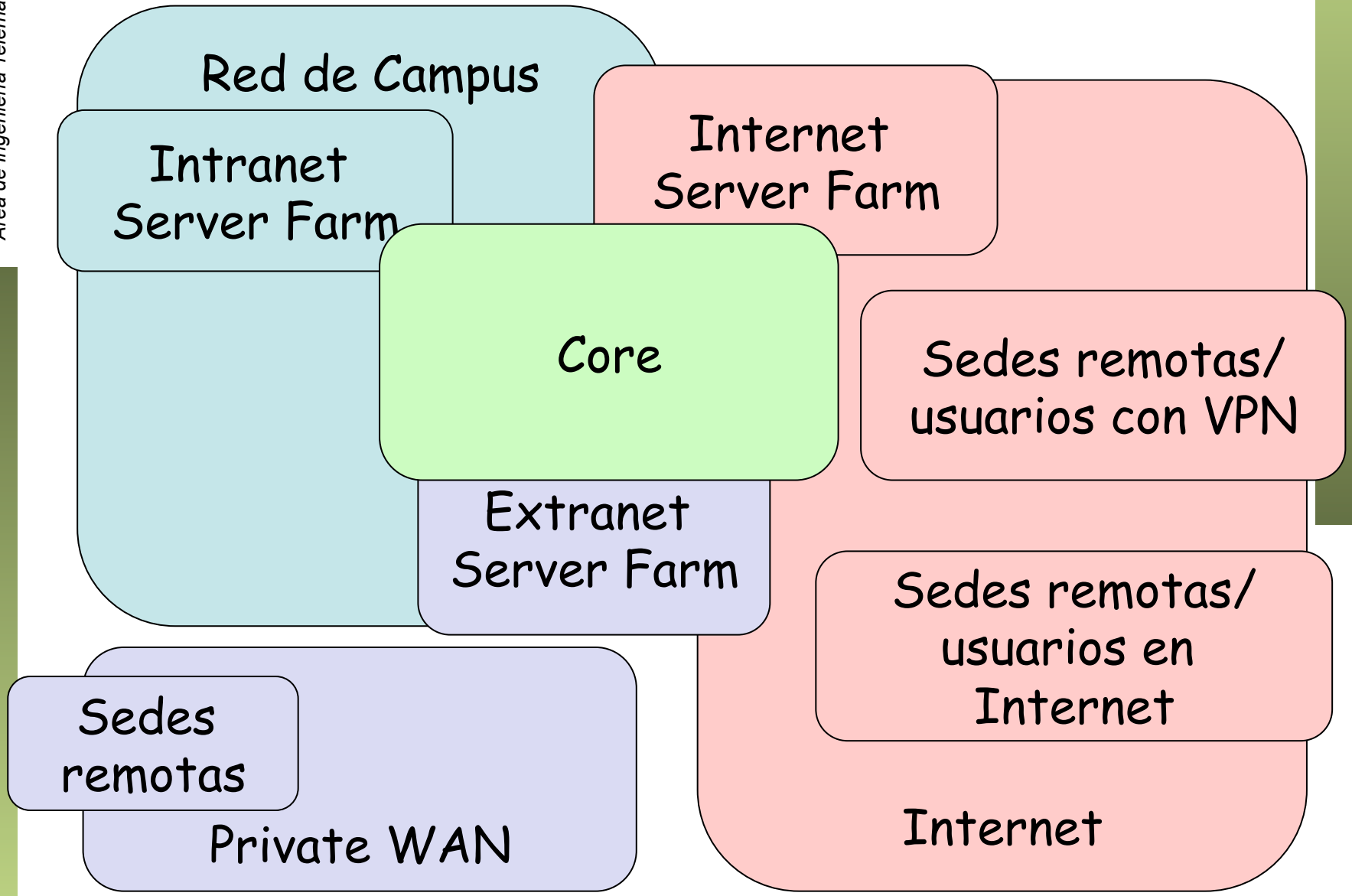
Enterprise network

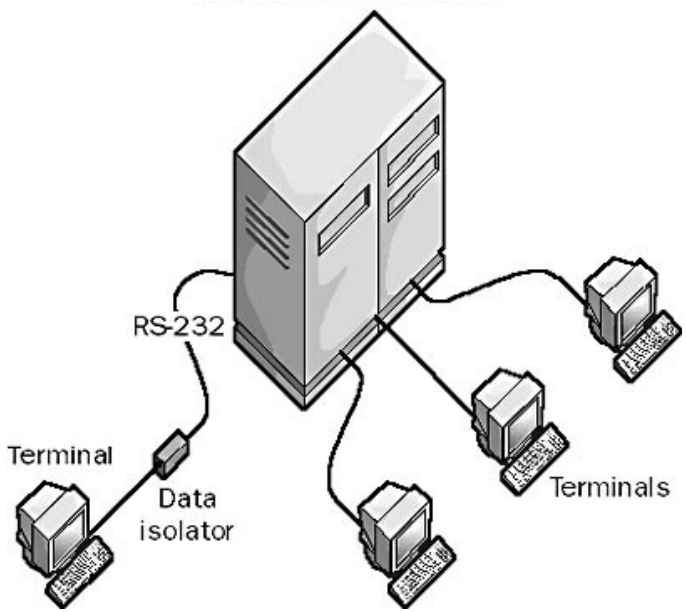
Elementos en la red empresarial



Arquitectura del servicio

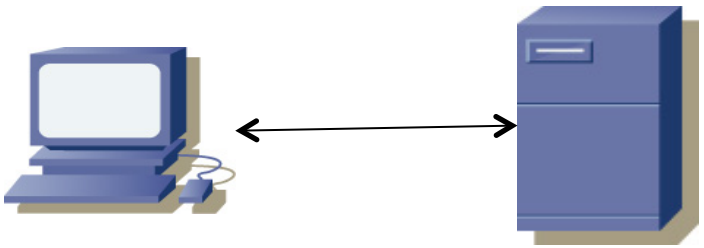
Arquitectura del servicio

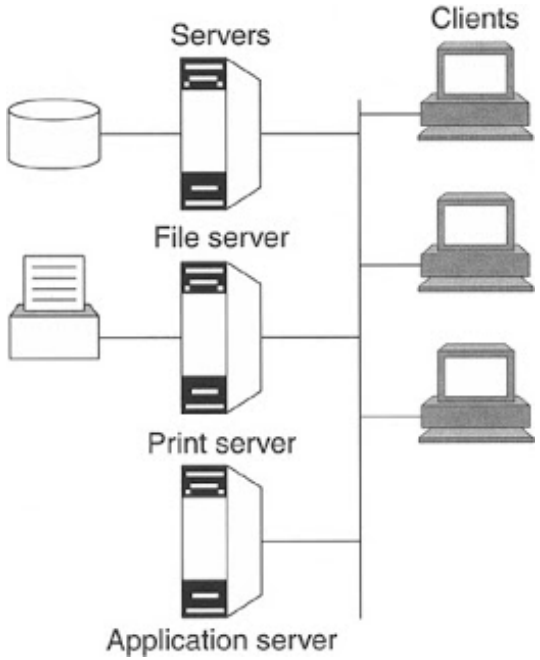
- Antes de las LANs y la arquitectura PC el *mainframe* era accedido desde terminales
- Los terminales (centenares de miles) eran *thin clients*, el trabajo pesado lo hacía el *mainframe*
- Está más orientado al trabajo en bloques (*batch*)
