**Arquitecturas de los sistemas distribuidos**

La organización de los sistemas distribuidos depende mayormente de los componentes de software que constituyen al sistema. Estas arquitecturas de software establecen como son organizados varios componentes del software y cómo interactúan entre ellos.

La implementación de un sistema distribuido requiere de la división e identificación de los componentes de software y su instalación en máquinas reales. La implementación e instalación final de la arquitectura de software se conoce como *arquitectura de software*.

Existen varias configuraciones de componentes y conectores que definen el estilo arquitectónico de un sistema distribuido. Los estilos más importantes son:

- *Arquitecturas en capas*
- *Arquitecturas basadas en objetos*
- *Arquitecturas centradas en datos*

La idea básica tras el *estilo arquitectónico en capas* es simple: los componentes están organizados en forma de capas, en la que un componente en una determinada capa puede llamar a componentes en la capa inmediata inferior. Una observación clave es que el control generalmente fluye de capa en capa: las peticiones van de arriba abajo y los resultados de abajo a arriba, tal como se puede observar en la **Figura 1(a)**.

Una organización, por mucho más suelta, se tiene en *arquitecturas basadas en objetos*, tal como se muestra en la **Figura 1(b)**. En esencia, cada objeto corresponde a lo que hemos definido como componente, y estos componentes están conectados mediante un mecanismo RPC. No es de sorprender que esta arquitectura de software se adapte al *modelo cliente-servidor* que trataremos más adelante.

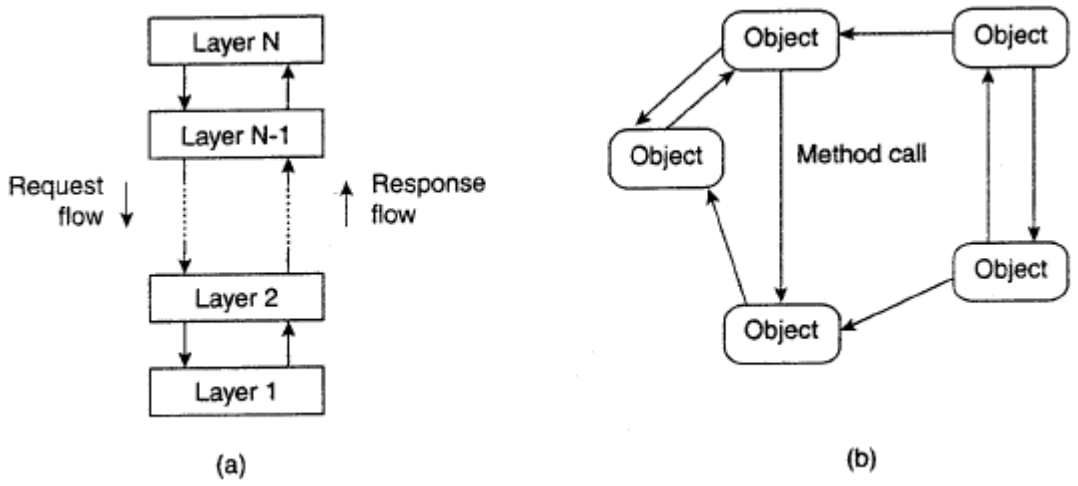



**Figura 1.** (a) estilo arquitectónico en capas; (b) estilo arquitectónico basado en objetos.

Las *arquitecturas centradas en datos* evolucionan en torno a la idea de que los procesos se comunican a través de un repositorio o medio común (ver **Figura 2**). Por ejemplo, una cantidad importante de aplicaciones distribuidas en las que la comunicación se establece por medio de un archivo compartido a través de un sistema de archivos distribuidos.

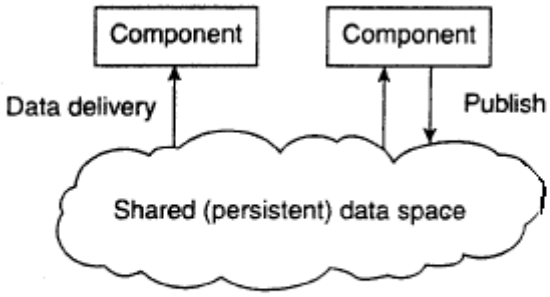



**Figura 2.** arquitectura centrada en datos

**Instancias de arquitecturas**

Ya que se ha discutido brevemente sobre algunos *estilos arquitectónicos* comunes, se verá cómo muchos sistemas distribuidos están organizados, considerando la manera en que sus componentes de software fueron establecidos. El determinar que componentes de software se usarán, cómo interactuarán y cómo se distribuirán es lo que se conoce como una *instancia de arquitectura* también llamada *arquitectura de sistema*.

