



Arquitecturas de sistemas distribuidos



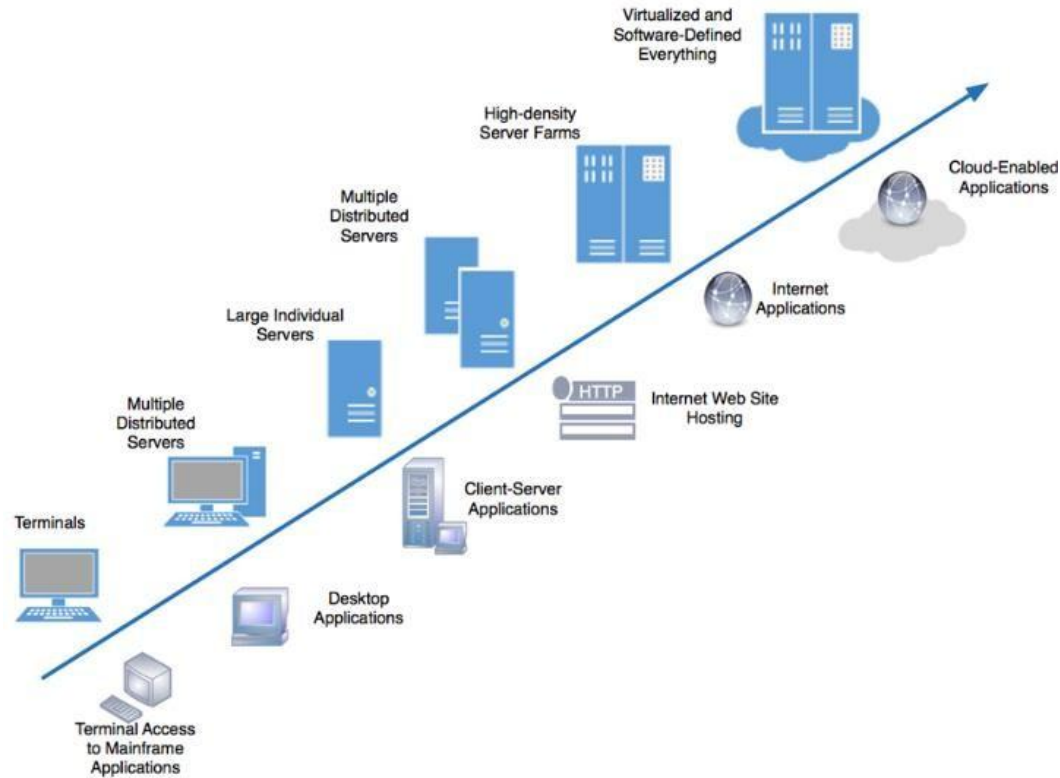
UNIVERSIDAD
NEBRIJA

Índice

- Clasificación general
- Arquitecturas
- Mecanismos de interacción con el middleware
- Autoadministración de arquitecturas



Clasificación general



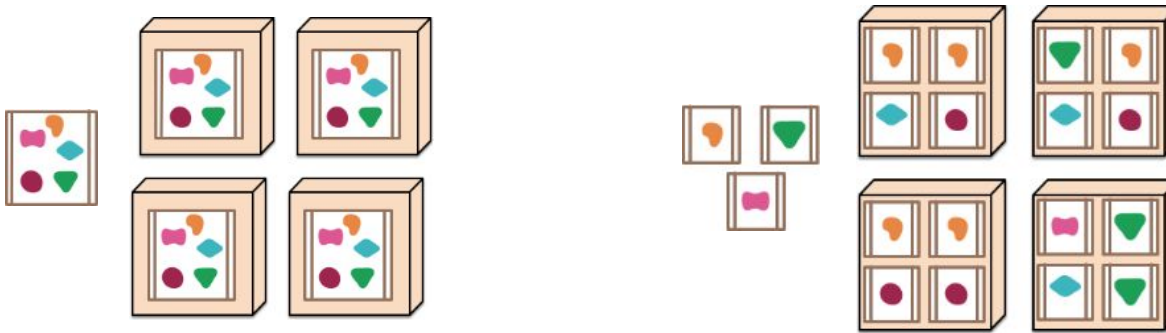
Clasificación general

- **Arquitectura monolítica:** Consiste en desarrollar las aplicaciones software como un único componente que realiza una tarea concreta
 - Un sistema aislado que trabaja de forma autónoma en un entorno que le provee todo lo que necesita para su ejecución, incluidos los datos
 - La aplicación la vemos como un único programa
 - El acceso a los datos es local. Esto quiere decir, que se encuentran en algún recurso (fichero, stream) accesible sin mediar ningún tipo de comunicación
 - Realiza una computación local e independiente
 - Escalabilidad: depende fuertemente de los recursos físicos
 - Fiabilidad: depende íntegramente del programa en sí mismo
 - Seguridad: centrada en la entrada de datos y en la autenticación de usuarios



