ARQUITECTURA DE UN SISTEMA DE INFORMACIÓN

Maestría en Bioinformática – Marzo 2010

Contenidos

- ¿Qué es la Arquitectura de un Sistema?
- Patrones de Arquitectura
- Arquitectura de un Sistema de Información

Arquitectura de una casilla de perro



Puede ser construída por una sola persona
Requiere
- Modelado mínimo
- Procesos simples
- Herramientas simples

Arquitectura de una casa



Para construirla en forma eficiente se precisa un equipo
Requiere
- Modelado
- Procesos bien definidos
- Herramientas especializadas

Definiciones de arquitectura

- El arte o ciencia de la construcción de edificios para uso humano
- La acción o proceso de construir
- El “estilo” en los detalles de la estructura y la ornamentación
- La estructura conceptual y la organización lógica de una computadora o de un sistema basado en computadoras, desde el punto de vista de su uso o diseño; una realización particular de esto.

Oxford English Dictionary, 2nd ed.

Arquitectura de sistemas físicos

- Gobernados por leyes físicas
- Los sistemas físicos maduros tienen arquitecturas estables
  - Aviones, autos y barcos
  - Puentes y edificios
- Dichas arquitecturas surgen de procesos en los que hay experiencia
  - Ensayo y error
  - Reutilización y refinamiento de soluciones probadas
- Resultados predecibles (por suerte ☺)

La arquitectura del software es diferente

- No hay reglas equivalentes a las de la física
  - Es necesario definir reglas
- No es tangible
  - Hay que hacer modelos
- Es compleja
  - Muchos estados!
- Cambios en los requerimientos y la tecnología
  - Adaptabilidad, evolución
- Bajos costos de replicación y distribución

Definiciones de arquitectura (II)

- IEEE 1471-2000 **Booch, Kruchten, Reitman, Bittner, and Shaw**
  - La arquitectura del Software es la organización fundamental del sistema que incluye a sus componentes, sus relaciones entre ellos y el ambiente y los principios que dictan su diseño y evolución
  - Involucra un conjunto de decisiones significativas acerca de la organización del sistema
  - Selección de sus elementos estructurales y sus interfaces
  - Comportamiento, especificado en función de la colaboración de los elementos
  - Composición de sub-sistemas mas grandes a partir de elementos estructurales y elementos con comportamiento

Definiciones de arquitectura (III)

- La arquitectura de software también involucra:
  - Funcionalidad
  - Usabilidad
  - Tolerancia a cambios
  - Performance
  - Reutilización
  - Restricciones económicas y tecnológicas (equilibrio)
  - Aspectos estéticos

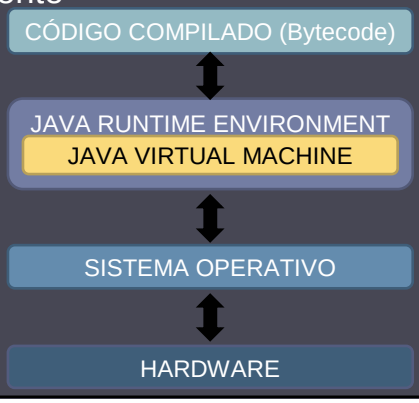
Patrones

- Un patrón es una solución conocida a un problema recurrente
- Un patrón codifica conocimiento específico recogido de la experiencia en un dominio
- Todos los sistemas bien estructurados tienen patrones:
  - EJ: “A language pattern”, C. Alexander, 1977
- Los **patrones de arquitectura** son esquemas base de organización estructural para sistemas de software.
  - Definen algunos subsistemas con sus responsabilidades
  - Incluyen reglas y pautas para organizar las relaciones entre ellos

Pattern-Oriented Software Architecture, Volume 1, A System of Patterns

Algunos patrones de arquitectura (I)

- Arquitectura en capas:
  - Organiza los componentes en grupos de sub tareas, donde cada grupo se encuentra en un nivel de abstracción diferente
- Ejemplo:
  - Virtualización, Java Virtual Machine como caso particular



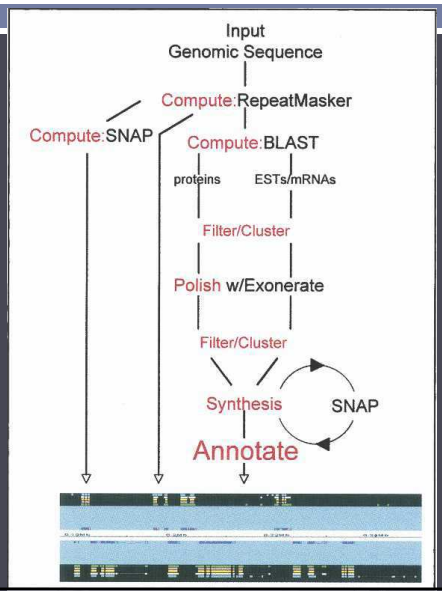
Algunos patrones de arquitectura (II)

