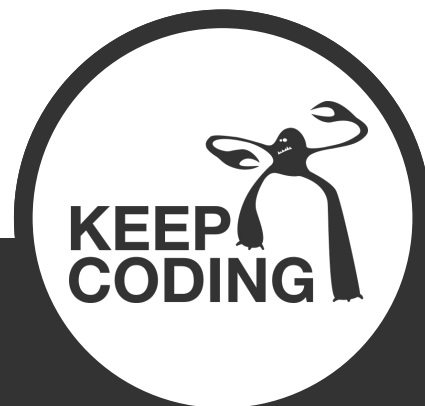


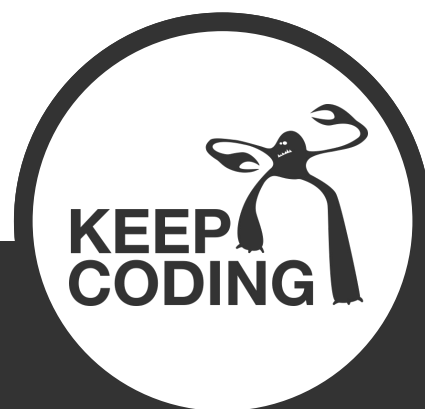
# ■ DevOps

Introducción y un poco de historia

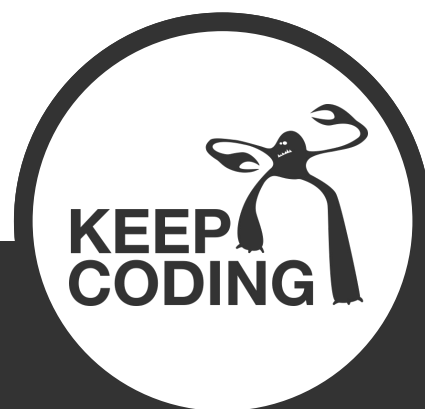
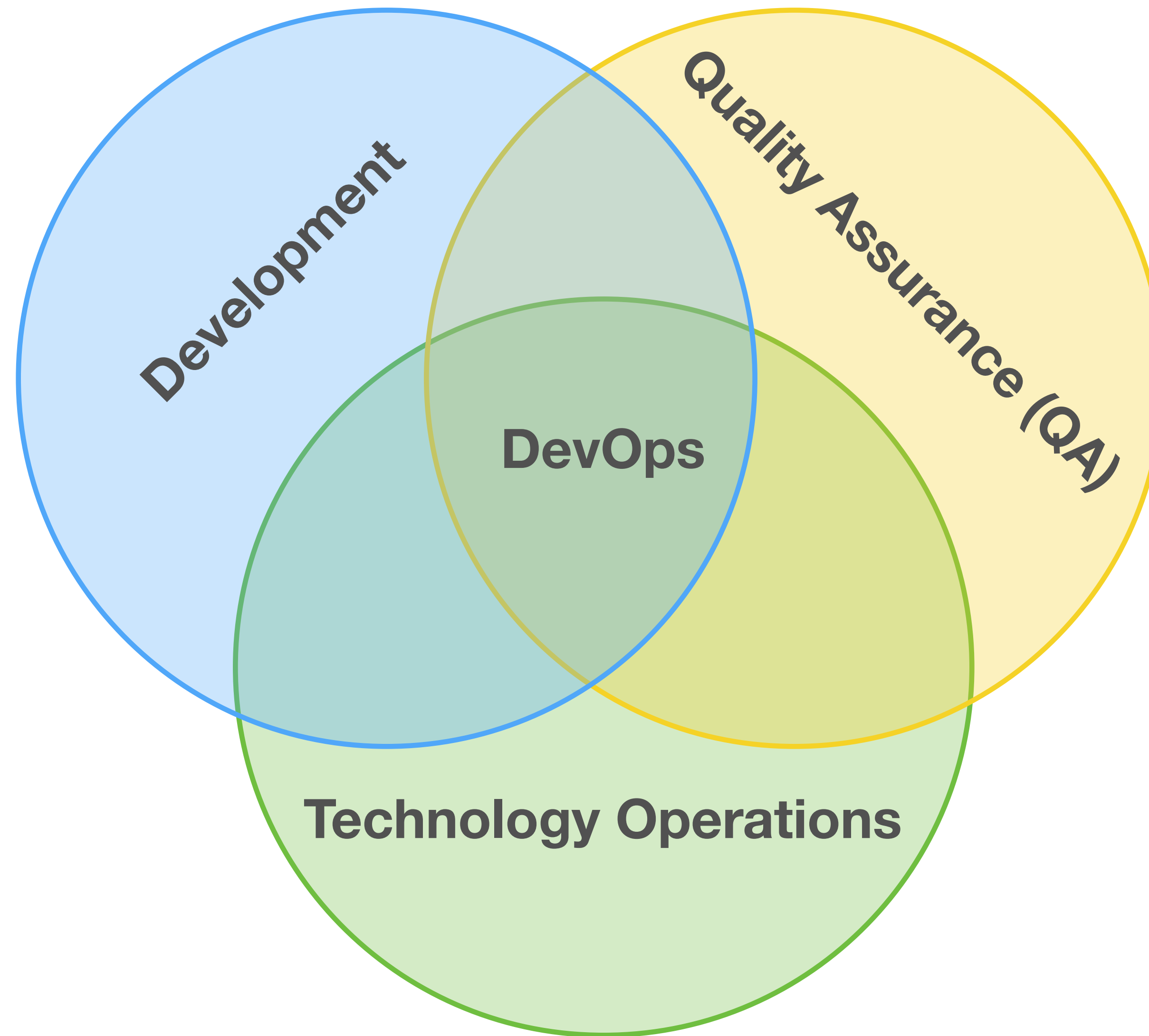


# ■ ¿Qué es DevOps?

Es una **metodología** de desarrollo de software que se centra en la **comunicación, colaboración, integración y automatización** entre los **desarrolladores de software y profesionales de IT.**

