

Arquitecturas de los Sistemas Distribuidos

Diego Alberto Rincón Yáñez MSc

drincony@poligran.edu.co

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO
GRANCOLOMBIANO

Facultad de Ingeniería

Agenda

Definiciones

Ventajas

Desventajas

Arquitecturas

Cliente/Servidor

Objetos Distribuidos

Orientado a Servicios

Multiprocesador

Peer to Peer



Definiciones Generales

Una arquitectura dentro del mundo de la tecnología está orientada a encontrar patrones de agrupación o configuración de componentes, estas arquitecturas generalmente están dictan parámetros necesarios para brindar una solución, ya sea por medio de un listado de requisitos o también en casos más específicos apoyarse en arquitecturas de referencia(Bass L & P, 2010).

Definiciones - Propósitos

Intercambio de información de los componentes:

Información necesaria relacionada con los datos de la aplicación, resultados de bases de datos, llamado a métodos de los servicios remotos.

Coordinación y articulación de componentes: cada una de las arquitecturas presentadas en esta unidad tienen mecanismos de sincronización, son muy necesarios, debido a que generalmente los componentes existen en contextos de memoria diferentes y no pueden comunicarse fácilmente.

Definiciones - Propósitos

Protocolos de comunicaciones: Tal y como se expresó en la primera unidad de este curso los protocolos de comunicaciones de red son la base fundamental para el trabajo de diferentes computadores en red, el cual se convierte en la base habilitadora del desarrollo de un sistema distribuido.

Tipos de Transmisión

Transmisión Sincrónica: Hace referencia al modo de transmisión el cual tanto emisor como receptor intercambian mensajes intercaladamente, esto quiere decir que una vez se inicia la conexión el emisor envía el mensaje y el receptor acusa el recibo y de la misma manera el receptor envía el mensaje y el emisor confirma su recibido, esto sucede hasta que alguno de ellos finaliza la conversación según el funcionamiento del protocolo específico.

Tipos de Transmisión

Transmisión Asincrónica: A diferencia del anterior en el que ambos participantes de la comunicación deben estarse respondiendo para que el protocolo pueda seguir su curso, no se necesitan confirmaciones de recepción ni acuerdos en el sostenimiento de la comunicación, esto quiere decir que en cualquier momento sin aviso previo cualquier participante puede abandonar la conversación.

Ventajas de las Arquitecturas Distribuidas

Compartir recursos. Un sistema distribuido permite compartir recursos hardware y software (discos, impresoras, ficheros y compiladores) que se asocian con computadoras de una red.

Apertura. Son normalmente sistemas abiertos: se diseñan sobre protocolos estándares que permiten combinar equipamiento y software de diferentes vendedores.

Concurrencia. Varios procesos pueden operar al mismo tiempo sobre diferentes computadoras de la red. Hasta pueden comunicarse con otros durante su funcionamiento.

Ventajas de las Arquitecturas Distribuidas

Escalabilidad. Los sistemas distribuidos son escalables mientras la capacidad del sistema pueda incrementarse, añadiendo nuevos recursos para cubrir nuevas demandas sobre el sistema.

Tolerancia a defectos. Contar con varias computadoras y el potencial para reproducir información significa que los sistemas distribuidos pueden ser tolerantes a algunas fallas de funcionamiento del hardware y del software.

Desventajas de las Arquitecturas Distribuidas

Complejidad: Los sistemas distribuidos son más complejos que los sistemas centralizados; lo que hace más difícil comprender sus propiedades emergentes y probar estos sistemas.

Por ejemplo, en vez de que el rendimiento del sistema dependa de la velocidad de ejecución de un procesador, depende del ancho de banda y de la velocidad de los procesadores de la red.

Mover los recursos de una parte del sistema a otra puede afectar de forma radical al rendimiento del sistema.

Desventajas de las Arquitecturas Distribuidas

Seguridad: Puede accederse al sistema desde varias computadoras diferentes, y el tráfico en la red puede estar sujeto a escuchas indeseadas.

Es más difícil mantener la integridad de los datos en el sistema y que los servicios del sistema no se degraden por ataques.

Manejabilidad: Las computadoras en un sistema pueden ser de diferentes tipos y ejecutar versiones diferentes de sistemas operativos.

Los defectos en una máquina pueden propagarse a otras, con consecuencias inesperadas.

Esto significa que se requiere más esfuerzo para gestionar y mantener el funcionamiento del sistema.

Desventajas de las Arquitecturas Distribuidas

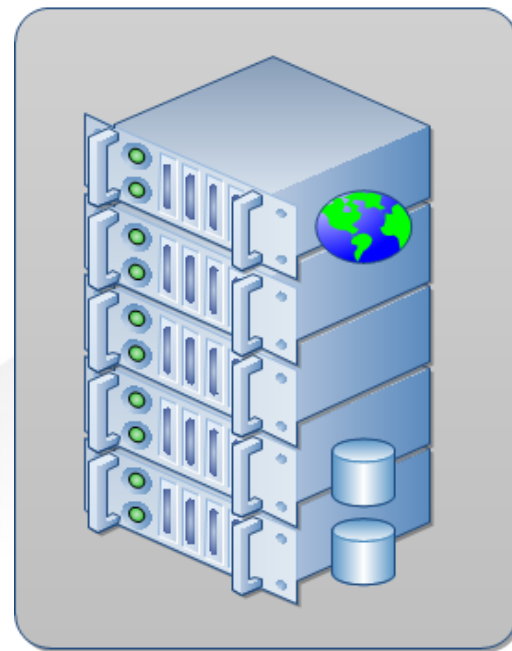
Impredecibilidad: Los sistemas distribuidos tienen una respuesta impredecible.

La respuesta depende de la carga total en el sistema, de su organización y de la carga de la red.

