridad recomendadas
8. Conclusiones

# Asegurar un servidor

Proceso en el que eliminamos

- características no necesarias,
- servicios,
- configuraciones e
- información de seguridad del servidor,

de forma que sólo se dejen las aplicaciones, servicios y puertos realmente necesarios.

# Asegurar un servidor

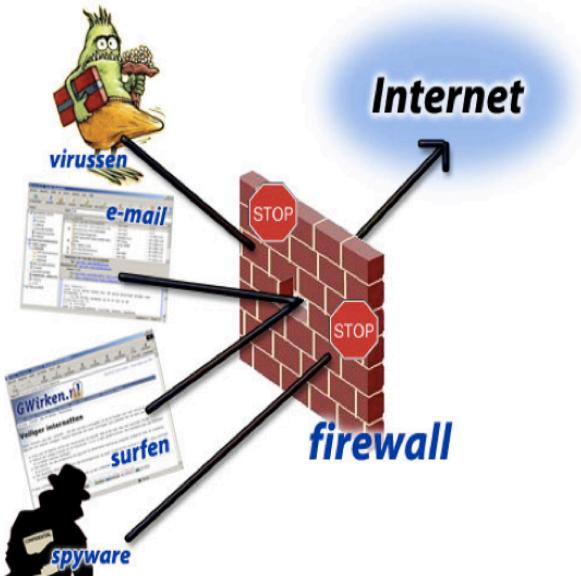
## Dos fases:

(1) una vez que el servidor ha sido montado hacer **cambios de configuración** (instalación limpia).

- Eliminar cuentas y grupos de usuarios no necesarios
- Renombrar las cuentas de administrador e invitado
- Eliminar servicios no necesarios
- Poner filtros TCP/IP
- Equipo de seguridad al día

(2) mantenimiento continuo para proteger de los ataques que van surgiendo.

# Índice



1. Introducción
2. Defensa en profundidad
3. Políticas de seguridad
4. Asegurar un servidor
5. Cortafuegos
6. Evitar ataques
7. Prácticas de seguridad recomendadas
8. Conclusiones

# Cortafuegos

Un cortafuegos protege el sistema de accesos indebidos.

En un sistema sin cortafuegos, otros elementos del sistema quedarán expuestos a diferentes riesgos.

Es el guardián de la puerta al sistema, permitiendo el tráfico autorizado y denegando el resto.



# Cortafuegos

El cortafuegos más efectivo:



# Cortafuegos

Colocados entre subredes para realizar diferentes tareas de manejo de paquetes.

## Tareas que realizan:

- **Bloquear y filtrar paquetes** de red inspeccionando las direcciones y puertos de cada paquete enviado entre las subredes que separa y controla.  
Por defecto, un cortafuegos debería prohibir el tráfico, y en el proceso de configuración se establecerán reglas para permitir cierto tipo de tráfico.

# Cortafuegos

Tareas que realizan:

- **Controlar protocolos de aplicación**, como HTTP, FTP, ssh o telnet. Esto se consigue configurando reglas relativas a ciertos puertos.
- **Control del tráfico de red a nivel de protocolo de red** (TCP o UDP). Así, si las reglas permiten la comunicación entre dos servidores, el tráfico (paquetes) fluirán entre ambos mientras la conexión permanezca abierta.

# Cortafuegos

Tareas que realizan:

- **Ocultar la verdadera dirección del servidor**, actuando como un proxy. De esta forma traduce la información de dirección de los mensajes entrantes y salientes reenviándolos a su destino.
- **Proteger los servidores y aplicaciones de ataques y uso indebido** controlando el flujo de información. Sin el cortafuegos, todos los servidores de la red serían accesibles para cualquier usuario

# Cortafuegos

La implementación y configuración del cortafuegos es **compleja**, pero aporta beneficios:

- Evita el consumo excesivo de recursos, reduciendo el tráfico global que un servidor recibirá.
- Oculta los servidores finales a otras redes.
- Protege los servidores de múltiples ataques.
- Oculta información de los servidores a otras redes (evitamos escaneo de puertos).
- Avisa de posibles ataques.

# Cortafuegos

Construir el conjunto de reglas de la siguiente forma:

- Crear grupos de reglas para conjuntos de servidores que deben responder a diferente tipo de tráfico.
- **Por defecto**, establecer reglas para **denegar el tráfico** que no esté permitido explícitamente.
- **Permitir el tráfico en el sentido necesario** (un servidor web no necesita navegar por Internet).

# Cortafuegos

## Recomendaciones:

1. Configurar el cortafuegos completamente independiente del resto de recursos.
2. La máquina cortafuegos no debe ejecutar otro software salvo el del cortafuegos.
3. Eliminar cualquier servicio accesorio en el cortafuegos.

# Cortafuegos

## Recomendaciones:

4. Blindar el cortafuegos para que no acepte conexiones directas a él (se comporte como un paso más en el camino y el atacante no se dé cuenta de que está ahí).
5. No registrar la IP del cortafuegos en ningún servicio de DNS, ya que su IP no es necesaria para que los clientes accedan a la granja web.
6. No permitir acceso desde Internet para administrar el cortafuegos, ya que un hacker podría conseguir acceso al mismo → VPN

# Cortafuegos

## Configurar el cortafuegos en Linux con iptables

<http://www.cyberciti.biz/tips/linux-iptables-examples.html>

<http://bit.ly/17Vqwi3>

[http://www.linuxtotal.com.mx/?cont=info\\_seyre\\_002](http://www.linuxtotal.com.mx/?cont=info_seyre_002)

<https://openwebinars.net/como-configurar-en-linux-firewall-basico-con-iptables/>

<http://es.tldp.org/Manuales-LuCAS/doc-iptables-firewall/doc-iptables-firewall-html/>

## Configurar el cortafuegos en Linux con ufw

<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-16-04>

<https://ubuntuforums.org/showthread.php?t=1876124>

## Usar nmap

<http://bencane.com/2013/02/25/10-nmap-commands-every-sysadmin-should-know/>

<https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/>

# Configurar el cortafuegos con iptables

## (proteger un servidor web):

```
iptables -F
```

```
iptables -X
```

```
iptables -Z
```

```
iptables -t nat -F
```

Eliminar todas las  
reglas (configuración  
 limpia)

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT DROP
```

```
iptables -P FORWARD DROP
```

Política por defecto:  
denegar todo el  
tráfico

```
iptables -A INPUT -i lo -j ACCEPT
```

```
iptables -A OUTPUT -o lo -j ACCEPT
```

Permitir cualquier  
acceso desde localhost  
(interface lo)

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

```
iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT
```

Abrir los puertos  
HTTP (80) de  
servidor web

# Configurar el cortafuegos con iptables

## (reiniciar configuración):

```
iptables -F
```

```
iptables -X
```

```
iptables -Z
```

```
iptables -t nat -F
```

Eliminar todas las  
reglas (configuración  
 limpia)

```
iptables -P INPUT ACCEPT
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -P FORWARD ACCEPT
```

Permitir cualquier  
acceso (todo el tráfico  
está permitido)

```
iptables -L -n -v
```

Examinar las reglas  
que hay establecidas

# Cortafuegos: iptables

## Configurar el cortafuegos con iptables (ejemplos):

Examinar las reglas configuradas en este momento:

```
iptables -L -n -v
```

Guardar/restaurar las reglas configuradas en este momento:

```
iptables-save > ~/reglas.iptables  
iptables-restore < ~/reglas.iptables
```

Evitar el acceso a www.facebook.com:

```
iptables -A OUTPUT -p tcp -d 69.171.224.0/19 -j DROP
```

También se puede usar el nombre de dominio:

```
iptables -A OUTPUT -p tcp -d www.facebook.com -j DROP  
iptables -A OUTPUT -p tcp -d facebook.com -j DROP
```

# Cortafuegos: ufw

## Configurar el cortafuegos con ufw:

Para hacer una configuración lo más segura posible, dejando acceso a SSH, HTTP, HTTPS:

```
ufw default deny incoming  
ufw default allow outgoing  
ufw allow ssh  
ufw enable  
ufw allow http  
ufw allow https  
ufw status verbose
```

Para comprobar la configuración de la red, podemos usar netstat, lsof, nc, nmap

```
netstat -natopu  
lsof -i -P -n  
nc -vn -w 1 105.21.19.6 22  
nmap 105.21.19.6  
nmap -O 105.21.19.6  
nmap -sL 105.21.19.0/24
```

# Cortafuegos

## Configurar el cortafuegos con iptables (ejemplos):

### Comprobación del funcionamiento del cortafuegos

Con la siguiente orden, comprobaremos qué puertos hay abiertos y cuáles cerrados:

```
netstat -tulpn
```

Para asegurarnos del estado del puerto 80 (abierto/cerrado), ejecutar:

```
netstat -tulpn | grep :80
```

Para ver las conexiones abiertas en el puerto 80 ejecutar:

```
netstat -an | grep :80 | sort
```

```
netstat | grep http | wc -l
```

# Cortafuegos

## Configurar el cortafuegos con iptables (ejemplos):

### Comprobación del funcionamiento del cortafuegos

```
[jmsoto@m3-jmsoto:~$ sudo netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 0.0.0.0:80                0.0.0.0:*              LISTEN    820/nginx: master p
tcp      0      0 127.0.0.53:53              0.0.0.0:*              LISTEN    656/systemd-resolve
tcp      0      0 0.0.0.0:22                0.0.0.0:*              LISTEN    816/sshd
tcp      0      0 0.0.0.0:443               0.0.0.0:*              LISTEN    820/nginx: master p
tcp6     0      0 :::22                   ::::*                  LISTEN    816/sshd
udp      0      0 127.0.0.53:53              0.0.0.0:*              LISTEN    656/systemd-resolve
udp      0      0 192.168.56.103:68            0.0.0.0:*              LISTEN    654/systemd-network
udp      0      0 10.0.2.15:68                0.0.0.0:*              LISTEN    654/systemd-network
[jmsoto@m3-jmsoto:~$ 
[jmsoto@m3-jmsoto:~$ 
[jmsoto@m3-jmsoto:~$ sudo netstat -tulpn | grep :80
tcp      0      0 0.0.0.0:80                0.0.0.0:*              LISTEN    820/nginx: master p
```

# Cortafuegos

## Configurar el cortafuegos con iptables (ejemplos):

### Comprobación del funcionamiento del cortafuegos

```
[jmsoto@m1-jmsoto:~$ sudo netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp    0      0 0.0.0.0:111                0.0.0.0:*
                                         LISTEN     464/rpcbind
tcp    0      0 127.0.0.53:53              0.0.0.0:*
                                         LISTEN     647/systemd-resolve
tcp    0      0 0.0.0.0:22                0.0.0.0:*
                                         LISTEN     804/sshd
tcp6   0      0 :::111                  ::::*
                                         LISTEN     464/rpcbind
tcp6   0      0 :::80                   ::::*
                                         LISTEN     845/apache2
tcp6   0      0 :::22                   ::::*
                                         LISTEN     804/sshd
tcp6   0      0 :::443                  ::::*
                                         LISTEN     845/apache2
tcp6   0      0 :::3306                 ::::*
                                         LISTEN     844/mysqld
udp    0      0 127.0.0.53:53              0.0.0.0:*
                                         LISTEN     647/systemd-resolve
udp    0      0 192.168.56.101:68            0.0.0.0:*
                                         LISTEN     646/systemd-network
udp    0      0 10.0.2.15:68               0.0.0.0:*
                                         LISTEN     646/systemd-network
udp    0      0 0.0.0.0:111                0.0.0.0:*
                                         LISTEN     464/rpcbind
udp    0      0 0.0.0.0:640                0.0.0.0:*
                                         LISTEN     464/rpcbind
udp6   0      0 :::111                  ::::*
                                         LISTEN     464/rpcbind
udp6   0      0 ::::640                 ::::*
```

# Índice



1. Introducción
2. Defensa en profundidad
3. Políticas de seguridad
4. Asegurar un servidor
5. Cortafuegos
6. Evitar ataques
7. Prácticas de seguridad recomendadas
8. Conclusiones

# Evitar otros tipos de ataques

El balanceador de carga puede evitar cierto tipo de ataques:

- denegación de servicio
- TCP SYN
- *ping of death*
- *Teardrop*
- *Smurf*
- *IP spoofing*
- Phishing

Para saber más sobre el funcionamiento de estos ataques:

[https://es.wikipedia.org/wiki/Ataque\\_de\\_denegaci%C3%B3n\\_de\\_servicio](https://es.wikipedia.org/wiki/Ataque_de_denegaci%C3%B3n_de_servicio)  
[http://es.ciberseguridad.wikia.com/wiki/Ataques\\_TCP/IP](http://es.ciberseguridad.wikia.com/wiki/Ataques_TCP/IP)

# Tipos de ataques

## Denegación de servicio:

- denegación de servicio por saturación
- denegación de servicio por explotación de vulnerabilidades

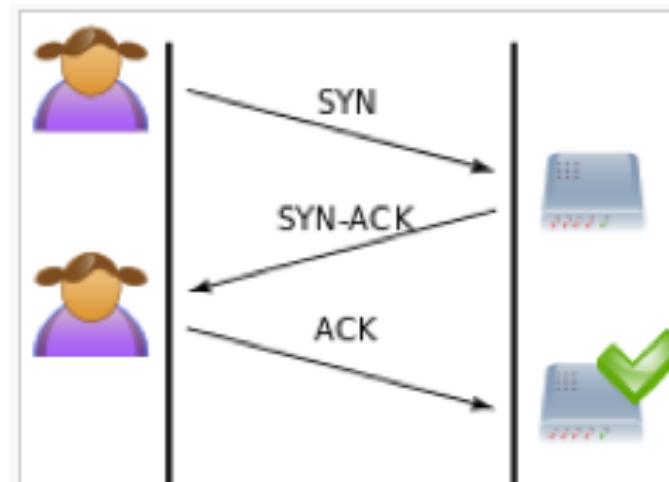
Los ataques por denegación de servicio envían paquetes IP o datos de tamaños o formatos raros que saturan los equipos de destino o los vuelven inestables.

DDoS, Distributed Denial of Service: sistema distribuido de denegación de servicio (participan varios equipos en la denegación de servicio).

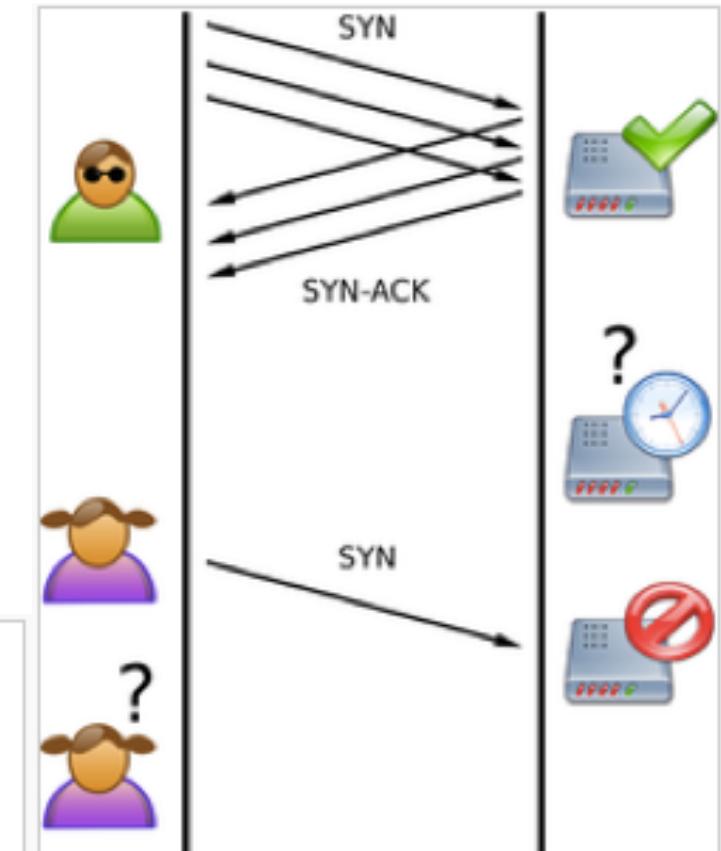
# Tipos de ataques

## TCP SYN o SYN flood (denegación de servicio):

Saturar el tráfico de la red aprovechando el mecanismo de negociación de tres vías del protocolo TCP.



A normal connection between a user (Alice) and a server. The three-way handshake is correctly performed.



SYN Flood. The attacker (Mallory) sends several packets but does not send the "ACK" back to the server. The connections are hence half-opened and consuming server resources. Alice, a legitimate user, tries to connect but the server refuses to open a connection resulting in a denial of service.

# Tipos de ataques

## TCP SYN o SYN flood:

aprovecha el mecanismo de negociación de tres vías del protocolo TCP:

Se envía una gran cantidad de solicitudes SYN a través de un ordenador con una dirección IP inexistente o no válida, de forma que el equipo atacado no puede recibir un paquete ACK. Así, quedarán las conexiones abiertas en cola en la estructura de memoria esperando la recepción de un paquete ACK.

# Tipos de ataques

## Ping de la muerte (ping of death):

El principio de este ataque consiste simplemente en crear un datagrama IP cuyo tamaño total supere el máximo autorizado (65.536 bytes). Cuando un paquete con estas características se envía a un sistema que contiene una pila vulnerable de protocolos TCP/IP, éste produce la caída del sistema.

Los sistemas actuales ya no son vulnerables a este ataque.

# Tipos de ataques

## Ataque por fragmentación (teardrop):

Se aprovecha del protocolo para fragmentar paquetes grandes en varios paquetes IP más pequeños. Cada uno de ellos tiene un número de secuencia y un número de identificación común para ensamblarlos.

El ataque se basa en introducir información falsa en los paquetes fragmentados para que queden fragmentos vacíos o superpuestos que pueden desestabilizar el sistema.

Los sistemas actuales ya no son vulnerables a esto.

# Tipos de ataques

## Ataque pitufo (Smurf):

Se basa en el uso de servidores de difusión (capacidad de duplicar un mensaje y enviarlo a todos los equipos de una misma red).

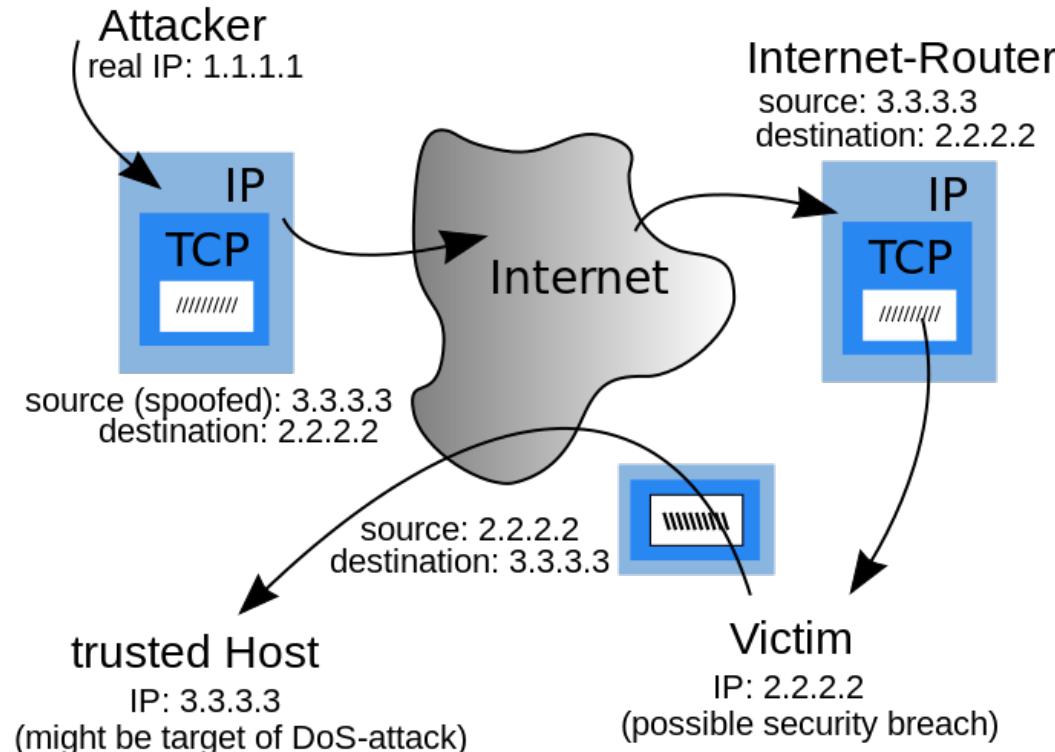
El equipo atacante envía una solicitud de ping a varios servidores de difusión falsificando las direcciones IP de origen y proporciona la dirección IP de un equipo de destino (atacado). El servidor transmite la solicitud a toda la red. Todos los equipos de la red envían una respuesta al servidor de difusión, que redirecciona las respuestas al equipo de destino.

# Tipos de ataques

## IP spoofing:

Consiste en crear paquetes con la IP de origen falsa.

Así se consigue que las respuestas que genere el equipo de destino vayan a otro equipo (el objetivo del ataque).



# Tipos de ataques

## Suplantación de identidad (Phishing):

Es una técnica de "ingeniería social", lo que significa que no aprovecha una vulnerabilidad en los ordenadores sino un "fallo humano" al engañar a los usuarios de Internet con un correo electrónico que aparentemente proviene de una empresa fiable, comúnmente de una página Web bancaria o corporativa.

# ¿Cómo de difícil es hacer un ataque?

¿Alguien ha pensado en hacer un ataque?

Código C:

<http://www.binarytides.com/syn-flood-dos-attack/>

```
118 //Uncomment the loop if you want to flood :)
119 //while (1)
120 //{
121     //Send the packet
122     if (sendto (s,      /* our socket */
123                 datagram, /* the buffer containing
```

Código Perl:

<http://www.binarytides.com/perl-syn-flood-program-raw-sockets-linux/>

La herramienta hping:

<http://www.binarytides.com/tcp-syn-flood-dos-attack-with-hping/>