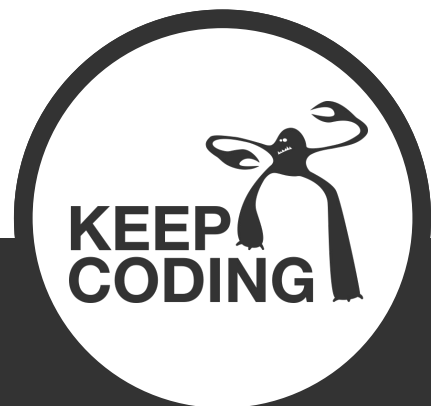


# ■ ¿Qué es DevOps?

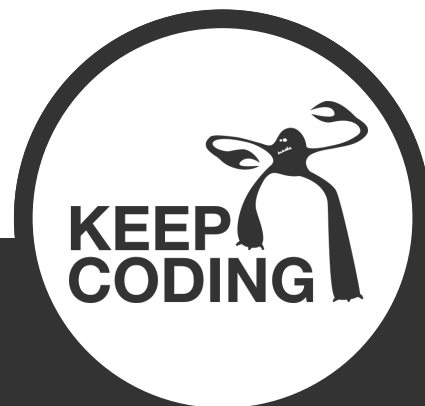


# ■ ¿Por qué DevOps?

- Aparición de las **metodologías ágiles y lean**
- El desarrollo ahora es más rápido, pero no se puede poner tan rápido en producción
- Para poder realizar despliegues continuos (**continuous delivery** o **continuous deployment**) es necesario acercar a desarrolladores (devs) y personal de operaciones IT (ops)
- DevOps nace como una necesidad de acercar los mundos del desarrollo y los sistemas para mejorar la competitividad



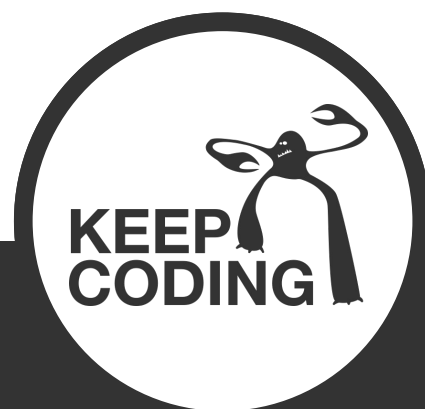
■ The easy way: seguir siendo Dev



# ■ The easy way

Cuando despliegas tu app en la nube adquieres una “deuda técnica” porque delegas a un tercero se delegas en un servicio de tercero el montaje del servidor delegas una deuda técnica con ese tercero. Si el día de mañana cambian las condiciones (de cualquier tipo) de ese tercero, tu te verás afectado y pagarás esa deuda. O pagas tu el servidor o pagas para invertir horas en la nueva versión que va a funcionar.

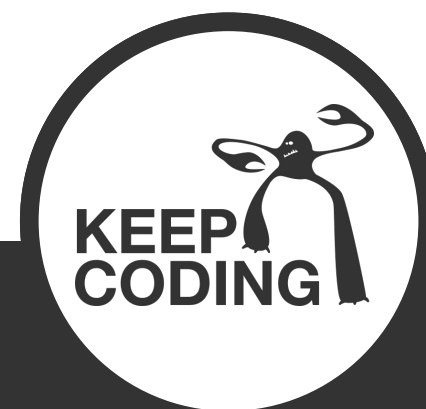
- Para trabajar bien con Azure y node usando Web Apps, hay que utilizar Azure PowerShell
- Sobre todo cuando hay que hacer despliegues avanzados
- No tenemos control sobre la versión de node instalada (no suele ser la última)
- Mi experiencia personal con estos sistemas:





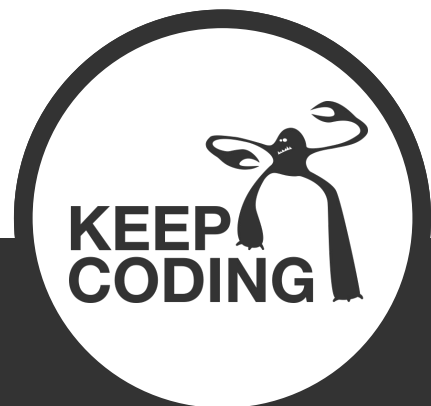
# ■ The rockstar way

De Developer a DevOps

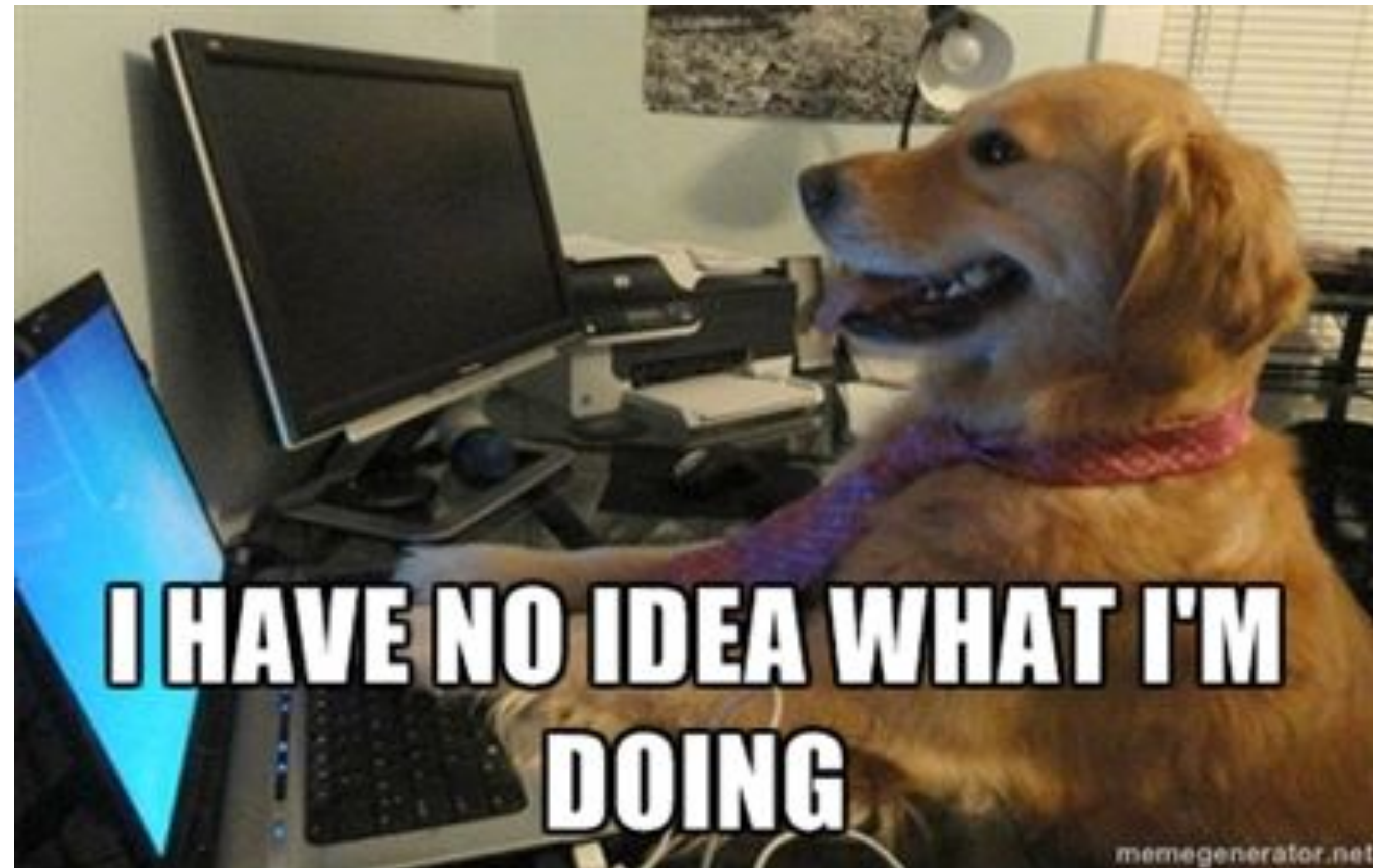


# ■ De Developer a DevOps

- A los desarrolladores, generalmente, no nos gusta la administración de sistemas.
- Acercarnos al mundo DevOps, supone **salir de nuestra zona de confort**
- Debemos pasar de crear nuestro software a usar, aprender y configurar software creado por otros