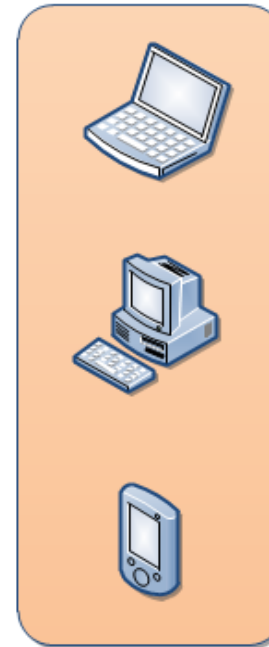
Como todos ellos pueden cambiar rápidamente, el tiempo requerido para responder a una petición de usuario puede variar drásticamente, de una petición a otra.

Arquitectura Cliente - Servidor

Un servidor es el que proporciona los o el servicio a los clientes, estos clientes generalmente están conectados a una red la cual tiene conexión o visibilidad con los servidores para poder utilizar el servicio que estos prestan.



Lado Servidor



Lado Cliente



Arquitectura Cliente - Servidor

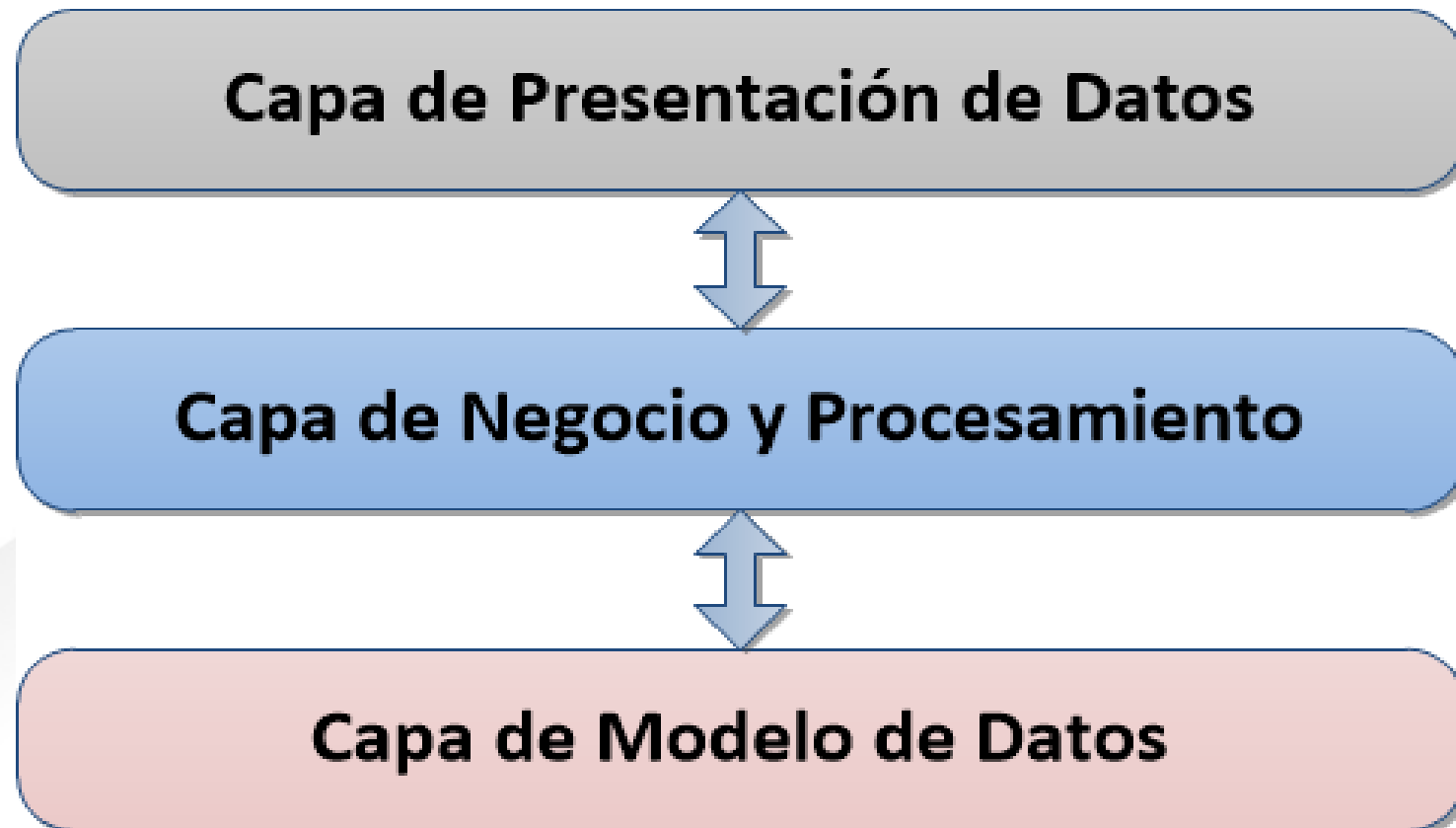
Servidor: Conjunto de procesos, componentes de software, y componentes de hardware que en conjunto se presenta como una sola entidad y presta un servicio

Cliente: Conjunto de módulos, librerías programas, interfaces y dispositivos los cuales se conectan al servicio prestado por el servidor.