```
sudo apt-get install hping3
```

```
sudo hping3 -i u1 -S -p 80 192.168.1.1
```

# Evitar otros tipos de ataques

El balanceador puede mantener **listas negras**.

Limitar o denegar completamente el acceso a listas de IP monitorizando el origen, destino o puerto del tráfico.

Se pueden incluir **rangos completos de IP**.

Se pueden evitar ataques de sitios concretos, actuando como sistema adicional de detección de intrusos.

# Evitar otros tipos de ataques

Posibilidad de **crear listas de control de acceso** (access control list, ACL) y realizar filtrado a partir de ellas.

Definir las aplicaciones (servicios o puertos) a los que puede acceder un grupo. El administrador de red puede permitir o denegar el acceso a ciertas funcionalidades (aplicaciones) a rangos de IP.

- El balanceador sólo **complementa/ayuda al cortafuegos**, ya que tiene capacidad limitada para bloquear o filtrar.

# Índice



1. Introducción
2. Defensa en profundidad
3. Políticas de seguridad
4. Asegurar un servidor
5. Cortafuegos
6. Evitar ataques
7. Prácticas de seguridad recomendadas
8. Conclusiones

# Prácticas de seguridad recomendadas

## Copias de seguridad:

 menéame ▾ login registrarse ☰

**183 meneos**  
**menéalo**  
**2568 clics**

Gitlab.com fundido por borrar un directorio incorrecto, los backups fallan [ENG]

por **ccguy** a **theregister.co.uk** 09:08  
publicado: 11:50



Gitlab.com, una empresa de gestión de código fuente, ha perdido datos de producción. Un administrador cansado ejecutó un rm -rf sobre un directorio incorrecto que contenía 300 GB. Cuando se dio cuenta sólo quedaban 4.5 GB de datos. Los sistemas de backups que tienen no estaban probados y no han servido para nada. Se están recuperando los datos de un snapshot con algunas horas de antigüedad que además no es completo. La pérdida afecta a la base de datos (incidencias y merge requests) pero no a los repositorios.

**etiquetas:** gitlab, rm, backup, directorio, datos

# Prácticas de seguridad recomendadas

## Copias de seguridad:

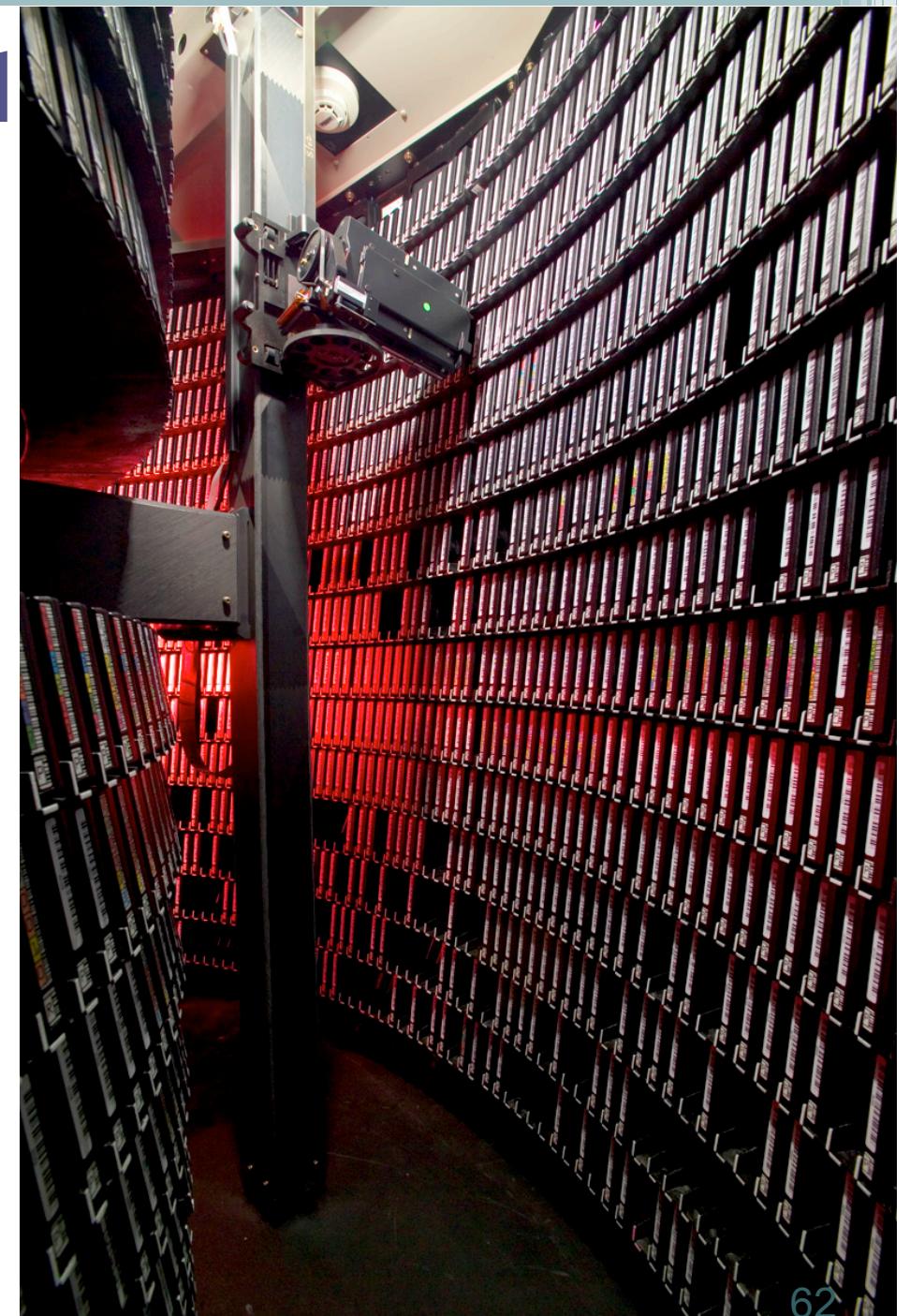
Tener un sistema de **copias de seguridad automatizado** es indispensable para asegurar la disponibilidad de los datos en nuestro sistema.

El software de copia de seguridad debe verificar los datos una vez grabados.

Las copias de seguridad deben guardarse en un lugar seguro, en un local diferente al que alberga los servidores.

# Prácticas de seguridad

## Copias de seguridad:



## Ejemplos de funcionamiento:

<https://www.youtube.com/watch?v=d-eWDuEo-3Q>

<https://www.youtube.com/watch?v=GwMn7YpF8r8>

# Prácticas de seguridad recomendadas

## Imágenes de los servidores:

También conviene disponer de **imágenes de instalación** de los propios sistemas.

Podremos restaurar una máquina rápida y fácilmente.

Opciones: desde usar el comando dd de Linux hasta usar software propietario como Intelligent Disaster Recovery (Veritas Backup-Exec) o Take Two (Adaptec).

# Prácticas de seguridad recomendadas

## Imágenes de los servidores. Intelligent Disaster Recovery

Backup Exec Intelligent Disaster Recovery  
for Windows (2000/XP/Server 2003/Vista/7/Server 2008/Server 2008 R2)  
Copyright (c) 2011 Symantec Corporation. All rights reserved.

You have successfully loaded a Backup Exec Disaster Recovery CD/Tape image.

If you are testing the bootable media, the computer successfully booted the image. Remove the boot media and press <Esc> to stop the recovery.  
**DO NOT PRESS <ENTER>.**

If you are performing a disaster recovery, press <Enter> to start the disaster recovery process, which will repartition and reformat the computer's hard disks and **DESTROY ALL EXISTING DATA**. The Windows setup program and the Backup Exec Disaster Recovery Wizard are then loaded.

# Índice



1. Introducción
2. Defensa en profundidad
3. Políticas de seguridad
4. Asegurar un servidor
5. Cortafuegos
6. Evitar ataques
7. Prácticas de seguridad recomendadas
- [8. Conclusiones]**

# Conclusiones

**El éxito de un sitio web depende de la seguridad.**

La seguridad no se puede pasar por alto !

**Aspecto crítico** en un sistema web para mantener a salvo de ataques los recursos de la empresa.

Hay que establecer unas **políticas de seguridad**, y **mantenerse al día** de vulnerabilidades del software, de posibles ataques, de actualizaciones de software, etc.

# Conclusiones

**La defensa en profundidad** implica mantener diferentes capas de seguridad, independientes entre ellas, de forma que si un atacante consigue pasar una, tendrá otra que superar.

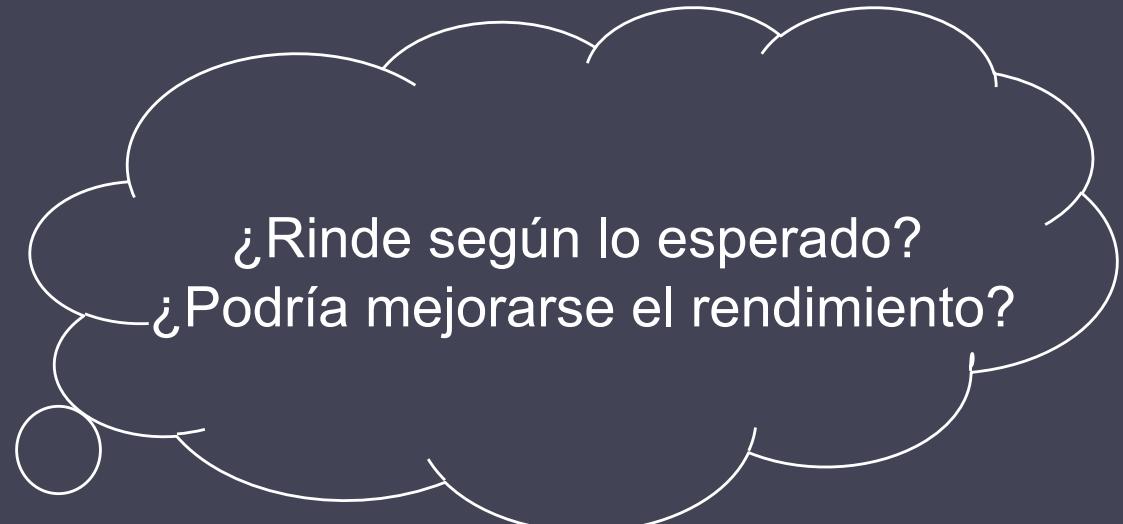
Así se dificulta en gran medida la consecución final de un ataque.

Se diseñarán **diferentes tipos de acceso** y se configurará el sistema para facilitar esos accesos exclusivamente, denegando cualquier otro.

# TEMA 5

## Medir prestaciones de la granja web

SWAP



José Manuel Soto Hidalgo  
Dpto. Arquitectura y Tecnología de Computadores  
Universidad de Granada

jmsoto@ugr.es

# Índice



- [ [1. Introducción](#) ]
- [2. Conexiones por segundo](#)
- [3. Número de conexiones concurrentes](#)
- [4. Rendimiento, en bits por segundo](#)
- [5. Tipos de tráfico](#)
- [6. Límite en las prestaciones](#)
- [7. Software](#)
- [8. Apache benchmark](#)
- [9. httpperf](#)
- [10. OpenWebLoad](#)
- [11. Siege](#)

# Introducción

**Medir las prestaciones** de nuestro sistema web:

- los servidores finales
- los dispositivos de balanceo
- elementos de red

**Objetivo:** Comprobar si cumplen unos mínimos requisitos de rendimiento.

Aplicar una metodología de test de prestaciones para detectar posibles problemas de rendimiento.

# Introducción

Principal necesidad de hacer los tests:

- No son exclusivamente las caídas o errores de programación factores que influyen en el rendimiento.
- Detectar posibles cuellos de botella e ineficiencias.
- Detectar límite del sistema

Limitaciones de los tests:

- Dificultad para hacer pruebas en un entorno de producción.
- Dificultad para simular el comportamiento de los usuarios.

# Introducción

Muy importante medir las prestaciones de los dispositivos de balanceo.

Según el sitio web, recomendaremos diferentes métricas:

- conexiones por segundo
- número total de conexiones concurrentes
- rendimiento (en bits por segundo)

# Índice



1. Introducción
2. Conexiones por segundo
3. Número de conexiones concurrentes
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software
8. Apache benchmark
9. httpperf
10. OpenWebLoad
11. Siege

# Conexiones por segundo

Es una de las métricas más importantes cuando hablamos del rendimiento de servidores web.

Hace referencia al número de **conexiones de entrada** que cierto dispositivo puede manejar por segundo.

También se llama *transacciones por segundo* o *sesiones por segundo*.



# Conexiones por segundo

Es un factor determinante, ya que **abrir y cerrar conexiones HTTP resulta muy costoso.**

En el nivel que estamos tratando, es la operación principal.

Para enviar datos hay que llevar a cabo una serie de pasos que pueden llegar a **sobrecargar el dispositivo de red.**

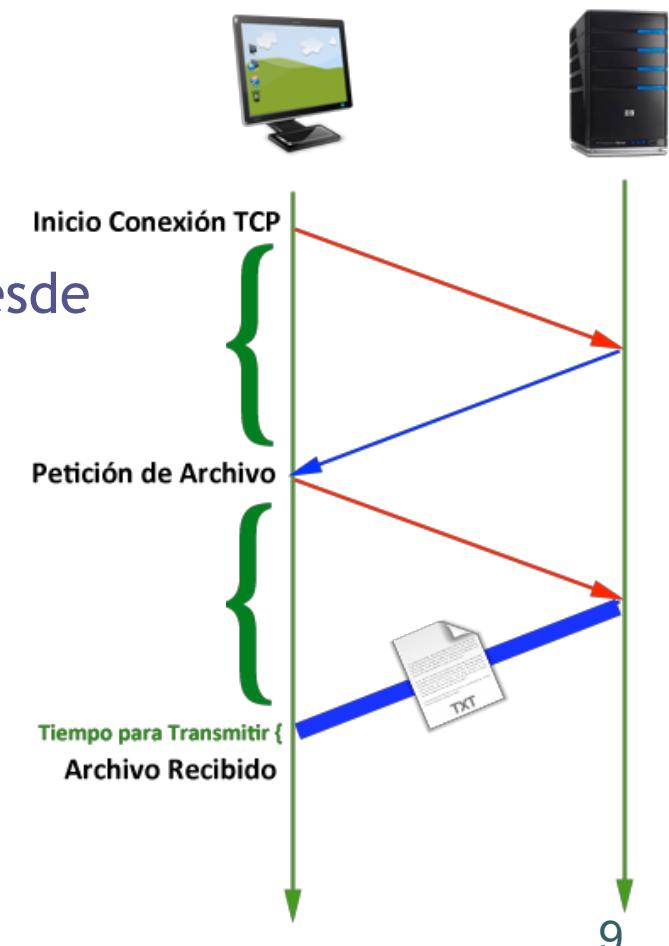
Las aperturas y cierres de **conexiones consumen muchos recursos.**

# Conexiones por segundo

**Pasos para establecer una conexión HTTP:**

- el cliente inicia la conexión HTTP enviando un paquete TCP SYN al puerto 80 del servidor web
- el servidor web envía un paquete ACK al cliente seguido de otro SYN
- el cliente envía un paquete ACK como respuesta

Ahora ya pueden comenzar a enviarse datos desde el servidor al cliente (normalmente será una página web).



# Conexiones por segundo

La **velocidad** a la que se gestionan las aperturas y cierres de conexiones es fundamental.

Si cierto servidor web tiene un tráfico HTTP alto (muchas peticiones), conexiones por segundo será la métrica más importante.



# Conexiones por segundo

Ver el número de conexiones por segundo.

```
netstat -an | grep :80 | sort  
netstat | grep http | wc -l  
netstat -n -p | grep SYN_REC | sort -u
```

Habilitar stats en Nginx y ver conexiones activas y conexiones por segundo:

```
location /  
{  
    proxy_pass http://balanceo_jmsoto;  
    proxy_set_header Host $host;  
    proxy_set_header X-Real-IP $remote_addr;  
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    proxy_http_version 1.1;  
    proxy_set_header Connection "";}  
  