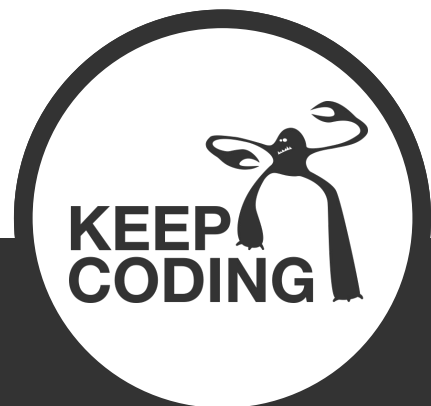




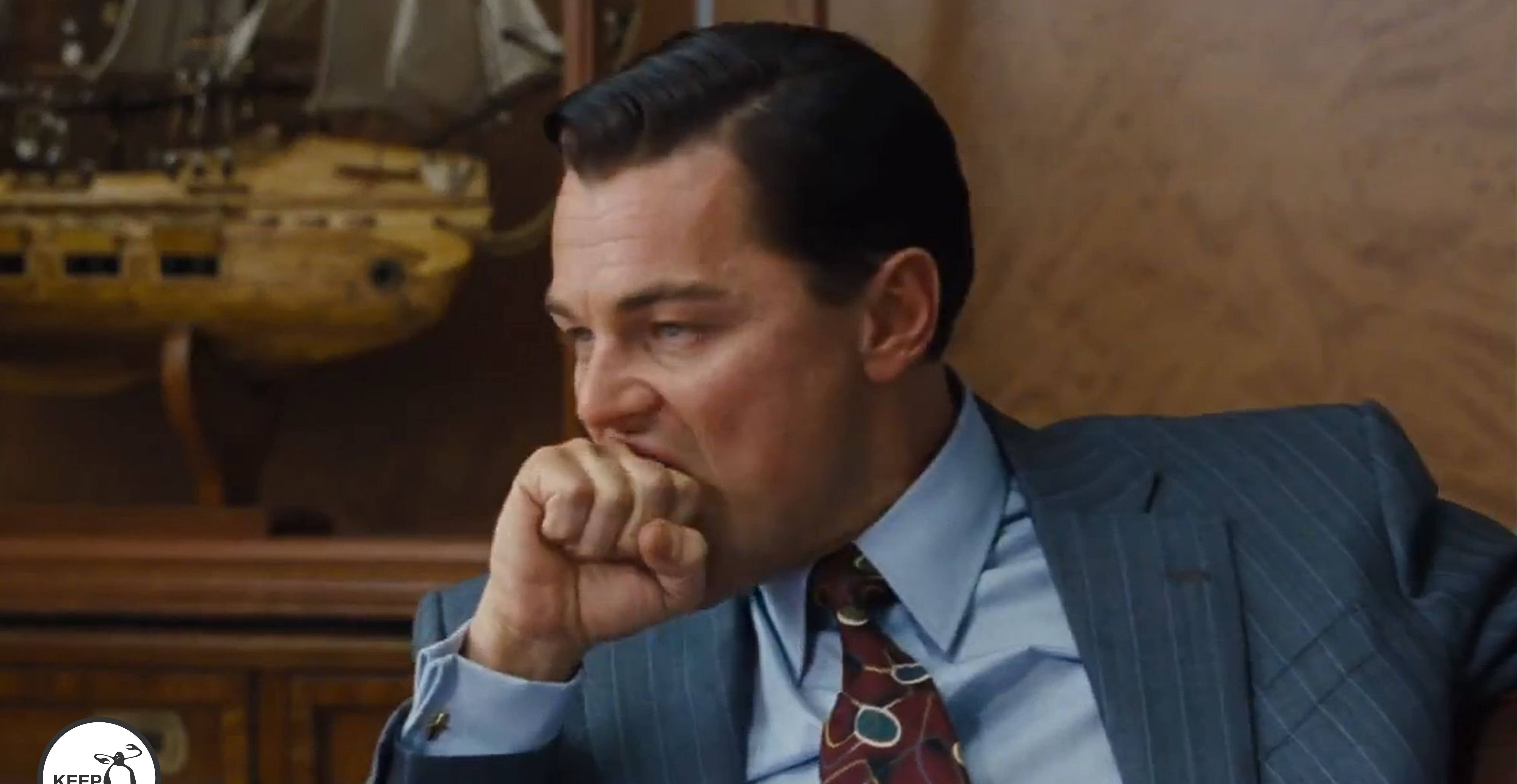


# ■ De Developer a DevOps

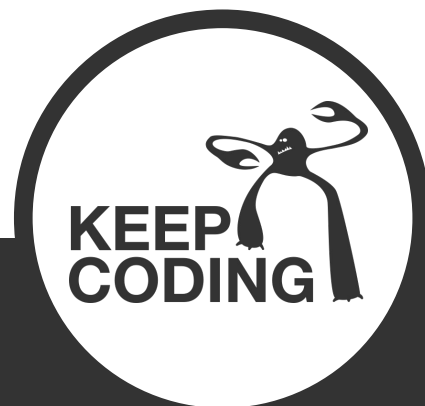
Y encima, desde la **consola de comandos**  
(porque usaremos Linux)