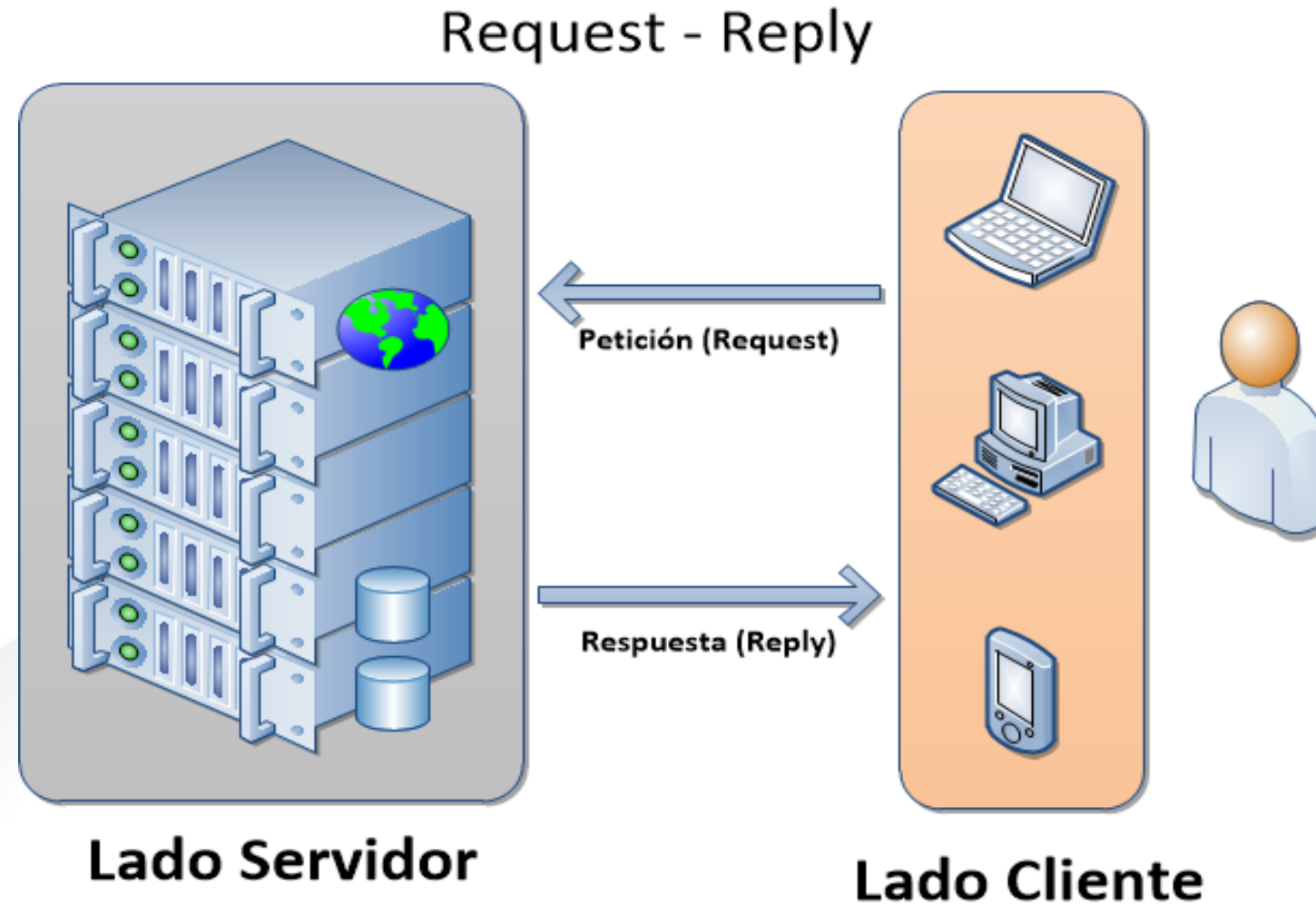
Este tipo de arquitectura, recalca que cada uno de estos procesos, cliente y servidor, son diferentes y deben correr en instancias diferentes.