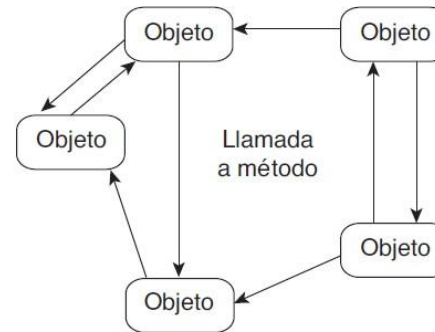
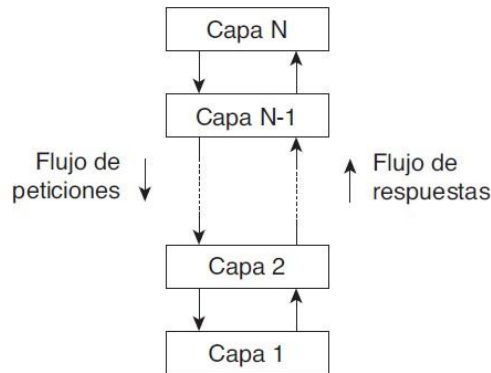
Clasificación general

- **Arquitectura basada en microservicios:** se desarrollan e implementan servicios de manera independiente
 - Mayor flexibilidad al diseñar y escalar el sistema
 - Mayor complejidad al tratarse de un sistema distribuido
 - Aumenta el número de horas de diseño con el fin de planificar la correcta división de los diferentes servicios



