Indice

3. Big Data

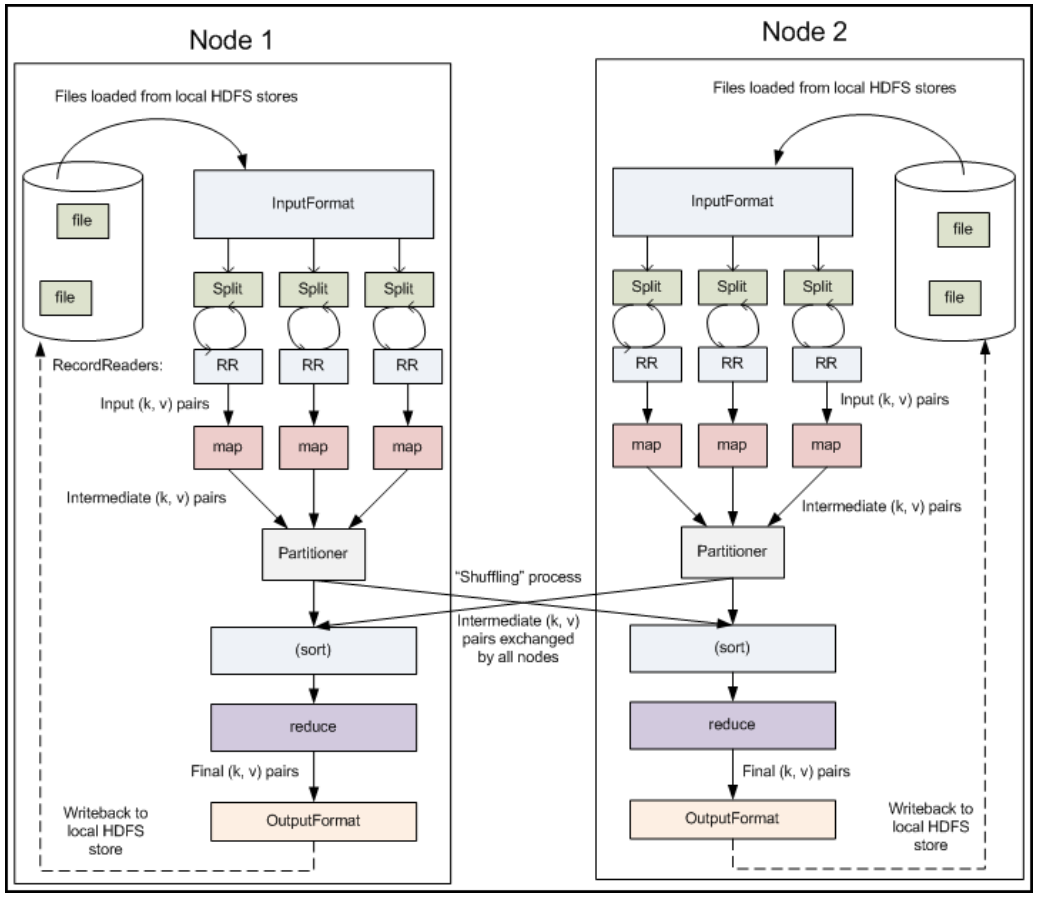
- En cualquier servicio se genera mucha información que se almacena en bruto de forma masiva porque no se es capaz de saber a priori en que información depurada se está interesado
  - Por si acaso se guarda todo
  - Supone un problema cada vez más importante e interesante el ser capaz de extraer información interesante de cantidades de datos masivos
- Sistemas que manipulan grandes conjuntos de datos.
- Grandes conjuntos de datos relacionados, por ejemplo:
  - Correlar los hábitos de compra de usuarios
  - Analizar el comportamiento de compras con tarjeta de crédito
  - Identificar perfiles de usuarios según sus búsquedas en Internet
  - Análisis de negocio
  - Evaluación de datos de enfermedades infecciosas
  - Meteorología, genómica, etc.
- De utilidad cada vez mayor en cualquier tipo de empresa