Arquitectura Cliente - Servidor

Arquitectura de 3 Capas

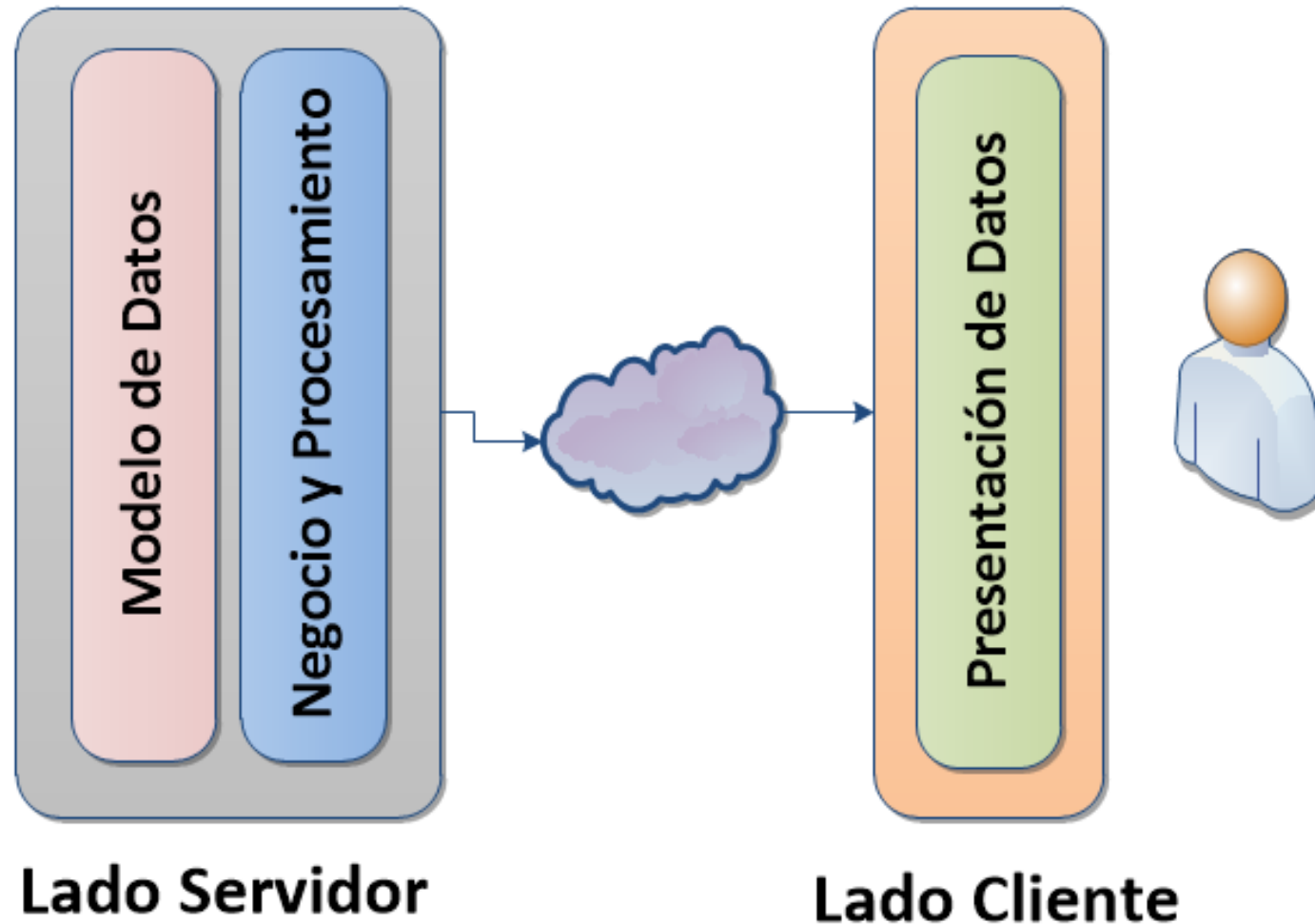


Arquitectura Cliente - Servidor



Arquitectura Cliente – Servidor

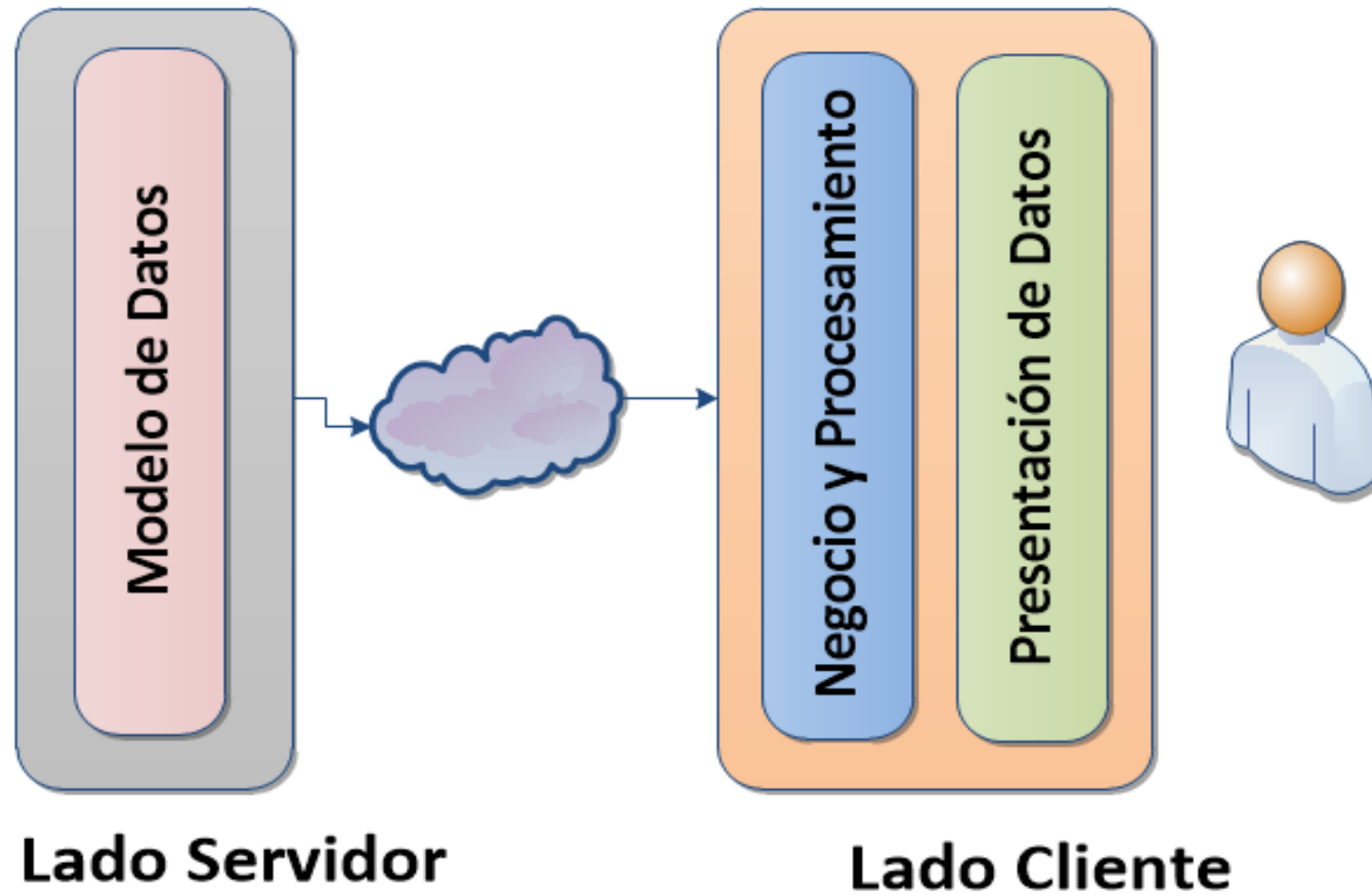
Modelo de Cliente Ligerito/Delgado/Pobre



Sitios Web
Sistemas Legacy
Estaciones “Brutas”
Terminales de Acceso
Aplicaciones Móviles “Ligth”

Arquitectura Cliente – Servidor

Modelo de Cliente Pesado/Gordo/Rico



Aplicaciones
Móviles

Cajeros
Automáticos

Usadas dentro de
Redes de Área
Local (LAN)

Sistemas
accesados por una
VPN.

Arquitecturas de Objetos Distribuidos

El modelo **Cliente – Servidor** funciona bien para muchos tipos de aplicaciones.

Sin embargo, limita la flexibilidad del diseñador, que debe decidir dónde se proporciona cada servicio.

También debe planificar la escalabilidad y proporcionar algún medio para distribuir la carga sobre los servidores, cuando más clientes se añadan al sistema.