location /nginx_status{  
    stub_status;  
    allow 192.168.56.1;  
    deny all;  
}
```



# Índice



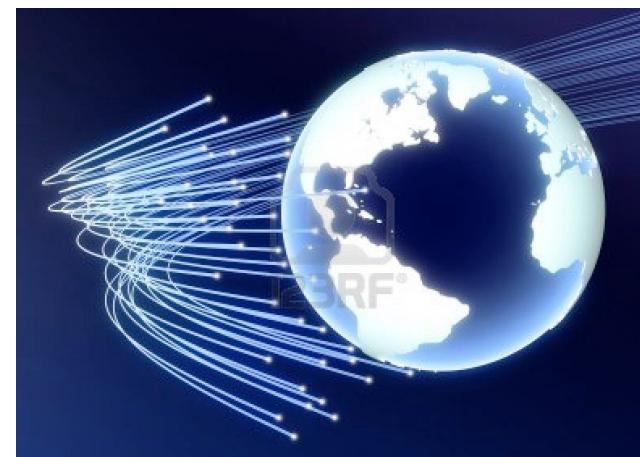
1. Introducción
2. Conexiones por segundo
- 3. Número de conexiones concurrentes**
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software
8. Apache benchmark
9. httpperf
10. OpenWebLoad
11. Siege

# Número de conexiones concurrentes

Métrica para determinar **cuántas sesiones de usuario TCP** puede manejar el balanceador al **mismo tiempo**.

Limitado por la memoria o el procesador del dispositivo.

Varía desde varios miles hasta millones.  
Límite teórico (realmente no es tan alta).



# Número de conexiones concurrentes

Esto en cuanto a las conexiones TCP...

Sin embargo, para el tráfico UDP (streaming o tráfico DNS) el número de conexiones concurrentes no es un factor que afecte, ya que se trata de un protocolo “sin conexión”:

- el receptor no reconoce haber recibido paquetes.

No hay una fase de establecimiento de la conexión.

No se mantiene información de estado.

TCP mantiene información sobre el estado de la conexión para garantizar un servicio fiable de transferencia de datos y control de congestión.

# Número de conexiones concurrentes

Hoy en día, las webs de vídeos como Youtube o Vimeo, utilizan TCP ya que algunas organizaciones bloquean el tráfico UDP por cuestiones de seguridad.

También se usa TCP para no colapsar el servidor ya que TCP provee control de congestión.

# Índice



1. Introducción
2. Conexiones por segundo
3. Número de conexiones concurrentes
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software
8. Apache benchmark
9. httpperf
10. OpenWebLoad
11. Siege

# Rendimiento, en bits por segundo

Hace referencia a la **velocidad** a la que el balanceador maneja y pasa el tráfico.

Todos los dispositivos tienen una serie de factores que acaban limitando las prestaciones, basados en la estructura interna (hardware y software).

Algunos desarrolladores de balanceadores de carga sólo soportan Fast Ethernet, limitándolos así a 100Mbps.

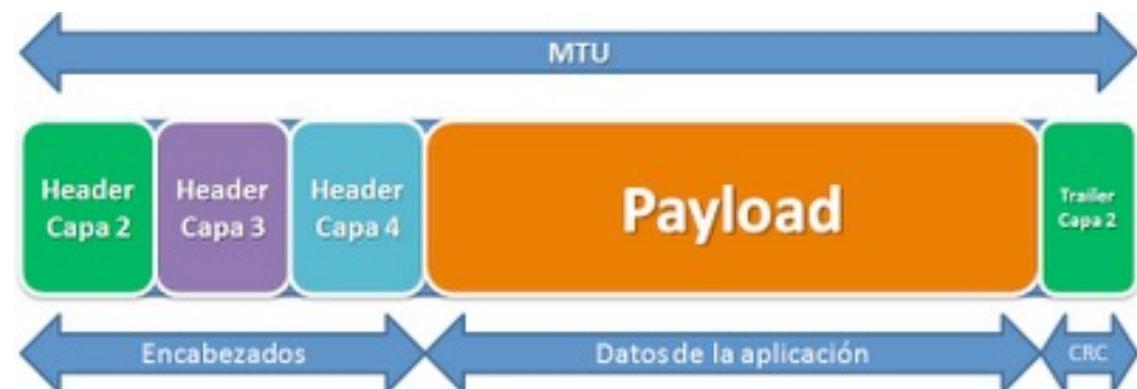
Algunas implementaciones no tienen el hardware o el software adecuado, con lo que quedan limitados a transferencias máximas de 80Mbps.

# Rendimiento, en bits por segundo

Se mide en bits por segundo. Es combinación de las variables *"tamaño del paquete"* y *"paquetes por segundo"*.

El paquete típico tiene un tamaño máximo (MTU, Maximum Transmittable Unit) de 1.5KB.

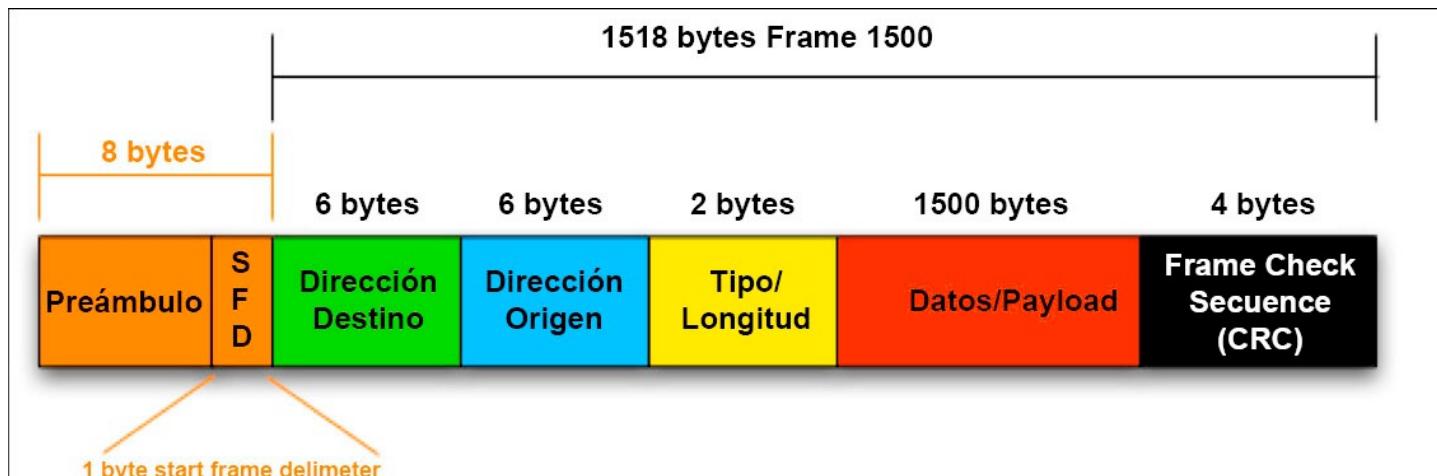
Si hay que enviar más datos, se trocean en paquetes de este tamaño máximo.



# Rendimiento, en bits por segundo

## Ejemplo:

- un acceso por HTTP usando el método GET a un recurso de 100 bytes podrá servirse en un solo paquete.
- un acceso por GET a un archivo de 32KB necesitará 21 paquetes, con 1.5KB de información útil (*payload*) en cada uno.

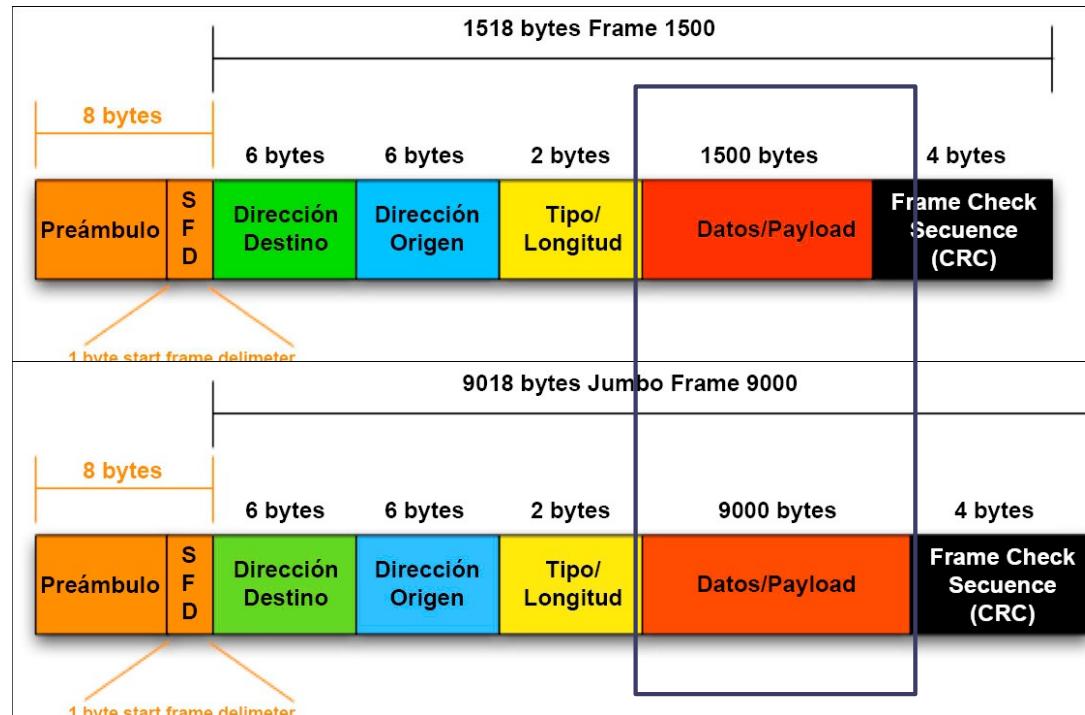


# ¿Y para grandes transferencias?

¡Jumbo Frame!

<http://www.mundonas.com/2013/05/jumbo-frames.html>

“con la salida de **las redes Gigabit** se implementó la posibilidad de modificar el tamaño de esos paquetes para optimizar tanto tráfico como transferencia de información”



# ¿Y para grandes transferencias?

## ¡Jumbo Frame! - Ventajas

- Al ser paquetes más grandes requieren de menor dedicación de CPU para su procesado en tarjetas de red, routers, etc, además de llegar antes
- Se minimiza el uso de bytes para la asignación de los datos que acompañan a la MTU, dedicando más bytes a datos enviados.
- Aliviamos la carga de protocolo y menor fragmentación de la información por la red.
- Se mejora la carga en los programas que gestionan el tráfico de red, como son los Firewalls (Cortafuegos).

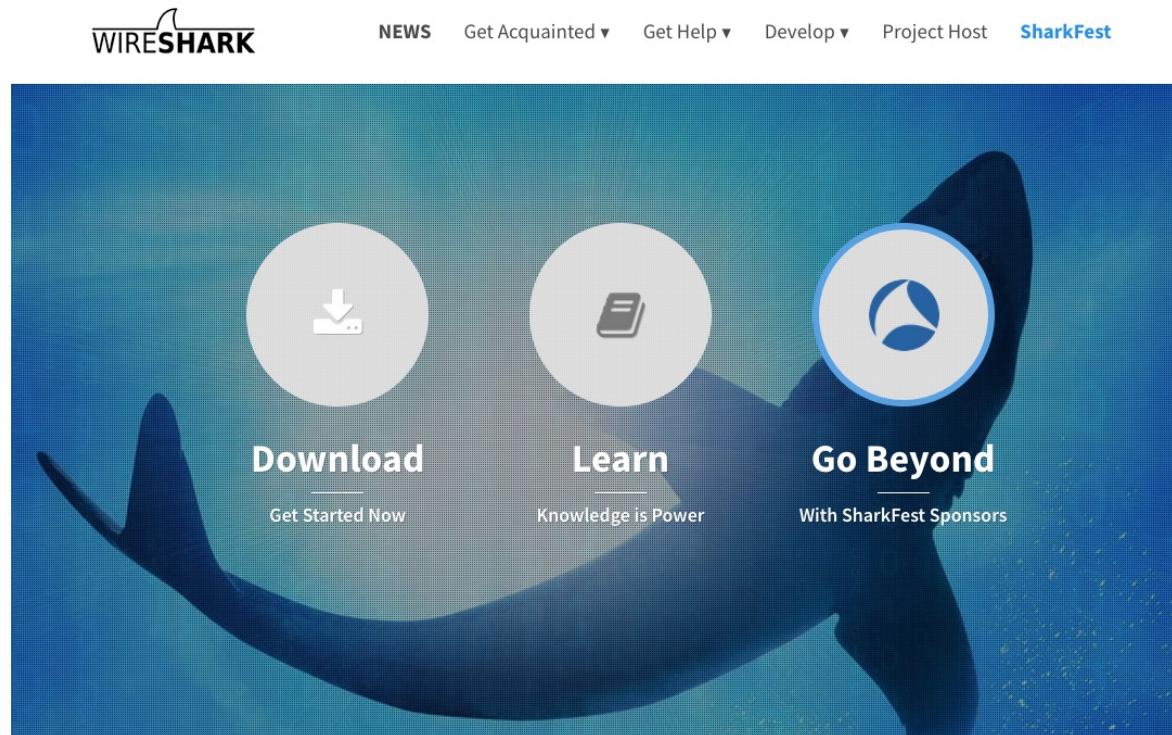
# ¿Y para grandes transferencias?

## ¡Jumbo Frame! - Desventajas

- Al aumentar el tamaño de los paquetes aumenta la latencia puesto que el tiempo de envío de los mismos es mayor, no siendo conveniente en redes con fuerte streaming.
- En redes de baja/media calidad provoca que si un paquete no pasa la comprobación de errores debe volverse a reenviar, siendo mayor la información que se reenvía y el tiempo que se pierde en la tarea.
- No es muy recomendable forzar el uso de Jumbo Frames (paquetes de información grandes) con redes que manejan paquetes pequeños (P2P) puesto que no se rinde al 100% de la configuración, es conveniente probar la administración por parte del router de esta mezcla.

# Análisis del tráfico con Wireshark

<https://www.wireshark.org/>



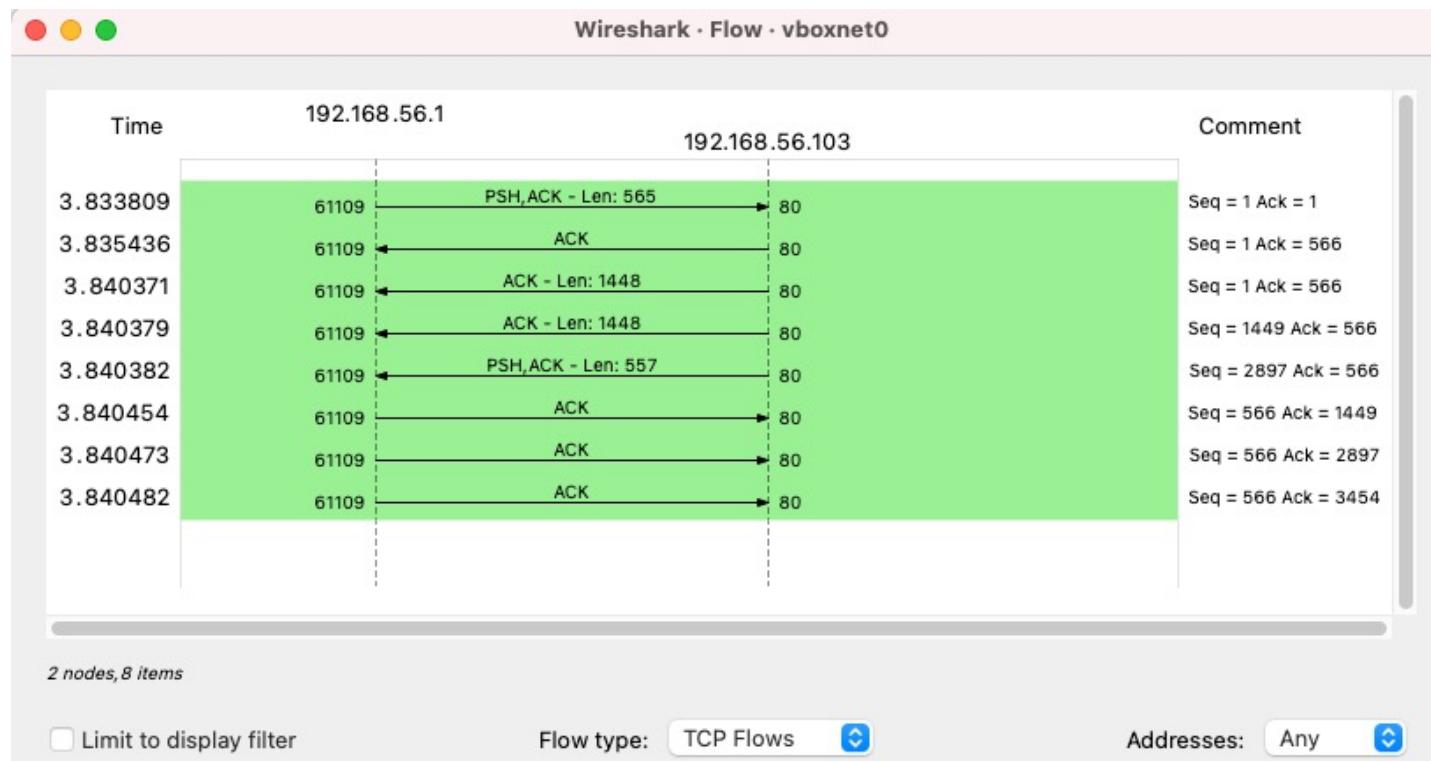
Es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para análisis de datos y protocolos

# Análisis del tráfico con Wireshark

<https://www.wireshark.org/>

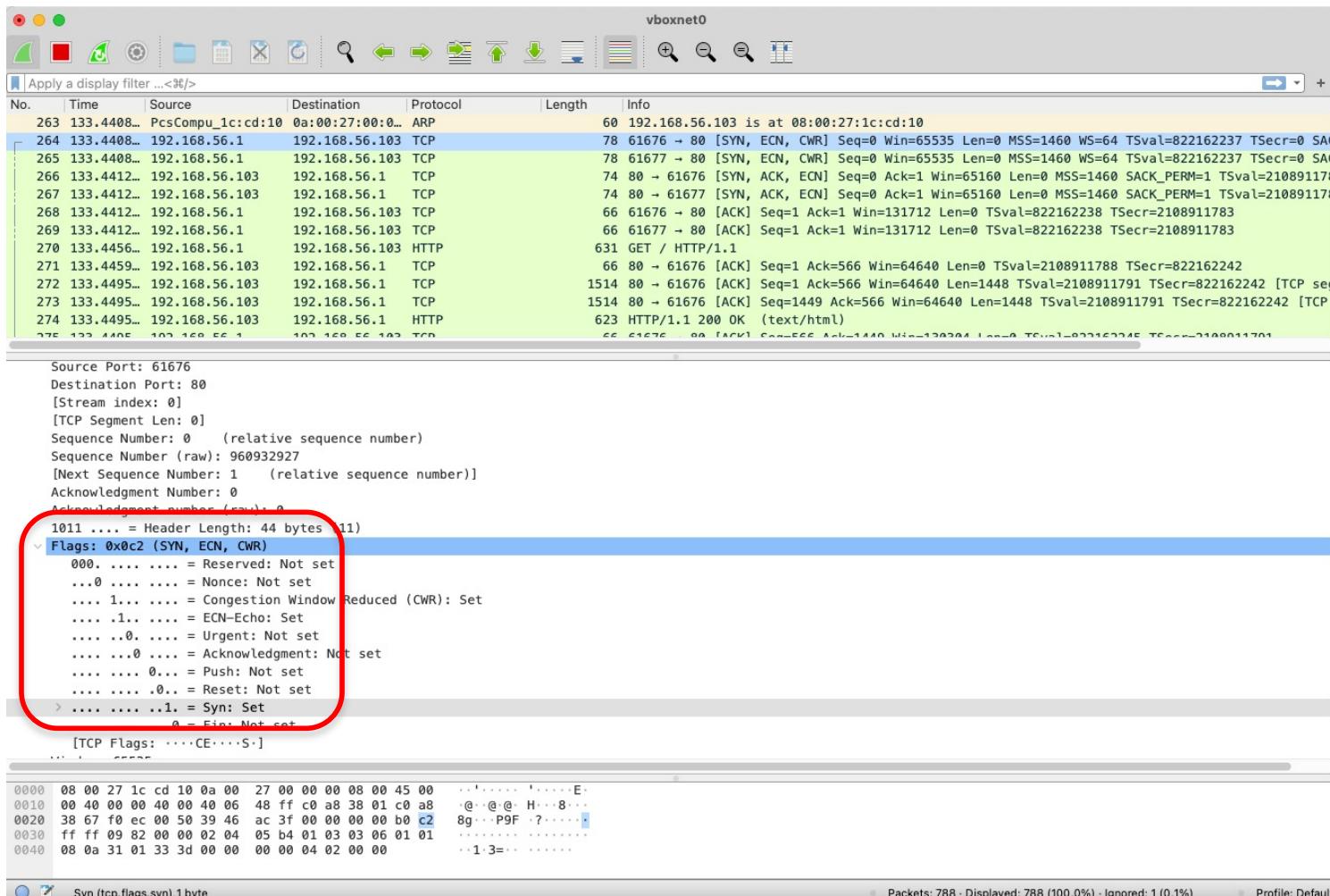


- *Establecimiento de conexión (handshake de tres vías)*



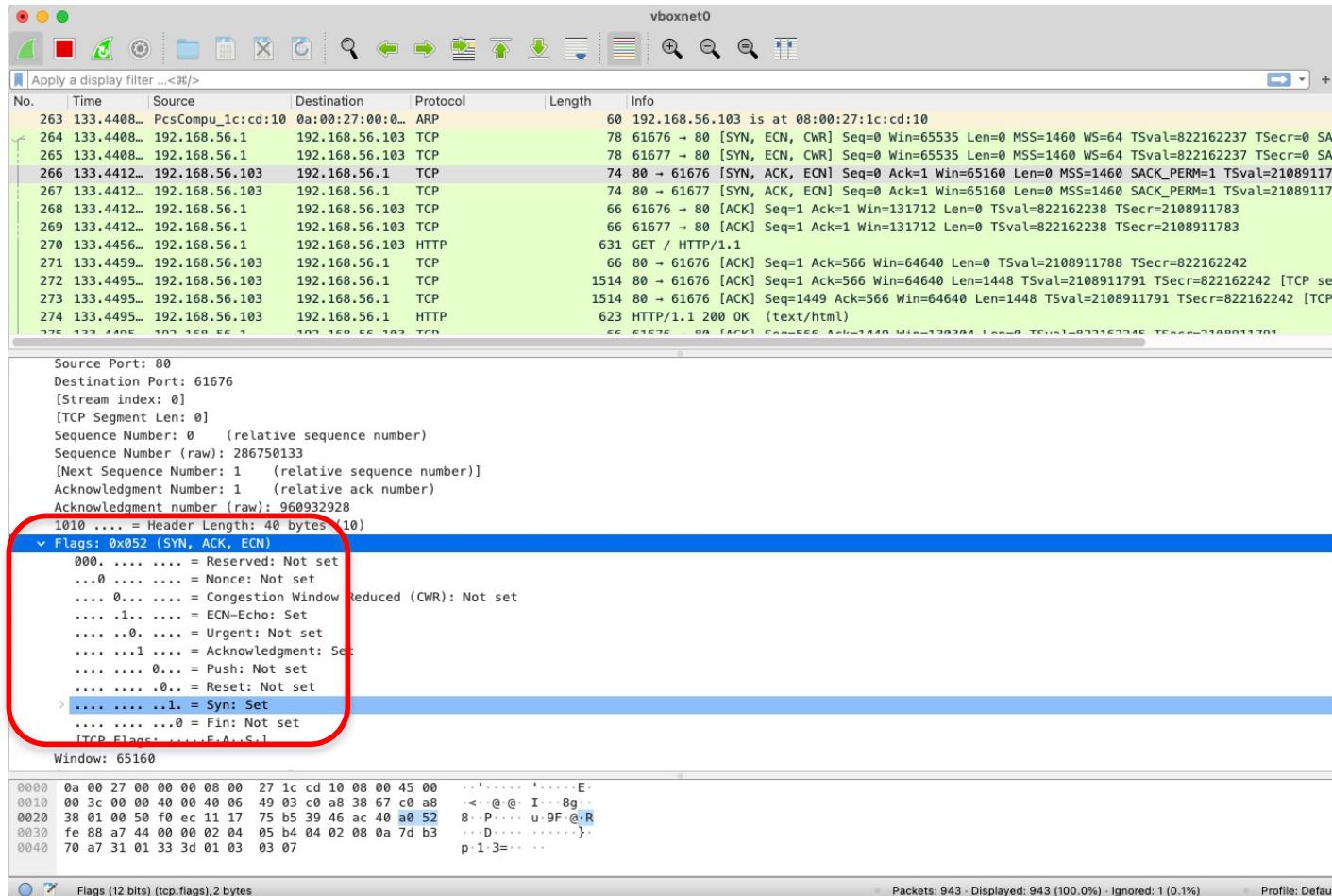
# Análisis del tráfico con Wireshark

Ej: Establecimiento de conexión (handshake de tres vías)



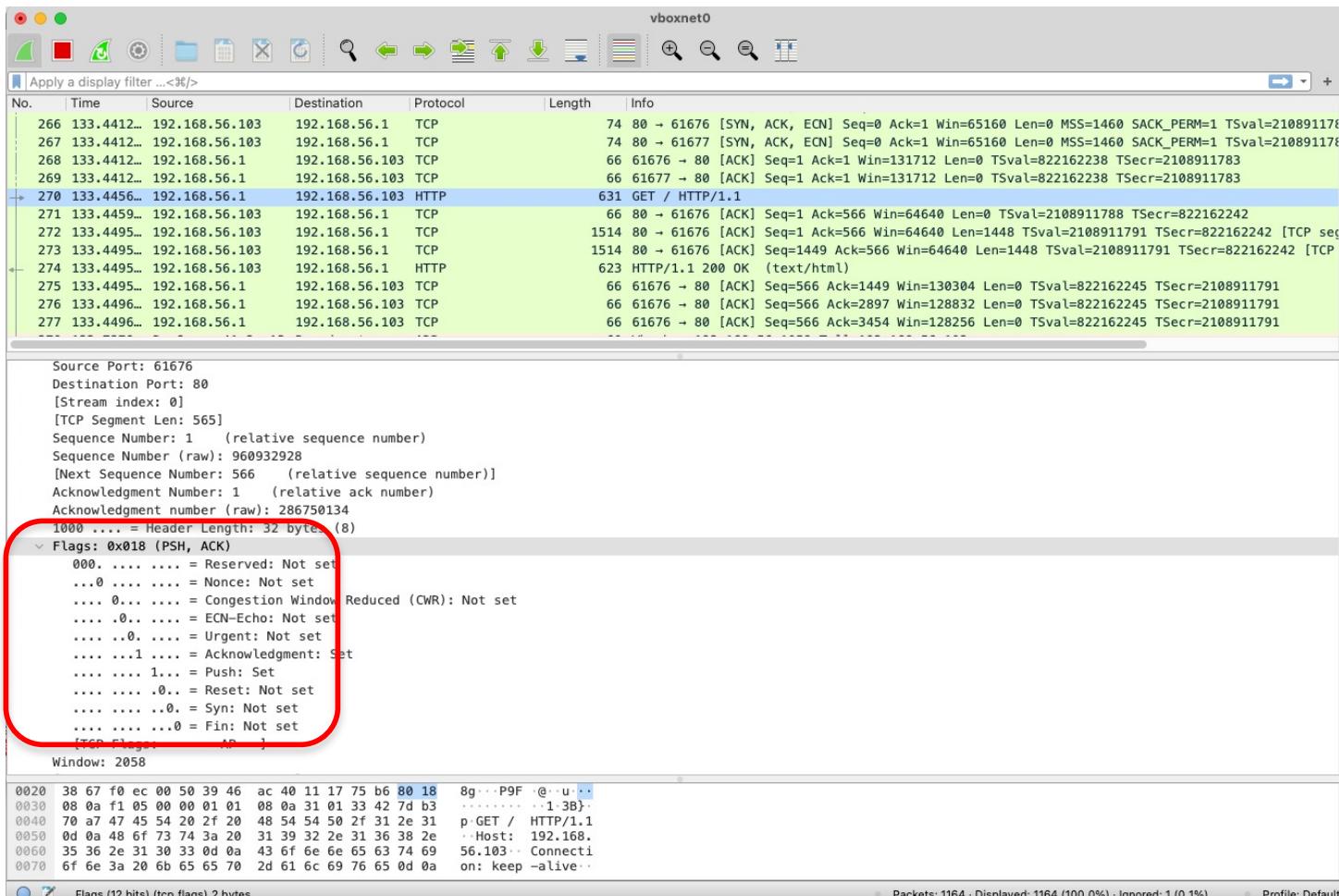
# Análisis del tráfico con Wireshark

Ej: Establecimiento de conexión (handshake de tres vías)



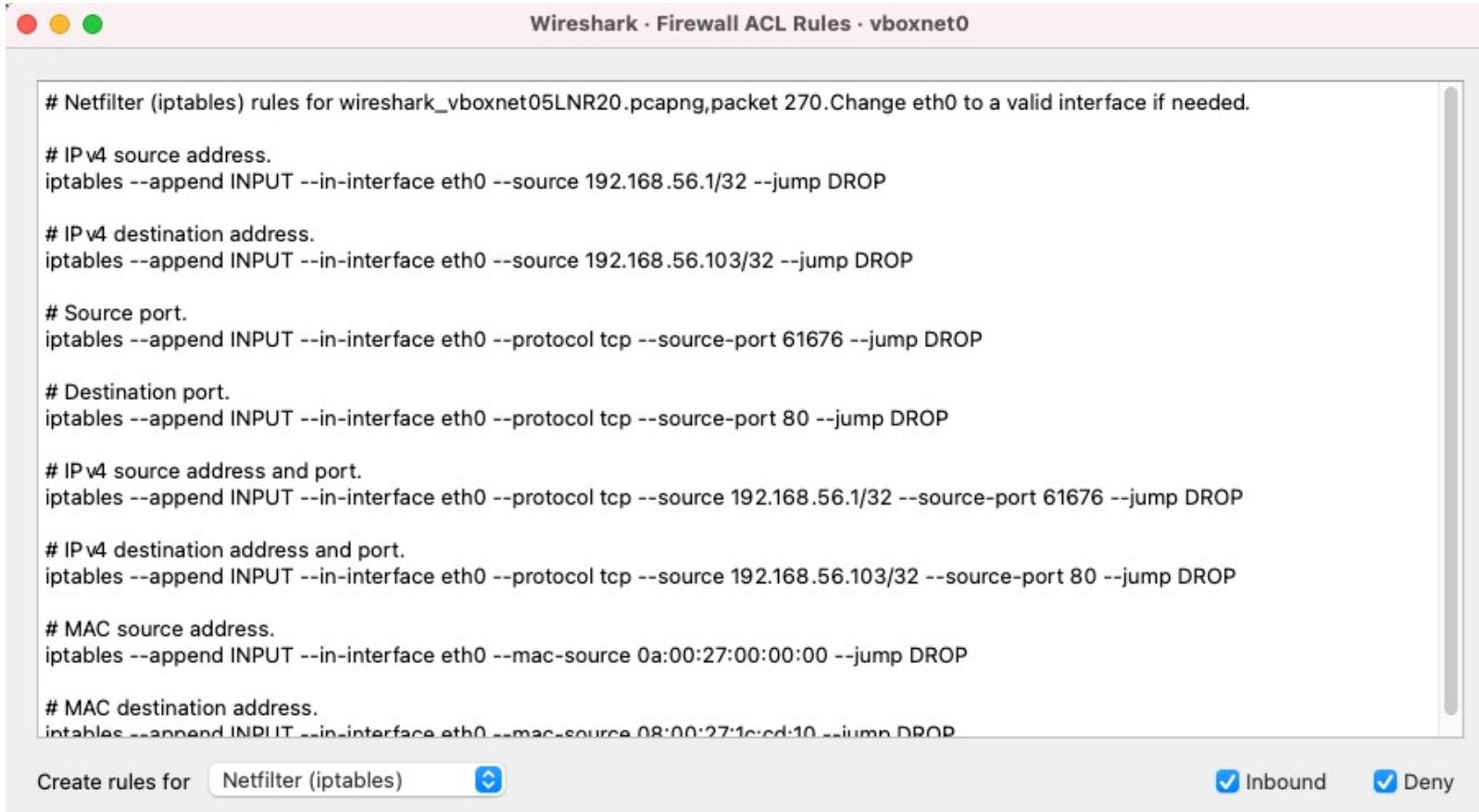
# Análisis del tráfico con Wireshark

Ej: Establecimiento de conexión (handshake de tres vías)



# Análisis del tráfico con Wireshark

Ej: Seguridad con Wireshark. IPTABLES



The screenshot shows the Wireshark Firewall ACL Rules interface for the interface 'vboxnet0'. The window title is 'Wireshark - Firewall ACL Rules - vboxnet0'. The content area displays the following iptables rules:

```

# Netfilter (iptables) rules for wireshark_vboxnet05LNR20.pcapng, packet 270. Change eth0 to a valid interface if needed.

# IP v4 source address.
iptables --append INPUT --in-interface eth0 --source 192.168.56.1/32 --jump DROP

# IP v4 destination address.
iptables --append INPUT --in-interface eth0 --source 192.168.56.103/32 --jump DROP

# Source port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source-port 61676 --jump DROP

# Destination port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source-port 80 --jump DROP

# IP v4 source address and port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source 192.168.56.1/32 --source-port 61676 --jump DROP

# IP v4 destination address and port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source 192.168.56.103/32 --source-port 80 --jump DROP

# MAC source address.
iptables --append INPUT --in-interface eth0 --mac-source 0a:00:27:00:00:00 --jump DROP

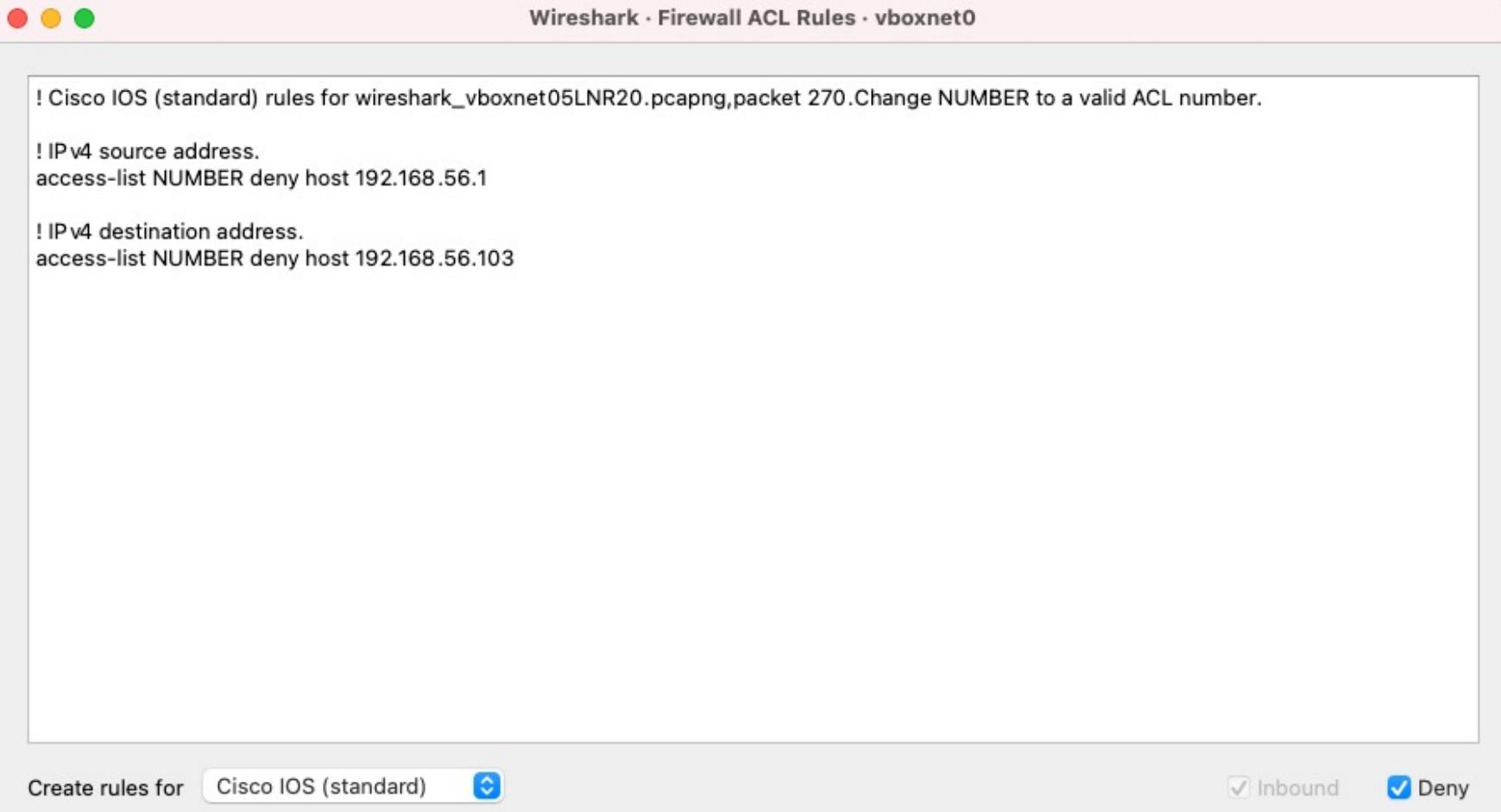
# MAC destination address.
iptables --append INPUT --in-interface eth0 --mac-source 08:00:27:1c:cd:10 --jump DROP

```

At the bottom, there are buttons for 'Create rules for' (set to 'Netfilter (iptables)'), 'Inbound' (checked), and 'Deny' (checked).

# Análisis del tráfico con Wireshark

Ej: Seguridad con Wireshark. ACL standard



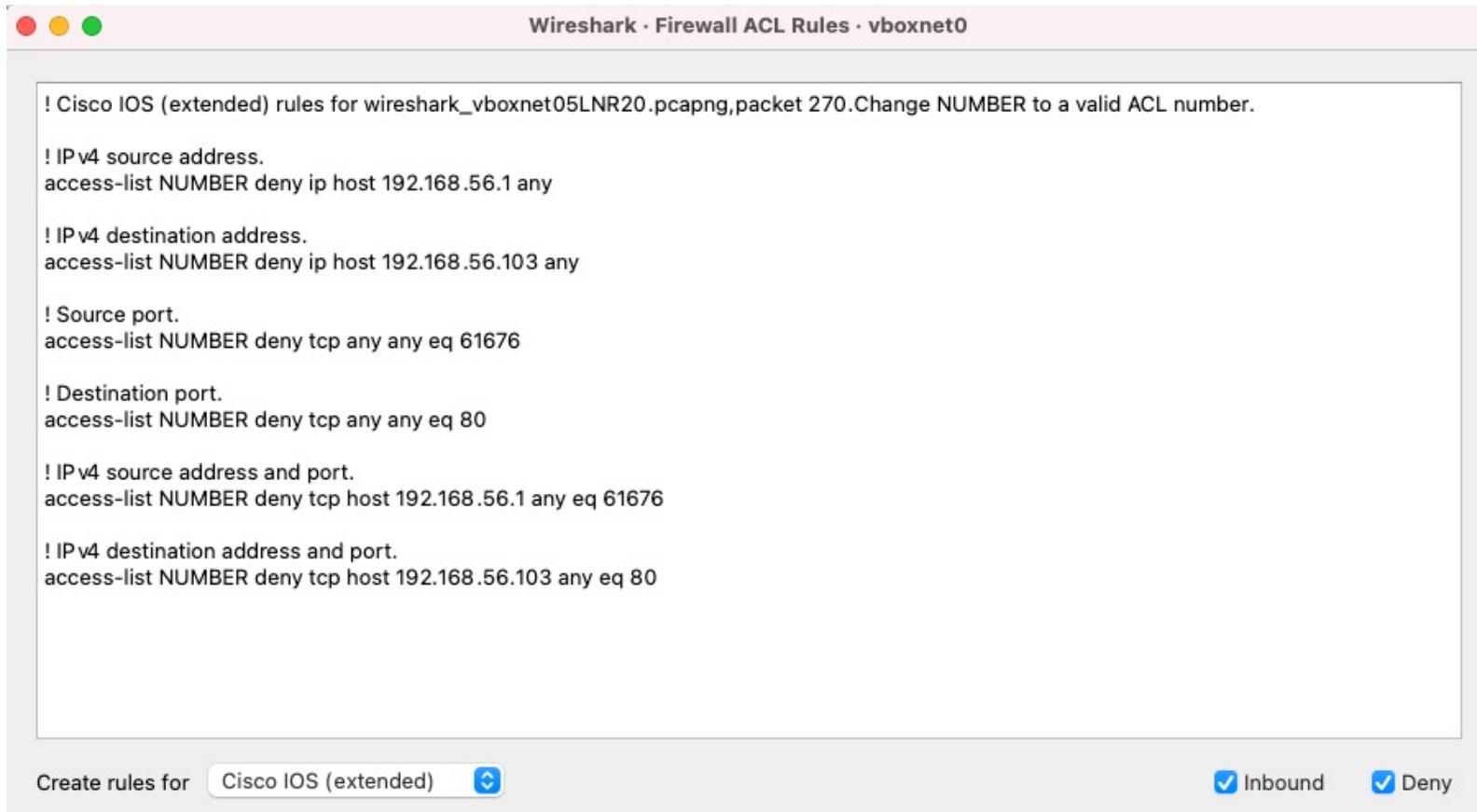
The screenshot shows the Wireshark Firewall ACL Rules interface for the interface `vboxnet0`. The title bar reads "Wireshark · Firewall ACL Rules · vboxnet0". The main pane displays the following Cisco IOS standard ACL rules:

```
! Cisco IOS (standard) rules for wireshark_vboxnet05LNR20.pcapng,packet 270.Change NUMBER to a valid ACL number.  
!  
! IP v4 source address.  
access-list NUMBER deny host 192.168.56.1  
  
!  
! IP v4 destination address.  
access-list NUMBER deny host 192.168.56.103
```

At the bottom of the window, there are buttons for "Create rules for" (set to "Cisco IOS (standard)"), "Inbound" (checked), and "Deny" (checked).

# Análisis del tráfico con Wireshark

Ej: Seguridad con Wireshark. ACL extendida



The screenshot shows the Wireshark Firewall ACL Rules interface for the 'vboxnet0' interface. The title bar reads 'Wireshark - Firewall ACL Rules - vboxnet0'. The main window displays the following Cisco IOS (extended) ACL rules:

```
! Cisco IOS (extended) rules for wireshark_vboxnet05LNR20.pcapng,packet 270.Change NUMBER to a valid ACL number.