Big Data

- Las dificultades más habituales en estos casos se centran en
  - la captura
  - el almacenado
  - la búsqueda
  - la compartición
  - el análisis
  - la visualización
- Nuevas herramientas: distribución de tareas en nodos trabajadores y sistemas de ficheros
  - Google para MapReduce y Google File System (GFS)
  - Apache Hadoop

Big Data

- MapReduce es un modelo de programación diseñado para procesar grandes volúmenes de datos en paralelo gracias a dividir el trabajo en un conjunto de tareas independientes



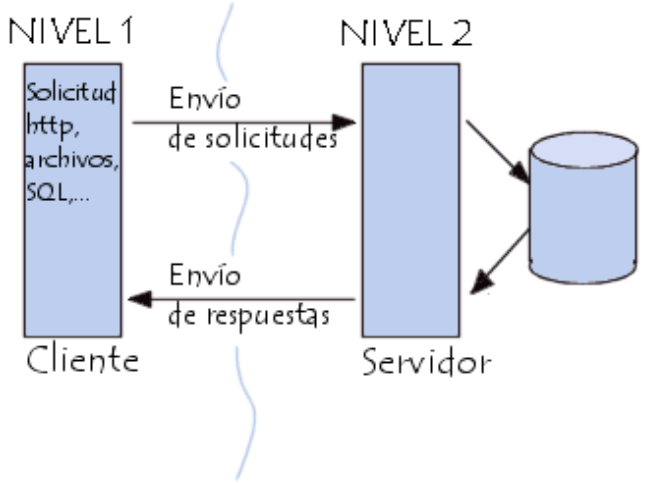
4. Arquitecturas de servicio

- A la hora del diseño de una arquitectura de servicio sobre la red podemos distinguir dos puntos de vista:
  - Capas: segmentación desde el puntos de vista lógico
    - Cómo se organiza desde el punto de vista funcional
  - Niveles: segmentación desde el puntos de vista físico
    - Cómo se distribuyen en diferentes servidores
- Propuestas de arquitectura de servicios
  - Arquitectura a 2 capas
  - Arquitectura a 3 capas

4.1 Arquitecturas a 2 capas

- Capa cliente
  - Solicitan y presentan la información
- Capa servidor
  - Almacenan y procesan la información

- Los servidores realizan múltiples tareas
- A nivel de máquinas y servicios el mapeo es directo



Arquitecturas a 2 capas

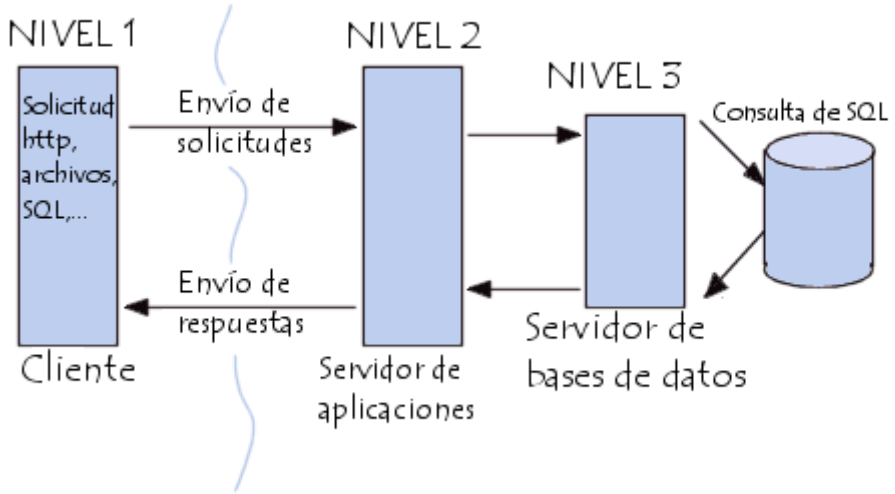
- Dificultades de la arquitectura de 2 capas
  - Cambios en una capa afectan a la otra
  - Dificultad al compartir procesos comunes: escalabilidad
  - Tareas de servidor no distribuibles
  - Problemas de seguridad, etc.