- "Pipes and Filters":
  - Sirve en sistemas que procesan un flujo de datos.
  - Cada paso es encapsulado en un "filtro".
  - Los "tubos" pasan datos de un filtro a otro
  - Al recombinar filtros puedo construir familias de sistemas relacionados



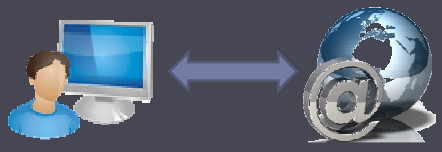
Ejemplo: pipelines de anotación

Extraído de "MAKER: An easy-to-use annotation pipeline designed for emerging model organism genomes",
Cantarel et al., Genome Res. 2008 January; 18(1): 188–196.



Algunos patrones de arquitectura (III)

- Cliente – Servidor:
  - Permite estructurar sistemas distribuidos con componentes desacoplados que interactúan por medio de invocaciones remotas a servicios.
- Ejemplo: navegación web
  - Cliente: navegador, cliente "fino" (poca lógica)
  - Servidor: servidores web



Sistemas cliente-servidor

- El procesamiento se distribuye entre:
  - clientes
  - servidores
- Los procesos **cliente** requieren de recursos provistos por los **servidores**
- Clientes y servidores pueden residir en el mismo host o en hosts diferentes conectados a traves de una red.

Recordemos... subsistemas de un SI

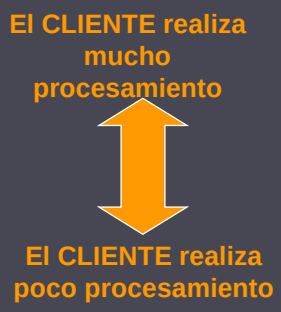
- Presentación o E/S:
  - Responsable del formato y la presentación de los datos en la pantalla del usuario (u otro dispositivo de salida)
  - Gestiona las entradas que hace el usuario desde el teclado (u otro dispositivo)
- Procesamiento:
  - Procesamiento de datos (ej: validación e identificación de errores)
  - Lógica de las reglas de negocio
- Almacenamiento:
  - Responsable del almacenamiento y la recuperación de datos desde los medios físicos de almacenamiento

Usando el patrón cliente-servidor

- Distribuyendo estos sub sistemas mediante el patrón cliente-servidor obtengo varias arquitecturas cliente-servidor para Sistemas de Información

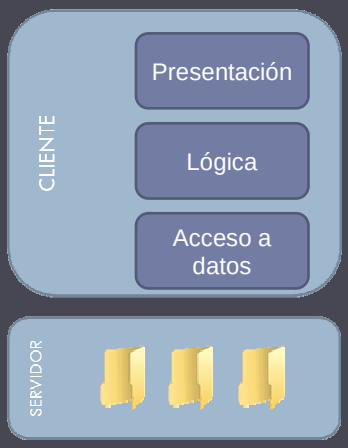
Arquitecturas cliente-servidor

- Servidor de archivos
- Servidor de Base de datos
- Arquitectura en tres niveles



Arquitectura con servidor de archivos (I)

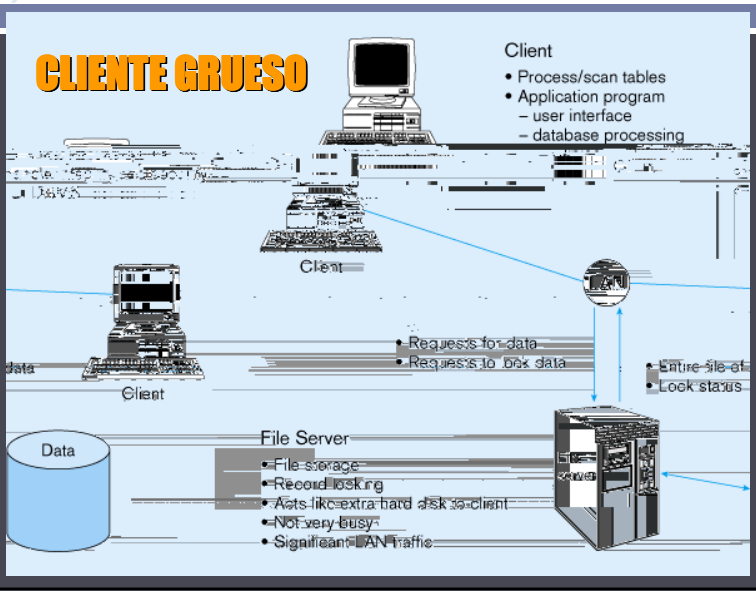
- Todo el procesamiento se realiza en el PC que solicita los datos.
- Un servidor de archivos es un dispositivo que gestiona operaciones sobre archivos y es compartido por cada uno de los PCs clientes en esa red.
- Cliente GRUESO