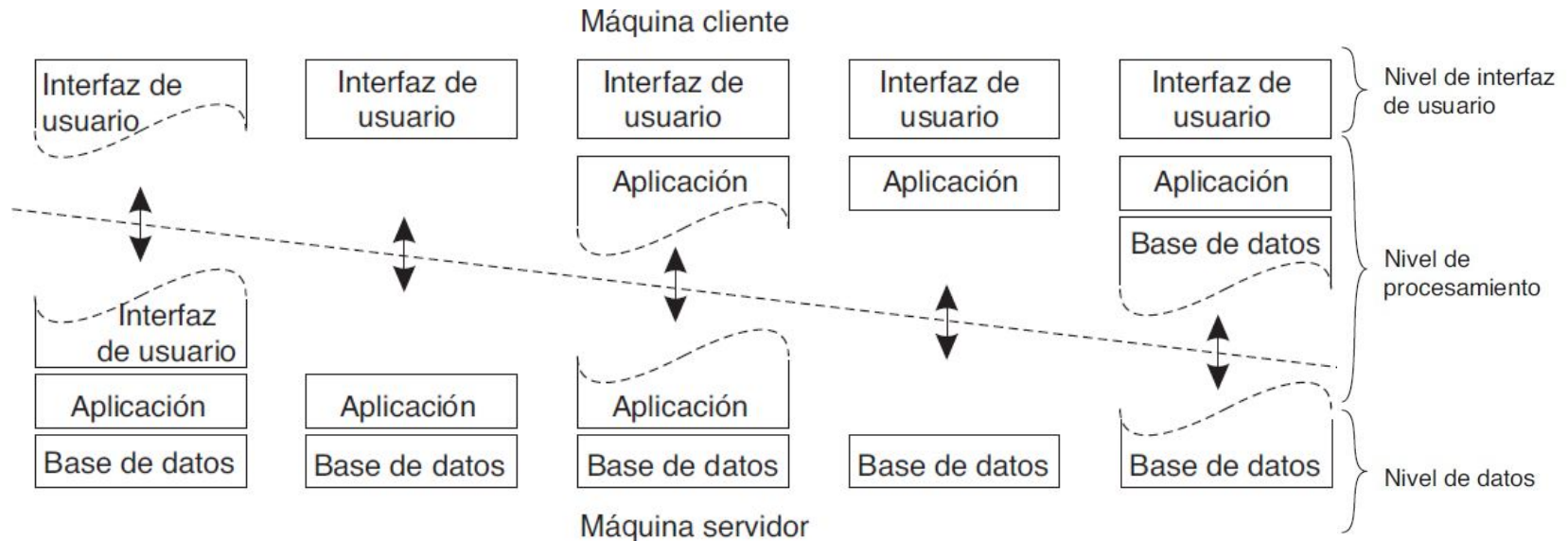
Clasificación general

- **Arquitecturas en capas:** componente de una capa se le permite llamar a componentes de la capa subyacente, pero no del resto de capas
- **Arquitecturas basadas en objetos:** cada objeto corresponde a lo que hemos definido como componente, y estos componentes se conectan a través de un mecanismo de llamadas a procedimientos (remotos)