4.2 Arquitecturas a 3 capas

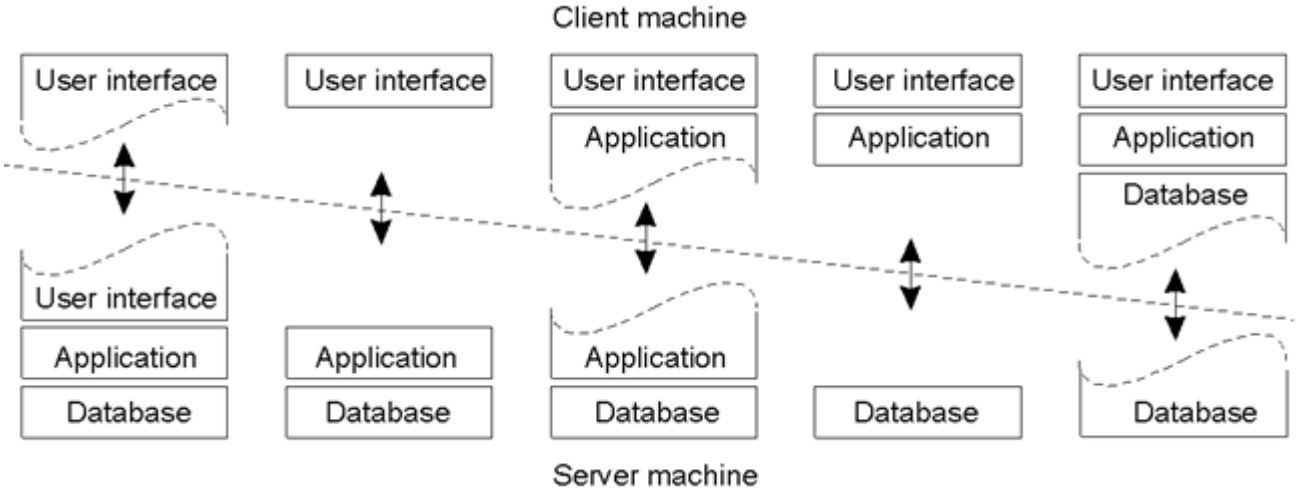
- Capa de presentación
  - Respuesta al usuario, formularios, informes
- Capa de negocios
  - Reglas del negocio, flujos, cálculos, validaciones, middlewares
- Capa de datos
  - Base de datos, sistemas de ficheros, procedimientos almacenados
- Se intenta especializar la labor de cada servidor
- A nivel de máquinas y servicios el mapeo no es tan directo
  - No tiene por qué ser un mapeo 1:1 entre capa y nivel