- El *mainframe* sigue vivo, aunque lo podemos ver como un servidor con grandes capacidades de virtualización
- Ahora el *mainframe* también está en red



Arquitectura C/S

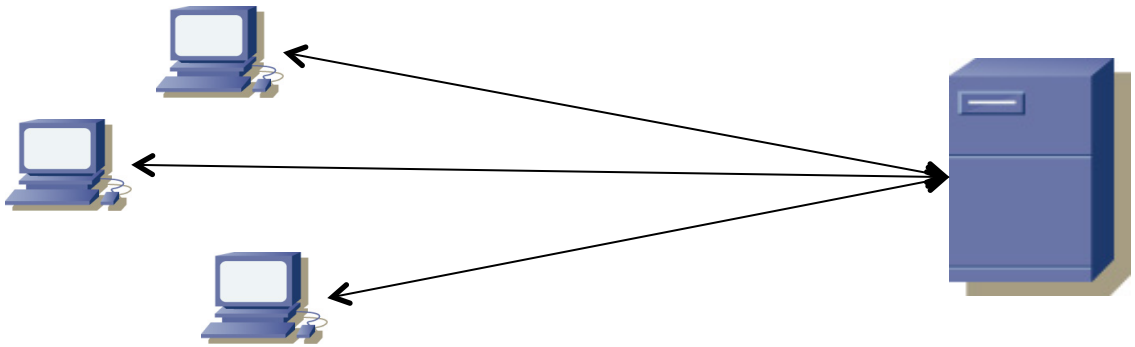
- Cliente-servidor
- Servidores de menor capacidad que *Mainframe*
- Clientes de mayor capacidad que terminales “tontos” (*thick client*)
- Servidores como hardware independiente
- Servidores distribuidos por la red de la empresa
- Interfaces propietarios hasta llegar la web
- Se migra de una arquitectura básica c/s a una basada en web
- Se sigue lo que se conoce como el modelo *n-Tier*