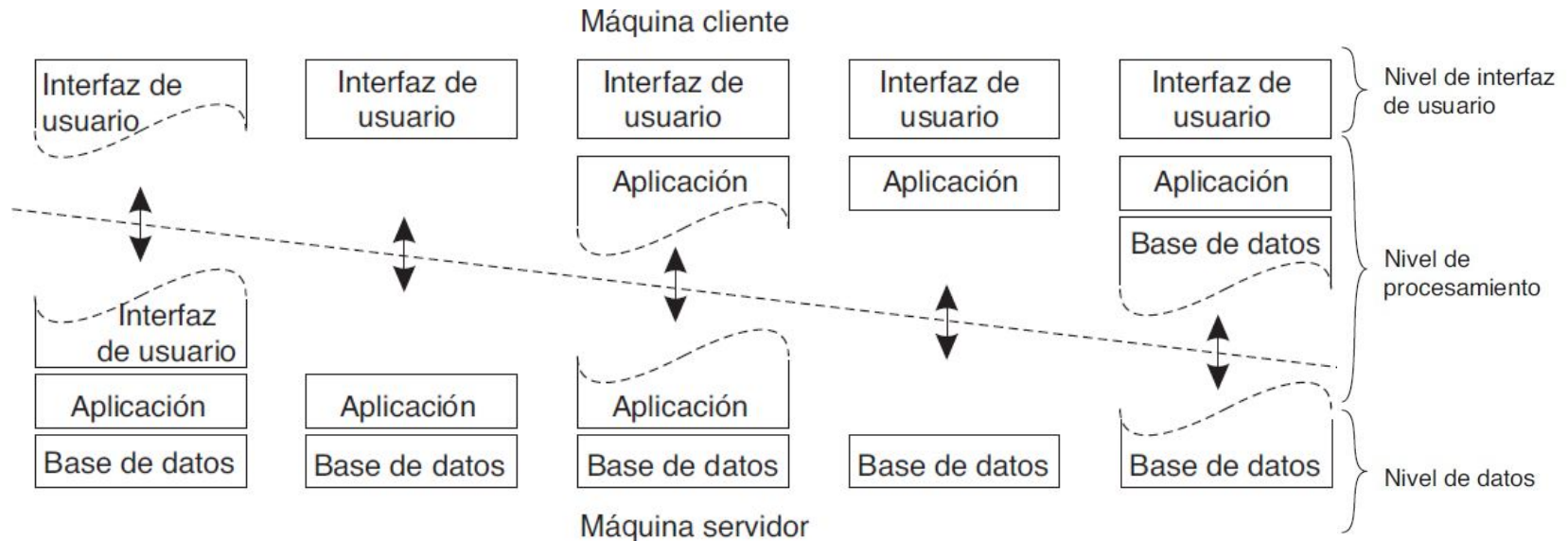
Clasificación general

- Arquitecturas en capas



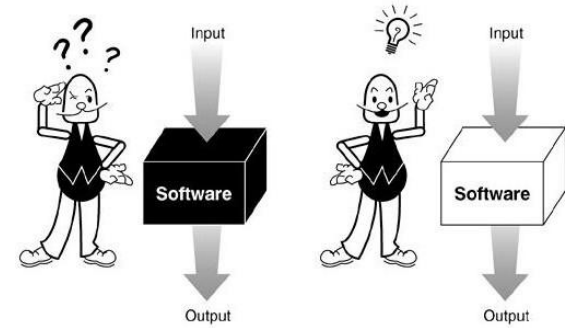
Clasificación general

- Arquitecturas en capas



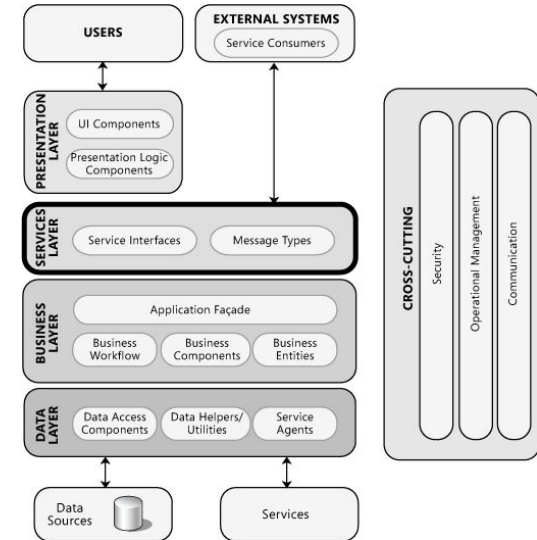
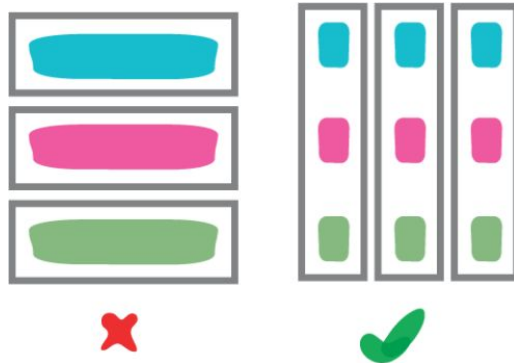
Clasificación general

- Arquitecturas en capas
- Tipos de arquitecturas por capas:
 - **Arquitectura de capas estricta:** Cada capa únicamente puede comunicarse con la directamente inferior en la jerarquía
 - **Arquitectura de capas no estricta:** Cada capa puede comunicarse con capas no adyacentes
 - **Arquitectura de capas black-box:** Cada capa es un sistema con entradas y salidas, no se conoce su implementación interna. Sólo se puede realizar la comunicación mediante su interfaz
 - **Arquitectura de capas white-box:** Entre las capas encapsuladas (black-boxes) existen componentes software cuyo código e implementación interna se conoce y puede modificarse



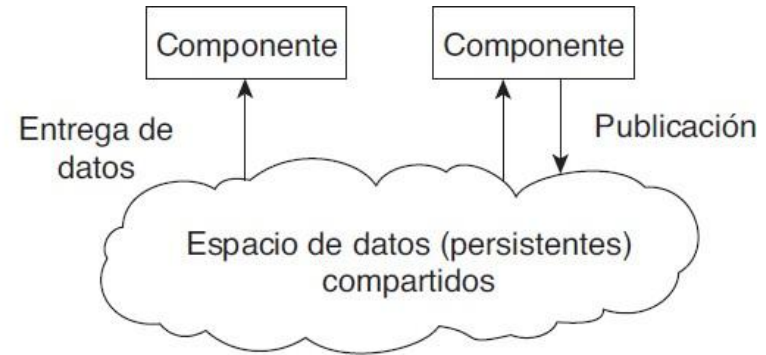
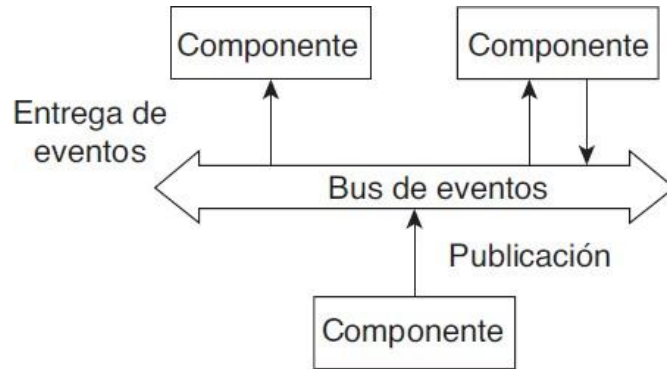
Clasificación general

- **Arquitecturas por niveles:** Varias capas pueden estar lógicamente relacionadas, ya que se encargan de diversos aspectos de una misma funcionalidad de la aplicación. Lo usual en estos casos es agrupar esas capas en un nivel