! IP v4 source address.
access-list NUMBER deny ip host 192.168.56.1 any

! IP v4 destination address.
access-list NUMBER deny ip host 192.168.56.103 any

! Source port.
access-list NUMBER deny tcp any any eq 61676

! Destination port.
access-list NUMBER deny tcp any any eq 80

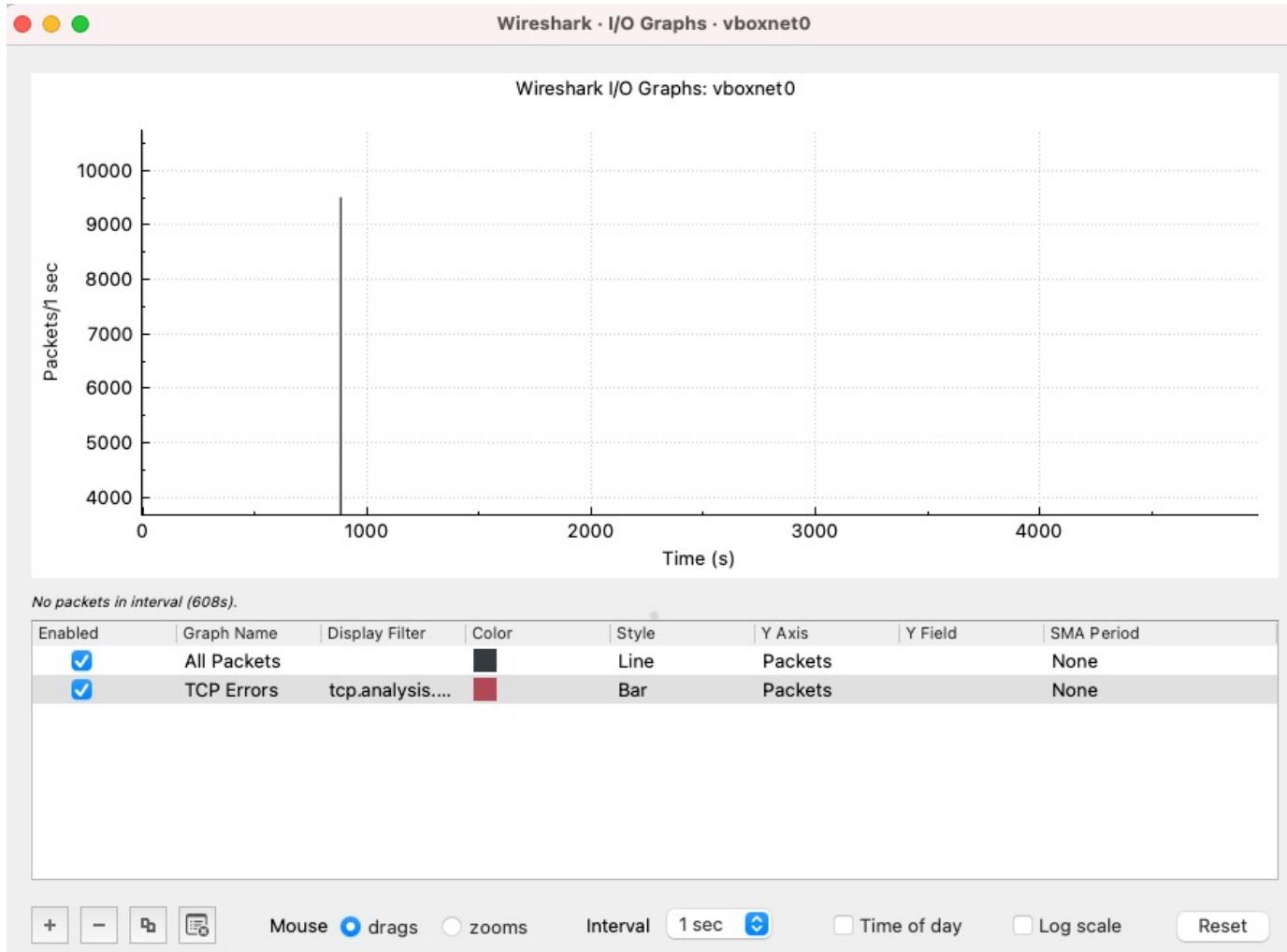
! IPv4 source address and port.
access-list NUMBER deny tcp host 192.168.56.1 any eq 61676

! IPv4 destination address and port.
access-list NUMBER deny tcp host 192.168.56.103 any eq 80
```

At the bottom of the interface, there are buttons for 'Create rules for' (set to 'Cisco IOS (extended)'), 'Inbound' (checked), and 'Deny' (checked).

# Análisis del tráfico con Wireshark

Ej: I/O Graphs



# Índice



1. Introducción
2. Conexiones por segundo
3. Número de conexiones concurrentes
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software
8. Apache benchmark
9. httpperf
10. OpenWebLoad
11. Siege

# Tipos de tráfico

Hay patrones de tráfico muy comunes en sitios web:

- HTTP
- FTP o streaming
- tienda web

Patrón de tráfico	Métrica más importante	Segunda métrica más importante	Métrica menos importante
HTTP	Conexiones por segundo	Rendimiento	Total de conexiones concurrentes
FTP/streaming	Rendimiento	Total de conexiones concurrentes	Conexiones por segundo
Tienda web	Total de conexiones concurrentes	Conexiones por segundo	Rendimiento

# Tipos de tráfico

Tráfico HTTP:

Consumo ancho de banda intensivamente y genera muchas conexiones por segundo.

HTTP 1.0, se necesita una conexión para cada objeto.  
HTTP 1.1 envía con una sola conexión varios objetos.

Necesidad de hacer las páginas web ligeras, de forma que los usuarios puedan cargarlas rápidamente.

La programación web es importante!!

# Tipos de tráfico

Tráfico FTP / streaming:

Tras una conexión inicial (ya que usa UDP como protocolo),  
se envía una gran cantidad de información.

El número de conexiones para este tipo de tráfico es muy bajo comparado con la cantidad de información enviada.

Consumen el ancho de banda máximo rápidamente.

# Tipos de tráfico

Tráfico tipo “tienda web”:

La velocidad es el factor más importante.

Buena experiencia de usuario: si el usuario se desespera navegando en la tienda web, gastará poco dinero...

No se necesita un alto ancho de banda, ni tampoco va a haber demasiadas conexiones por segundo.

Sin embargo, el sitio debe dar soporte al máximo de usuarios navegando en sesiones largas al mismo tiempo.

# Índice



1. Introducción
2. Conexiones por segundo
3. Número de conexiones concurrentes
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software
8. Apache benchmark
9. httpperf
10. OpenWebLoad
11. Siege

# Límite de las prestaciones

**Existe un límite de tráfico de red suficientemente alto que produce una degradación grave en las prestaciones.**

En unos casos ese límite es más fácil de alcanzar que en otros.

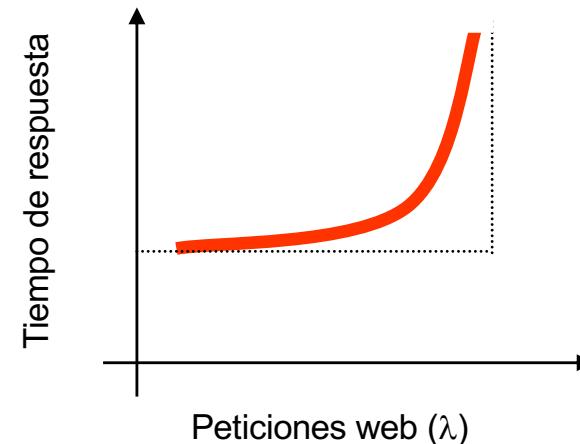
Llegado a ese límite los tiempos de respuesta en las conexiones HTTP se degradan completamente, haciendo imposible la conexión → perjudicando la disponibilidad

# Límite de las prestaciones

Si lo representamos gráficamente:

Esta degradación en las prestaciones se debe a los cuellos de botella.

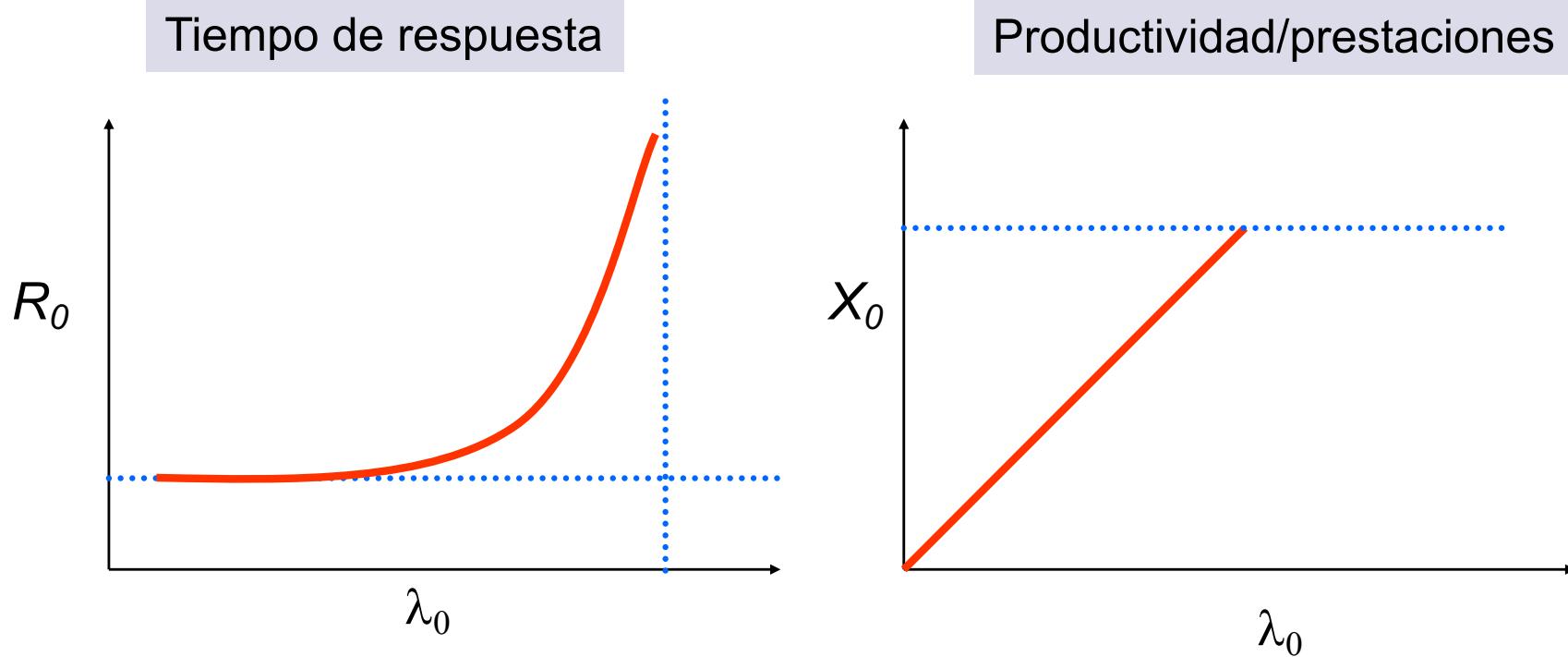
Hay que estudiar los datos de la monitorización durante las pruebas para determinar las carencias.



# Límite de las prestaciones

## Ejemplo 1: curva característica

Al subir la carga, las prestaciones/productividad se degradan y dejan de crecer.

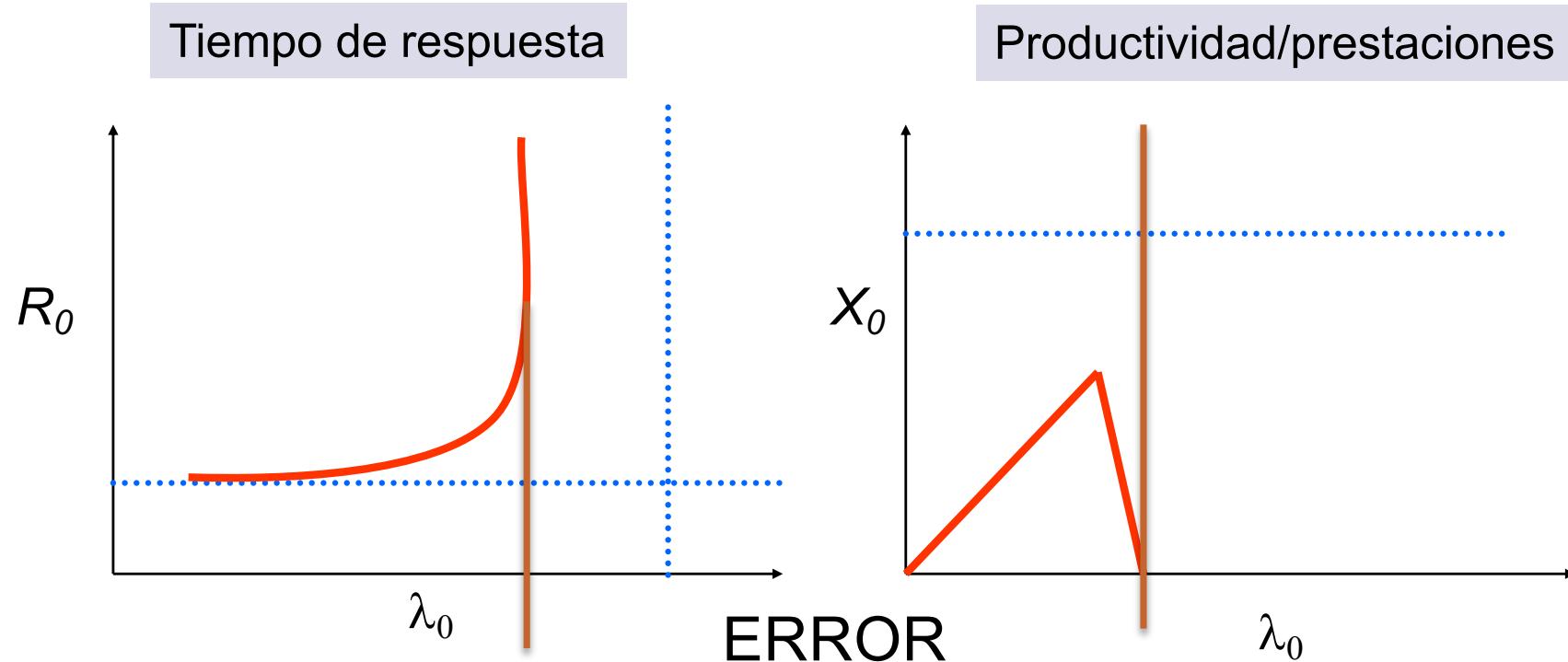


# Límite de las prestaciones

## Ejemplo 2: cuando ocurre algún problema...

Al fallar algún servicio, caen las prestaciones. --> También los tiempos de respuesta, al terminar las transacciones rápidamente con un error.

En estos casos es interesante examinar los *logs* (acceso y error).



# Límite de las prestaciones

Cada dispositivo de red puede comportarse de forma diferente, llegando a **cuelgues o reinicios**.

Estos límites son difíciles de alcanzar...  
pero a mayor número de características activas en un balanceador, será más fácil de alcanzar el límite:



Si es capaz de procesar tráfico a 90Mbps (suponemos máximo 100Mbps), puede ver reducido su rendimiento a la mitad si le pedimos que haga análisis de URLs y que de soporte de cookies (requieren uso más intensivo de la CPU para inspeccionar los paquetes completos y no solo la cabecera).



# Índice

1. Introducción
2. Conexiones por segundo
3. Número de conexiones concurrentes
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software para hacer tests
8. Apache benchmark
9. httpperf
10. OpenWebLoad
11. Siege

# Software para los tests - Benchmark

Necesarias **herramientas** para ejecutar en máquinas clientes y crear una carga HTTP específica.

Se suelen usar benchmarks como **SPECweb** o **WebStone** para simular un número determinado de clientes:

- <http://www.spec.org/benchmarks.html>
- <http://sourceforge.net/projects/webstone/>

El número de usuarios de un servidor web puede ser del orden de los millones de usuarios, así es que **simular un número pequeño de clientes no es realista** !



# ¿Cómo hacer los tests?

Consideraciones a tener en cuenta cuando vamos a evaluar el rendimiento de un sitio web real:

1. **Primero fijar un número alto de usuarios.** Calcular el tiempo medio cuando hay un alto número de usuarios haciendo peticiones al sitio web.
2. **Después, evaluar cómo se comporta el servidor cuando tiene el doble de usuarios.** Un servidor que tarda el doble en atender al doble de usuarios será mejor que otro que al doblar el número de usuarios (la carga) pase a tardar el triple.

# ¿Cómo hacer los tests?

En sistemas críticos, en lugar de usar (o desarrollar) una herramienta para generar la carga para los tests, se le puede encargar a una empresa externa especializada.

Algunas empresas ofrecen su herramienta y realizan los tests:

- Micro Focus Intl. - Segue Software (SilkPerformer)
- HP (LoadRunner)
- Micro Focus Intl. - Compuware (QALoad)
- Rational (SiteLoad)
- Radvie (WebLoad)

# Tipos de pruebas

Tenemos que elegir correctamente el tipo de pruebas:

- **Humo (Smoke)**: pruebas preliminares para comprobar que el sistema está listo para los siguientes tests.
- **Carga (Load)**: cargas lo más parecidas a la real. Se ejecutan en periodos cortos (1h). Para determinar los tiempos de respuesta que tendrán los usuarios.
- **Capacidad (Capacity)**: actividad creciente hasta detectar el punto de saturación.

# Tipos de pruebas

Tipos de pruebas (II):

- Estrés (Stress): para analizar el efecto de aplicar de forma continua una carga por encima de la capacidad del sistema.
- Sobrecarga (Overload): aplicar fuertes picos de carga durante cortos periodos.
- Estabilidad (Stability): cargas lo más similares posibles a la real, aplicadas durante 1 día o 1 semana.

# Durante los tests, monitorización

Durante la sesión de pruebas, recoger mediciones que nos indiquen lo que está ocurriendo en el sistema en cada momento y cómo reacciona éste en función de la carga introducida:

- Medidas de la calidad de servicio ofrecida por el sistema a los usuarios (estadísticas proporcionadas por la misma herramienta de simulación de carga)
- Medidas relativas al consumo de recursos del sistema (utilizando las herramientas del sistema operativo)

# Software para los tests

Diversas **herramientas** para comprobar el rendimiento de servidores web. Línea de comandos y de interfaz gráfica:

- Apache Benchmark <http://httpd.apache.org/docs/2.2/programs/ab.html>
- Siege <https://www.joedog.org/siege-home/>
- Weighttp <https://redmine.lighttpd.net/projects/weighttp/wiki>
- httpperf
- OpenWebLoad
- The Grinder
- OpenSTA
- JMeter
- Webstone (Mindcraft) <http://mindcraft.com/webstone/>

Las de línea de comandos sobrecargan menos las máquinas.

# Software para los tests

Estas herramientas permiten comprobar el rendimiento de cualquier servidor web (Apache, MS Internet Information Services -IIS-, nginx, Cherokee, Tomcat, lighttpd, httpd, etc).

**Comprobar el rendimiento del hardware, software o de alguna modificación que le hayamos hecho.**

Interesante combinar con WireShark!

# Índice



1. Introducción
2. Conexiones por segundo
3. Número de conexiones concurrentes
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software
8. Apache benchmark
9. httpperf
10. OpenWebLoad
11. Siege

# Apache Benchmark

ab no simula con total fidelidad el uso del sitio web que pueden hacer los usuarios habitualmente.

Pide la misma página repetidamente. Los usuarios reales no solicitan siempre la misma página.

Las medidas dan una idea aproximada del rendimiento del sitio, pero no reflejan el rendimiento real.

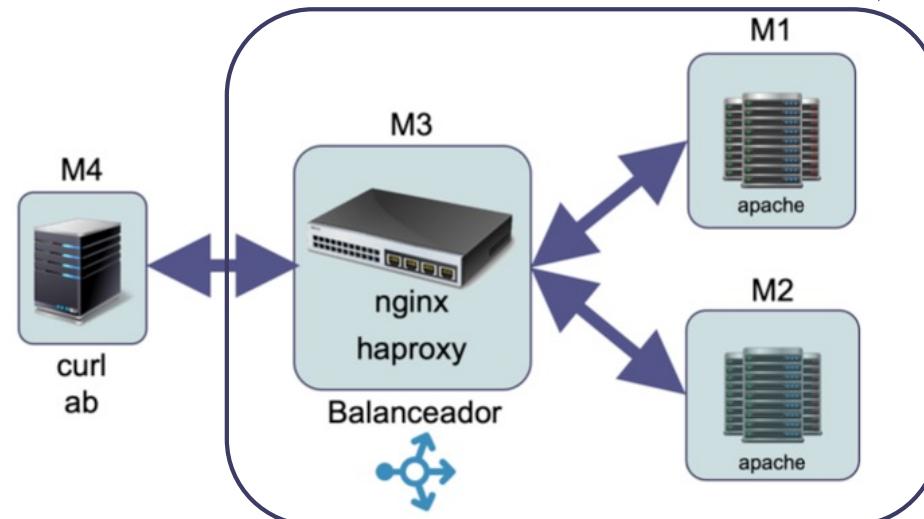
Va bien para testear cómo se comporta el servidor antes y después de modificar cierta configuración.

Teniendo los datos del “estado base”, podemos comparar cómo afecta una nueva configuración.

# Apache Benchmark

Debemos ejecutar el benchmark en otra máquina diferente a la que hace de servidor web.

Ambos procesos no deben consumir recursos de la misma máquina (veríamos un menor rendimiento).



Sin embargo, al hacerlo remotamente, introducimos cierta latencia debido a las comunicaciones.

# Apache Benchmark

Cada vez que ejecutemos el test obtendremos resultados ligeramente diferentes.

Esto es debido a que en el servidor hay diferente número de procesos en cada instante, y además la red puede encontrarse más sobrecargada en un momento que en otro.

Lo ideal es hacer al menos 30 ejecuciones, sacar resultados en **media y desviación estándar**, y representarlo gráficamente de forma adecuada.

# Apache Benchmark. Ejemplo

Para ejecutar el benchmark, usamos la sintaxis:

```
ab -n 1000 -c 5 http://maquina.com/prueba.html
```

-n 1000 => se solicita mil veces en total la URL

-c 5 => se hacen peticiones de 5 en 5 (conurrencia)

```
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
```

```
...
```

Concurrency Level: 5

Time taken for tests: 0.474 seconds

Complete requests: 1000

Failed requests: 0

Write errors: 0

```
...
```

Requests per second: 2109.82 [#/sec] (mean)

Time per request: 4.740 [ms] (mean)

Time per request: 0.474 [ms] (mean, across all concurrent requests)

Transfer rate: 733.49 [Kbytes/sec] received

# Índice



1. Introducción
2. Conexiones por segundo
3. Número de conexiones concurrentes
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software
8. Apache benchmark
9. httpperf
10. OpenWebLoad
11. Siege

# httpperf

httpperf es una herramienta para medir el rendimiento de sitios web.

Originalmente se desarrolló en los laboratorios de investigación de Hewlett-Packard.

Si tenemos varios clientes, deberíamos hacer la ejecución en todos simultáneamente.

De todas formas, puesto que los tests tardan varios minutos, que la ejecución comience con un segundo de diferencia, no afectará significativamente al resultado final.

# Httpperf. Ejemplo

La sintaxis de ejecución es:

```
httpperf --server maquina.com --uri /prueba.html --port 80 \
--num-conn 5000 --num-call 10 --rate 200 --timeout 5
```

Test del servidor *maquina.com*, puerto 80. Pedirá, de forma repetida, la página llamada "prueba.html"

Abrirá un total de **5000 conexiones** TCP para hacer con cada una de ellas peticiones HTTP (implica hacer la petición y esperar la respuesta).

Hará **10 peticiones por conexión**, y las hará a **200 conexiones por segundo** (implica **2000 peticiones/seg**).

*timeout* = segundos que el cliente esperará respuesta. Si pasa ese tiempo, considerará que la llamada habrá fallado.

# Httpperf. Ejemplo

La salida será similar a:

```
Total: connections 4986 requests 39620 replies 39620 test-duration 29.294 s
```

```
Connection rate: 170.2 conn/s (5.9 ms/conn, <=1022 concurrent connections)
```

```
Connection time [ms]: min 922.1 avg 4346.7 max 8045.6 median 4414.5 stddev 1618.6
```

```
Connection time [ms]: connect 643.6
```

```
Connection length [replies/conn]: 10.000
```

**Request rate: 1352.5 req/s** (0.7 ms/req)

```
Request size [B]: 58.0
```

```
Reply rate [replies/s]: min 1195.0 avg 1344.7 max 1393.1 stddev 84.1 (5 samples)
```

```
Reply time [ms]: response 370.3 transfer 0.0
```

```
Reply size [B]: header 167.0 content 2048.0 footer 0.0 (total 2215.0)
```

```
Reply status: 1xx=0 2xx=39620 3xx=0 4xx=0 5xx=0
```

```
CPU time [s]: user 1.35 system 27.95 (user 4.6% system 95.4% total 100.0%)
```

**Net I/O: 3002.2 KB/s** (24.6\*10<sup>6</sup> bps)

**Errors: total 1038** client-timo 1024 socket-timo 0 connrefused 0 connreset 0