Algunos problemas en esta arquitectura

- Se transfieren muchos datos.
- Problemas de acceso concurrente a los archivos
- Problemas de integridad

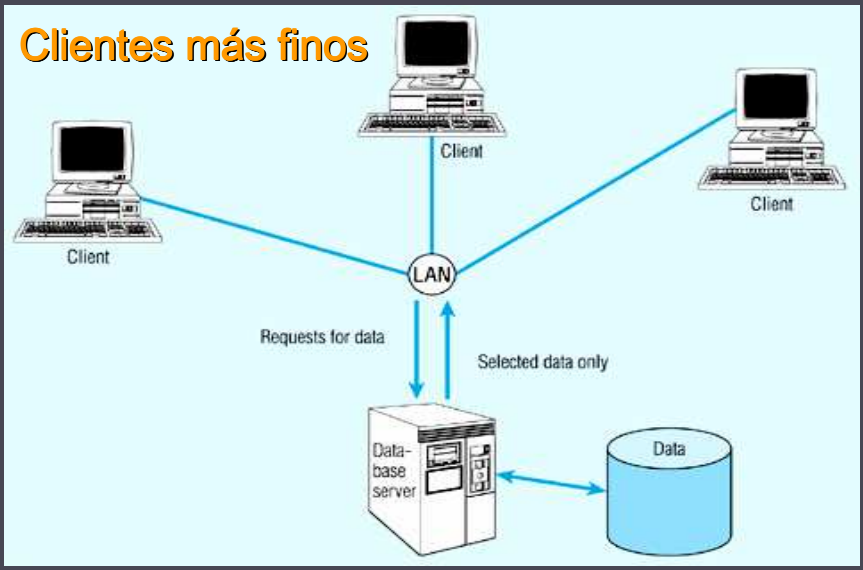
Arquitectura con servidor de archivos (II)



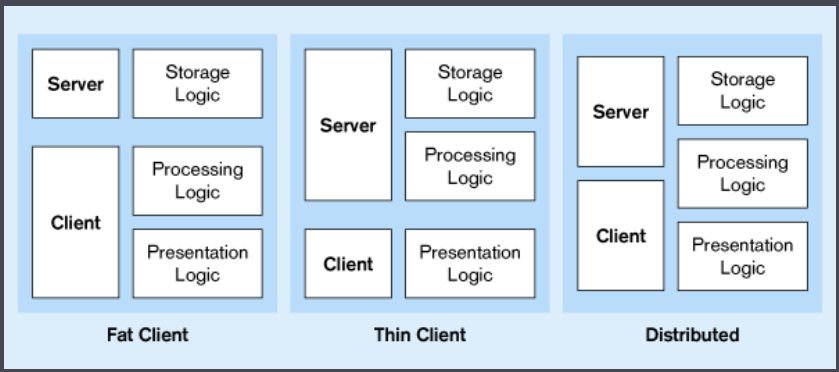
Arquitectura con servidor de bases de datos

- Surgen arquitecturas en 2 niveles
- El cliente es responsable de la interfaz de usuario, el porcesamiento de E/S, el procesamiento de los datos y algo de las reglas de negocio
- El servidor de bases de datos se encarga del almacenamiento y el acceso a los datos
- Ventajas:
  - Los PC clientes pueden ser tener menor capacidad de cálculo y almacenamiento.
  - Se reduce el tráfico de datos en la red, sólo se transfiere los datos solicitados.
  - Se mejora la integridad de los datos debido a la centralización

Arquitectura con servidor de bases de datos



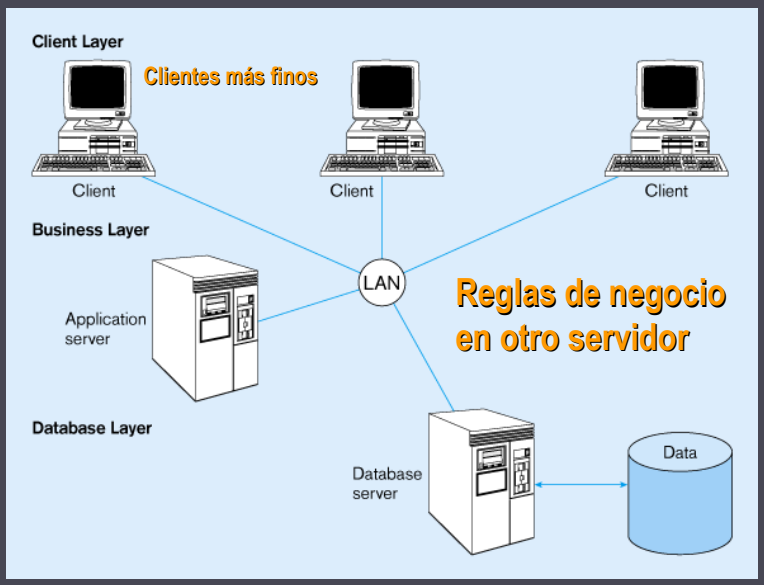
Arquitectura en 2 niveles



Arquitectura en 3 niveles (I)