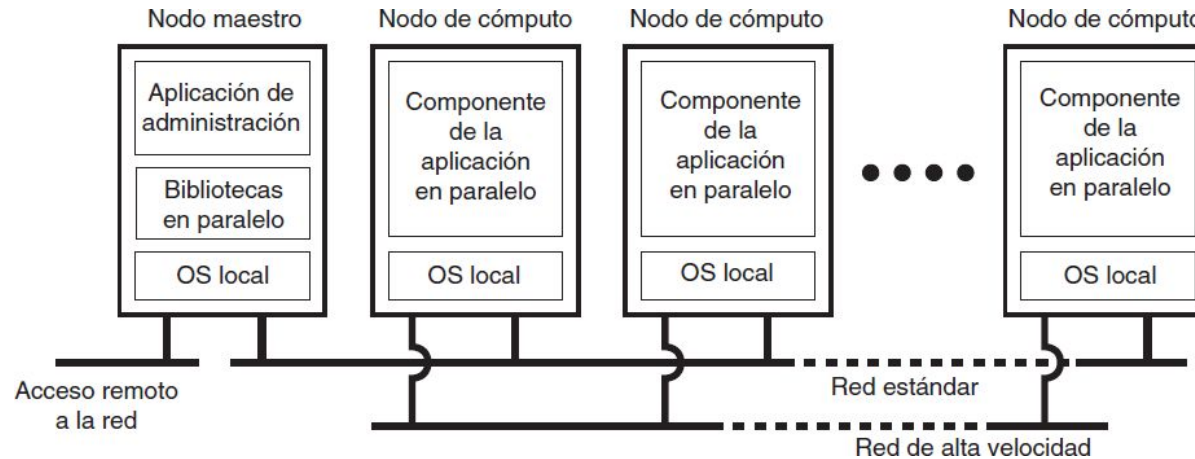
Clasificación general

- **Arquitecturas centradas en datos:** se basan en un sistema de archivos distribuidos compartidos donde casi todas las comunicaciones se realizan a través de archivos
- **Arquitecturas basadas en eventos:** se comunican a través de la propagación de eventos, los que opcionalmente transportan datos (publicación-suscripción)



Arquitecturas

- **Clúster:** se utiliza para la programación en paralelo donde un solo programa (de cálculo intensivo) corre paralelamente en múltiples máquinas



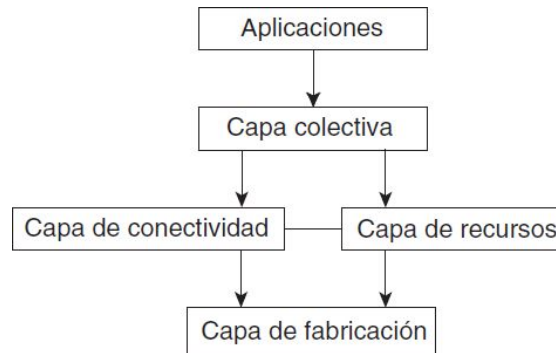
Arquitecturas

- **Clúster:** se utiliza para la programación en paralelo donde un solo programa (de cálculo intensivo) corre paralelamente en múltiples máquinas
 - El nodo maestro ejecuta el middleware necesario para la ejecución de programas y la administración del clúster
 - El middleware está formada por las bibliotecas necesarias para ejecutar programas paralelos
 - Los nodos de cómputo no requieren nada más que un sistema operativo
 - Una característica del cómputo basado en cluster es su homogeneidad



Arquitecturas

- **Grid:** se utiliza para reunir los recursos de diferentes organizaciones para permitir la colaboración de un grupo de personas o instituciones
- Tiene un alto grado de heterogeneidad: no se hacen suposiciones de ninguna índole con respecto al hardware, sistemas operativos, redes, dominios, políticas de seguridad, etc.



Arquitecturas

- **Grid:** se utiliza para reunir los recursos de diferentes organizaciones para permitir la colaboración de un grupo de personas o instituciones
 - *Capa de fabricación:* proporciona interfaces para recursos locales ubicados en un sitio específico
 - Funciones para consultar el estado y las capacidades de un recurso
 - Funciones para la administración real de un recurso (bloqueo/concurrencia)



Arquitecturas

- **Grid:** se utiliza para reunir los recursos de diferentes organizaciones para permitir la colaboración de un grupo de personas o instituciones
 - *Capa de conectividad:* protocolos de comunicación para dar soporte a las transacciones del grid que abarcan el uso de múltiples recursos
 - Contiene protocolos de seguridad
 - *Capa de recursos:* utiliza las funciones proporcionadas por la capa de conectividad y llama de manera directa a las interfaces puestas a disposición mediante la capa de fabricación
 - Responsable del control de acceso con el apoyo de la capa de conectividad