```
Errors: fd-unavail 14 addrunavail 0 ftab-full 0 other 0
```

# Httpperf. Ejemplo

El ratio de peticiones (request rate) es menor de 2000 (sale 1352.5 peticiones/seg).

O bien el servidor está saturado y no soporta 2000 peticiones por segundo, o bien el cliente no puede llegar a hacerlas.

Ya sabemos el límite de nuestro servidor.

Ha habido 1038 errores:

- 1024 *timeouts* (tardó más de 5seg en llegar la respuesta a httpperf)
- 14 *fd-unavail*: httpperf intentaba abrir otro descriptor de fichero y no podía (se alcanzó el límite de descriptores abiertos por proceso establecido en el kernel de Linux, que es precisamente 1024).

La línea *Net I/O* muestra 24.6 Mbps (muy por debajo de 100 Mbps). Si estuviese cerca de 90Mbps habría que mejorar el ancho de banda.

# Índice



1. Introducción
2. Conexiones por segundo
3. Número de conexiones concurrentes
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software
8. Apache benchmark
9. httpperf
- [ 10. OpenWebLoad ]**
11. Siege

# OpenWebLoad

OpenWebLoad es otra herramienta de línea de comandos para medir el rendimiento de servidores web:

```
openload [options] http://maquina.com 10
```

El programa recibe **dos parámetros**, muy similares a los de las herramientas anteriores:

- La URL de la página en el servidor.
- El número de clientes simultáneos que simularemos (es un parámetro opcional y el valor por defecto es 5).

El programa ofrece más opciones:

[http://openwebload.sourceforge.net/cmd\\_parms.html](http://openwebload.sourceforge.net/cmd_parms.html)

# OpenWebLoad. Ejemplo

## Sintaxis de uso:

```
openload maquina.com 10
```

## Salida del benchmark:

URL: <http://maquina.com:80/>

Clients: 10

MaTps 355.11, Tps 355.11, Resp Time 0.015, Err 0%, Count 511
MaTps 339.50, Tps 199.00, Resp Time 0.051, Err 0%, Count 711
MaTps 343.72, Tps 381.68, Resp Time 0.032, Err 0%, Count 1111
MaTps 382.04, Tps 727.00, Resp Time 0.020, Err 0%, Count 1838
MaTps 398.54, Tps 547.00, Resp Time 0.018, Err 0%, Count 2385
MaTps 425.78, Tps 670.90, Resp Time 0.014, Err 0%, Count 3072

**Total TPS: 452.90**

**Avg. Response time: 0.021 sec.**

Max Response time: 0.769 sec

# Índice



1. Introducción
2. Conexiones por segundo
3. Número de conexiones concurrentes
4. Rendimiento, en bits por segundo
5. Tipos de tráfico
6. Límite en las prestaciones
7. Software
8. Apache benchmark
9. httpperf
10. OpenWebLoad
11. Siege

# Siege

Siege es una herramienta de generación de carga HTTP para benchmarking parecida a Apache Benchmark:

```
siege -b -t60S -v http://maquina.com
```

El programa recibe **varios parámetros**:

- -b para indicar que haga los tests de forma continua, y no interactivos.
- -t para indicar el tiempo que queremos que esté en ejecución el programa.
- -v para indicar que genere una salida detallada.
- La URL a la que queremos “atacar”.

Por defecto usará 15 conexiones concurrentes durante ese tiempo indicado.

# Siege. Ejemplo