Arquitecturas a 3 capas

- 3 capas en 3 niveles

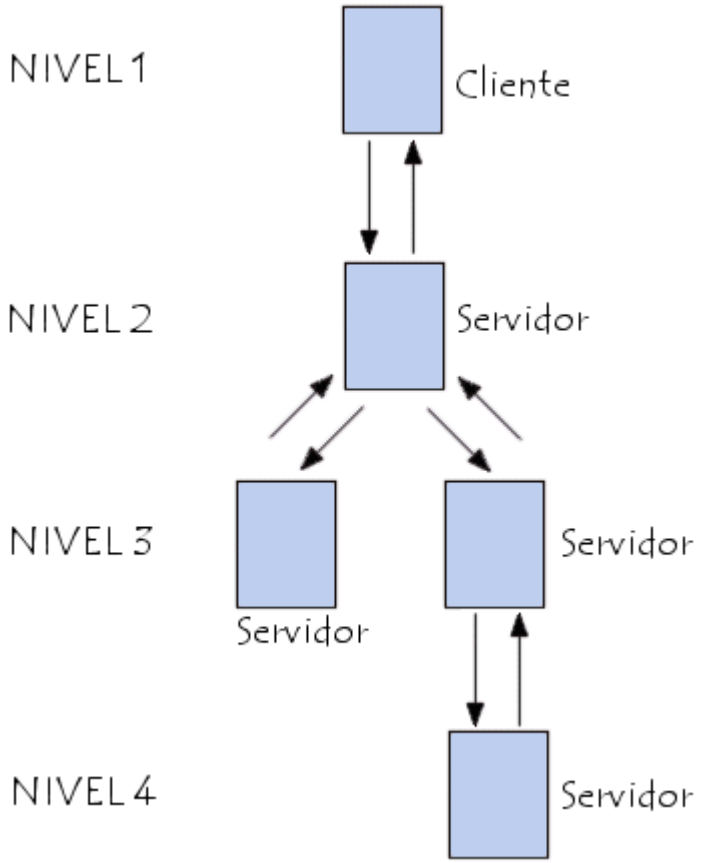


- 3 capas en 2 niveles



Arquitecturas a 3 capas

- Con múltiples servidores (N-niveles), cada servidor se especializa y optimiza para su tarea
- Ventajas
  - Más fácil de mantener
  - Los componentes son reusables
  - Desarrollo más rápido (división del trabajo implícito en la propia arquitectura)
  - Escalable
  - Seguridad: el cliente no puede acceder nunca a los datos en bruto

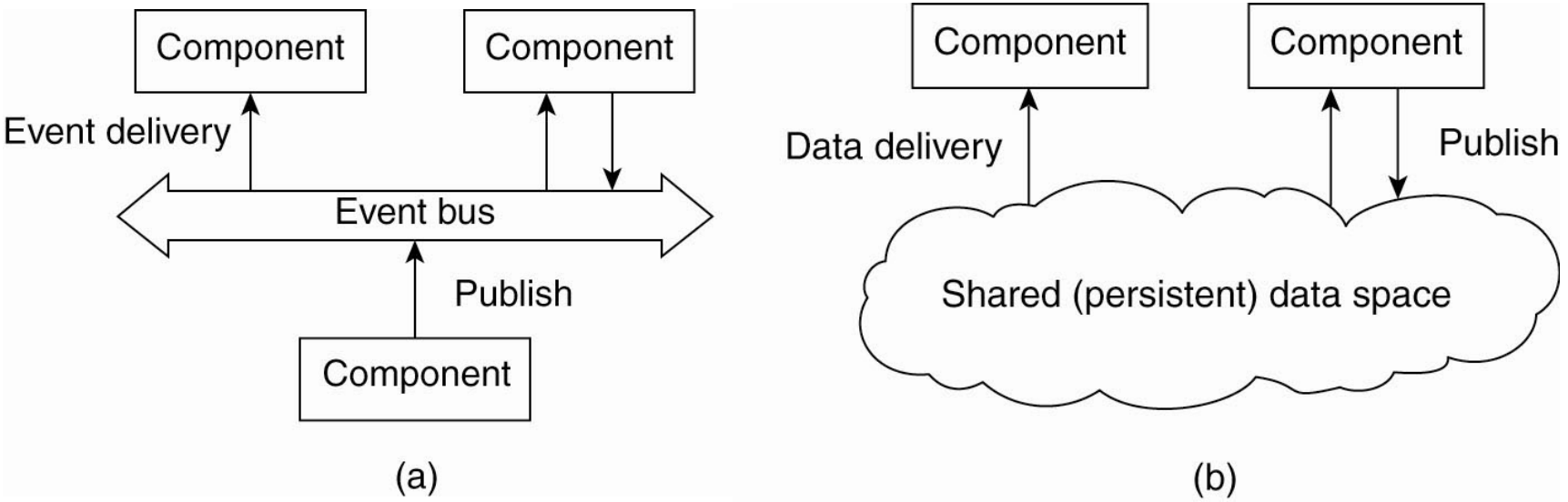


Front-End/Back-End

- Típicamente se habla de
  - Front-End: la capa de interacción con el cliente: usuario (persona) o con otras aplicaciones (web service)
  - Back-End: la capa interna de negocio (procesado) y de acceso a datos. También recibe el nombre de Back-Office