Arquitectura Objetos Distribuidos

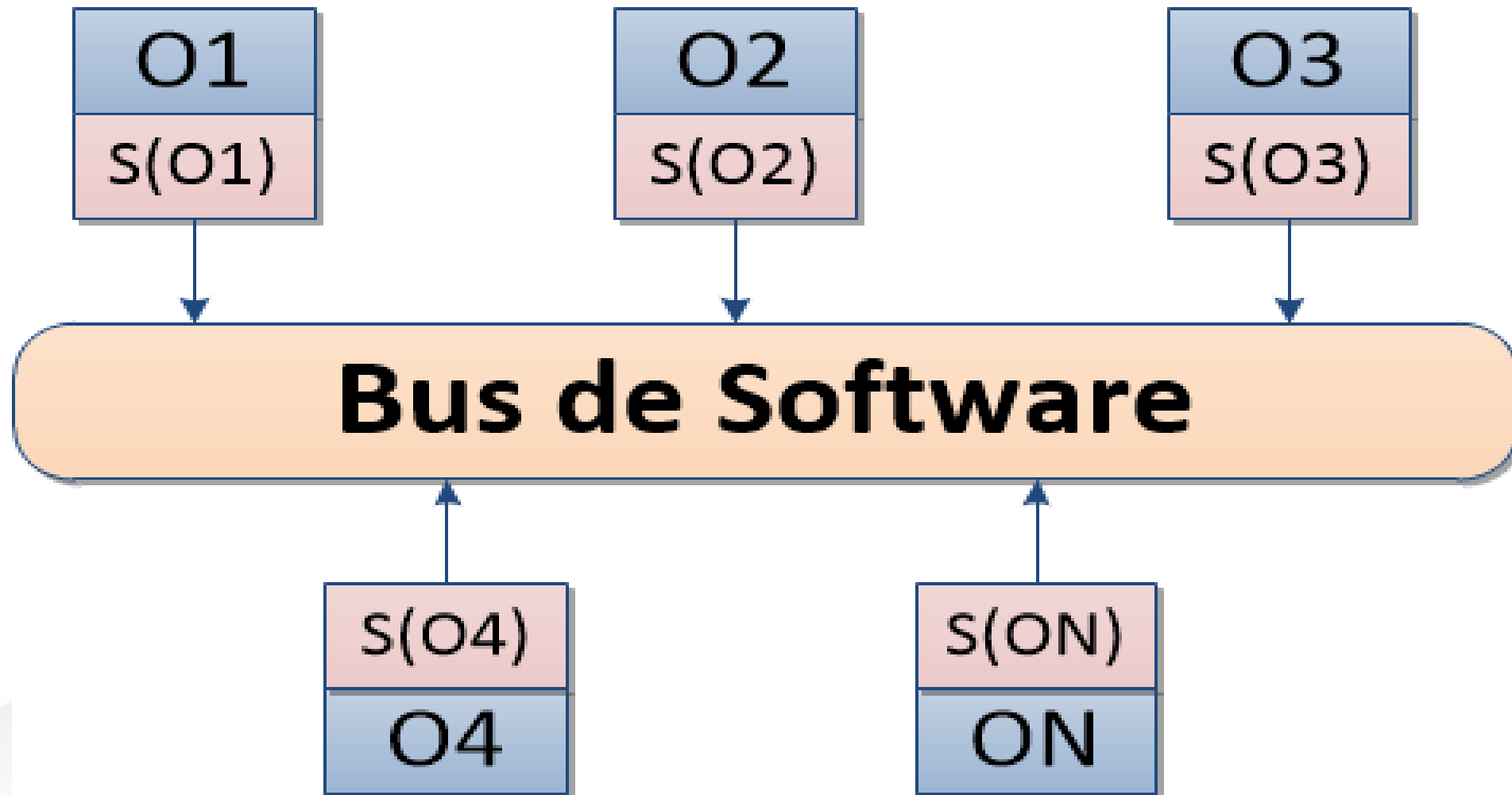
Modifican el paradigma de la definición de responsabilidades y permite que tanto clientes como servidores intercambien servicios, esto quiere decir, que un cliente puede consumir servicios de un servidor y ese mismo servidor, volverse cliente para poder consumir servicios del cliente que a su vez tiene la posibilidad de prestar servicios.

Arquitectura Objetos Distribuidos

En este caso y debido a que no hay una distinción entre el concepto de cliente y de servidor, las entidades que existen en esta arquitectura se conocen como Objetos. Los objetos realizan llamadas a los servicios sin tener una frontera entre lo que es un cliente y un servidor, en el caso de esta arquitectura se conocen como:

- Receptor de un Servicio
- Proveedor de un Servicio

Arquitectura Objetos Distribuidos



Arquitectura Objetos Distribuidos

El middleware se convierte entonces para la arquitectura de Objetos Distribuidos en una interfaz, la cual le garantiza transparencia de localización y de comunicación a los Objetos que componen los servicios o funcionalidades del sistema (Coulson & Baichoo, n.d.)

Arquitectura Orientada a Servicios

Noción de Servicio Web: Por ellos, las organizaciones que quieren hacer accesible su información a otros programas, **definen y publican una interfaz de servicio web.**

Esta interfaz define los datos disponibles y cómo se puede acceder a ellos.

Un **Servicio Web** es una **representación estándar para cualquier recurso computacional o de información que pueda ser usado por otros programas.**

La esencia de un servicio, por lo tanto, es que **la provisión de servicio es independiente de la aplicación que utiliza el servicio.**

Arquitectura Orientada a Servicios

Los proveedores de servicios pueden desarrollar servicios especializados y ofertarlos a un cierto número de usuarios de servicios de diferentes organizaciones.

Existen varios modelos de servicios, aunque todos ellos operan de acuerdo al modelo de la Figura siguiente.

Un proveedor de servicio oferta un servicio definiendo su interfaz y definiendo la funcionalidad del servicio.