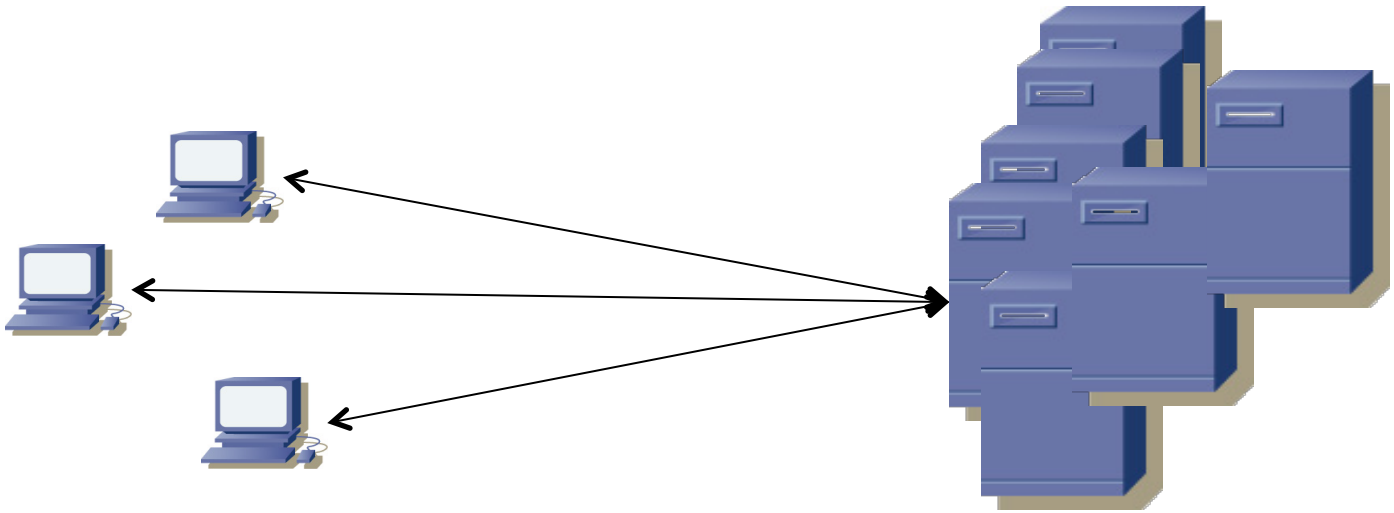
Arquitectura del servicio

- La aplicación puede estar centralizada en un servidor
- Pero seguramente con usuarios remotos
- ¿La aplicación está solo en un servidor? (…)