```
** Preparing 15 concurrent users for battle.  
The server is now under siege...  
HTTP/1.1 200 0.03 secs: 9405 bytes ==> GET /  
... ... ... ...  
HTTP/1.1 200 0.02 secs: 9405 bytes ==> GET /  
[error] socket: 208723968 connection refused.: Connection refused  
HTTP/1.1 200 5.63 secs: 9405 bytes ==> GET /  
... ... ... ...  
HTTP/1.1 200 0.01 secs: 9405 bytes ==> GET /  
Lifting the server siege... done.
```

<b>Transactions:</b>	17317 hits
<b>Availability:</b>	99.99 %
<b>Elapsed time:</b>	59.97 secs
Data transferred:	155.35 MB
<b>Response time:</b>	0.05 secs
<b>Transaction rate:</b>	288.76 trans/sec
Throughput:	2.59 MB/sec
<b>Concurrency:</b>	13.95
Successful transactions:	17317
<b>Failed transactions:</b>	1
<b>Longest transaction:</b>	20.44
Shortest transaction:	0.01

# TEMA 7

## Almacenamiento en la granja web

SWAP



¿Cómo se almacenan los datos de la  
granja web?  
¿Será robusto y escalable?



José Manuel Soto Hidalgo  
Dpto. Arquitectura y Tecnología de Computadores  
Universidad de Granada

jmsoto@ugr.es

# Índice



- [ 1. Introducción ]
- 2. Tecnologías hardware para BD
- 3. Tecnología RAID
- 4. SSA
- 5. SAN
- 6. NAS
- 7. Conclusiones

# Introducción

El sistema de almacenamiento de datos resulta clave en un sistema web de altas prestaciones.

Parte del sistema donde se guarda la información, ya sea en una BD o en archivos.



# Introducción

Diseñar teniendo en mente ciertos requisitos en cuanto a escalabilidad es esencial

Todo usuario que llegue al sistema accederá a los datos almacenados, y debemos estar preparados para servir datos a un número creciente de usuarios.



# Introducción

Los gestores de BD y el diseño de éstas deben ser **robustas** para soportar **múltiples accesos concurrentes**.

Podemos mejorar las prestaciones de los sistemas de almacenamiento:

- **ampliación vertical** (adquirir un mejor hardware más rápido y actualizado)
- **ampliación horizontal** (replicar el almacenamiento entre varios servidores); puede resultar más efectivo en cuanto a la escalabilidad

# Introducción

Posibles problemas de realizar la replicación y repartir la carga:

- El coste de nuevos servidores y almacenamiento.
- La configuración de métodos y rutinas de replicación y sincronización.
- La latencia en los procesos de replicación.
- La necesidad de un sistema de balanceo de carga adecuado entre los servidores de BD.

# Introducción

**Estrategias alternativas a la replicación completa para mejorar el sistema de almacenamiento y BD:**

**Realizar distribución funcional:**

dividir la BD global en varias secciones relativas a aplicaciones diferentes (p.ej. inventario, usuarios, mensajería, etc).

Configurar varios servidores que hospedarán cada sección de la BD.

Es complicado mantener la integridad de los datos entre las diferentes secciones.

# Introducción

**Estrategias alternativas a la replicación completa para mejorar el sistema de almacenamiento y BD:**

## Segmentar la BD:

hacer una división lógica de la BD, p.ej. en función del tipo de clientes o según periodos contables.

Cada segmento queda almacenado en un servidor de BD, quedando repartida así la carga.

Es complicado mantener la integridad de los datos entre las diferentes divisiones.

# Introducción

Existen productos de **BD propietarios** en los cuales se pueden usar **extensiones** que facilitan la interacción entre varios servidores para gestionar una sola gran BD.

Suelen depender estrechamente de un sistema operativo o de un sistema de distribución muy concretos.

- Oracle11g <http://www.oracle.com/us/products/database/overview/index.html>
  - SQL Server 2008 R2 <http://msdn.microsoft.com/en-us/library/ms191440.aspx>
  - Apache Cassandra <http://cassandra.apache.org/>
- 
- Configurar con MySQL un cluster de BD (práctica 5).

# Índice



1. Introducción
2. Tecnologías hardware para BD
3. Tecnología RAID
4. SSA
5. SAN
6. NAS
7. Conclusiones

# Tecnologías hardware

El sistema de almacenamiento y de BD es un punto fundamental en cualquier sistema web actual.

Una mala configuración afectará a las prestaciones.

Hay que ser cuidadosos con el hardware y software.

El hardware del resto del sistema web puede actualizarse en cualquier momento casi sin que los usuarios lo noten.

El de la BD es crítico, ya que no se podrá actualizar de forma fácil una vez que esté en funcionamiento.

# Tecnologías hardware

Factores a tener en cuenta al diseñar la arquitectura de BD:

- El número de sesiones concurrentes en la BD puede afectar al rendimiento de la granja web completa (conexiones costosas).
- El tipo de accesos a la BD también influye.
- Las búsquedas que devuelvan resultados muy grandes afectarán al rendimiento de CPU, almacenamiento y red.

# Tecnologías hardware

Factores a tener en cuenta al diseñar la arquitectura de BD:

- El tamaño total de la BD determinará el espacio para almacenamiento, y el tiempo necesario para hacer copias de seguridad y restaurarlas.
- Conviene utilizar hardware redundante para los servidores.
- Una gran cantidad de accesos a la BD por cada petición HTTP puede sobrecargar la conexión de red entre los servidores web y de BD.

# Tecnologías hardware

Factores a tener en cuenta al diseñar la arquitectura de BD:

- Arquitectura de la BD basada en un cluster.
- Una BD se podrá escalar en el futuro si desde el principio se instaló hardware con capacidad de ampliación (CPU, memoria, etc) y se configuró el software de forma adecuada.



# Índice

1. Introducción
2. Tecnologías hardware para BD
3. Tecnología RAID
4. SSA
5. SAN
6. NAS
7. Conclusiones

# Almacenamiento basado en RAID

¿Qué es RAID y qué no es RAID?



# Almacenamiento basado en RAID

¿Qué es RAID y qué no es RAID?

*Redundant Array of Independent Disks*

<http://es.wikipedia.org/wiki/RAID>



# Almacenamiento basado en RAID

RAID (*conjunto redundante de discos independientes*) es un sistema de almacenamiento que usa múltiples discos duros entre los que se distribuyen o replican los datos.

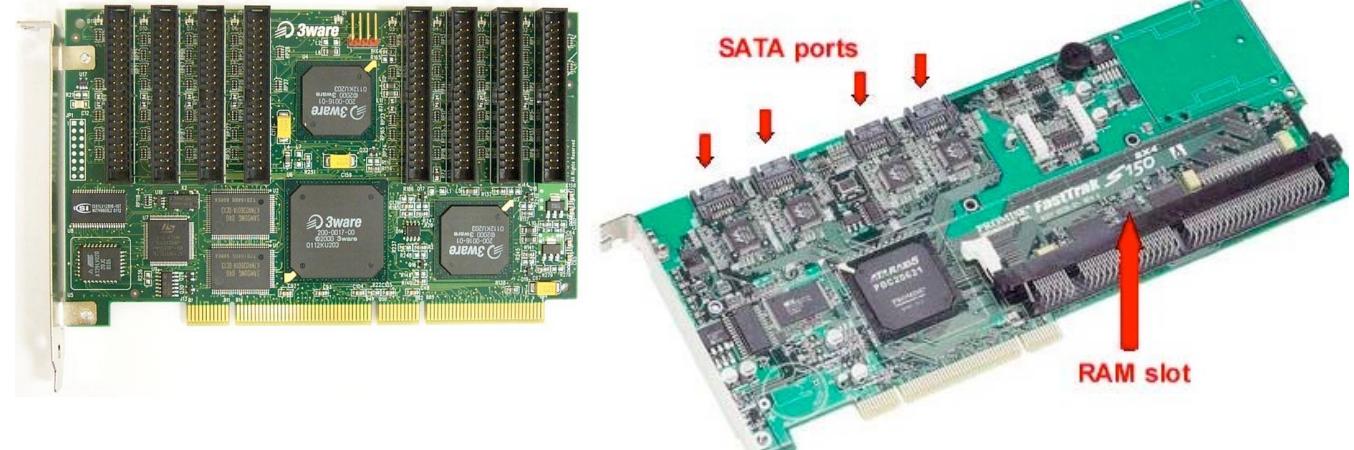
Ofrece mayor integridad, mayor tolerancia a fallos, mayor rendimiento y mayor capacidad.

La idea inicial es combinar varios dispositivos en un conjunto que ofrece mayor capacidad, fiabilidad y velocidad que un solo dispositivo de última generación más caro.

# Almacenamiento basado en RAID

Un RAID por hardware es mucho más rápido que uno configurado por software.

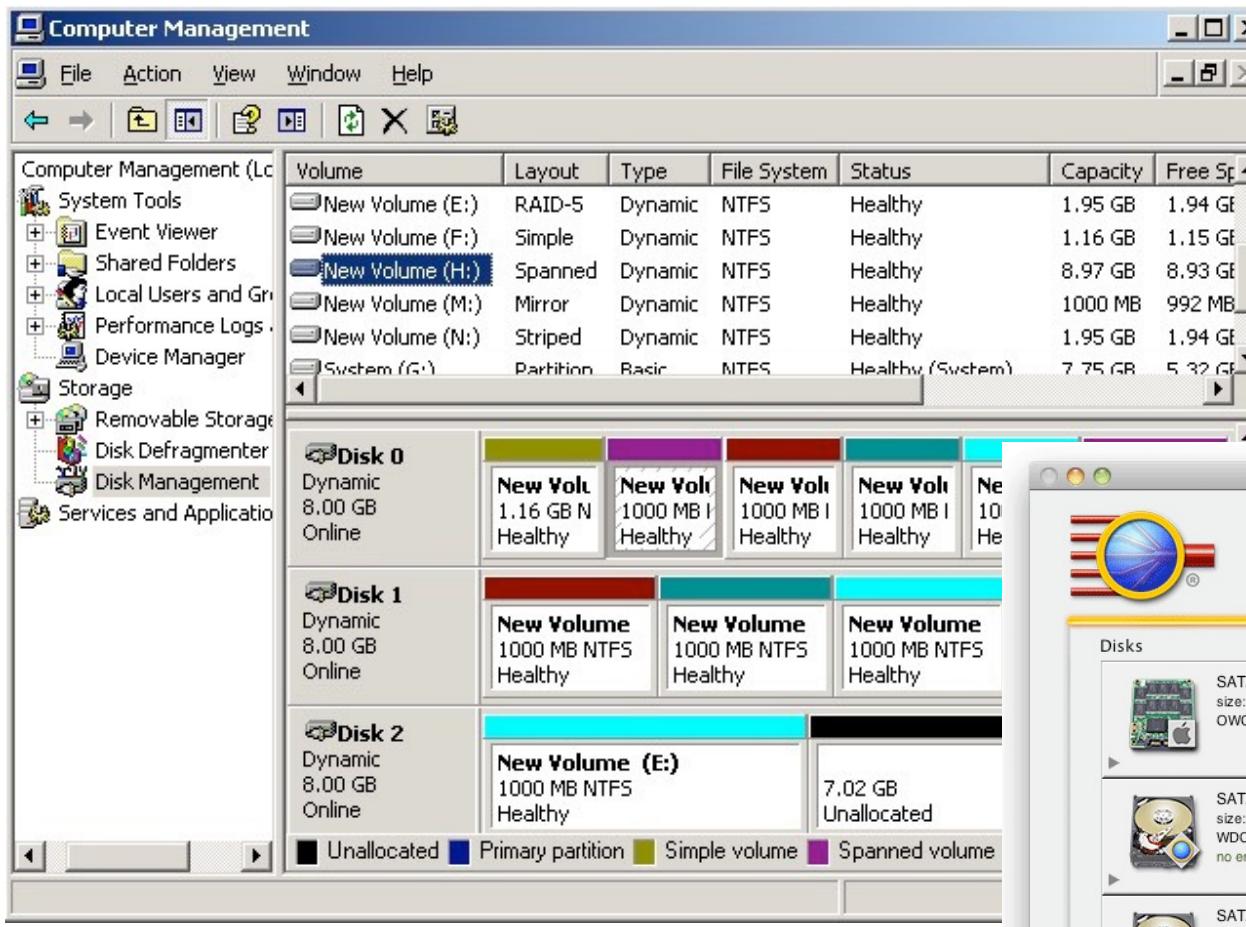
From Computer Desktop Encyclopedia  
© 2004 The Computer Language Co. Inc.



Por software son mucho más flexibles:

- permiten construir RAID de particiones en lugar de discos completos
- agrupar en un mismo RAID discos conectados en varias controladoras.

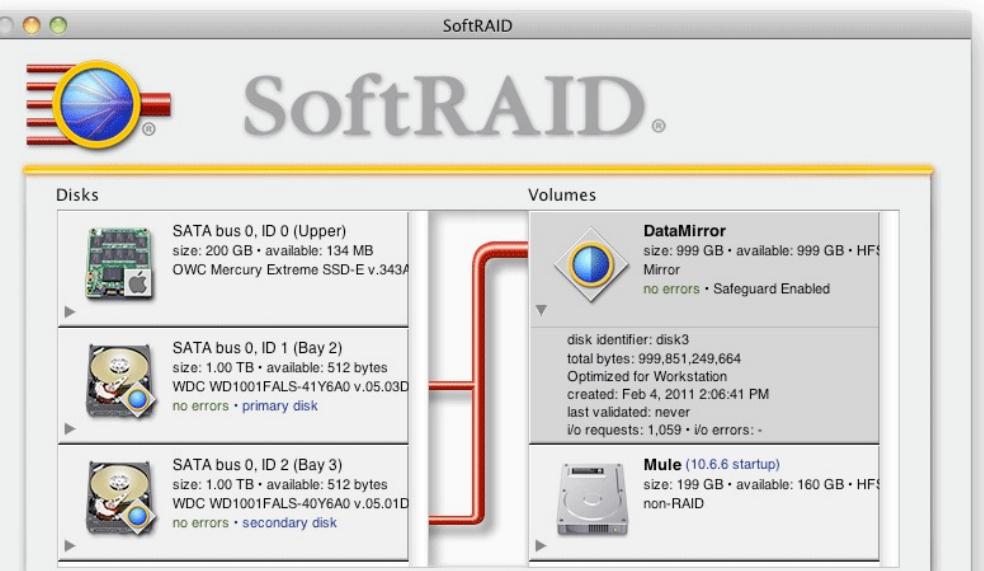
# Almacenamiento basado en RAID



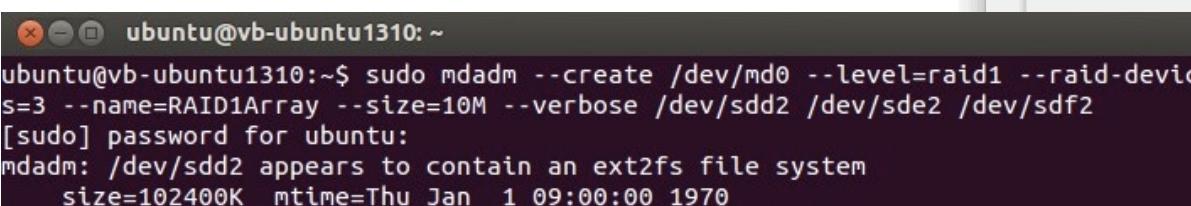
The screenshot shows the Windows Computer Management Disk Management interface. The left sidebar lists storage-related tools like Event Viewer, Shared Folders, Local Users and Groups, Performance Logs, Device Manager, Storage, Removable Storage, Disk Defragmenter, and Disk Management. The main pane displays a table of volumes and their properties:

Volume	Layout	Type	File System	Status	Capacity	Free Sp.
New Volume (E:)	RAID-5	Dynamic	NTFS	Healthy	1.95 GB	1.94 GB
New Volume (F:)	Simple	Dynamic	NTFS	Healthy	1.16 GB	1.15 GB
<b>New Volume (H:)</b>	Spanned	Dynamic	NTFS	Healthy	8.97 GB	8.93 GB
New Volume (M:)	Mirror	Dynamic	NTFS	Healthy	1000 MB	992 MB
New Volume (N:)	Striped	Dynamic	NTFS	Healthy	1.95 GB	1.94 GB
System (G:)	Partition	Basic	NTFS	Healthy (System)	7.75 GB	5.32 GB

The Disk Management interface also shows three disks (Disk 0, Disk 1, Disk 2) with their volume configurations. Disk 0 has four simple volumes. Disk 1 has three simple volumes. Disk 2 has one simple volume (E:) and one unallocated 7.02 GB space.



The SoftRAID application window shows a RAID setup. It lists three physical disks (SATA bus 0, ID 0, 1, 2) and a single DataMirror volume (disk identifier: disk3). The DataMirror volume is a mirror of the first disk and is optimized for a workstation. The Mule volume is a non-RAID volume.

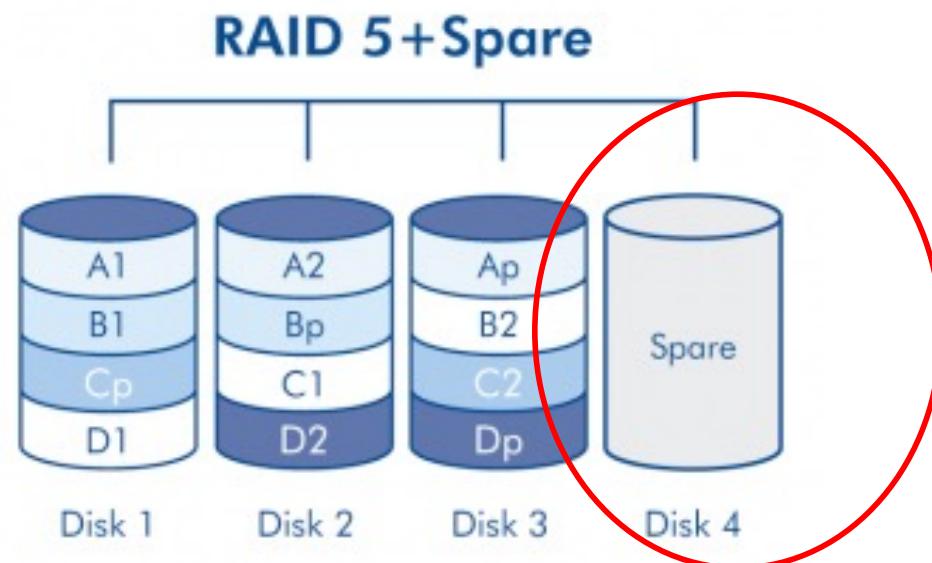


The terminal window shows the execution of the `sudo mdadm --create /dev/md0 --level=raid1 --raid-device=s=3 --name=RAID1Array --size=10M --verbose /dev/sdd2 /dev/sde2 /dev/sdf2` command. The output indicates that `/dev/sdd2` appears to contain an ext2fs file system with a size of 102400K and a mtime of Thu Jan 1 09:00:00 1970.

# Almacenamiento basado en RAID

La tecnología RAID soporta el uso de varios discos de reserva (*hot spare*), para usarse inmediatamente y de forma automática tras el fallo de uno de los discos.

Esto reduce el tiempo del período de reparación al acortar el tiempo de reconstrucción del RAID.



# Almacenamiento basado en RAID

## Niveles RAID

Hay diversos métodos de almacenamiento, llamados niveles, con diferente complejidad:

- RAID 0: Conjunto dividido
- RAID 1: Conjunto en espejo
- RAID 5: Conjunto dividido con paridad distribuida

[http://en.wikipedia.org/wiki/Standard\\_RAID\\_levels](http://en.wikipedia.org/wiki/Standard_RAID_levels)

**Podemos anidar niveles RAID:** que un RAID pueda usarse como elemento básico de otro en lugar de discos físicos.

# Almacenamiento basado en RAID

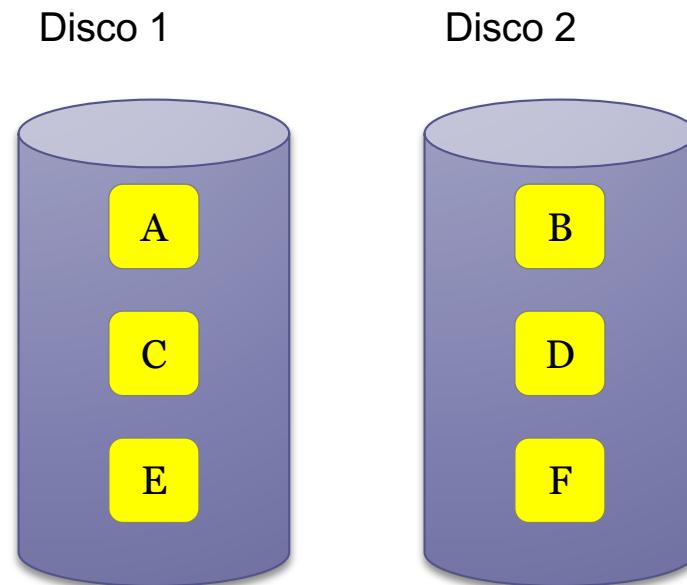
## RAID 0

Reparte los datos entre varios discos => incremento de la velocidad de lectura y escritura.

Se puede acceder a varios bloques consecutivos al mismo tiempo.

Esta configuración **no ofrece protección contra fallos** en los discos, ya que no se escribe información duplicada o información de paridad.

striping



# Almacenamiento basado en RAID

## RAID 0

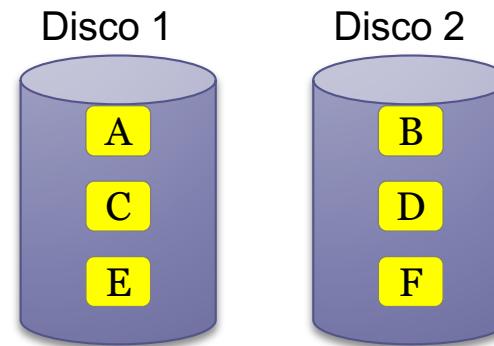
La velocidad de transferencia (ideal) se puede ver como la suma de las velocidades de transferencia de todos los discos.

Se suele usar en configuraciones de servidor NFS.

# Almacenamiento basado en RAID

## Ejercicio:

*¿Qué tamaño de unidad de unidad RAID se obtendrá al configurar un RAID 0 a partir de dos discos de 100 GB y 100 GB?*



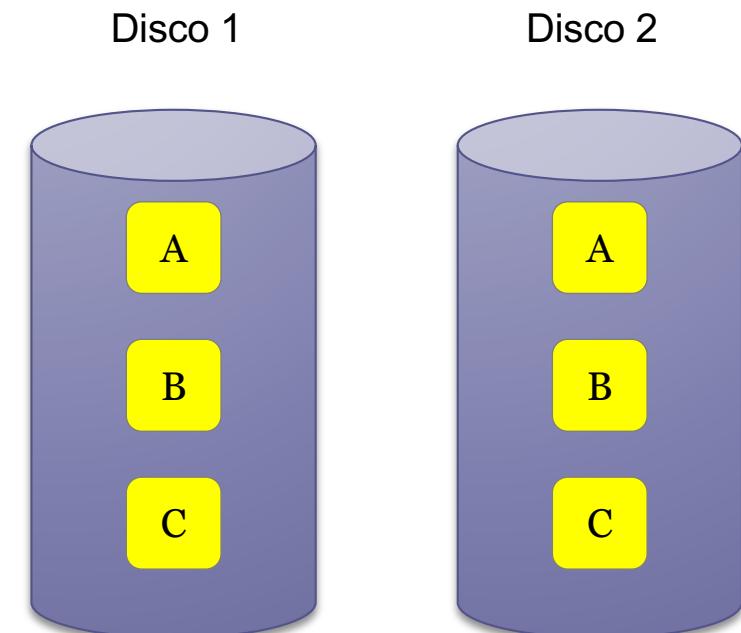
*¿Qué tamaño de unidad de unidad RAID se obtendrá al configurar un RAID 0 a partir de tres discos de 200 GB cada uno?*

# Almacenamiento basado en RAID

## RAID 1

Crea una copia exacta (o espejo) de un conjunto de datos en dos o más discos. Ofrece gran fiabilidad, ya que para que el conjunto falle es necesario que lo hagan todos sus mirroring.

Como los discos que forman el RAID 1 tienen hardware independiente, se puede leer simultáneamente dos datos diferentes en dos discos diferentes, por lo que su rendimiento se duplica.



# Almacenamiento basado en RAID

## RAID 1

Útil si la seguridad de los datos es más importante que la capacidad de almacenamiento total.

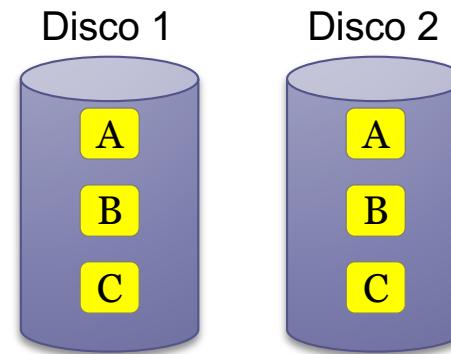
Se recomienda tener controladoras independientes para cada disco.

Ventajas desde el punto de vista administrativo: se puede poner un disco inactivo para hacer backup de los datos, mientras que el otro sigue dando servicio.

# Almacenamiento basado en RAID

## Ejercicio:

*¿Qué tamaño de unidad de unidad RAID se obtendrá al configurar un RAID 1 a partir de dos discos de 100 GB y 100 GB?*



*¿Qué tamaño de unidad de unidad RAID se obtendrá al configurar un RAID 1 a partir de tres discos de 200 GB cada uno?*

# Almacenamiento basado en RAID

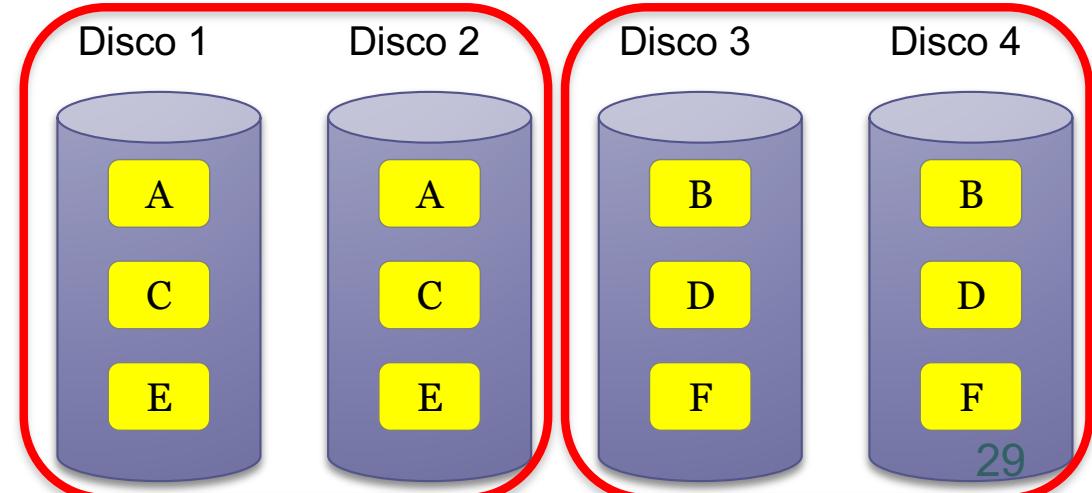
RAID 10 o RAID 1+0 es una **división de espejos**

La **duplicación** (RAID 1) significa grabar los datos en dos o más discos al mismo tiempo.

Si un disco falla por completo, la duplicación conserva la información.

El **reparto de los datos** (RAID 0) divide los datos en fragmentos y los graba en conjuntos distintos de forma sucesiva.

Mejora el rendimiento porque el equipo puede obtener datos de más de un conjunto a la vez.



RAID Level	Description	Performance and Fault Tolerance
10 (1+0)	RAID 0 (stripe) built with RAID 1 (mirror) arrays	<p>RAID 1+0 provides high levels of I/O performance, data redundancy, and disk fault tolerance. Because each member device in the RAID 0 is mirrored individually, multiple disk failures can be tolerated and data remains available as long as the disks that fail are in different mirrors.</p> <p>You can optionally configure a spare for each underlying mirrored array, or configure a spare to serve a spare group that serves all mirrors.</p>
10 (0+1)	RAID 1 (mirror) built with RAID 0 (stripe) arrays	<p>RAID 0+1 provides high levels of I/O performance and data redundancy, but slightly less fault tolerance than a 1+0. If multiple disks fail on one side of the mirror, then the other mirror is available. However, if disks are lost concurrently on both sides of the mirror, all data is lost.</p> <p>This solution offers less disk fault tolerance than a 1+0 solution, but if you need to perform maintenance or maintain the mirror on a different site, you can take an entire side of the mirror offline and still have a fully functional storage device. Also, if you lose the connection between the two sites, either site operates independently of the other. That is not true if you stripe the mirrored segments, because the mirrors are managed at a lower level.</p> <p>If a device fails, the mirror on that side fails because RAID 1 is not fault-tolerant. Create a new RAID 0 to replace the failed side, then resynchronize the mirrors.</p>

# Almacenamiento basado en RAID

## ¿Qué puede y qué no puede hacer RAID?



- Permite acceder a los datos aunque falle un disco.
- Puede mejorar el rendimiento de ciertas aplicaciones (para archivos grandes mantiene tasas de transferencia altas).



- No protege los datos (p.ej. por virus).
- No simplifica la recuperación de un desastre.
- No mejora el rendimiento para todas las aplicaciones.
- No facilita el traslado del almacenamiento a un sistema nuevo.

# Tutoriales

Instalación Ubuntu server con RAID1

<http://www.youtube.com/watch?v=DS4uKJ9pfnk>

Instalación ubuntu server 12.04 LTS precise pangoline con RAID 1 software

<http://www.youtube.com/watch?v=y17EfNs0TBc>

Como crear un RAID 1 en Windows

<http://www.youtube.com/watch?v=g5I-1IXgwRo>

Raid 1 - Sincronización espejo en Windows Server 2008

<http://www.youtube.com/watch?v=k92yKphhKYE>

Instalación y configuración RAID de dos discos duros en un Mac PRo

<http://www.youtube.com/watch?v=O5VuJSRjLT8>

Xserve RAID Install

<http://www.youtube.com/watch?v=WxsQ2Y1iW7w>

# Índice

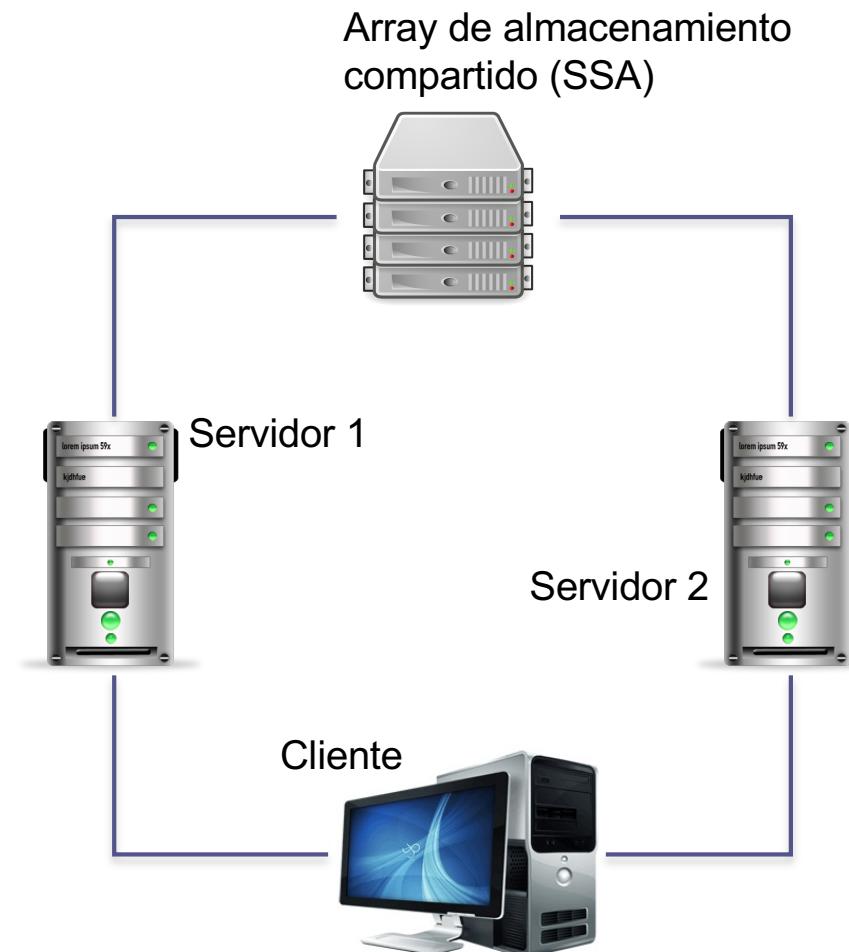


1. Introducción
2. Tecnologías hardware para BD
3. Tecnología RAID
4. SSA
5. SAN
6. NAS
7. Conclusiones

# Arrays de almacenamiento compartido: SSA

Forma simple de almacenamiento externo.

Dispositivo específico que incluye varios discos en rack:



# Arrays de almacenamiento compartido: SSA

- Posee una interfaz para conectar los discos a las controladoras (normalmente SCSI).
- Número limitado de puertos para hacer la conexión entre servidores y almacenamiento.
- Se suele usar para disponer del almacenamiento necesario para archivos y BD en clusters.
- La posibilidad de manejo y la flexibilidad de un SSA es limitada. Aceptan cambios en caliente de discos y varias configuraciones RAID.
- Dispositivos desarrollados por una empresa con unas especificaciones y herramientas propietarias.

# Índice



1. Introducción
2. Tecnologías hardware para BD
3. Tecnología RAID
4. SSA
5. SAN
6. NAS
7. Conclusiones

# Área de almacenamiento en red: SAN

**Red de almacenamiento especializada que conecta dispositivos de almacenamiento a los servidores.**

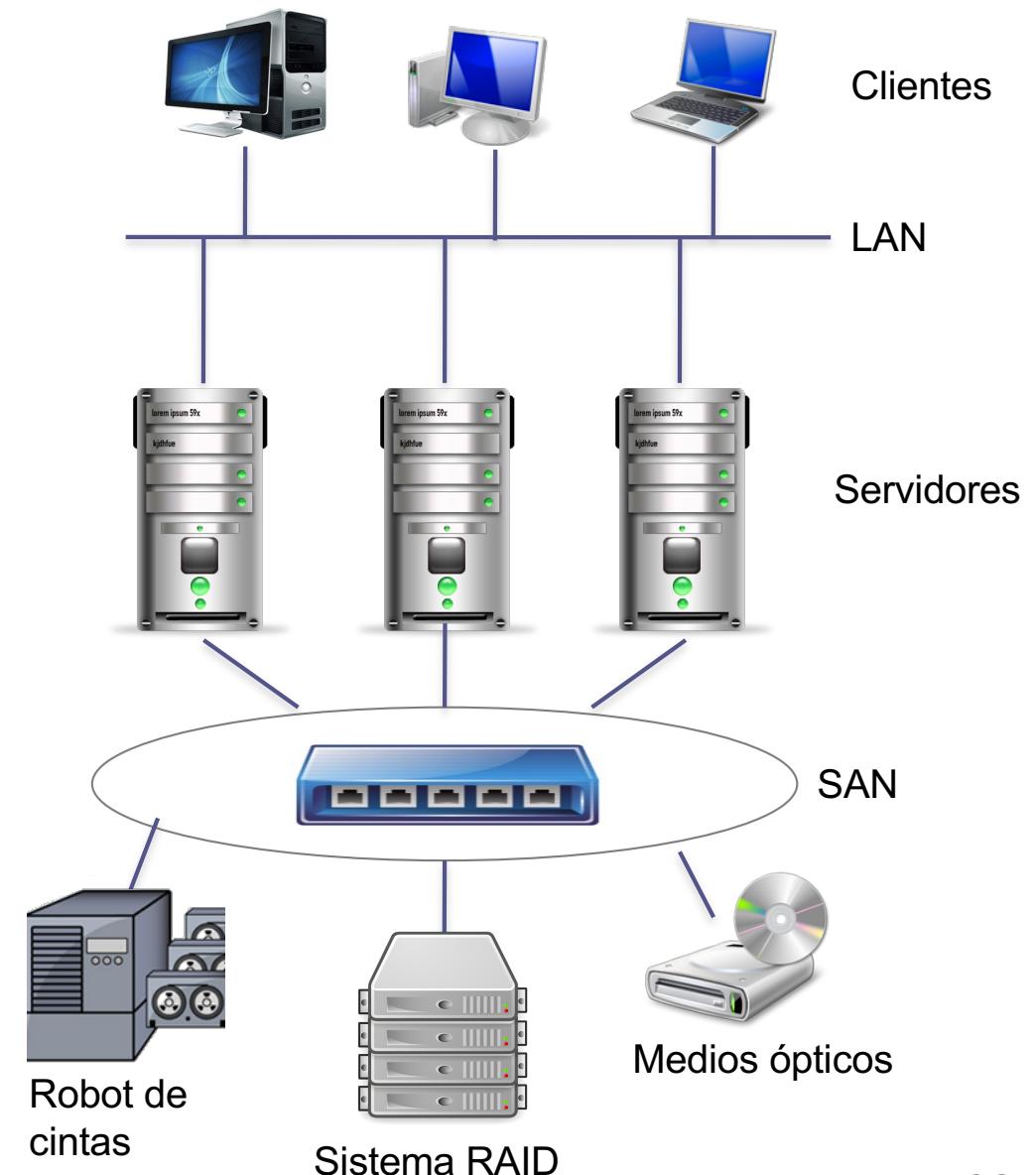
Conjunto de dispositivos interconectados (discos, cintas, etc.) y servidores conectados a un canal de comunicación e intercambio de datos común (concentrador de alta velocidad).

# Área de almacenamiento en red: SAN

Esquema:

Gran flexibilidad y  
facilidad de manejo del  
almacenamiento.

Se puede actualizar  
cualquier componente.



# Área de almacenamiento en red: SAN

- Red de alta velocidad (mínimo de 1Gbps). Es como un bus de un ordenador, pero compartido entre varias máquinas.
- Utiliza hardware de red muy especializado.
- Una SAN ofrece una capa de abstracción entre los dispositivos de almacenamiento y los servidores, y permite que el espacio físico de almacenamiento crezca.
- Se puede usar para almacenar archivos, compartir datos entre los servidores, mirroring de discos y backups.
- Puede operar con SSA y NAS.
- Permite que se añadan nuevos dispositivos al sistema (servidores o almacenamiento).



# Índice

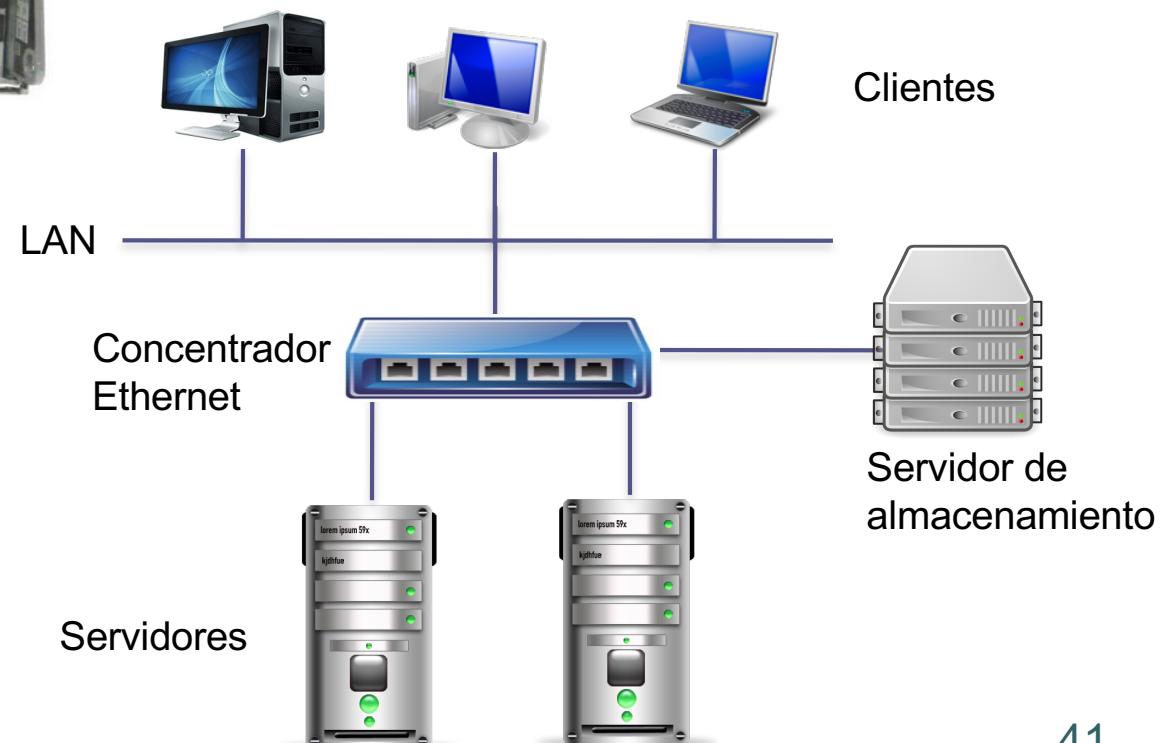
1. Introducción
2. Tecnologías hardware para BD
3. Tecnología RAID
4. SSA
5. SAN
6. NAS
7. Conclusiones

# Almacenamiento conectado a la red: NAS

Dispositivo que actúa como un servidor de ficheros, pero ahorrando los recursos de tener una máquina más.



para almacenar copias de seguridad, y para ofrecer espacio de almacenamiento compartido



# Almacenamiento conectado a la red: NAS

- Conjunto de discos organizados en un dispositivo de red con IP y que puede conectarse a una red Ethernet.
- Utilizando algún protocolo, como Internetwork Packet Exchange (de Microsoft), NetBEUI (de Microsoft), Network File System (NFS, de Sun) o IPE (de Novell).
- Aparece como otro servidor más en la red.
- Usan software específico para configurarlos y manejarlos (creación de unidades, gestión de permisos, etc).
- Utilizan configuraciones RAID.

# Ejemplo de NAS. *openmediavault*

Sistema de almacenamiento en red basado en Debian:

<http://www.openmediavault.org/>

<http://en.wikipedia.org/wiki/OpenMediaVault>

Distribución Linux basada en Debian pensada para configurar un NAS con un PC.

Servicios: ssh, sftp, smb/cifs, rsync

Requisitos hardware: 1GByte de RAM, 2 GByte de disco para el sistema operativo, y los discos duros que usen para el servicio de almacenamiento en red.

<https://ostechnix.wordpress.com/2013/01/17/openmediavault-setup-your-own-nasnetwork-attached-storage-box-in-minutes/>

# Ejemplo de NAS. *openmediavault*

The screenshot shows the OpenMediaVault web interface. The top banner features the OpenMediaVault logo and the tagline "The open network attached storage solution". Below the banner, the main window has a sidebar on the left containing a navigation tree:

- System
  - General Settings
  - Date & Time
  - Network
  - Notification
  - Power Management
  - Certificates
  - Cron Jobs
  - Update Manager
  - Plugins
- Storage
  - Physical Disks
  - RAID Management
  - Filesystems
  - S.M.A.R.T.
- Access Right Management
  - User
  - Group
  - Shared Folders
- Services
  - TFTP
  - SNMP
  - SSH

The main content area is titled "Storage | Physical Disks" and displays a table of physical disks:

Device	Model	Serial Number	Vendor	Capacity
/dev/sda	TRANSCEND	[REDACTED]	ATA	3.73 GiB
/dev/sdb	WDC WD15EARS-00Z5B1	[REDACTED]	ATA	1.36 TiB
/dev/sdc	WDC WD15EARS-00Z5B1	[REDACTED]	ATA	1.36 TiB
/dev/sdd	WDC WD15EARS-00Z5B1	[REDACTED]	ATA	1.36 TiB
/dev/sde	WDC WD15EARS-00MVWB0	[REDACTED]	ATA	1.36 TiB
/dev/sdf	WDC WD15EARS-00MVWB0	[REDACTED]	ATA	1.36 TiB

At the bottom of the page, there are navigation buttons for "Page 1 of 1" and a status message "Displaying items 44 of 6".

# Índice



1. Introducción
2. Tecnologías hardware para BD
3. Tecnología RAID
4. SSA
5. SAN
6. NAS
7. Conclusiones

# Conclusiones

**La arquitectura de almacenamiento de la granja web resulta fundamental para la disponibilidad de las aplicaciones.**

**El escalado del sistema de BD resultará muy importante a lo largo de la vida del sistema web.**

**La configuración de un sistema RAID supondrá una mejora en la disponibilidad y en la seguridad de nuestros datos.**

**Además, se mejorará la capacidad del sistema de entrada/salida a disco.**

# Conclusiones

Otras soluciones pasan por la instalación y configuración de **sistemas avanzados de almacenamiento** usando tecnologías de red (SSA, SAN y NAS).

Estas tecnologías ofrecen **flexibilidad y la posibilidad de escalar** el sistema de almacenamiento en el futuro.

En resumen, la **arquitectura de BD** del sistema web debe ser lo más robusta posible, con capacidad para crecer (**ampliable y escalable**).

Convendrá realizar un buen análisis, adquirir un buen hardware e instalar un buen software al principio.

# TEMA 8

## Requisitos hardware y software en la granja web

SWAP

¿Qué requisitos hardware deben tener  
los elementos de la granja web?  
¿Serán adecuados?



José Manuel Soto Hidalgo  
Dpto. Arquitectura y Tecnología de Computadores  
Universidad de Granada

jmsoto@ugr.es

# Índice

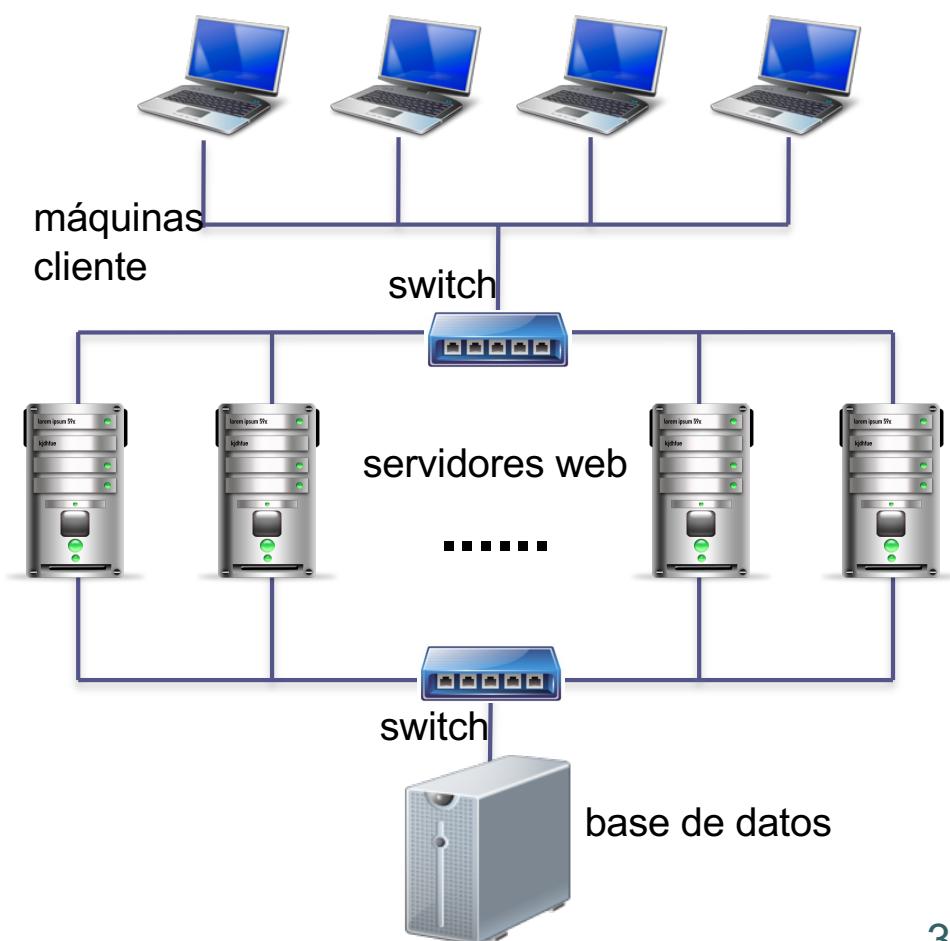
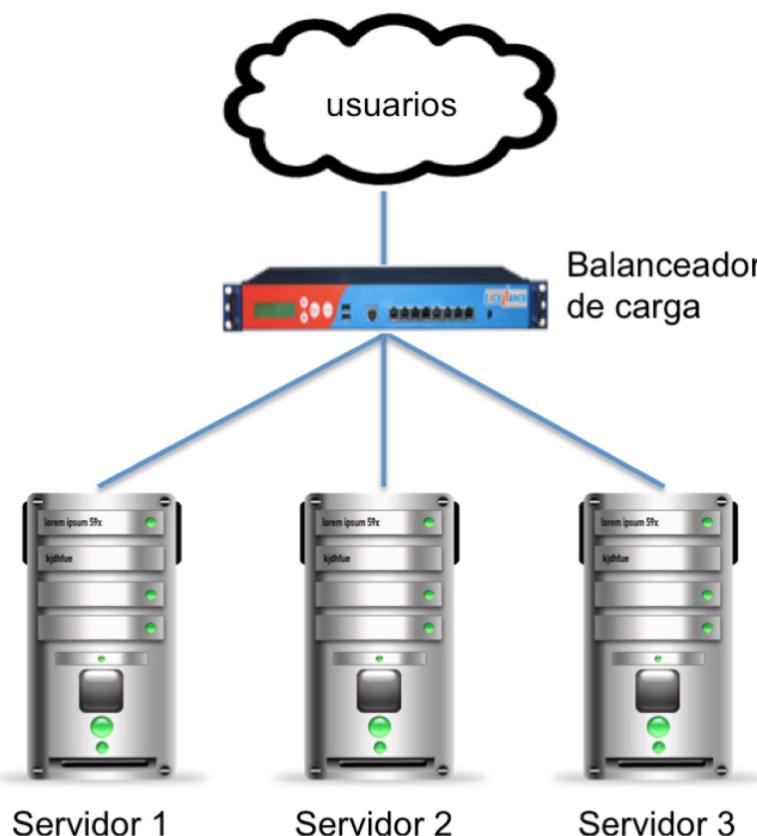
## [ 1. Introducción ]

2. Elementos de la granja web
3. Necesidades del servidor web
4. Hardware para servidores
5. Software para servidores
6. Conclusiones



# Introducción

Usar clústers web incrementa la capacidad de ofrecer servicios usando tecnologías hardware y software estándar.



# Introducción

## Clasificación:

- High Performance Computing Clusters (HPCC) o clústers de alto rendimiento.
- High Availability Computing Clusters (HACC) o clústers de alta disponibilidad.
- High Throughput Computing Clusters (HTCC) o clústers de alta eficiencia.

# Índice



1. Introducción
2. Elementos de la granja web
3. Necesidades del servidor web
4. Hardware para servidores
5. Software para servidores
6. Conclusiones

# Elementos de la granja web

La granja web necesita varios componentes de software y hardware para poder funcionar.

En cuanto al hardware, cabe destacar:

- servidores
- almacenamiento
- conexiones de red