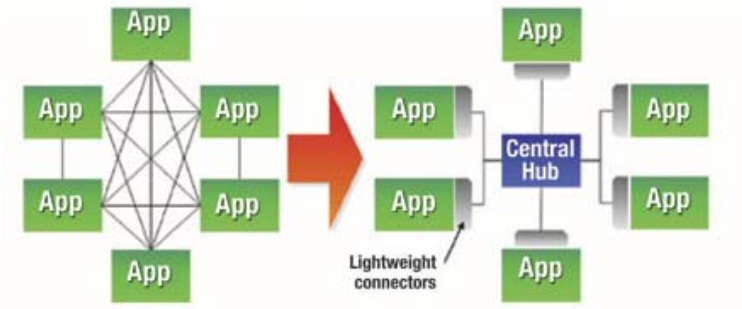
4.3 Comunicación intra/inter nivel

- Posibilidades
  - Punto a punto, síncrona
  - Arquitectura orientada a eventos (a)
    - Desacopla emisor-receptor
    - Permite broadcast, punto a punto, publicación/suscripción
  - Arquitectura de espacio de datos compartido (b)
    - Como la de eventos, añade persistencia (no es necesario que ambos extremo estén funcionando en un momento dado)

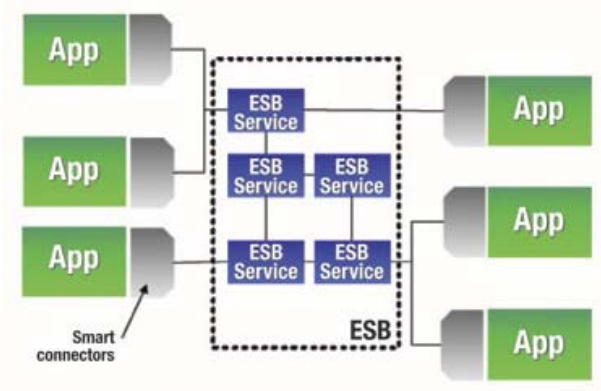


Comunicación intra/inter nivel

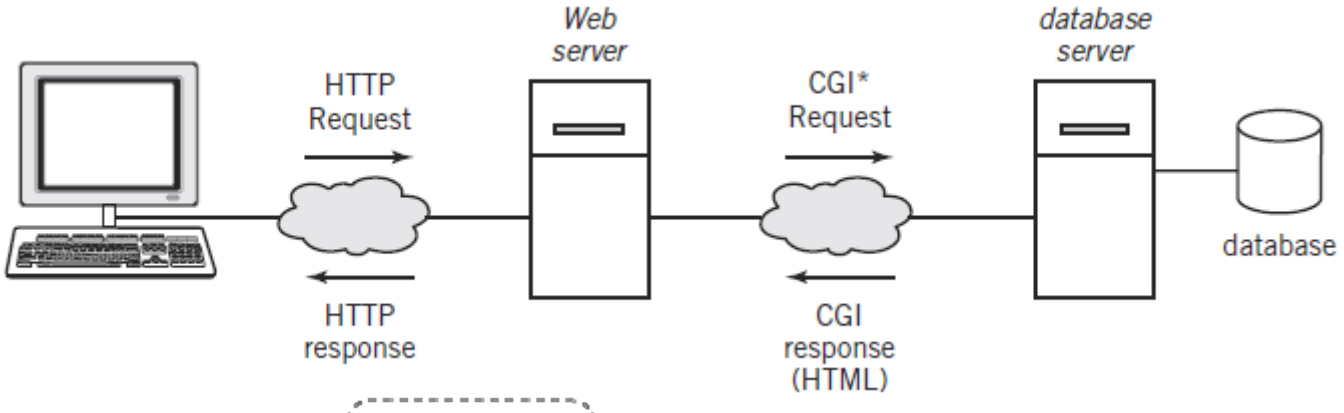
- Entreprise Service Bus (ESB)
  - Evita usar un nodo gestor central (central hub) que
    - Pudiese ser el punto de fallo
    - Exigiese interfaces estándar de conexión
    - Limitase la escaliabilidad
  - Infraestructura distribuida para la integración de servicios empresariales
  - Consiste en una arquitectura orientada a eventos con un bus de mensajes fiable
  - Se intercambian mensajes XML
  - El ESB provee servicios para trasformar y enrutar mensajes, así como para la administración centralizada del sistema