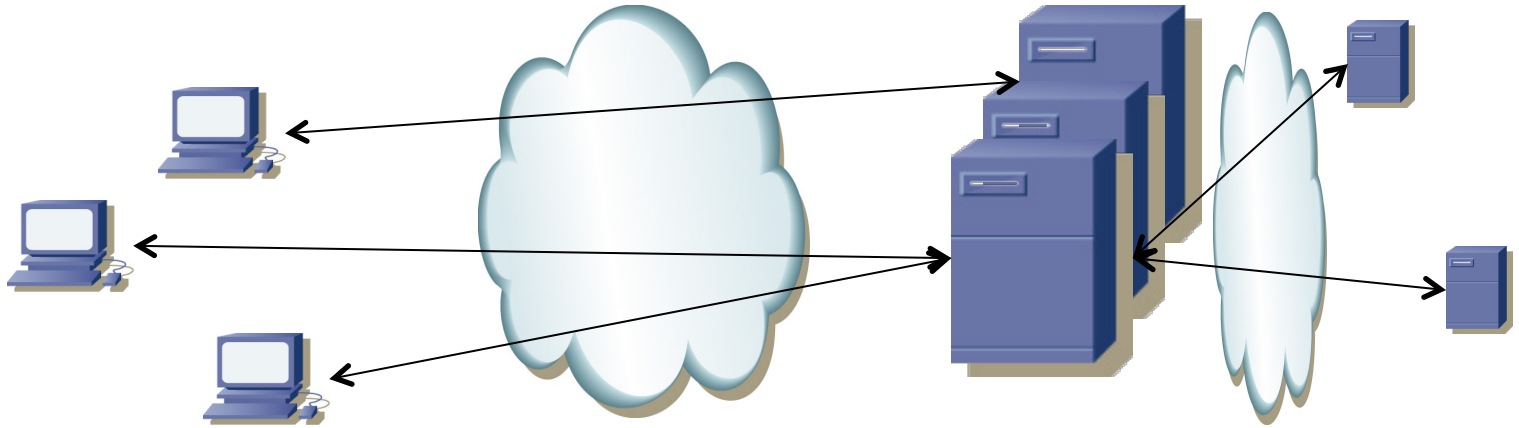
Arquitectura del servicio

- La aplicación puede estar centralizada en un servidor
- Pero seguramente con usuarios remotos
- ¿La aplicación está solo en un servidor?
- ¿Cómo escalar?
  - Verticalmente (***scale-up***)
  - Horizontalmente (***scale-out***) aumentando el número de servidores y repartiendo el trabajo entre ellos (…)
  - Pueden ser de capacidad moderada y bajo coste
  - Ahora se requiere una forma de repartir el trabajo entre ellos (más complejo)



Arquitectura del servicio

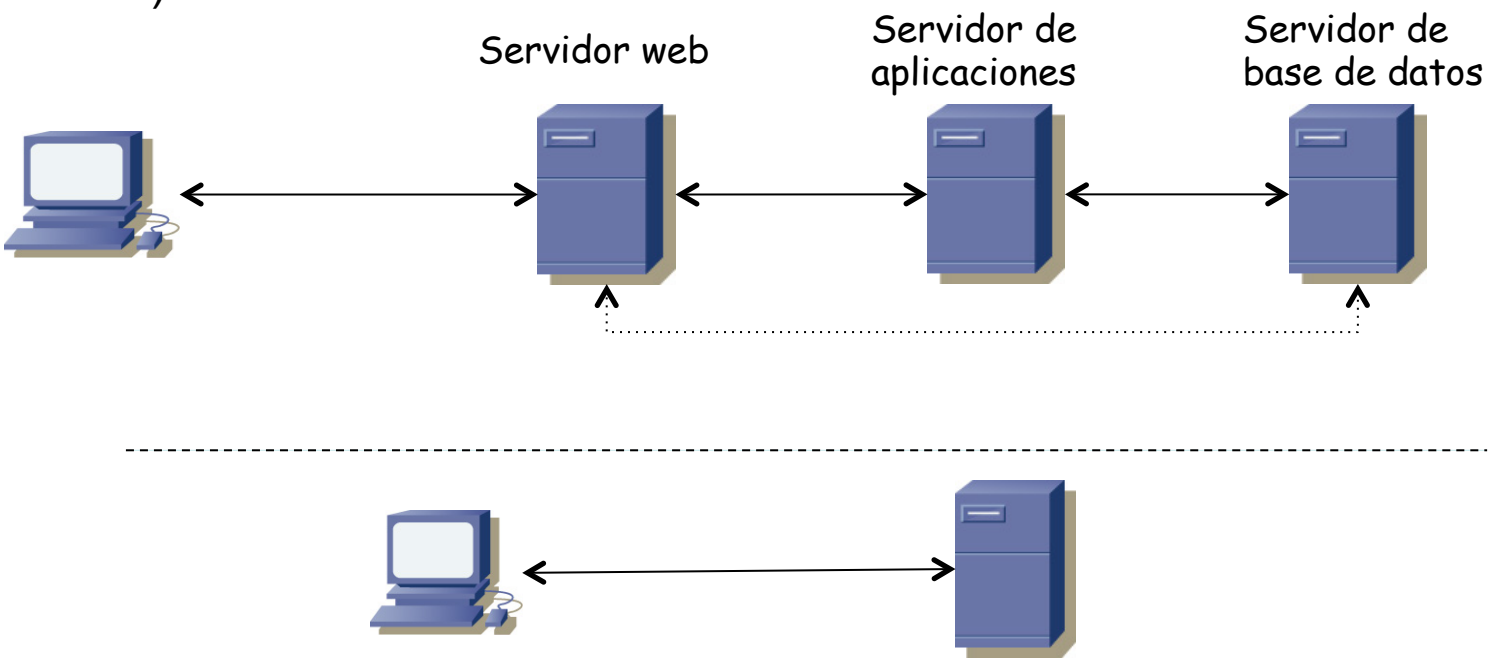
- La aplicación puede estar centralizada en un servidor
- Pero seguramente con usuarios remotos
- ¿La aplicación está solo en un servidor?
- ¿Cómo escalar?
- ¿Se comunica con otras aplicaciones/servidores? (¿por otro interfaz?)
- La red pretende darle un servicio a la aplicación
- La arquitectura de la aplicación/servicio condiciona cuál es un buen diseño de la red
- Para asegurar rendimiento y seguridad debemos conocer los caminos empleados