Arquitectura Orientada a Servicios

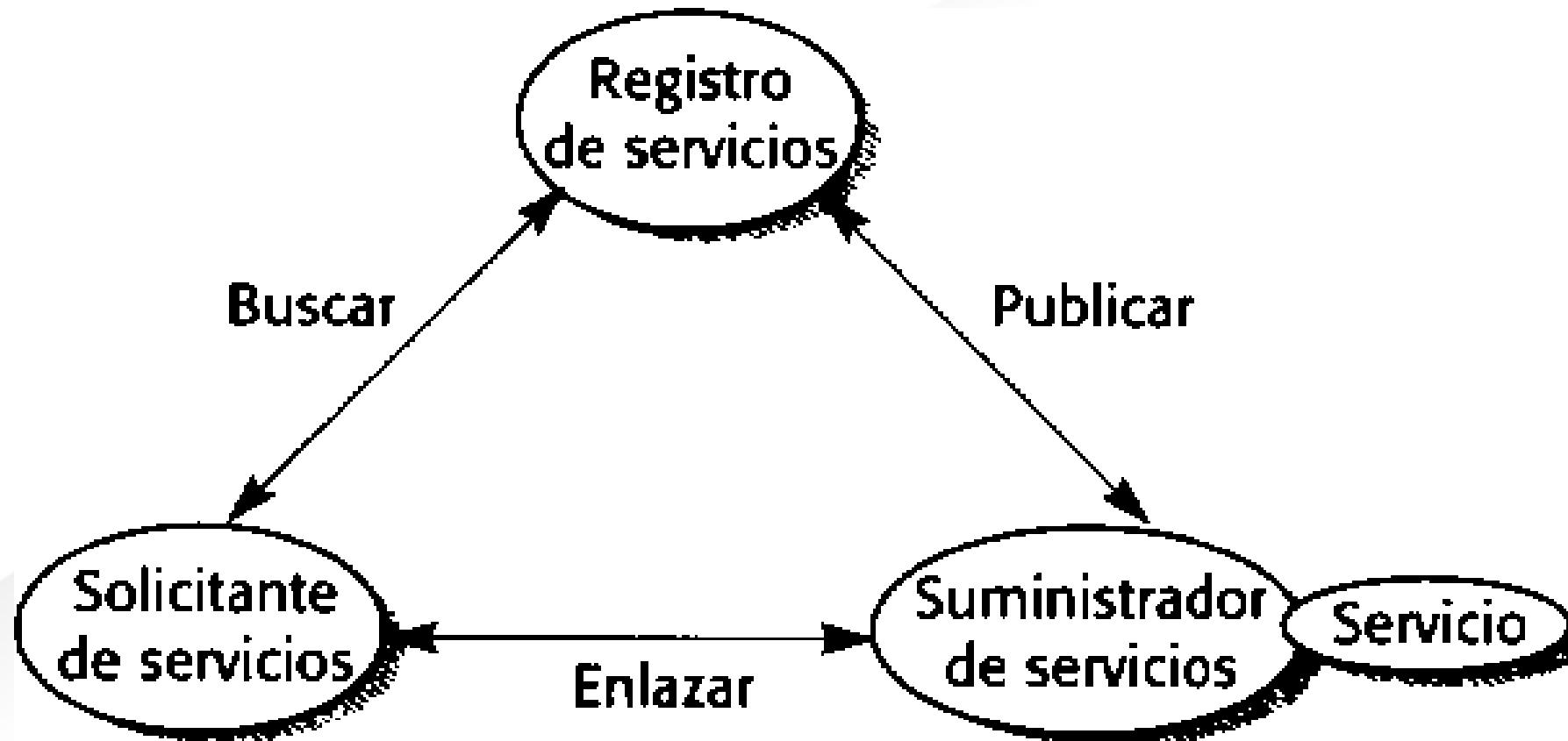
Un solicitante del servicio enlaza este servicio en su aplicación.

Es decir, la aplicación del solicitante incluye un código para llamar al servicio y procesa el resultado de esa llamada al servicio.

Para asegurar que el servicio puede ser accedido por usuarios externos, el proveedor de servicios registra una entrada en el servicio de registro, que incluye información sobre el servicio y lo que hace.

Arquitectura Orientada a Servicios

Arquitectura Conceptual de un Sistema orientado a Servicios:



Arquitectura Orientada a Servicios

Diferencias entre el **Modelo de Servicios** y la aproximación de **Objetos Distribuidos**:

Los servicios pueden ofertarse por cualquier proveedor de servicio dentro o fuera de una organización.

Las organizaciones pueden crear aplicaciones integrando servicios desde varios proveedores.

Por ejemplo, una compañía industrial puede enlazar directamente a los servicios de sus proveedores.

Arquitectura Orientada a Servicios

El proveedor de servicios hace pública la información sobre el servicio para que cualquier usuario autorizado pueda usarlo.

El proveedor de servicios y el usuario de los servicios no necesitan negociar sobre lo que el servicio hace.

Las aplicaciones pueden retrasar el enlace de los servicios hasta que éstas sean desplegadas o hasta que estén en ejecución.

Es posible la construcción de nuevos servicios.

Un proveedor de servicios puede reconocer nuevos servicios que se creen.

Arquitectura Orientada a Servicios

De este modo, los usuarios de los servicios pueden pagar por los servicios con arreglo a su uso en vez de a su provisión.

Por lo tanto, en lugar de comprar un componente de precio elevado que se usa muy raramente, el desarrollador puede usar un servicio externo por el que pagará solamente cuando sea requerido.

Las aplicaciones pueden ser reactivas y adaptar su operación de acuerdo con su entorno, enlazando con diferentes servicios según sea cada caso.

Arquitecturas Peer-to-Peer

Los sistemas peer-to-peer (p2p) son sistemas descentralizados, en los que los cálculos pueden llevarse a cabo en cualquier nodo de la red y, al menos **en principio, no se hacen distinciones entre clientes y servidores.**

En las aplicaciones peer-to-peer, el sistema en su totalidad se diseña para aprovechar la ventaja de la potencia computacional y disponibilidad de almacenamiento a través de una red de computadoras potencialmente enorme.