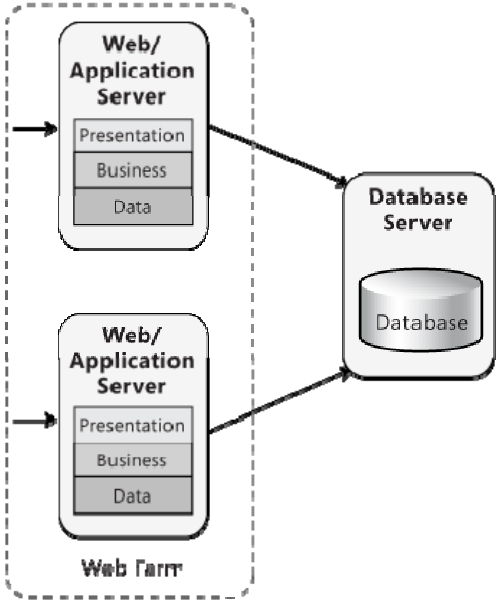


https://blogs.oracle.com/rtenhove/entry/what_is_enterprise_service_bus

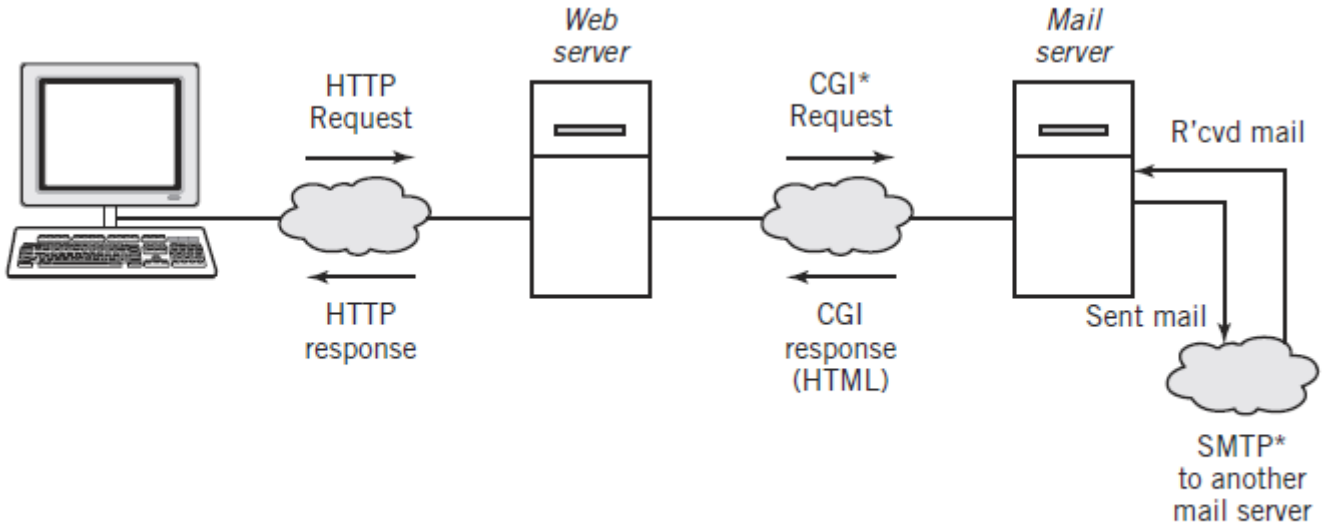


4.4 Ejemplo: Blog





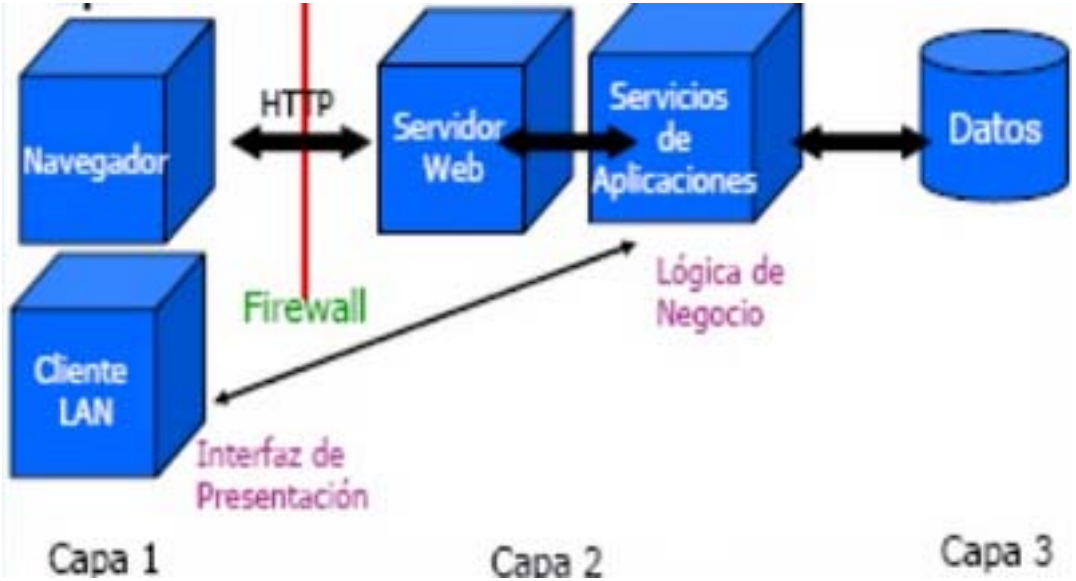
Ejemplo: Webmail



*SMTP: Simple Mail Transfer Protocol