n-Tier model

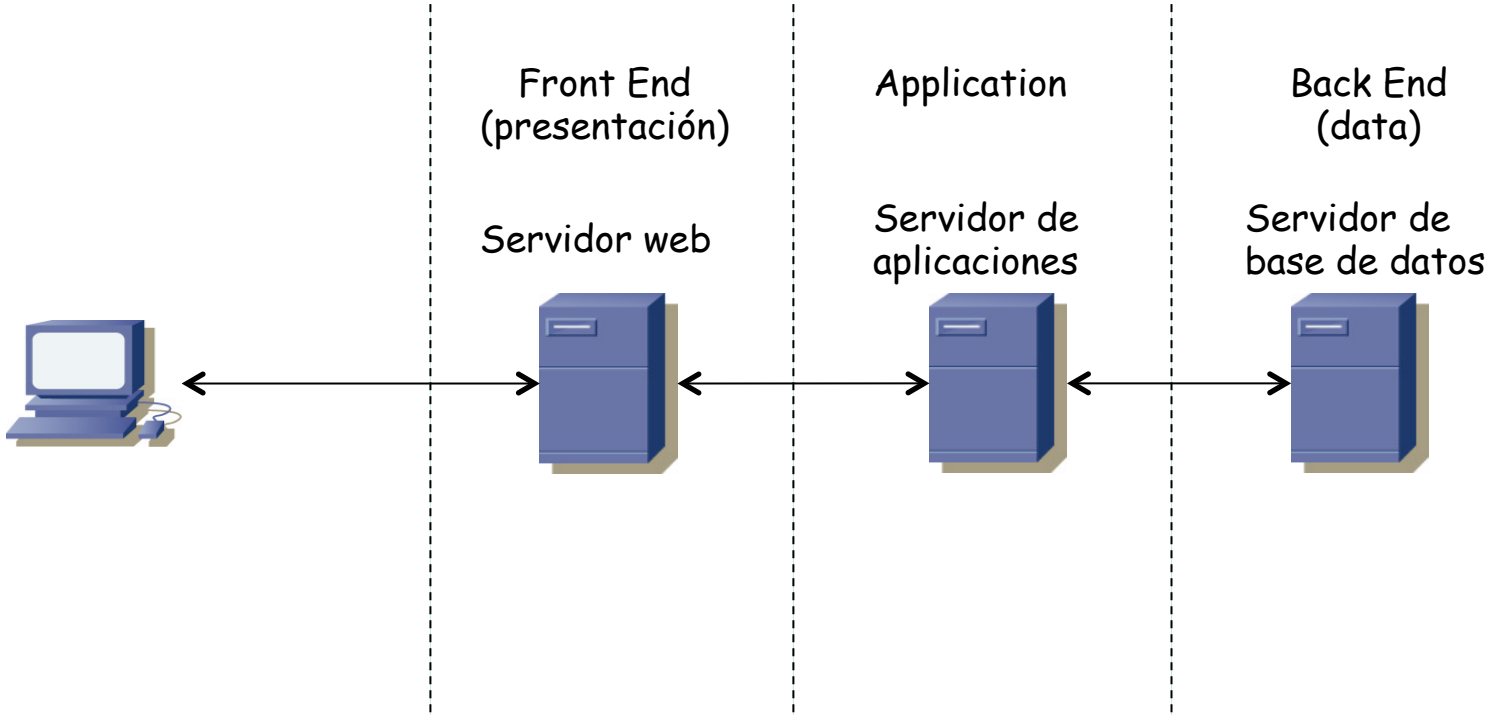
n-Tier model

- Las funcionalidades del servidor se dividen en niveles/capas/*tiers*
- Pueden distribuirse en servidores en 2, 3 o más capas
- El “pegamento” es el *middleware*
- Permite avanzar a una computación distribuida
- Eso permite escalar (*scale-out*) el sistema para mayores cargas
- Simplifica y distribuye el control de la aplicación
- Mejora la seguridad (intrusión en servidor web no implica en los demás)