El software de la granja web es muy importante (última sección del tema).

# Elementos de la granja web

Clientes

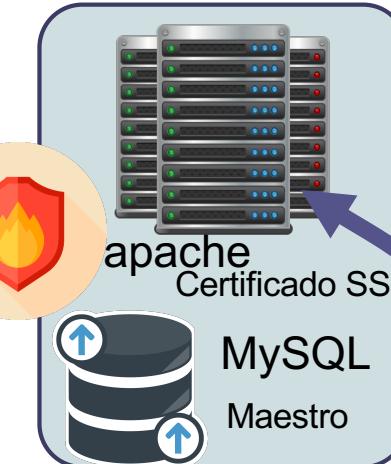


Balanceador

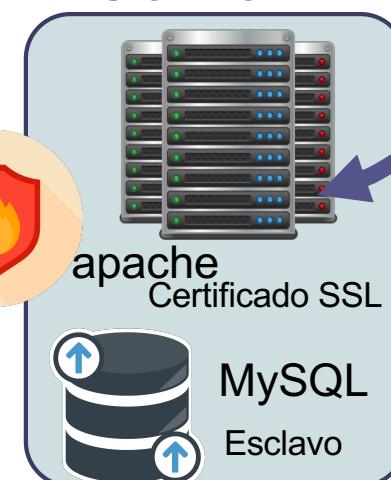


nginx  
Certificado SSL

Server 1



Server2



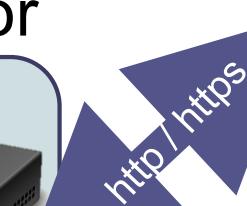
NFS



http  
https

http / https

http / https



# Elementos de la granja web

## Servidores



Los servidores web no necesitan una capacidad de procesamiento especialmente alta, ya que los servicios HTTP no consumen demasiada CPU.

La cantidad de memoria RAM es más importante.

Algunas páginas en un sitio serán servidas mucho más que el resto (regla 80-20: el 80% del tráfico corresponde al 20% de las páginas almacenadas).

Conviene *cachear* las imágenes más comunes. Así las páginas que las muestren se servirán más rápido, haciendo que el tiempo medio de servicio sea más rápido.

# Elementos de la granja web

## Almacenamiento



Se pueden evitar latencias usando **dispositivos de estado sólido** de alta velocidad (para hacer *caching*).

El almacenamiento puede consistir en un NAS (Almacenamiento conectado a la red), una SAN (Área de almacenamiento en red), o almacenamiento interno en el servidor.

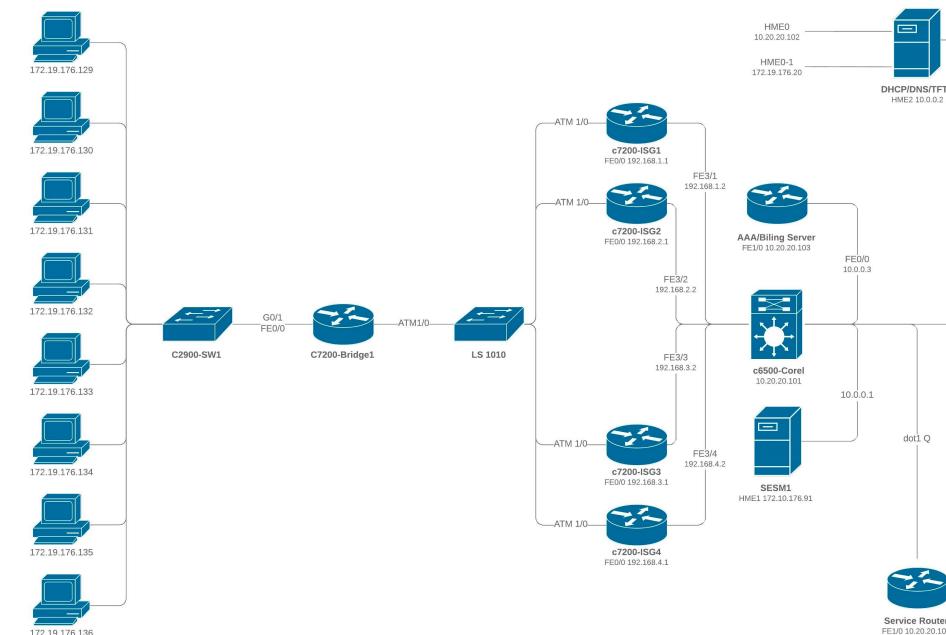
El protocolo más comúnmente utilizado es NFS, aunque se tiene a usar LUSTRE en su lugar.

# Elementos de la granja web

## Conexiones de red

Los nodos del clúster pueden conectarse mediante una simple red Ethernet con tarjetas de red comunes, o utilizarse tecnologías especiales de alta velocidad:

- Fast Ethernet,
- Gigabit Ethernet
- Myrinet,
- InfiniBand,
- SCI



# Índice



1. Introducción
2. Elementos de la granja web
3. Necesidades del servidor web
4. Hardware para servidores
5. Software para servidores
6. Conclusiones

# Necesidades del servidor web

Determinar las necesidades de un servidor es muy complejo.

El diseño debe permitir el **escalado del sistema**

y

debemos realizar continuamente **monitorización** para determinar cuándo y dónde escalar el sistema.

# Necesidades del servidor web

Para planear la capacidad del servidor (necesidades en cuanto a hardware y software) no existe una fórmula.

Hay que hacer un **primer estudio**, instalar y **configurar**, y una vez esté en funcionamiento, **monitorizar** constantemente el sistema para determinar si funciona correctamente.

No se pueden hacer suposiciones del tipo  
*“como la máquina va a servir páginas estáticas, necesitaremos una potencia de CPU de X”*, o bien  
*“como el tamaño medio de las páginas que vamos a servir será de 25KB, el ancho de banda que vamos a necesitar será Y”*.

# Necesidades del servidor web

Hay muchas variables a tener en cuenta => tarea compleja.

Hay cuatro tareas que pueden ayudar a determinar los requisitos de hardware del sistema:

1. Estimar el hardware en función de las necesidades de la empresa.
2. Utilizar un software de monitorización para buscar cuellos de botella en la configuración inicial.
3. Monitorizar el sistema durante todo el tiempo que esté en uso.
4. Utilizar técnicas de modelado para predecir capacidad

# Estimación de necesidad de CPU

**La máquina servidora llevará a cabo las siguientes tareas:**

- Ejecutar un sistema operativo
- Ejecutar servicios que no tienen que ver con el servidor web.
- Ejecutar los servicios del servidor web.
- Ejecutar programas externos necesarios para generar el contenido dinámico.

El sistema operativo debe consumir cerca del 10% de la capacidad de procesamiento del sistema.

Los procesos del servidor web necesitan poca capacidad de procesamiento.

El cuello de botella se presenta cuando el servidor recibe un alto número de conexiones.

# Estimación de necesidad de MEMORIA

El uso de memoria sigue patrones **similares** a los comentados **respecto a la CPU**. Así, el software en ejecución que demandará memoria es:

- El sistema operativo.
- Servicios adicionales al servidor web.
- Los servicios del servidor web.
- Programas externos necesarios para generar el contenido dinámico.

Un servidor con una gran capacidad de memoria puede albergar una caché de suficiente tamaño para almacenar y servir los contenidos estáticos más comunes.

# Estimación de necesidad de ANCHO DE BANDA

El ancho de banda debe soportar el **máximo de peticiones HTTP en momentos de picos de carga**. Para estimar el ancho de banda, primero hay que determinar:

- El número medio de clientes que se conectarán al servidor por segundo.
- El número medio de bytes que el cliente enviará al servidor en cada petición.
- El número medio de bytes que el servidor enviará al cliente en cada petición.

Multiplicar el número de clientes/seg por el total de bytes transferidos en cada petición.

La conexión debe tener más del doble del ancho de banda del que hayamos estimado en este punto.

# Buscar cuellos de botella

Una vez que hemos montado el sistema, importante monitorizar cómo está funcionando, si cubre nuestras necesidades, y detectar cuellos de botella.

Realizar **tests de carga** con benchmarks (apache benchmark, httpperf, u openwebload) o poner el sistema en producción de forma controlada (accesible a **betatesters**) para monitorizar y analizar cómo se comporta:

<http://www.thegeekstuff.com/2011/07/iostat-vmstat-mpstat-examples/>

<http://www.cyberciti.biz/tips/top-linux-monitoring-tools.html>

<http://www.tecmint.com/command-line-tools-to-monitor-linux-performance/>

# Monitorizar el resto del tiempo

Una vez en producción, debemos seguir monitorizando.

El análisis de los logs nos permitirá determinar qué contenido es el más demandado (hacerlo eficiente).

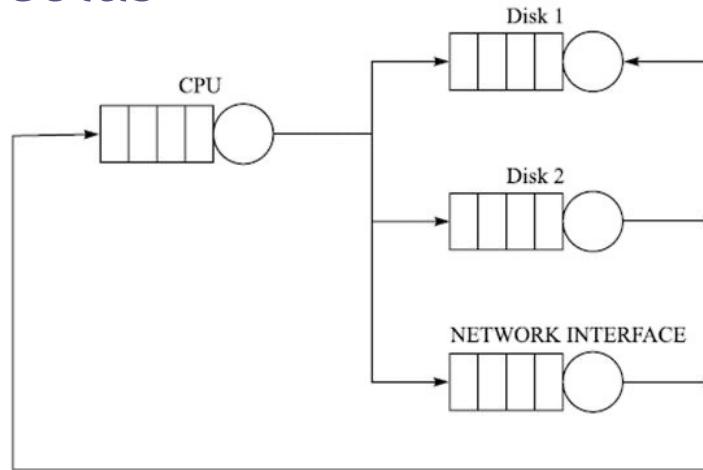
También nos permitirá hacer análisis del tipo de usuario y de los patrones de navegación (para marketing).

Podremos identificar fallos (errores tipo 404 ó 500).

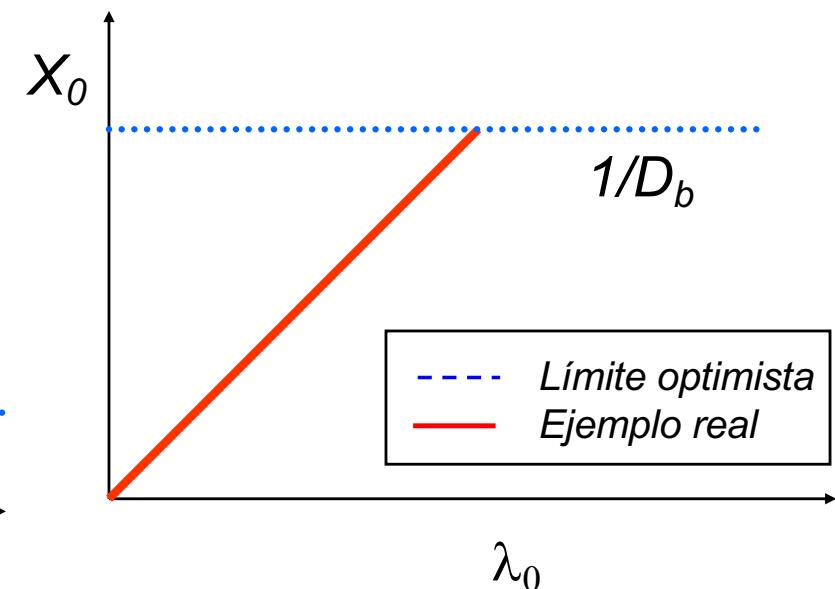
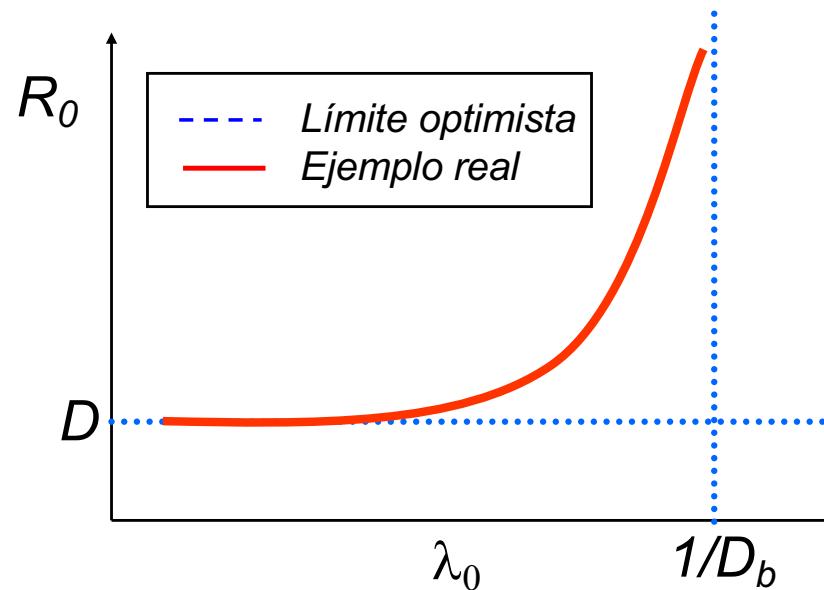
A partir de los logs podremos descubrir posibles ataques.

# Crear un modelo analítico

## Red de colas



Carga del sistema -----  
 $D = S \times V$



# Ejemplo de HW para servidor medio

Dependiendo del peso de las aplicaciones, para sitios con menos de un millón de páginas servidas al mes, una máquina tipo i7 con 16GB de RAM y un SO bien configurado puede ser suficiente.

Para volúmenes de tráfico mayores, esa configuración puede ser insuficiente.

También existen **herramientas propietarias** que pueden ayudar en la configuración de ciertos entornos, como en el caso de HP Sizer para Microsoft SharePoint.

O soluciones cloud: Amazon Web Services, Azure, etc.

# Herramientas automáticas para determinar las necesidades del servidor

## HP Sizer para Microsoft SharePoint



HP Sizer for Microsoft SharePoint 2010



**HP Sizer for Microsoft SharePoint 2010**

Solution Alternatives

Profile :	Intel	Price :	\$77,316
Web Front End and Query Search Server	1 x ProLiant BL460c Gen8 Server 2P Intel 6 - Core 2.3GHz / 15MB Cache 16,384 MB RAM (Per Server) Disk (DAS) - 2 Disks Spare ( 0 ) ( Per Server ) System Drive (S) - 2 Disks (72GB 6G SAS 15K 2.5in SC ENT HDD) RAID10 Spares ( 0 )	Recommended Configuration	
Index Search Server	1 x ProLiant BL460c Gen8 Server 2P Intel 6 - Core 2.3GHz / 15MB Cache 16,384 MB RAM (Per Server) Disk (DAS) - 2 Disks Spare ( 0 ) ( Per Server ) System Drive (S) - 2 Disks (72GB 6G SAS 15K 2.5in SC ENT HDD) RAID10 Spares ( 0 )		
SQL Server	1 x ProLiant BL460c Gen8 Server 2P Intel 6 - Core 2.3GHz / 15MB Cache 24,576 MB RAM (Per Server)		

# Índice



1. Introducción
2. Elementos de la granja web
3. Necesidades del servidor web
4. Hardware para servidores
5. Software para servidores
6. Conclusiones

# Hardware para servidores

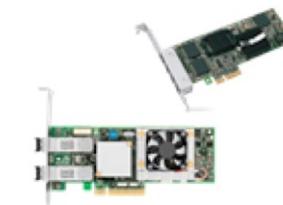
Revisar las soluciones de hardware propietario de diferentes vendedores y el que usan en otros grandes sistemas:

- Dell
- HP
- IBM
- Open Compute (Facebook)
- Google

# Hardware para servidores: Dell

Dell ofrece diversos productos para construir desde servidores de gama baja hasta grandes servidores de alta disponibilidad.

<http://www.dell.com/es/grandes-corporaciones/p/networking-products>



Modelos disponibles

## Switches PowerConnect

PERFECTO PARA:

La gama PowerConnect ofrece un conjunto de soluciones de switch flexibles, fáciles de gestionar e integrales que admiten hasta 10 Gigabit Ethernet

## PowerConnect Wireless Series

PERFECTO PARA:

Wireless networking access for large enterprise to small-office and branch deployments with extensive network mobility, security and remote networking requirements.

## Interconexiones Fibre Channel

PERFECTO PARA:

Complete su red de área de almacenamiento con switches y HBA que han sido probados y validados para funcionar con productos Dell/EMC.

## Tarjetas de interfaz de red

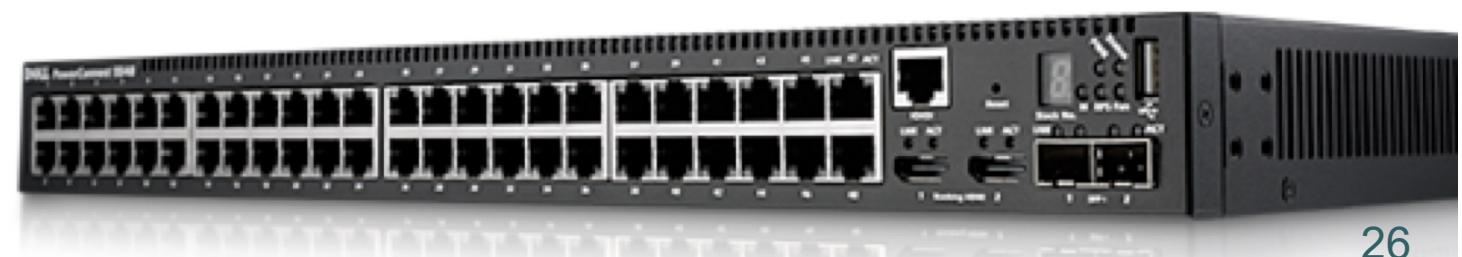
PERFECTO PARA:

Conexión a tiempo completo, dedicada y fiable a la red de área local. Con opciones inalámbricas o con cables para mejorar la capacidad de ampliación y aumentar los tiempos de actividad.

# Hardware para servidores: Dell

## Switch Gigabit Ethernet PowerConnect 5548

- Conmutación Gigabit flexible con funciones robustas de **seguridad**, apilamiento y gestión.
- **48 puertos Gigabit Ethernet** con velocidad de cable; se puede ampliar a medida que crece la red.
- Tiene 2 puertos de enlace ascendente 10GbE.
- Arquitectura **Energy Efficient Ethernet (EEE)** para reducir el consumo de energía de los puertos en modo suspendido o dejar de alimentar los puertos inactivos.
- Es posible gestionar hasta 8 switches con velocidad de cable como una sola unidad.



# Hardware para servidores: Dell

## Almacenamiento: MD3600i

- Ofrece alta disponibilidad y **alto rendimiento**.
- **Compatibilidad con niveles de RAID 0, 1, 10, 5 y 6**
- Hasta 120 discos físicos por grupo en RAID 0, 1 y 10
- Hasta 30 discos físicos por grupo en RAID 5 y 6
- Hasta 256 discos virtuales
- **Unidades intercambiables en caliente:**
  - MD3600i: hasta doce (12) unidades SAS, SAS nearline y SSD de 3,5"
  - MD3620i: hasta veinticuatro (24) unidades SAS, SAS nearline y SSD de 2,5"
  - MD3660i: hasta sesenta (60) unidades SAS, SAS nearline y SSD de 2,5" o 3,5"



# Hardware para servidores: Dell

## Servidor para rack Dell PowerEdge R810

- Procesadores Intel **Xeon** de ocho núcleos series 7500 y 6500.
- Sistema operativo: Windows Server, SUSE Linux, Red Hat Linux
- Opciones de **virtualización**: Citrix XenServer, Vmware
- Conjunto de chips Intel® 7500
- **Memoria**: Hasta 1 TB2 (32 ranuras DIMM): 1 GB/2 GB/4 GB/8 GB/16 GB/32 GB de DDR3 a 1066 MHz.
- **Almacenamiento**: Hasta 6 TB2 en RAID
  - Disco duro conectable en caliente. SSD SATA de 2,5", SAS (10 000 rpm, 15 000 rpm), SAS nearline (7200 rpm) y SATA (7200 rpm)
- Almacenamiento de estado sólido: Fusion-io 160IDSS, Fusion-io 640IDSS
- Tarjetas de red: Dos NIC Broadcom® 5709c Gigabit de dos puertos
- Dos fuentes de alimentación redundante conectables en caliente de 1.100 W
- Tarjeta de vídeo: Matrox® G200eW con 16 MB
- Chasis para rack

# Hardware para servidores: Dell

Servidor para rack Dell PowerEdge R810



# Hardware para servidores: HP

HP ofrece dispositivos de todo tipo para construir una red completa y por supuesto, un sistema web de altas prestaciones.

En la web de HP se encuentra información de todos los productos:

<http://welcome.hp.com/country/es/es/prodserv/servers.html>

Mostrar características generales de algunos productos (información extraída de la web de HP).

No pretendemos hacer una recopilación exhaustiva de los mismos.



# Hardware para servidores: HP

## Servidor HP ProLiant DL385 G7 6282SE

- Número de procesadores: hasta 16
- Memoria: RDIMM de 64 GB (8 x 8 GB). 24 ranuras DIMM 2R x4 PC3-10600R-9
- Ranuras de expansión: (6) PCIe.
- Controlador de red: Adaptador Ethernet NC382i multifunción de 1 Gb y 2 puertos por controlador.
- Tipo de fuente de alimentación: (2) kits de fuente de alimentación Platinum de 750 W de ranura común y conexión en caliente.
- Controlador de almacenamiento: (1) Smart Array P410i/1 GB FBWC
- Tipo de unidad óptica: SATA DVD-RW
- Formato: 2U



# Hardware para servidores: HP

**Almacenamiento: HP StorageWorks Network Storage System X1600 G2 - servidor NAS - 24 TB**

- Conectividad para Host: Gigabit Ethernet
- Formato de Montaje: en bastidor 2U
- Capacidad total de almacenamiento: 24 TB
- Dispositivos instalados / N° módulos: 12 (instalados) / 12 (máx.)
- Dimensiones: 44.8 cm x 69.9 cm x 8.8 cm . Peso: 19.2 kg
- Procesador: 1 x Intel Xeon E5520 2.26 GHz ( Quad-Core )
- Controlador de almacenamiento: RAID PCI Express 2.0 x8 - Serial ATA-300 / SAS 2.0. RAID 0, 1, 5, 6, 10, 50, 60
- Disco duro: 12 x 2 TB intercambio rápido (hot swap) Serial ATA-300
- Adaptador de red integrado Ethernet, Fast Ethernet, Gigabit Ethernet
- Sistema operativo de almacenamiento: Windows Storage Server 2008
- Alimentación redundante



# Hardware para servidores: IBM

En la web de la compañía se puede encontrar información detallada de todos sus productos, y en concreto de los que tienen que ver con servidores web de altas prestaciones:

<http://www-03.ibm.com/systems/es/bladecenter/?lnk=mprSS-blce-eses>

The image shows a screenshot of the IBM Systems website. At the top left, it says "IBM Systems > IBM Smarter Computing". The main title is "IBM System x y BladeCenter". Below the title, there are two subtitles: "Transforme TI en conocimiento y eficiencia" and "Las nuevas ofertas de x86 mejoran los aspectos económicos de TI". A "Más información" link is available. On the right side, there is a large graphic composed of colored geometric shapes (blue, yellow, green) arranged in a grid-like pattern. In the bottom center, there are three small icons: a blue and white square, a blue and white starburst, and a blue and white sun-like shape.

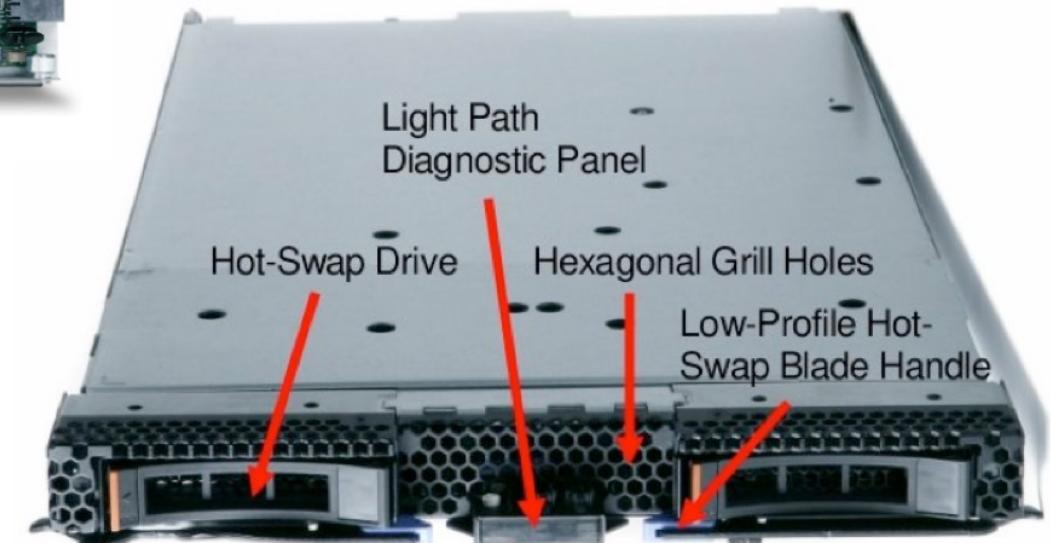
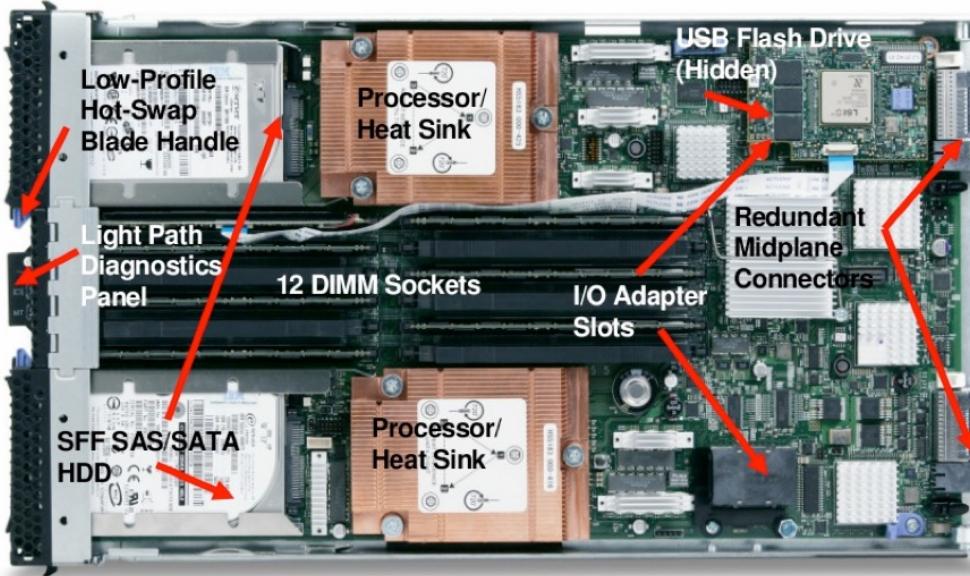
# Hardware para servidores: IBM

## Servidor: IBM BladeCenter HS22 Xeon 6C X5650 SAS

- 2 procesadores Intel Xeon 5600, de hasta 3,46 GHz
- Hasta 192 GB de memoria con 12 DIMM DDR-3 VLP.
- Una ranura CIOv (tarjeta secundaria PCIe de serie) y una ranura CFFh (tarjeta secundaria PCIe de alta velocidad).
- Adaptador Virtual Fabric integrado en algunos modelos.
- Tarjeta de interfaz de red (NIC) en la placa Broadcom 5709S con dos puertos Gigabit Ethernet (GbE) con TCP/IP Offload Engine (TOE).
- Trusted Platform Module (TPM) 1.2
- RAID -0, -1 y -1E (RAID-5 opcional con caché respaldada por batería).
- Compatibilidad con SSD o HDD (unidad de disco duro) SAS hot-swap.
- Compatibilidad con todos los chasis BladeCenter para oficinas y empresas.
- 3000 dólares cada máquina.

# Hardware para servidores: IBM

Servidor: IBM BladeCenter HS22 Xeon 6C X5650 SAS



# Hardware para servidores: IBM

## Almacenamiento: IBM System Storage SAN24B-4 Express

- Compatible con 8 Gbps Fibre Channel.
- De 8 a 24 puertos.
- Se puede usar con servidores Microsoft Windows, UNIX, Linux, IBM AIX y OS/400.



# Hardware para servidores: Open Compute

Facebook lanzó en 2011 el Open Compute Project con la idea de que las empresas pudieran desarrollar sus propias infraestructuras (servidores, almacenamiento, etc).

Evitar la dependencia de los grandes fabricantes.

Más información:

- <http://www.opencompute.org/>
- <http://alt1040.com/2013/02/open-compute-project-facebook>



# Hardware para servidores: Open Compute

Facebook recurre a fabricantes en Asia para que les fabriquen los servidores en base a sus especificaciones muy concretas.

Se busca eliminar lo superfluo para mejorar el rendimiento y optimizar el consumo de potencia.



# Hardware para servidores: Open Compute

La intención es ahorrar en cualquier aspecto posible: se han eliminado logotipos de plástico, tornillos y cualquier trozo de metal del chasis que no sea imprescindible:



# Hardware para servidores: Open Compute

Asimismo, se ha diseñado un rack para colocar hasta 30 servidores por columna:



# Hardware para servidores: Open Compute

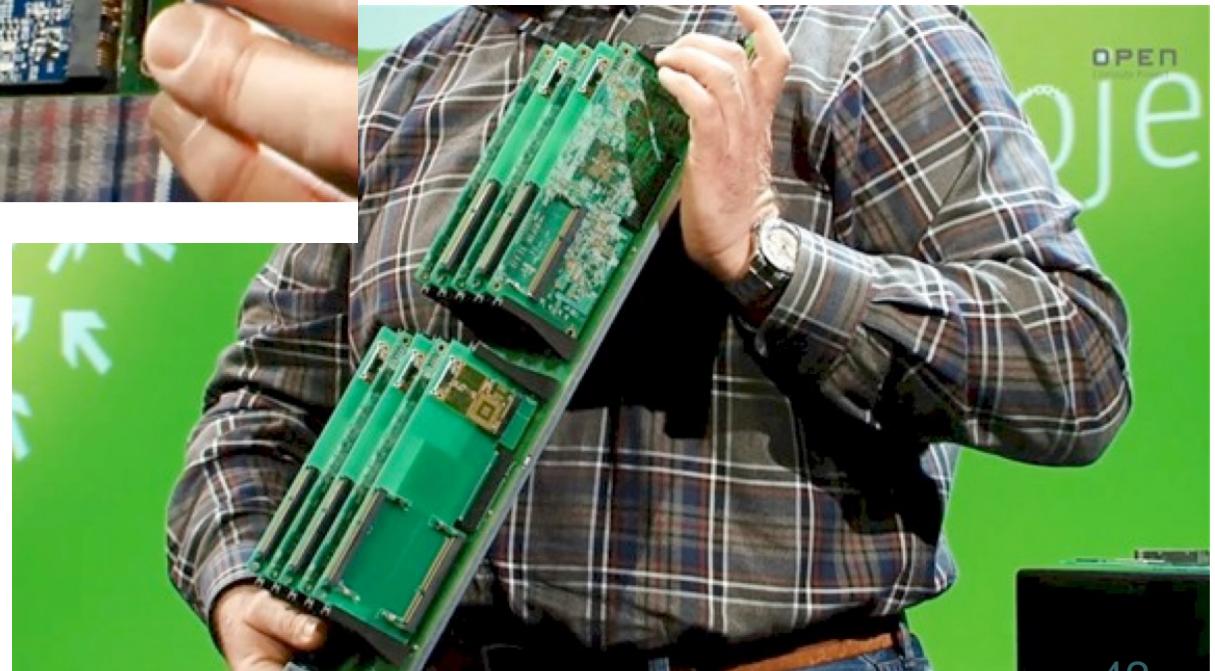
Hay un diseño con procesadores Intel (2 procesadores quad-core Xeon 5500 o six-core Xeon 5600) y otro con AMD (procesador Opteron 6100 con 8 o 12 cores ).

Las placas tienen 9 ranuras para memoria, con un máximo de 288GB. Tiene 6 puertos SATA-II y 3 puertos Gigabit



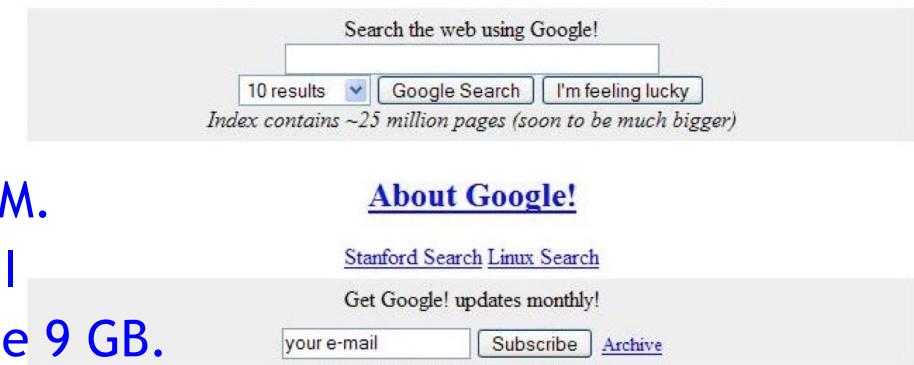
# Hardware para servidores: Open Compute

En un futuro cercano se espera un diseño de placa basado en conectar varios microservidores con conexión PCI-Express:



# Hardware para servidores: Google

El hardware original que usaron para el primer prototipo de Google desarrollado en la Universidad de Stanford se basó en el siguiente hardware:

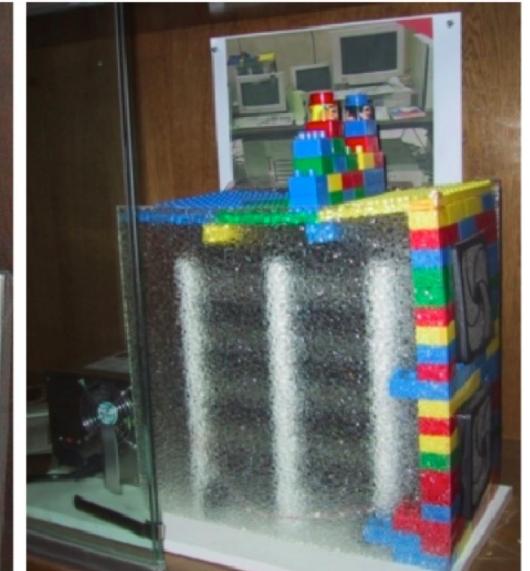
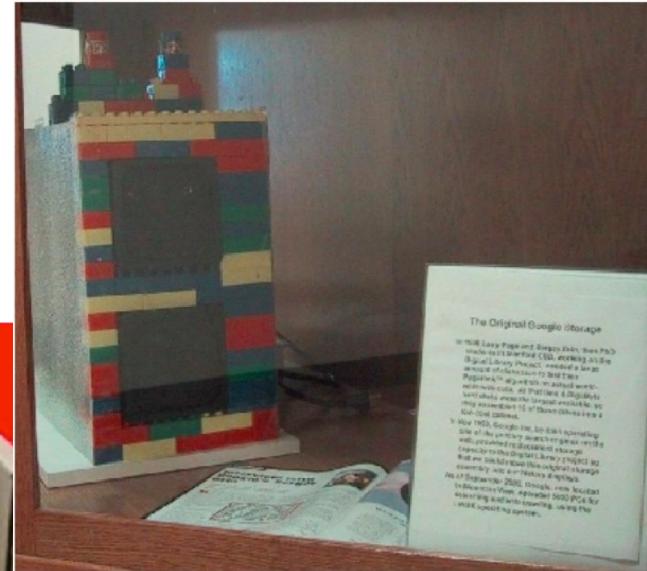


- Una máquina Sun Ultra II con dos procesadores a 200 MHz y 256 MB de RAM.
- Dos servidores biprocesadores Pentium II a 300 MHz, 512 MB de RAM y 10 discos de 9 GB.
- Una máquina IBM F50 IBM RS/6000 (4 procesadores, 512 MB de RAM y 8 discos de 9 GB).
- Dos máquinas adicionales con tres discos de 9 GB y seis discos de 4 GB.
- Una extensión de almacenamiento IBM con ocho discos de 9 GB.
- Una extensión de almacenamiento diseñada por Google con diez discos SCSI de 9 GB cada uno.

# Hardware para servidores: Google

[http://en.wikipedia.org/wiki/Google\\_platform](http://en.wikipedia.org/wiki/Google_platform)

<http://perspectives.mvdirona.com/2008/06/11/JeffDeanOnGoogleInfrastructure.aspx>



# Hardware para servidores: Google

- El hardware actual de los sistemas de Google se basa en PCs con arquitectura x86.
- Versión de Linux especialmente adaptada.
- Usar CPUs que den el máximo rendimiento por dólar.

Hacia 2009-2010 los servidores tenían 2 CPUs dual core, gran cantidad de memoria RAM y dos discos SATA; caja ATX no estándar; batería de 12 voltios para mejorar la eficiencia



# Hardware para servidores: Google

Las necesidades eléctricas entre los 500 y 700 megawatios.

Potencia computacional sobre los 100 petaflops.

Configuración de red: conjunto de concentradores y routers especialmente diseñados con una capacidad de 128 puertos de 10 Gigabit Ethernet.

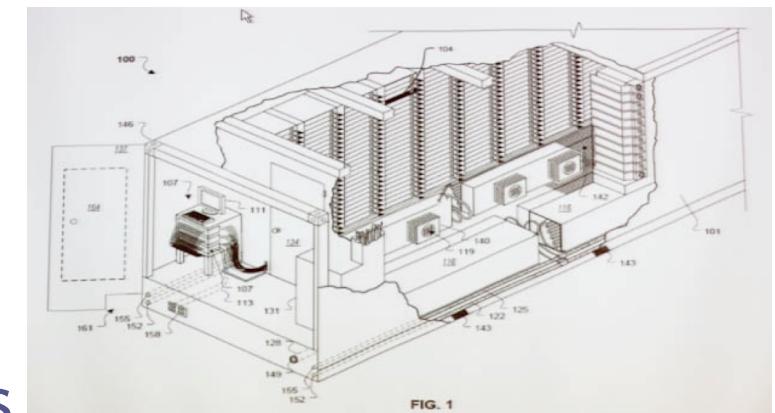
**Racks con diseño propio:** Contienen entre 40 y 80 servidores más un concentrador. Los servidores están conectados por un **enlace de 1 Gigabit Ethernet al concentrador del rack**. Estos concentradores (de cada rack) se conectan entre sí, y **hacia el exterior mediante 10 enlaces Gigabit**.

# Hardware para servidores: Google

Desde 2005, Google utiliza un modelo de centro de datos modularizado mediante contenedores (patente obtenida en 2003).

En cada contenedor hay alrededor de 1160 servidores.

En cada centro de datos hay decenas de contenedores .



# Hardware para servidores: Google

Hay que destacar que los centros de datos modulares no son exclusivos de Google, sino que otros fabricantes, como Sun Microsystems y Rackable Systems, también los desarrollan:



# Índice



1. Introducción
2. Elementos de la granja web
3. Necesidades del servidor web
4. Hardware para servidores
- [ 5. Software para servidores ]**
6. Conclusiones

# Software para servidores

En esta última sección vamos a revisar el software necesario para configurar un servidor web de altas prestaciones.

Concretamente necesitaremos:

- Sistema operativo
- Servidor web
- Cortafuegos
- Balanceadores de carga
- Software para monitorización

# Software para servidores: SO

Actualmente existe gran variedad de sistemas operativos que se utilizan para montar servidores web de altas prestaciones:

- GNU/Linux
- Unix: Solaris, HP-UX, AIX
- Windows: NT, 2000 server, 2008 server, 2010 server
- Mac OS X Server (xgrid)
- FreeBSD

El sistema operativo incluye ciertas herramientas para hacer balanceo de carga, cortafuegos o monitorización.

# Software para servidores: Cortafuegos

Implementados en hardware o software.

Cada sistema operativo tiene su propio cortafuegos incluido en la instalación básica:

- Firestarter
- ZoneAlarm
- Uncomplicated Firewall
- Gufw
- PF (OpenBSD)
- ipfw
- iptables
- Ipfilter
- Ufw
- Etc.

# Software para servidores: servidor web

Se pueden instalar en casi cualquier sistema operativo.

Los más conocidos:

- nginx
- apache
- Internet Information Services (IIS)
- cherokee
- Tomcat

Otros servidores, más simples pero más rápidos, son:

- lighttpd
- thttpd

# Software para servidores: servidor web

Además, existen soluciones para implementar servicios web sumamente eficientes:

- node.js (JavaScript)
- Tornado (Python),
- Twisted (Python),
- EventMachine (Ruby),
- Scale Stack (C++),
- Apache MINA (Java)
- Jetty (Java)

Se trata de una solución muy diferente a lo que conocemos que se puede hacer con Apache.

# Software para servidores: Balanceadores

Cada sistema operativo tiene soluciones (gratuitas o de pago):

- HaProxy: <http://haproxy.1wt.eu/>
- Pound: <http://www.apsis.ch/pound/>
- Varnish: <http://varnish-cache.org>
- NginX: <http://nginx.org/>
- Lighty: <http://www.lighttpd.net/>
- Apache: <http://httpd.apache.org/>
- NLB: <https://msdn.microsoft.com/en-us/library/bb742455.aspx>

# Software: Benchmarking

Basadas en interfaz de línea de comandos y de interfaz gráfica. Entre las más utilizadas destacan:

- Apache Benchmark
- httpperf
- openwebload
- the grinder
- OpenSTA
- JMeter
- Siege

# Software para servidores: Monitorización

En Linux, la herramienta más versátil es vmstat

<http://linux.die.net/man/8/vmstat>

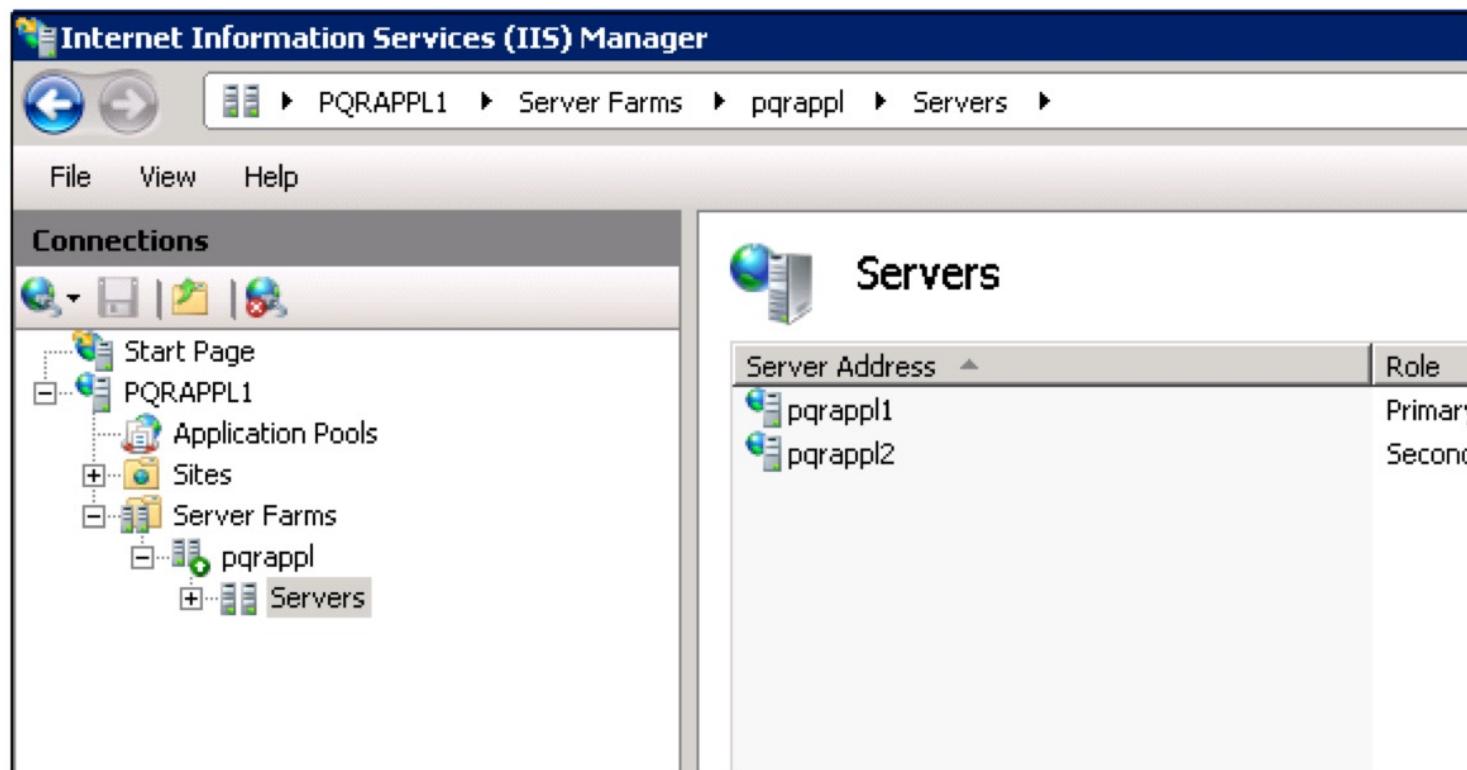
<http://storm.malditainternet.com/wp/2011/05/usando-y-entendiendo-vmstat/>

<http://www.cacaoadmin.com/2012/04/vmstat-linux-ejemplos-herramientas-de.html>