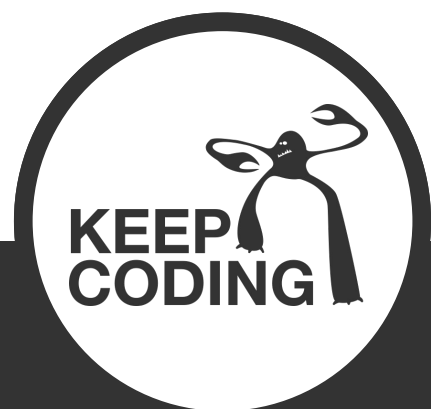


# ¿Linux? ¿Por qué no Windows Server?



# ■ ¿Linux? ¿Por qué no Windows Server?

- Precio de licencias
- Al no tener UI, más memoria para procesos del servidor











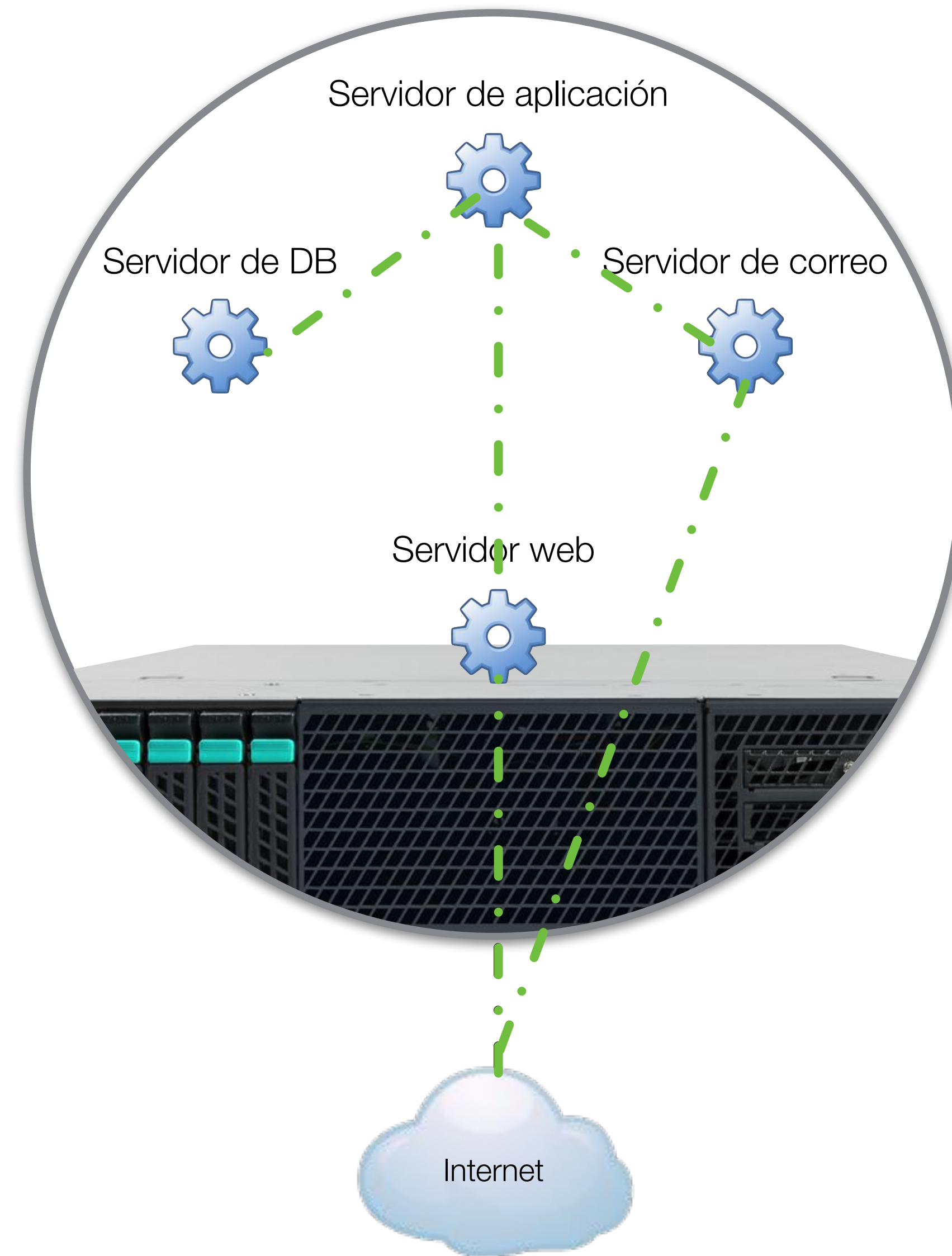


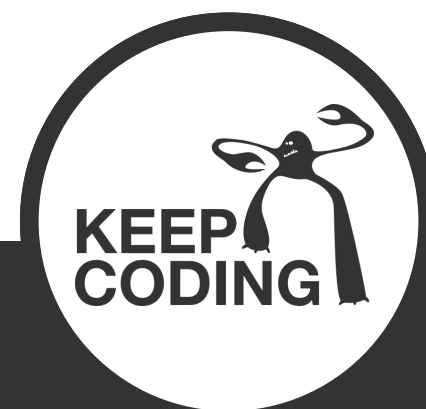
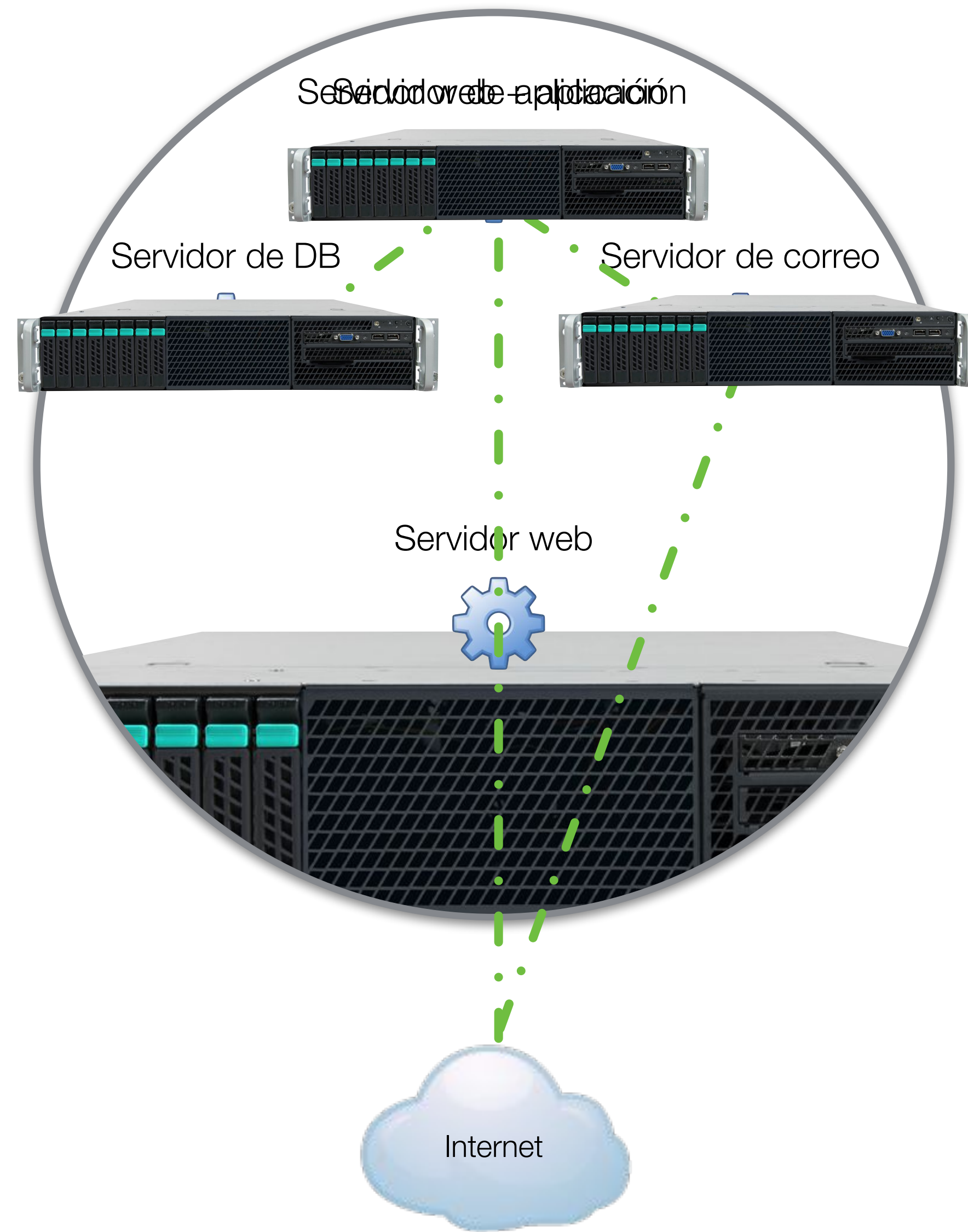