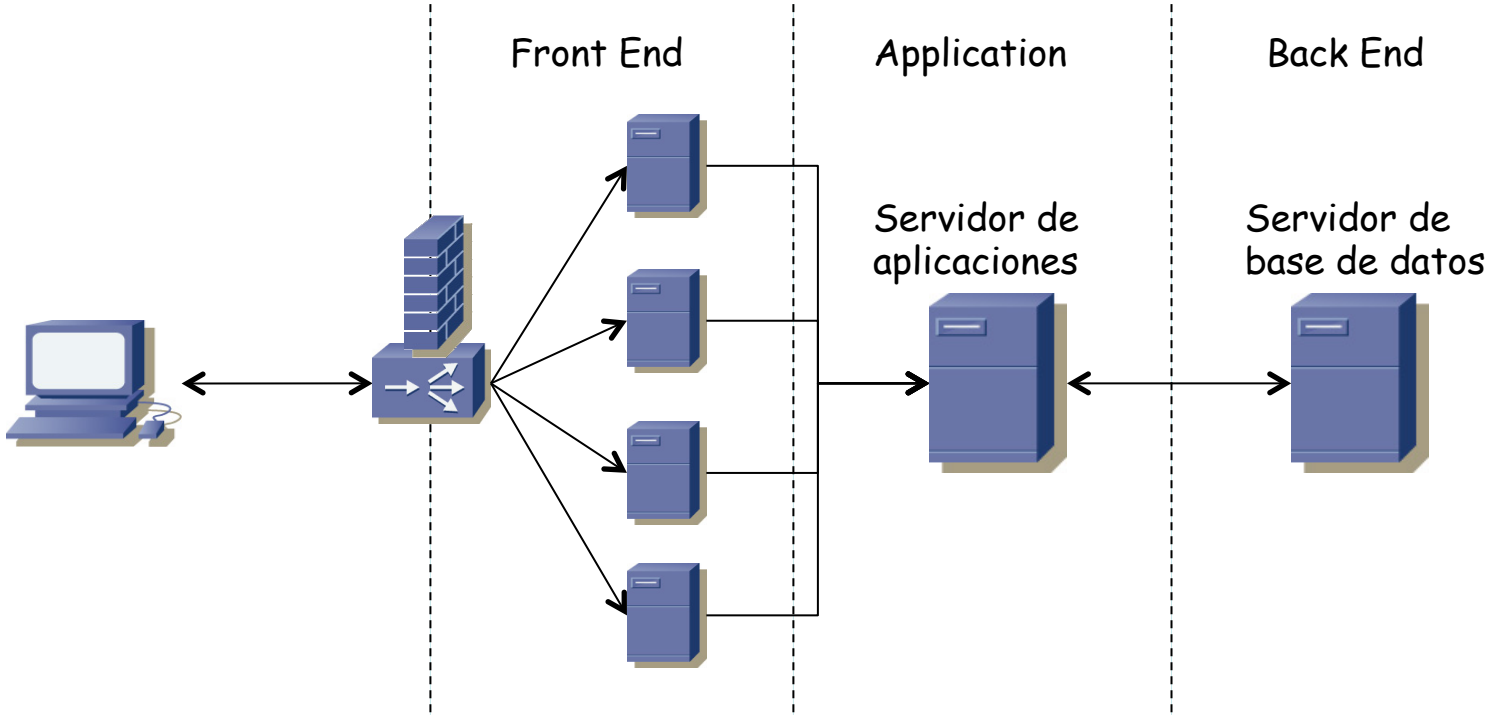
Arquitectura *multitier*

- La arquitectura de red ofrece separación física y lógica entre las capas de la aplicación
- Segmentos de red: Front End, Application y Back End