**Arquitecturas Cliente-Servidor**

A pesar de las diferencias en cuanto a varios aspectos de los sistemas distribuidos, solo hay un aspecto en los que muchos expertos coinciden: pensar en términos de *clientes* que solicitan servicios a *servidores* ayuda a entender y administrar la complejidad de los sistemas distribuidos.

En el *modelo básico cliente-servidor*, los procesos en un sistema distribuido están divididos en dos grupos, que posiblemente se traslapan. Un *servidor* es un proceso que implemente un servicio específico, por ejemplo, un servicio de sistema de archivos distribuido o de base de datos. Un *cliente* es un proceso que solicita un servicio a un servidor, enviándole una petición y subsecuentemente esperando la respuesta del servidor. La interacción *cliente-servidor*, también conocida como solicitud-respuesta, se muestra en la **Figura 3**.

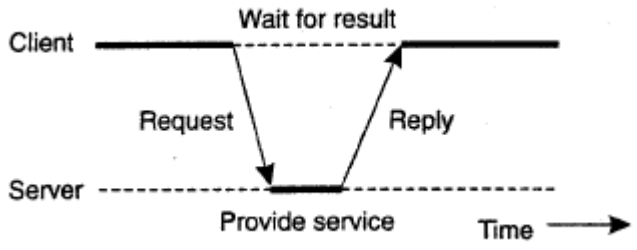



**Figura 3.** Interacción general entre un cliente y un servidor.

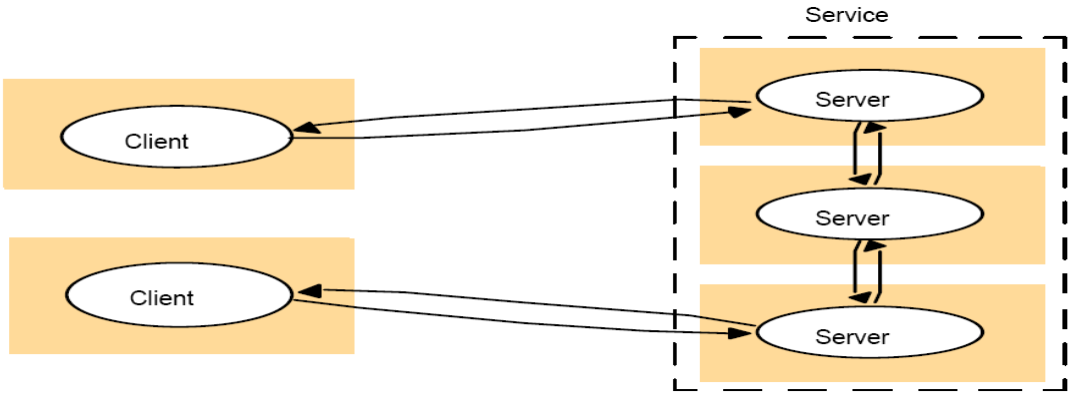
La comunicación entre un cliente y un servidor puede ser implementada por medio de un simple protocolo no orientado a la conexión (sin conexión) cuando la red subyacente es suficientemente confiable como es el caso de muchas redes de área local (LANs). En estos casos, cuando un cliente solicita un servicio, empaca simplemente el mensaje para el servidor, identificando el servicio que requiere y anexando los datos de entrada necesarios. El mensaje es posteriormente enviado al servidor. El servidor se encuentra continuamente en espera de recibir solicitudes, tras lo cual las procesa, empaqueta los resultados en un mensaje de respuesta, y finalmente envía este mensaje al cliente.

Una variante de la arquitectura Cliente-Servidor son los servicios proporcionados por múltiples servidores

Ejemplos: muchos servicios de comercio Web están implementados en diferentes servidores.

La motivación son el desempeño (ej. cnn.com, servidores para descarga, etc.) y la confiabilidad

Los servidores mantienen bases de datos replicadas o distribuidas. La siguiente figura muestra este caso.




***Implementación de aplicaciones en tres capas***

El modelo cliente-servidor ha sido sujeto de muchos debates y controversias a lo largo de los años. Una de las principales cuestiones es el cómo establecer una clara distinción entre un cliente y un servidor. No es de sorprender que en muchas ocasiones esta distinción no es tan clara. Por ejemplo, un servidor de una base de datos distribuida a través de la web puede actuar continuamente como cliente porque éste transfiere las solicitudes a varios servidores de archivos responsables de implementar las tablas de las bases de datos. En este caso, el servidor de base de datos por sí mismo no hace más que procesar las solicitudes de búsqueda o filtrado. La **Figura 4** muestra este caso.