```
sotillo19@m1:~$ vmstat 3 5
procs -----memory----- swap -----io----- system -----cpu-----
 r b swpd free buff cache si so bi bo in cs us sy id wa st
 1 0      0 112308 26832 535576  0    0 15453 7163 1213 2407 50 34 13 4 0
 0 0      0 77524 26840 570572  0    0 0 13767 3185 1322 21 37 42 0 0
 2 0      0 79956 26844 572124  0    0 1 16241 518 547 90 8 1 2 0
 0 1      268 141220 27416 500196  0    7 881 33 459 337 96 3 0 1 0
 2 0      524 77572 27904 524328  16   91 19707 2131 996 2360 56 17 0 28 0
```

# Software para servidores: Monitorización

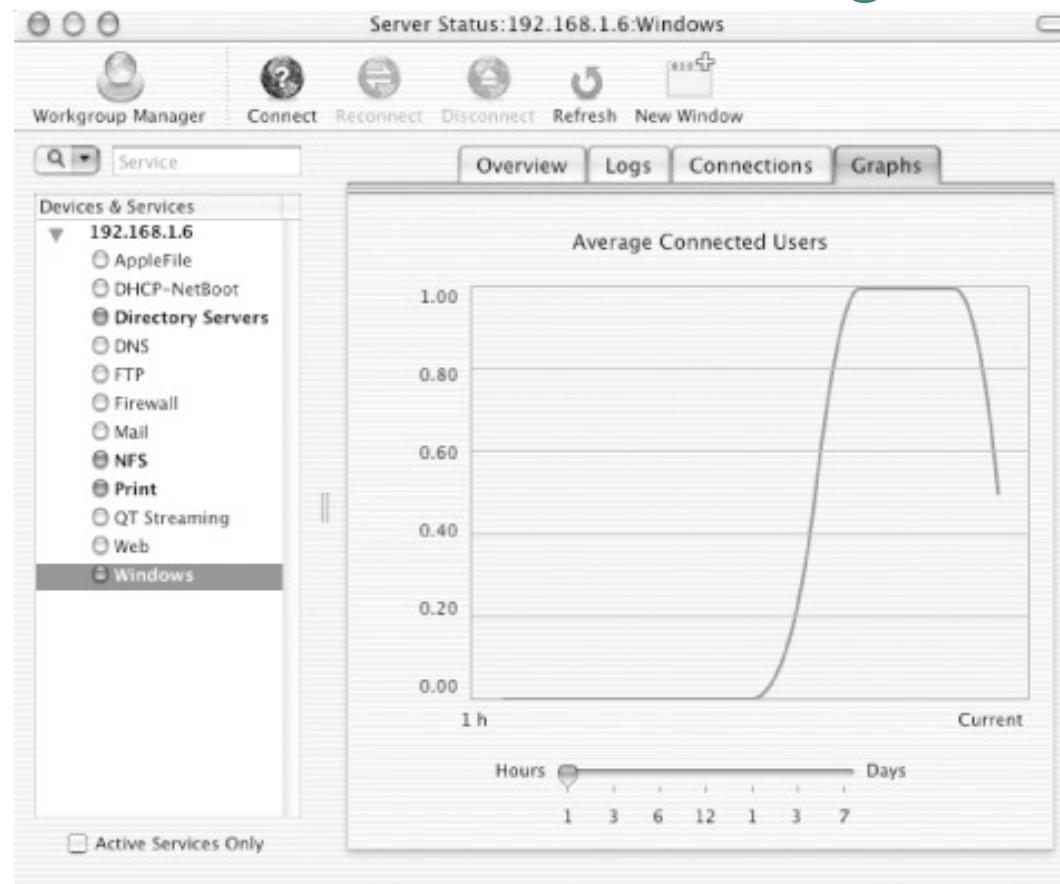
En el caso de sistemas basados en tecnologías Microsoft, disponemos de **Web Farm Framework (WFF)** como una solución para monitorizar y controlar una granja de servidores IIS.



# Software para servidores: Monitorización

Para OS X Server, tenemos:

- **Lithium5**, un software para monitorizar la red, los servidores y el almacenamiento.
- **Apple Mac OS X Server Monitoring Library**



# Software para servidores: Monitorización

Para **grandes sistemas**, existe software comercial, que normalmente se incluye en el sistema operativo.

- NetApp, usado en entornos corporativos de grandes empresas  
<http://www.netapp.com/es/products/management-software>
- Munin es una herramienta de monitorización del rendimiento de un sistema.  
<http://munin-monitoring.org>
- Nagios es una herramienta muy potente que ayuda a detectar y reparar problemas en la infraestructura del sistema.  
<http://www.nagios.org>
- GANGLIA: es un monitor sistemas de cómputo distribuidos (normalmente para altas prestaciones) que permite una gran escalabilidad.  
<http://ganglia.sourceforge.net>

# Índice



1. Introducción
2. Elementos de la granja web
3. Necesidades del servidor web
4. Hardware para servidores
5. Software para servidores
6. Conclusiones

# Conclusiones

**Determinar las necesidades hardware y software de un sistema antes de ponerlo en producción es una tarea muy compleja.**

Existe una gran variedad de dispositivos hardware de diferentes fabricantes para construir un sistema de cualquier tamaño.

**Importancia del software que elijamos para el sistema.**

**Necesidad de monitorizar continuamente el sistema para detectar cuellos de botella y fallos.**