*CGI: Common Gateway Interface

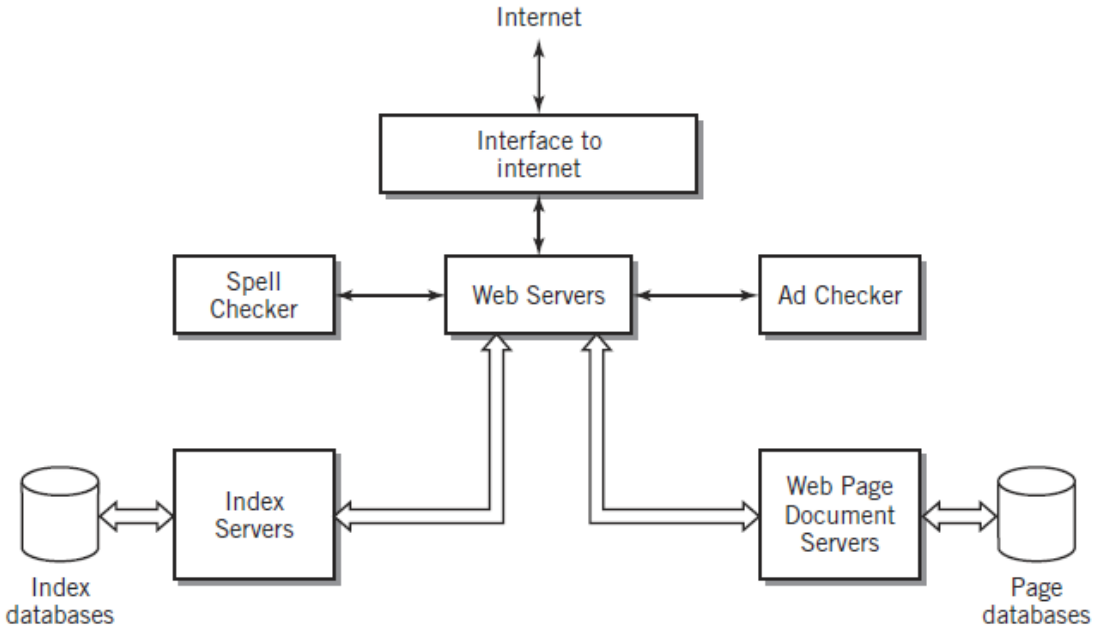
Ejemplo: aplicación de web empresarial

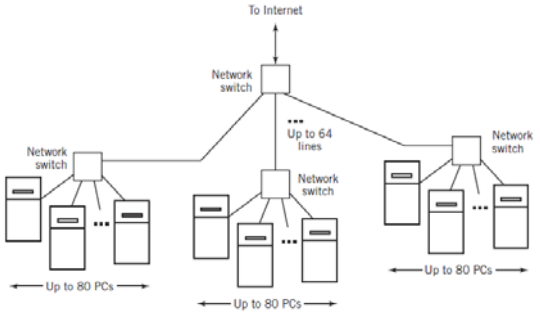
- Los servidores de aplicaciones permiten un acceso vía web o mediante interfaces específicos (SOAP)