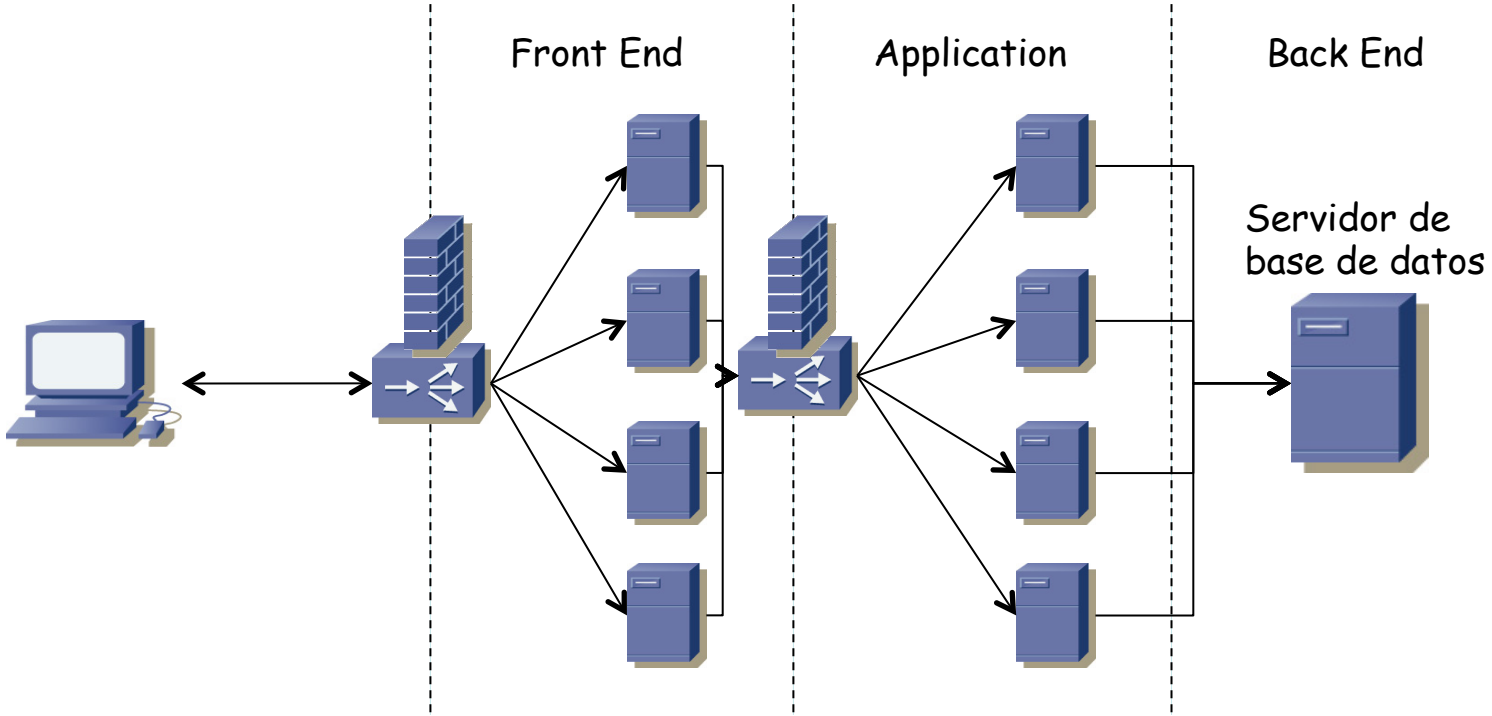
Arquitectura *multitier*

- El escalado se consigue mediante equipos que hacen balaceo de carga (y firewall) (...)