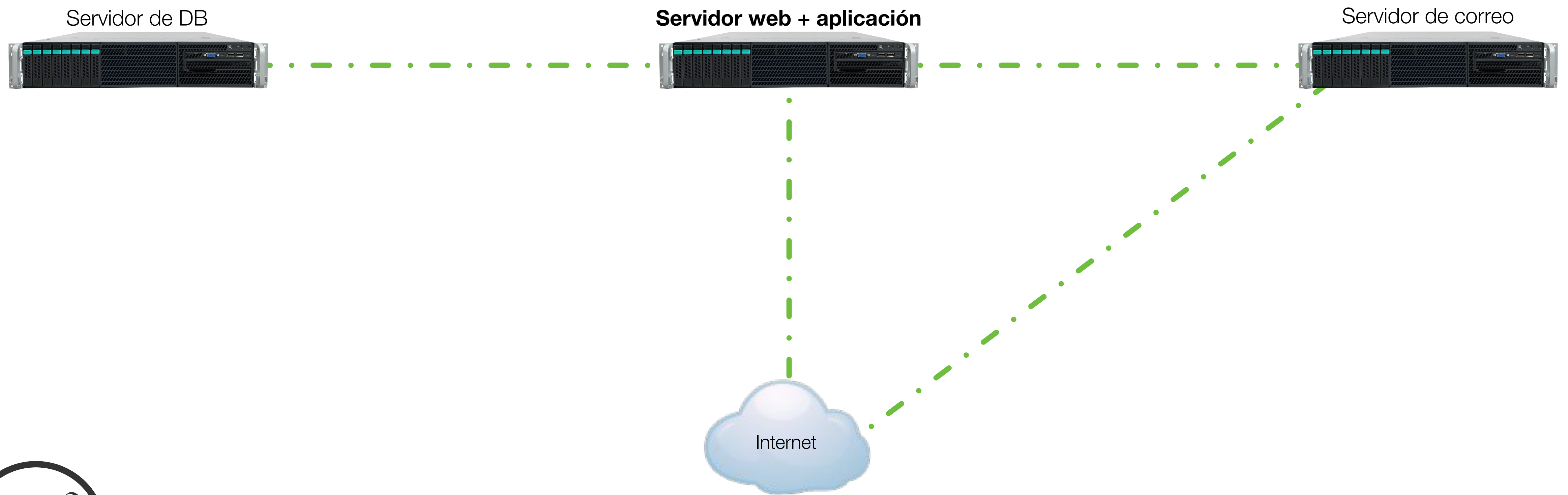


# ■ Ejemplos de Infraestructuras







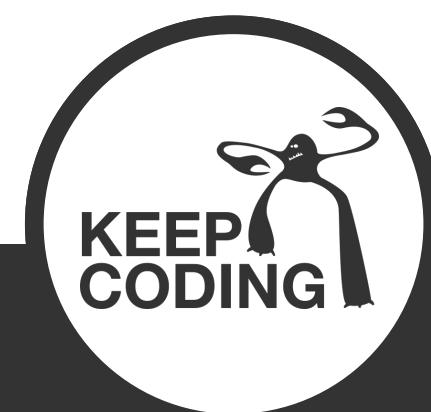




**Servidor web + aplicación**

Servidor de DB

Servidor de correo



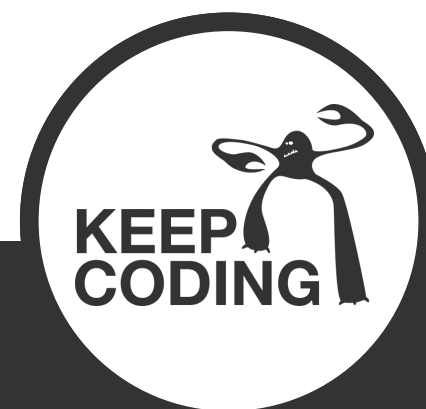