Arquitecturas Peer-to-Peer

Los estándares y protocolos que posibilitan las comunicaciones, a través de los nodos, están embebidos en la propia aplicación.

Cada nodo debe ejecutar una copia de dicha aplicación.

Las tecnologías peer-to-peer han sido mayormente usadas para sistemas personales.

Sin embargo, se está utilizando de forma creciente en empresas con potentes redes de PCs.

Arquitecturas Peer-to-Peer

Intel y Boeing han implementado sistemas p2p para aplicaciones que requieren computaciones intensivas.

Para aplicaciones cooperativas que soportan trabajo distribuido, es la tecnología más efectiva.

Hay dos tipos principales de arquitecturas lógicas de red que se pueden usar: arquitecturas descentralizadas y arquitecturas semicentralizadas.

Arquitecturas Peer-to-Peer

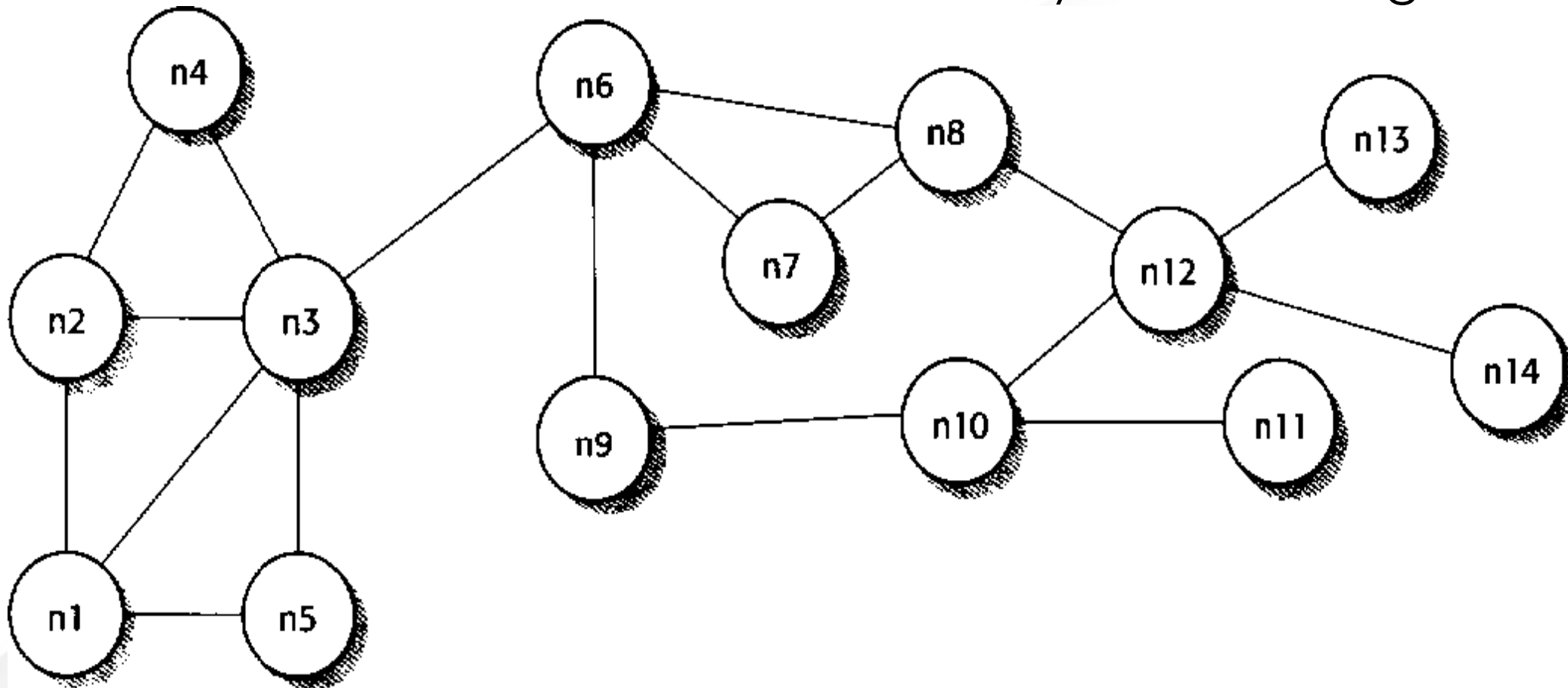
En teoría, en los sistemas peer-to-peer, cada nodo podría conocer cualquier otro nodo, conectarse con él, e intercambiar datos.

En la práctica, esto es imposible, ya que los nodos se organizan dentro de «localidades» con algunos nodos que actúan como puentes a otras localidades de nodos.

En una arquitectura descentralizada, los nodos en la red no son simplemente elementos funcionales, sino que también son interruptores de comunicaciones que pueden encaminar los datos y señales de control de un nodo a otro.

Arquitecturas Peer-to-Peer

Arquitectura p2p descentralizada: En la figura siguiente, si **n1** debe buscar un archivo que está almacenado en **n10**, esta búsqueda se encamina a través de los nodos **n3**, **n6** y **n9** hasta llegar a **n10**.