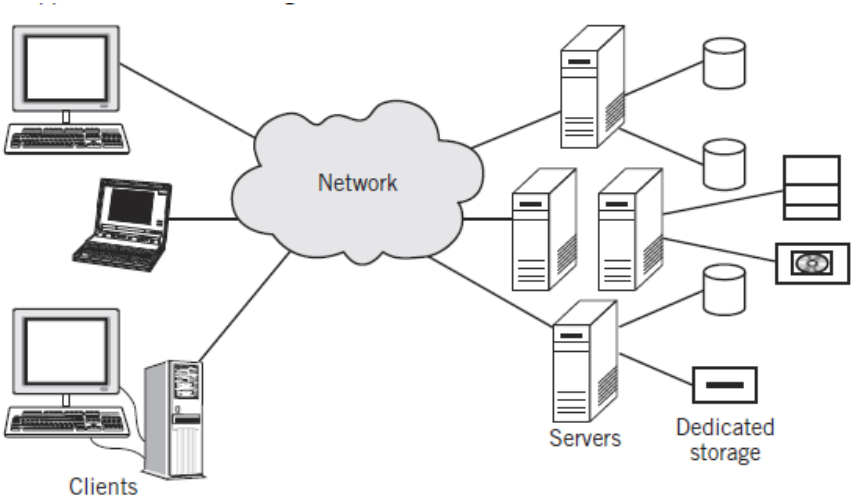
Ejemplo: búsquedas Google

- Versión simplificada de un servidor de un datacenter (redirigido a él por resolución de DNS)
- Donde dice Servers estamos hablando de cientos de servidores
- Cada datacenter despliega en sus servidores copia de la información del indexado de Google

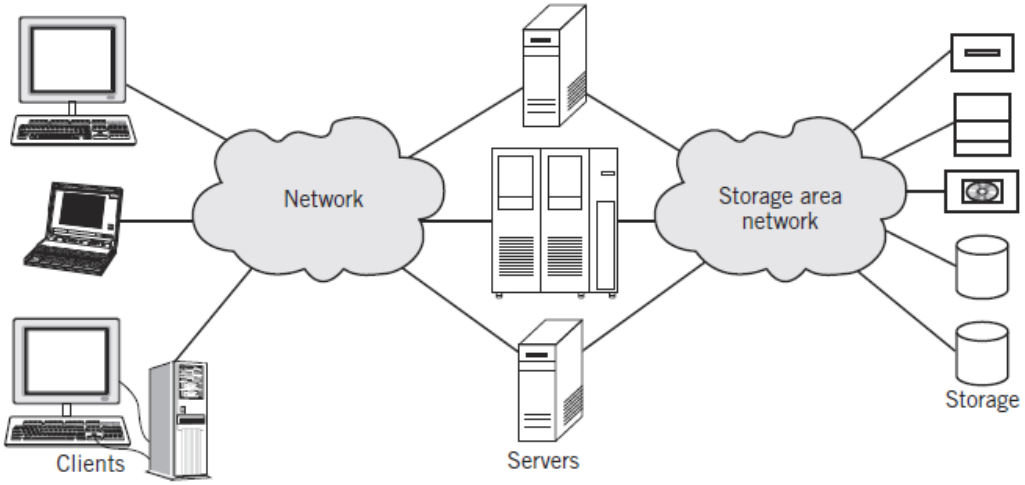




Ejemplo: almacenamiento



a. Standard client server configuration



b. Storage area network configuration

Referencias

- A. Gómez-Pérez, M. Fernández-López, O. Corcho. “Ontological Engineering”. Springer-Verlag, 2003
- Andrew S. Tanenbaum and Maarten Van Steen. Distributed Systems: Principles and Paradigms. Prentice Hall; 2 edition (October 12, 2006). ISBN-13: 978-0132392273
- Irv Englander. The Architecture of Computer Hardware, Systems Software, & Networking: An Information Technology Approach. Wiley; 4 edition (May 4, 2009). ISBN-13: 978-0471715429