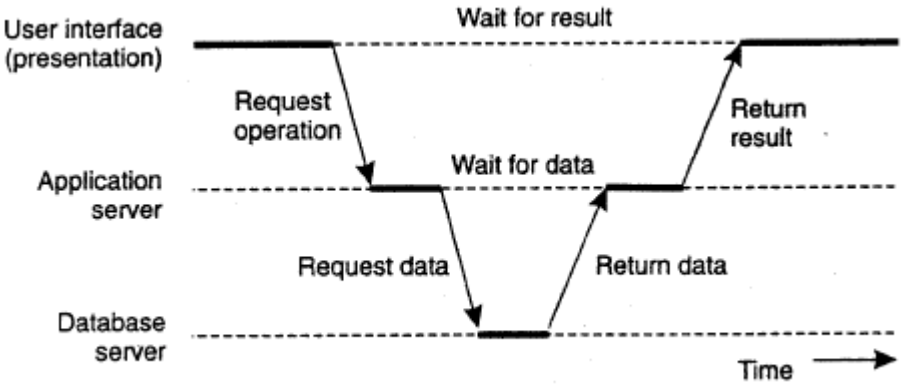



**Figura 4.** Ejemplo de servidor actuando como cliente.

Sin embargo, considerando que muchas aplicaciones cliente-servidor están orientadas a facilitar al usuario el acceso a la base de datos, mucha gente ha establecido una distinción entre los tres niveles siguientes, esencialmente usando el *estilo arquitectónico en capas* que se vio previamente:

1. *El nivel de interfaz de usuario*.
2. *El nivel de procesamiento*.
3. *El nivel de datos*.

El *nivel de interfaz de usuario* contiene todo lo necesario para establecer una interfaz directa con el usuario, tal como la administración del despliegue de la información. El *nivel de procesamiento* típicamente contiene las aplicaciones. El *nivel de datos* administra los datos sobre los cuales se está trabajando.

Los clientes normalmente implementan el *nivel de interfaz de usuario*. Este nivel consiste de los programas que permiten al usuario final interactuar con las aplicaciones. Hay una diferencia considerable en que tan sofisticada puede ser una interfaz de usuario. La más simple no es más que una simple pantalla de caracteres.

Como ejemplo considérese un motor de búsqueda en Internet. La interfaz es muy simple: un usuario introduce una cadena de palabras claves y subsecuentemente se le presenta una lista de títulos de páginas web. El extremo opuesto de la operación está constituido por una gran base de datos de páginas web, las cuales han sido extraídas e indexadas. El núcleo del motor de búsqueda es un programa que transforma la cadena de palabras claves que proporcionó el usuario en una o más peticiones de búsqueda a la base de datos. Subsecuentemente clasifica los resultados en una lista y transforma esta lista en una serie de páginas HTML. Dentro del modelo cliente-servidor, esta parte de extracción de información es típicamente localizada en el *nivel de procesamiento*. La **Figura 5** muestra esta organización.

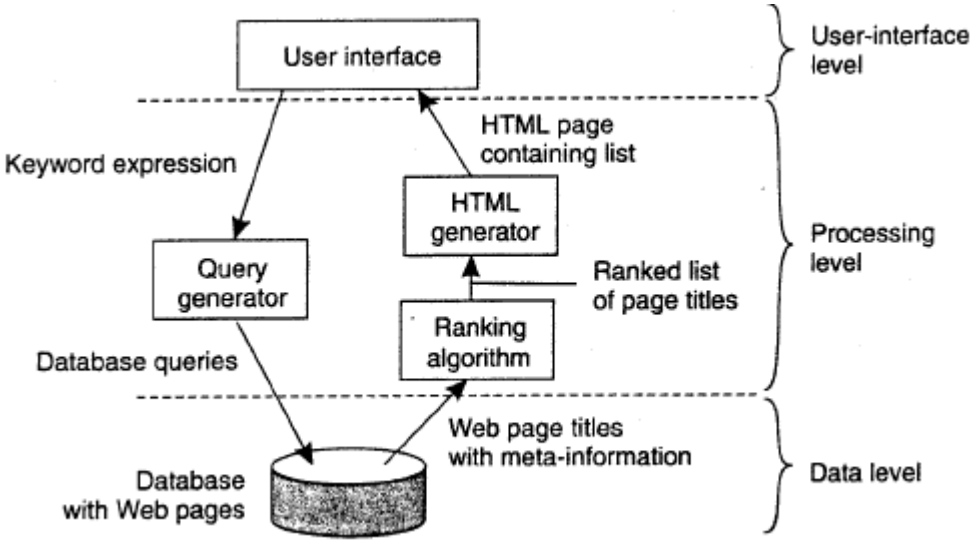



**Figura 5.** Organización simplificada en tres niveles diferentes de un motor de búsqueda.