Servidores web + aplicación

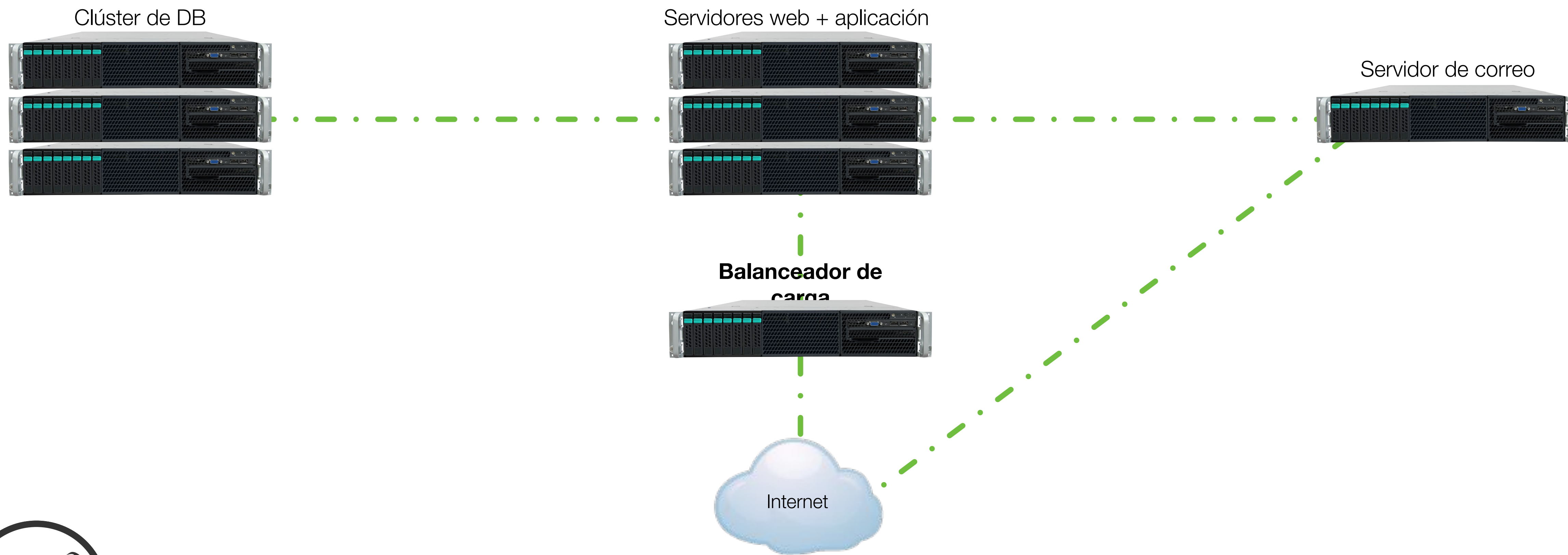
Servidor de DB

Servidor de correo

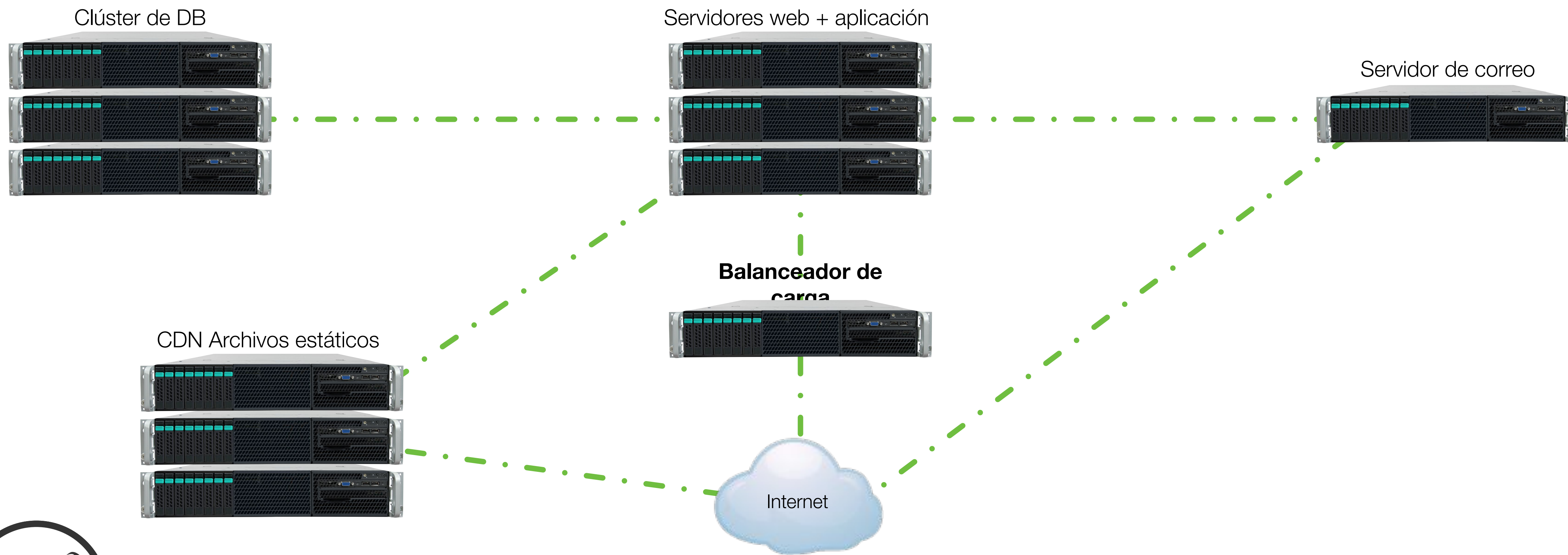
Balanceador de  
carga

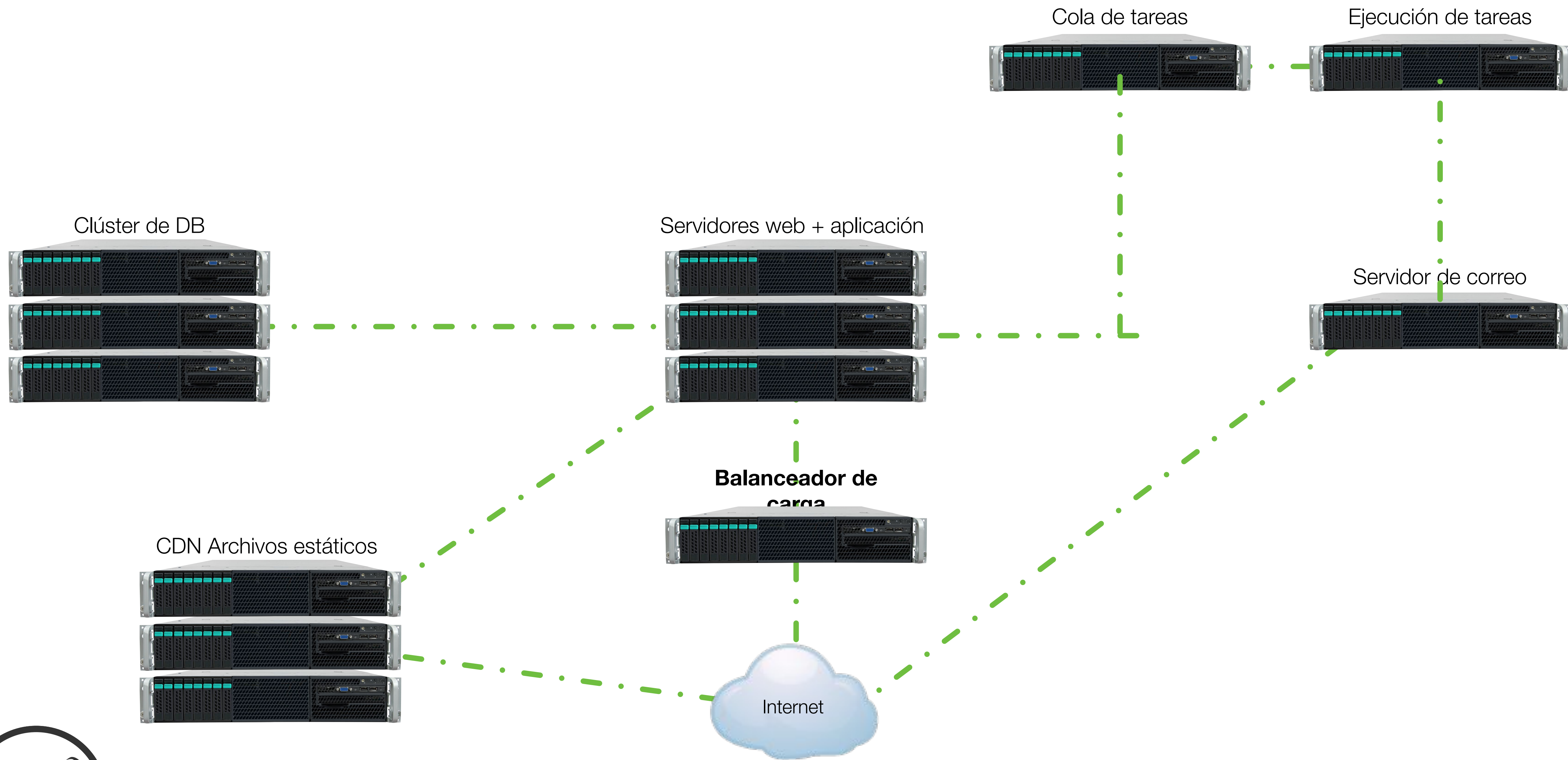
Internet













Servidores web + aplicación