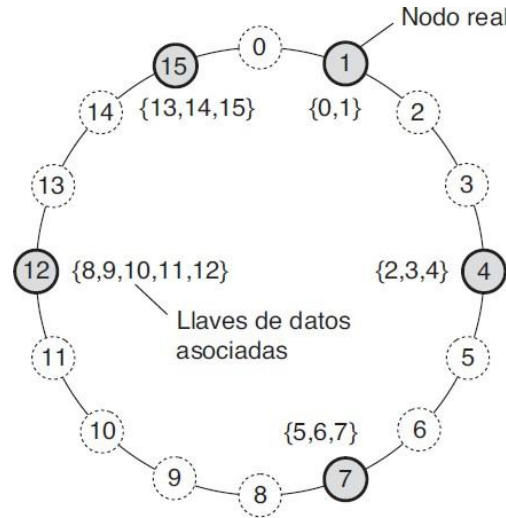
Arquitecturas

- **Grid:** se utiliza para reunir los recursos de diferentes organizaciones para permitir la colaboración de un grupo de personas o instituciones
 - *Capa colectiva:* consta de servicios para descubrir recursos, ubicación y calendarización de tareas dentro de múltiples recursos, replicación de datos, etc
 - Consta de una colección estándar de protocolos relativamente pequeña para muchos propósitos diferentes



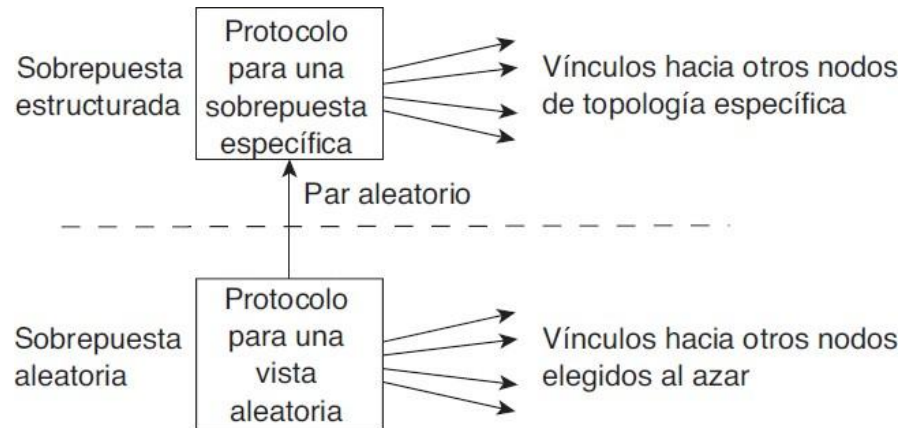
Arquitecturas

- **Sobrepuestas estructuradas:** se construye mediante un procedimiento determinista
 - Tabla hash distribuida (DHT): procedimiento para organizar los nodos



Arquitecturas

- **Sobrepuestas no estructuradas:** cada nodo mantiene una lista de vecinos, pero que esta lista se construye de manera más o menos aleatoria



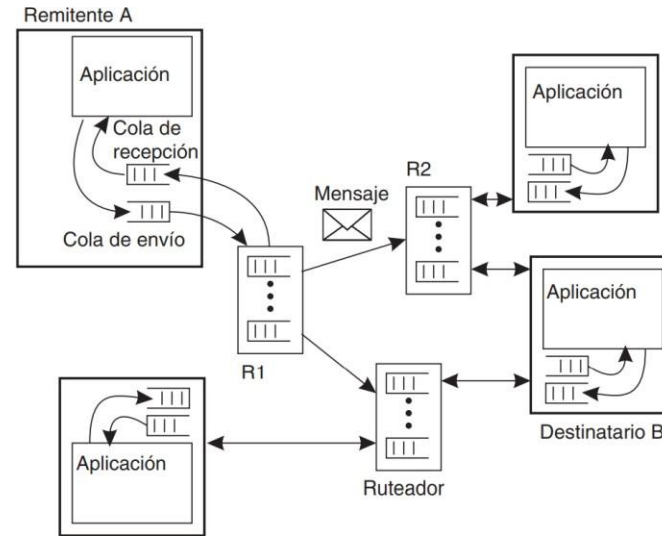
Arquitecturas

- **Sistema de colas de mensajes:** proporciona colas a los remitentes y destinatarios, y se encarga de que los mensajes se transfieran desde su cola fuente hasta su cola destino
 - La colección de colas se distribuye a través de diversas máquinas
 - Es necesario que se mantenga un mapa de las colas para localización en la red
 - Se debe mantener una base de datos distribuida de los nombres de las colas para localizarlas en la red
- **Hilo administrador de colas:** interactúa directamente con la aplicación que está enviando o recibiendo un mensaje
- **Hilos retransmisores:** retransmiten mensajes entrantes hacia otros administradores de colas (útiles por si algún nodo no tiene servicio DNS)
 - Se implementan en sistemas de seguridad y tolerancia a fallos