- Estas incluyen un tercer nivel entre el cliente y el servidor de bases de datos
- Este tercer nivel permite escalar el sistema
- ***Ejemplos:***
  - Un web donde se ubican programas de aplicación
  - Otro servidor de bases de datos que mantiene los datos locales mientras que en otros servidores se guardan los datos de toda la organización
- Por lo general el cliente es FINO, sólo se encarga de la interfaz de usuario y validación de las entradas.

Arquitectura en 3 niveles (II)



Arquitectura en 3 niveles (III)

Middleware

- El software que permite que las aplicaciones interoperen sin que el usuario deba preocuparse de esta interoperabilidad a bajo nivel
- Hay dos modos de comunicación:
  - Sincrónico: el sistema solicitante realiza un pedido y espera una respuesta en tiempo real.
    - EJ: llamada de teléfono
  - Asincrónico: el sistema solicitante realiza un pedido y no espera una respuesta en tiempo real.
    - EJ: envío de mensaje de texto

El pegamento que mantiene a las aplicaciones cliente-servidor funcionando

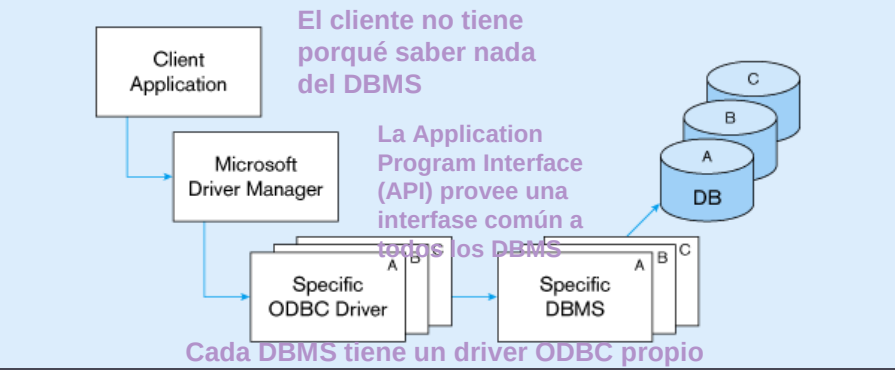
Algunos tipos de middleware

- 1. **Remote Procedure Call (RPC) sincrónico:**
  - se realizan pedidos a procesos o servicios en diferentes hosts y se espera por la respuesta. (EJ RMI en Java)
- 2. **Publicar/suscribir:**
  - los servidores monitorean y envian info a los clientes cuando esta está disponible.
  - Es asincrónico . Es útil para monitorear situaciones donde es necesario tomar acciones ante ciertos eventos.
- 3. **Middleware orientado a mensajes (MOM):**
  - Asincrónico
  - Se envían mensajes y estos se reciben y almacenan hasta que se procesan (EJ: JMS)

Middleware de bases de datos

- ODBC – Open Database Connectivity
  - Es un API que provee un lenguaje común para acceder a bases de datos en forma independiente del servidor de bases de datos particular
- JDBC – Java Database Connectivity
  - Similar a ODBC pero en lenguaje JAVA

Arquitectura ODBC



Client/Server security

- Network environment has complex security issues. Networks susceptible to breaches of security through eavesdropping, unauthorised connections or unauthorised retrieval of packets of information flowing round the network. Specific security issues include:
- System-level password security – user names and passwords for allowing access to the system. Password management utilities
- Database-level password security - for determining access privileges to tables; read/update/insert/delete privileges
- Secure client/server communication - via encryption – but encryption can negatively affect performance

Referencias

- *"Bioinformatics Computing"*, Bryan Bergeron, Pearson Education, 2003. (cap 2. y 3)
- "Fundamentals of Database Systems 5th Ed.", Elmasri & Navathe, Addison Wesley, 2007. (cap. 2)
- *"Pattern-Oriented Software Architecture, Volume 1, A System of Patterns"*, Frank Buschmann et al., Wiley, 1996. (cap. 2)
- *"Software Architecture",* Grady Booch, http://www.booch.com/architecture/blog/artifacts/Software%20Architecture.ppt