Clúster de DB



CDN Archivos estáticos



Balanceador de carga



Internet



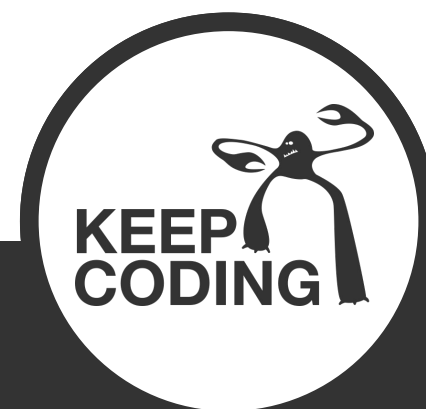
Cola de tareas



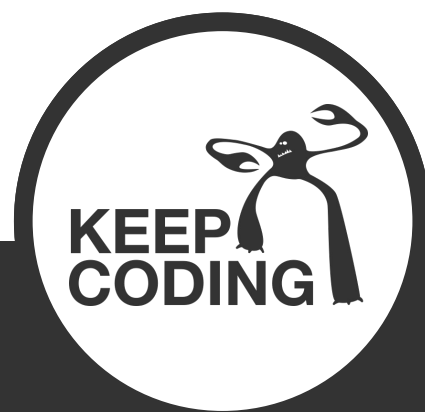
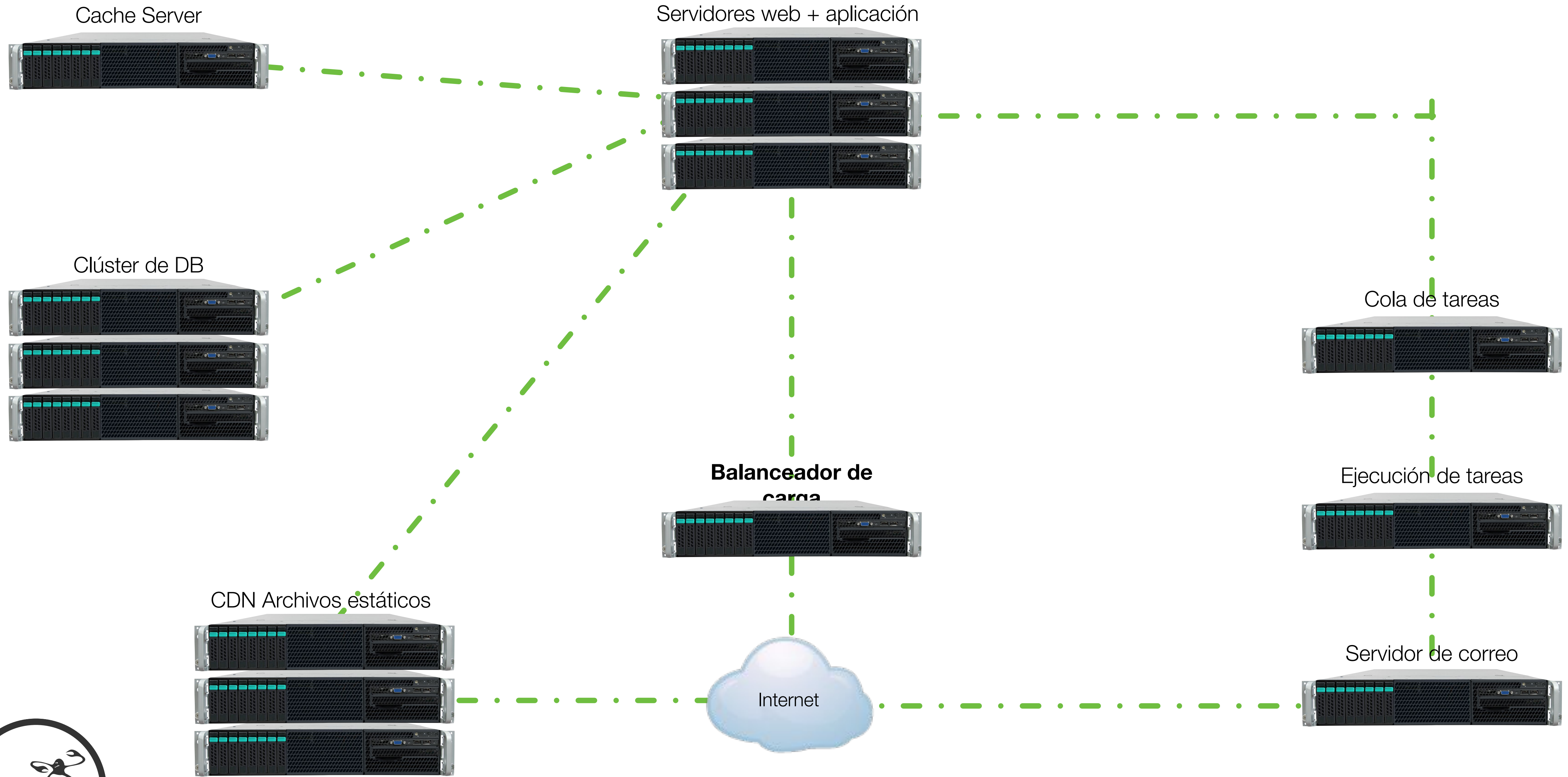
Ejecución de tareas