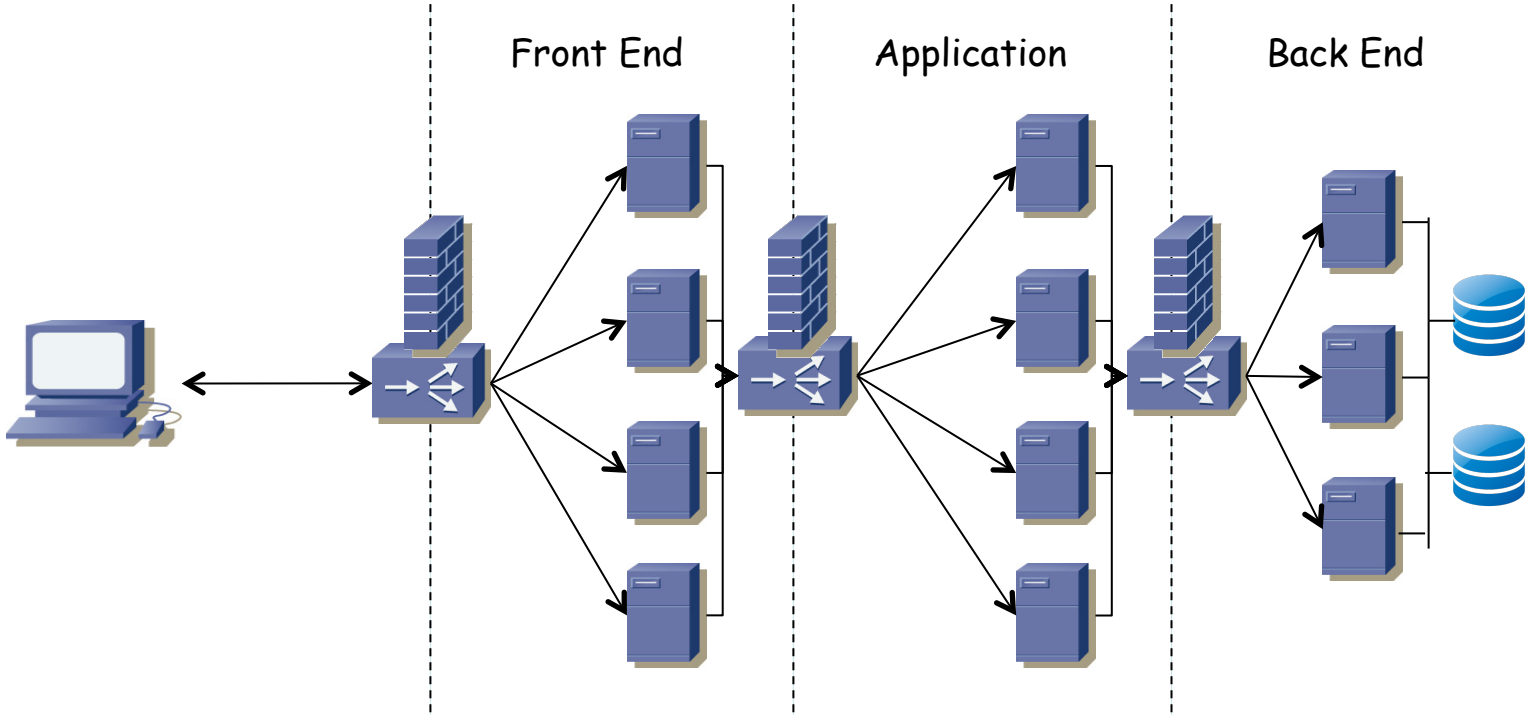
Arquitectura *multitier*

- El escalado se consigue mediante equipos que hacen balaceo de carga (y firewall) (...)



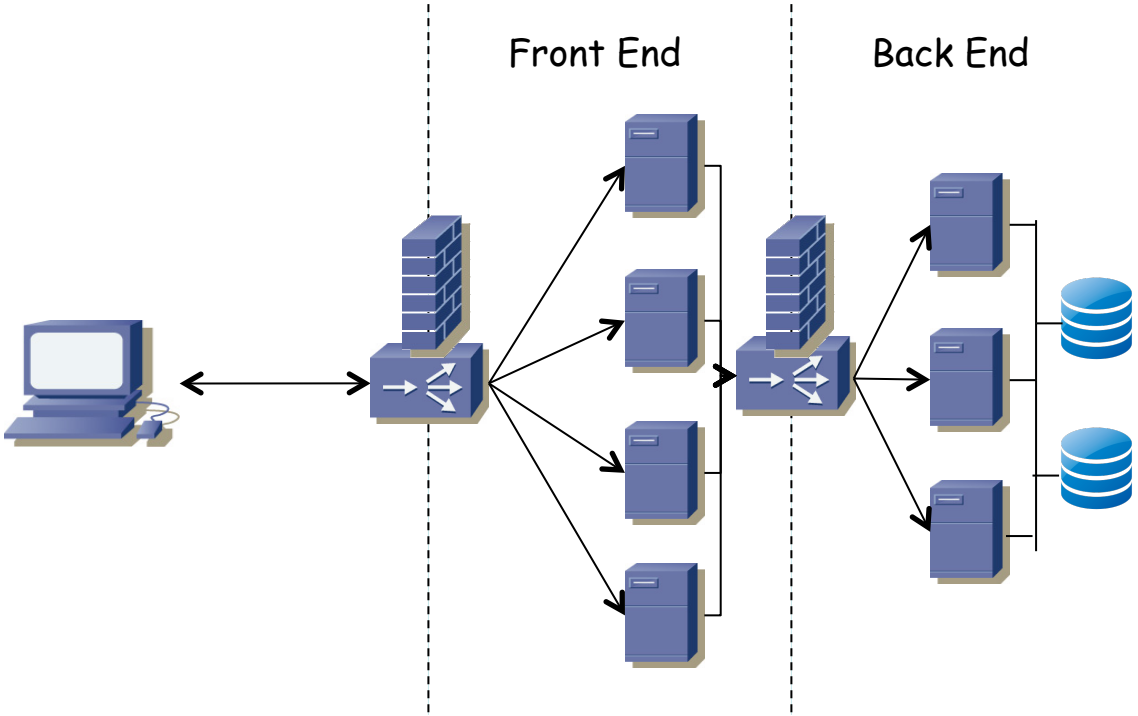
Arquitectura *multitier*

- El escalado se consigue mediante equipos que hacen balaceo de carga (y firewall)
- Otra opción serían técnicas de *clustering* entre esos servidores
- En ese caso el trabajo lo hace la aplicación en lugar de la red



2-tier vs 3-tier

- ¿Necesitamos 3 tiers?
- Depende de dónde necesitemos crecer al aumentar la carga



Arquitectura *multitier*

- ¿Cuellos de botella?
  - Servidor/es
  - Base de datos
  - Protocolos
  - Acceso a web services externos
  - (...)