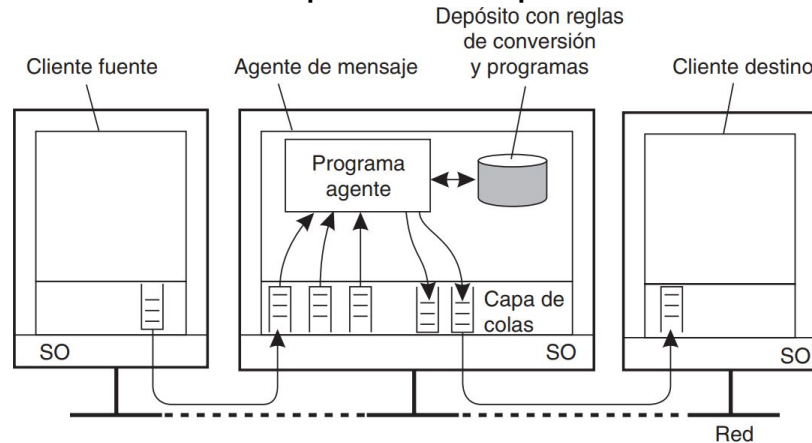
Arquitecturas

- **Sistema de colas de mensajes:** proporciona colas a los remitentes y destinatarios, y se encarga de que los mensajes se transfieran desde su cola fuente hasta su cola destino



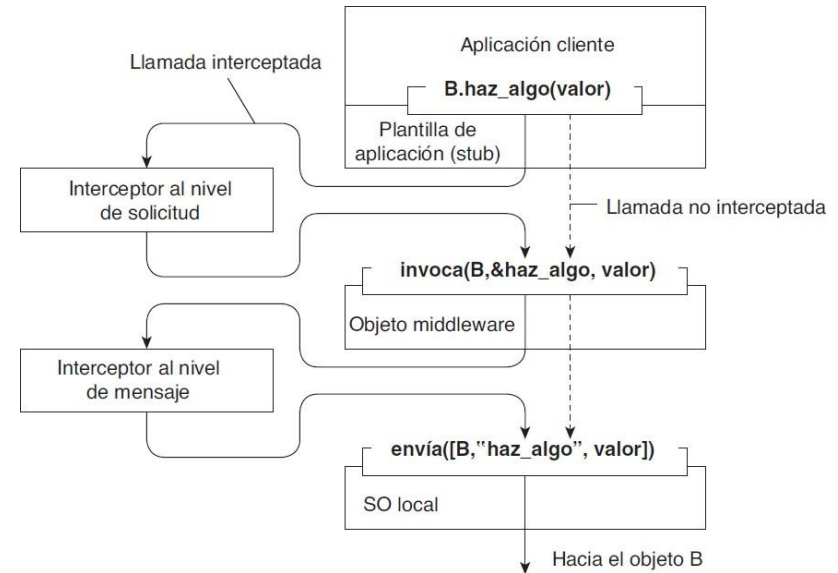
Arquitecturas

- **Sistema de colas de mensajes:** proporciona colas a los remitentes y destinatarios, y se encarga de que los mensajes se transfieran desde su cola fuente hasta su cola destino
 - **Agente de mensaje:** convierte los mensajes entrantes de tal manera que la aplicación de destino pueda comprenderlos