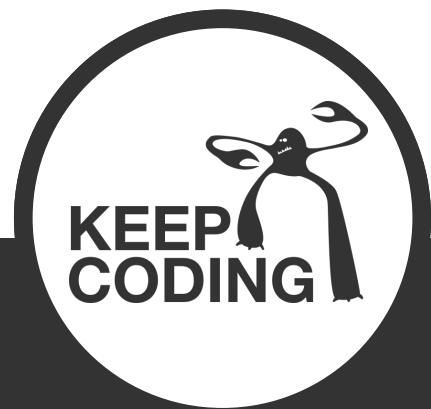
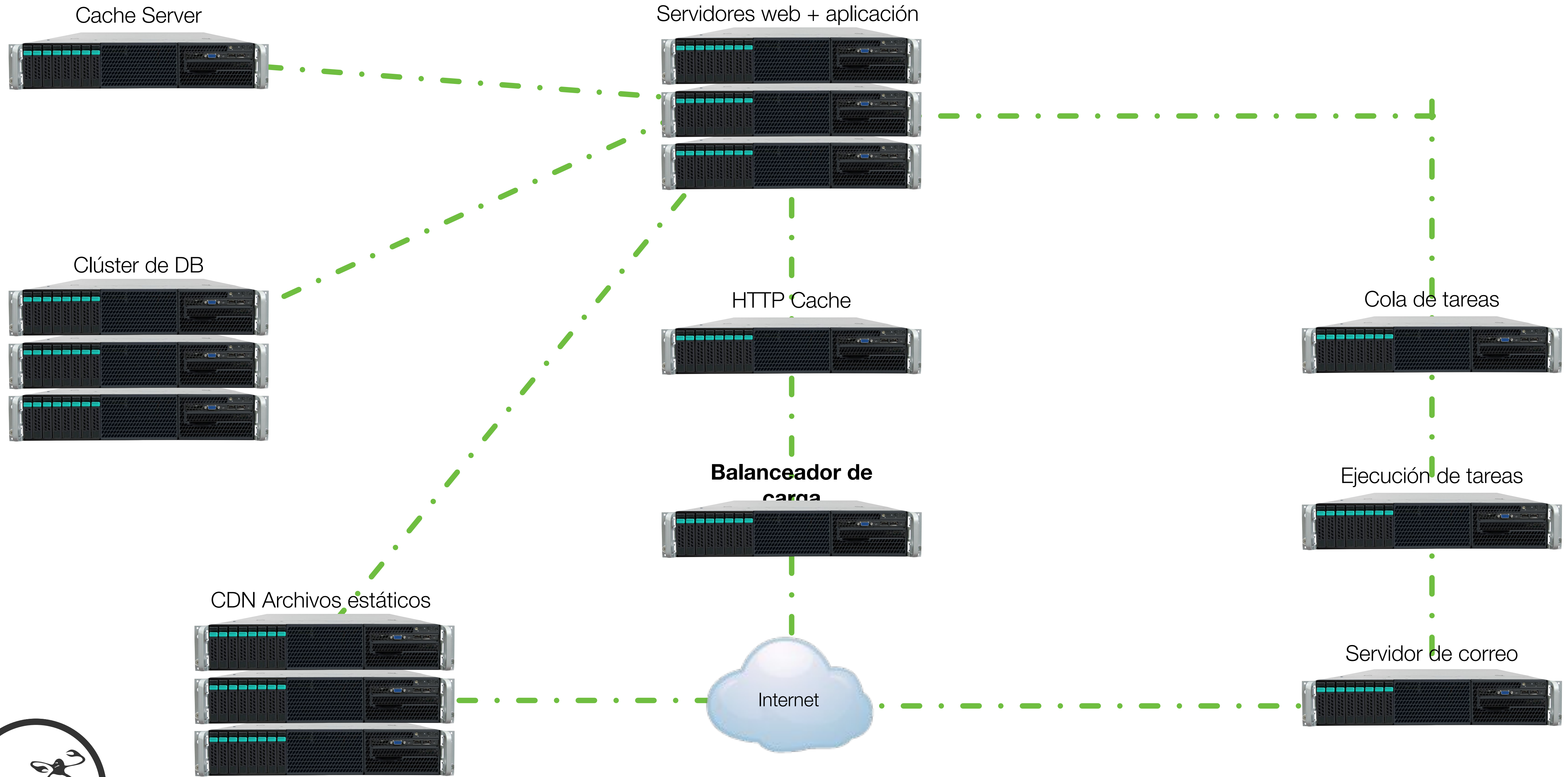
Servidor de correo













Servidores web + aplicación



HTTP Cache



Balanceador de carga



DNS Round Robin



Internet



Cache Server



Clúster de DB



CDN Archivos estáticos



Cola de tareas



Ejecución de tareas



Servidor de correo





Servidores web + aplicación



Node

Nginx + PHP-FPM/PHP Fast CGI  
Nginx + uWSGI/Gunicorn



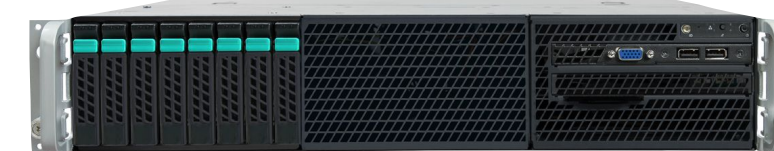
HTTP Cache

Varnish



Balanceador de carga

Nginx



DNS Round Robin

Bind



Internet



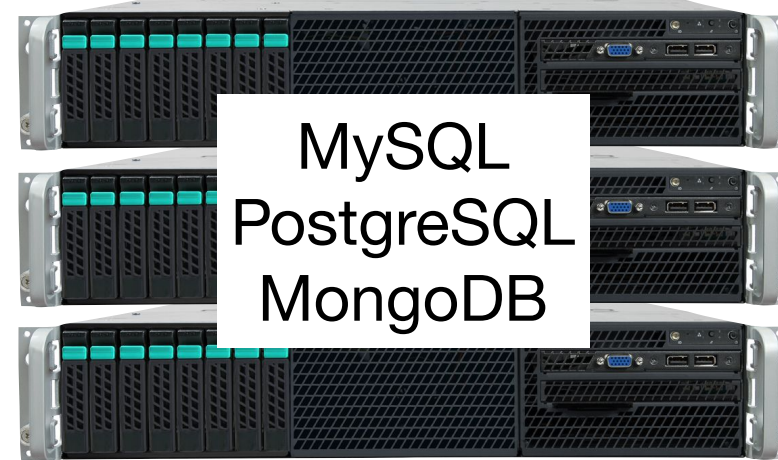
Cache Server

Memcached /  
Redis



Clúster de DB

MySQL  
PostgreSQL  
MongoDB



CDN Archivos estáticos

Azure CDN  
AWS CloudFront



Cola de tareas

RabbitMQ /  
Redis



Ejecución de tareas

Celery / Node



Servidor de correo

Postfix