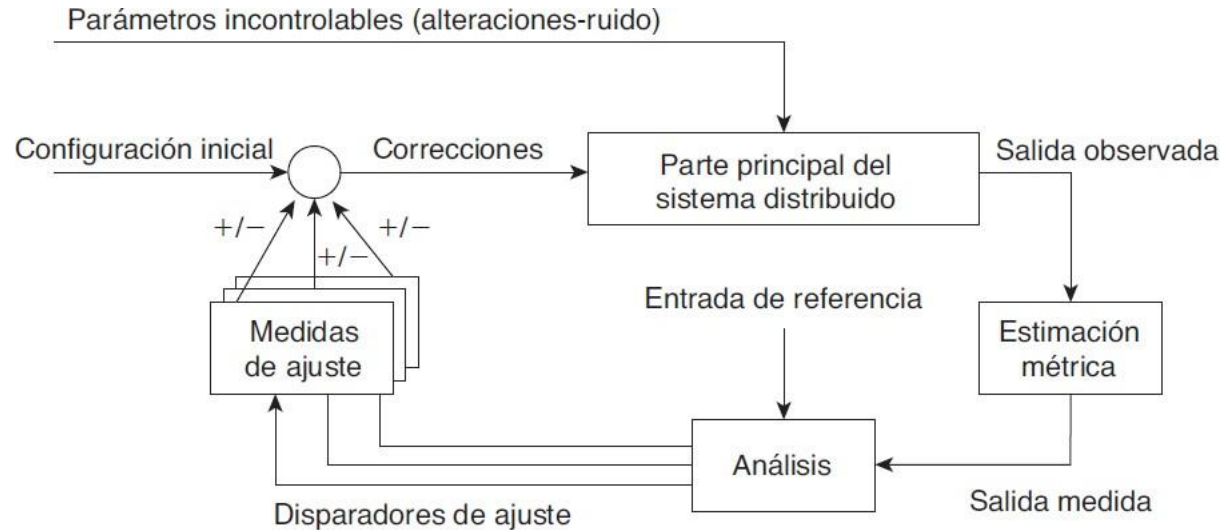
Mecanismos de interacción con el middleware

- **Interceptor:** software que rompe el flujo usual de control y permite que otra aplicación específica se ejecute para adaptar el middleware al entorno cambiante
 - Intercepción basada en objetos
 - Interceptor al nivel de petición
 - Interceptor al nivel de mensajes



Autoadministración de arquitecturas

- Sistemas de control de retroalimentación



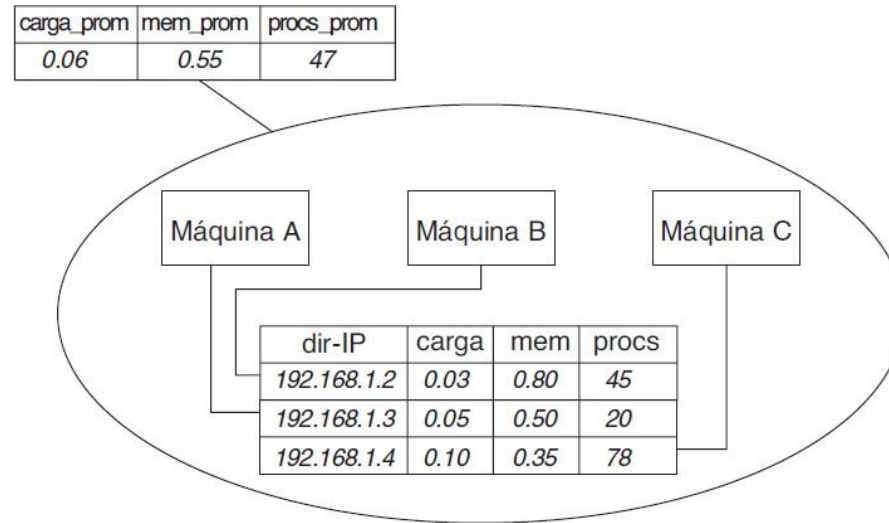
Autoadministración de arquitecturas

- **Sistemas de control de retroalimentación**
 - Astrolabe: monitoreo general de sistemas distribuidos muy grandes
 - Cada servidor:
 - Ejecuta un proceso Astrolabe, llamado agente, que colecciona información sobre las zonas en las que se encuentra ese servidor
 - El agente se comunica también con otros agentes con la intención de esparcir información (gossiping) de la zona a través de todo el sistema
 - Mantiene un conjunto de atributos para recopilar información local



Autoadministración de arquitecturas

- **Sistemas de control de retroalimentación**
 - Astrolabe: monitoreo general de sistemas distribuidos muy grandes



Autoadministración de arquitecturas

- **Sistemas de control de retroalimentación**
 - Globule cada servidor de origen rastrea patrones de acceso por página.
 - Los patrones de acceso son operaciones de lectura y escritura para una página
 - Se registra cada operación y la marca con un tiempo
 - Se registra la dirección IP
 - Con las mediciones de tiempos y número de peticiones se pronostica la ruta óptima del sistema distribuido