Arquitecturas Peer-to-Peer

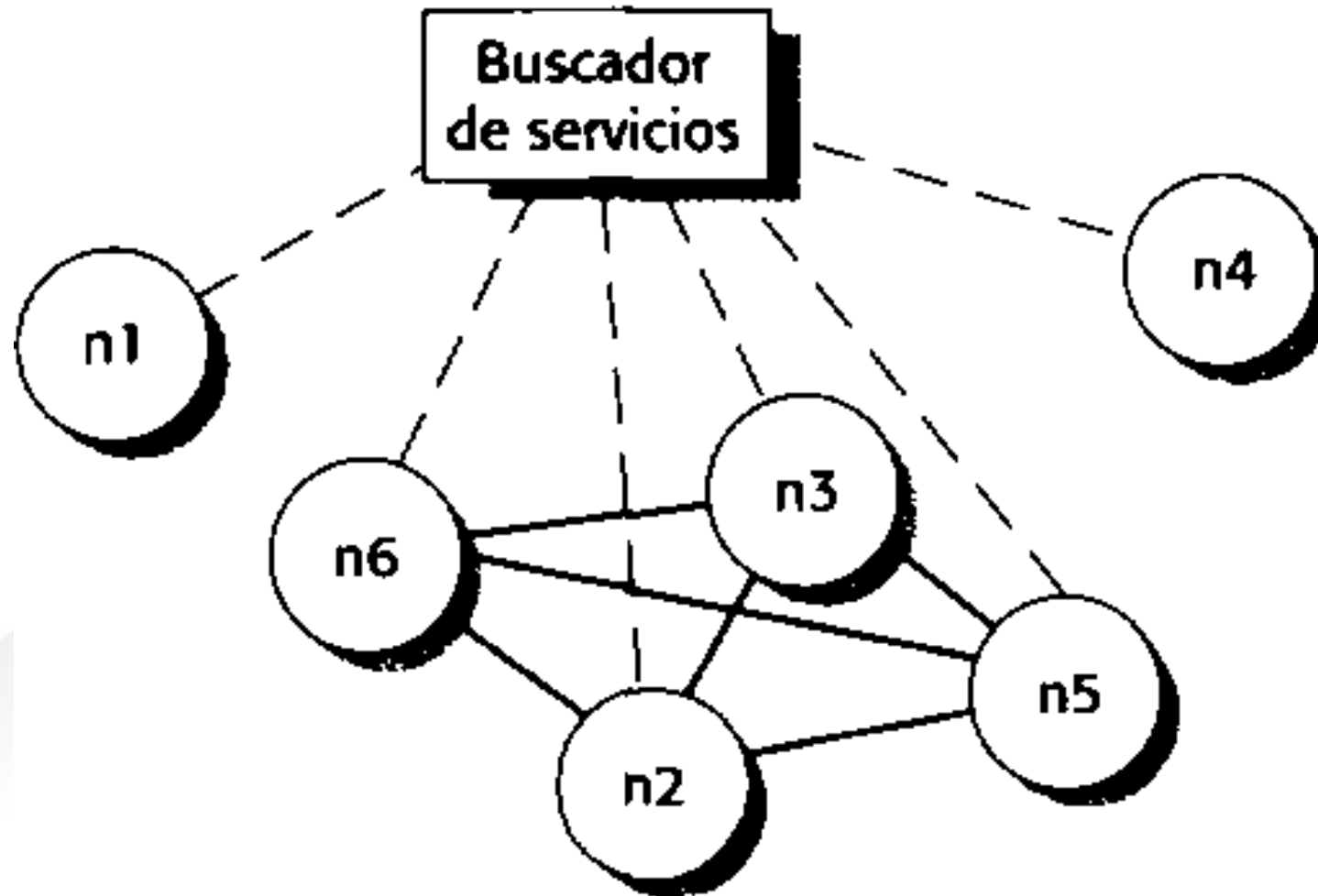
Esta arquitectura descentralizada tiene ventajas: es altamente redundante, y tolerante a defectos y a nodos desconectados de la red.

Sin embargo, existen sobrecargas obvias en el sistema ya que la misma búsqueda puede ser procesada por muchos nodos diferentes y hay una sobrecarga significativa en comunicaciones.

Un **modelo de arquitectura p2p alternativo** que parte de una **arquitectura p2p pura** es una **arquitectura semicentralizada** en la que, dentro de la red, uno o más nodos actúan como servidores para facilitar las comunicaciones entre los nodos.

Arquitecturas Peer-to-Peer

Arquitectura p2p SemiCentralizada



Arquitecturas Peer-to-Peer

En una arquitectura semicentralizada, el papel del servidor es ayudar a establecer contacto entre iguales en la red o para coordinar los resultados de un cálculo.

Por ejemplo, si la Figura anterior representa un **sistema de mensajería instantánea**, entonces los **nodos de la red** se comunican con el **servidor** (indicado por **líneas punteadas**), para encontrar qué otros nodos están disponibles.

Una vez que éstos son encontrados, **se pueden establecer comunicaciones directas** y la conexión con el servidor es innecesaria.

Por lo tanto, los nodos n2, n3, n5 y n6 están en comunicación directa.

Arquitecturas Peer-to-Peer

En un sistema p2p, donde un cálculo (que requiere un uso intensivo del procesador) **se distribuye a través de un gran número de nodos**, es normal que se distingan **algunos nodos cuyo papel es distribuir el trabajo a otros nodos y reunir y comprobar los resultados del cálculo.**

Si bien hay **sobrecargas obvias** en los sistemas peer-to-peer, **éstos son una aproximación mucho más eficiente para la computación interorganizacional que la aproximación basada en servicios** que veremos seguidamente.

Arquitecturas Peer-to-Peer

Todavía hay problemas en el uso de las **arquitecturas p2p**, ya que cuestiones tales como la **protección** y la **autenticidad** **no están del todo resueltas**.

Esto significa que los sistemas p2p se usan más en sistemas de información no críticos.

Arquitecturas Multiprocesador

El modelo más simple de un sistema distribuido es un sistema multiprocesador donde el software está formado por varios procesos que pueden (aunque no necesariamente) ejecutarse sobre procesadores diferentes.

Este modelo es común en sistemas grandes de tiempo real.

Estos sistemas recogen información, toman decisiones usando esta información y envían señales para modificar el entorno del sistema.

Arquitecturas Multiprocesador

Lógicamente, los procesos relacionados con la recopilación de información, toma de decisiones y control de actuadores podrían ejecutarse todos sobre un único procesador bajo el control de un planificador (**scheduler**).

El uso de múltiples procesadores mejora el rendimiento y adaptabilidad del sistema.

La distribución de procesos entre los procesadores puede ser predeterminada o puede estar bajo el control de un despachador (**dispatcher**) que decide qué procesos se asignan a cada procesador.

Arquitecturas Multiprocesador

Un ejemplo de este tipo de sistemas:

Es un modelo simplificado de sistema de control de tráfico.

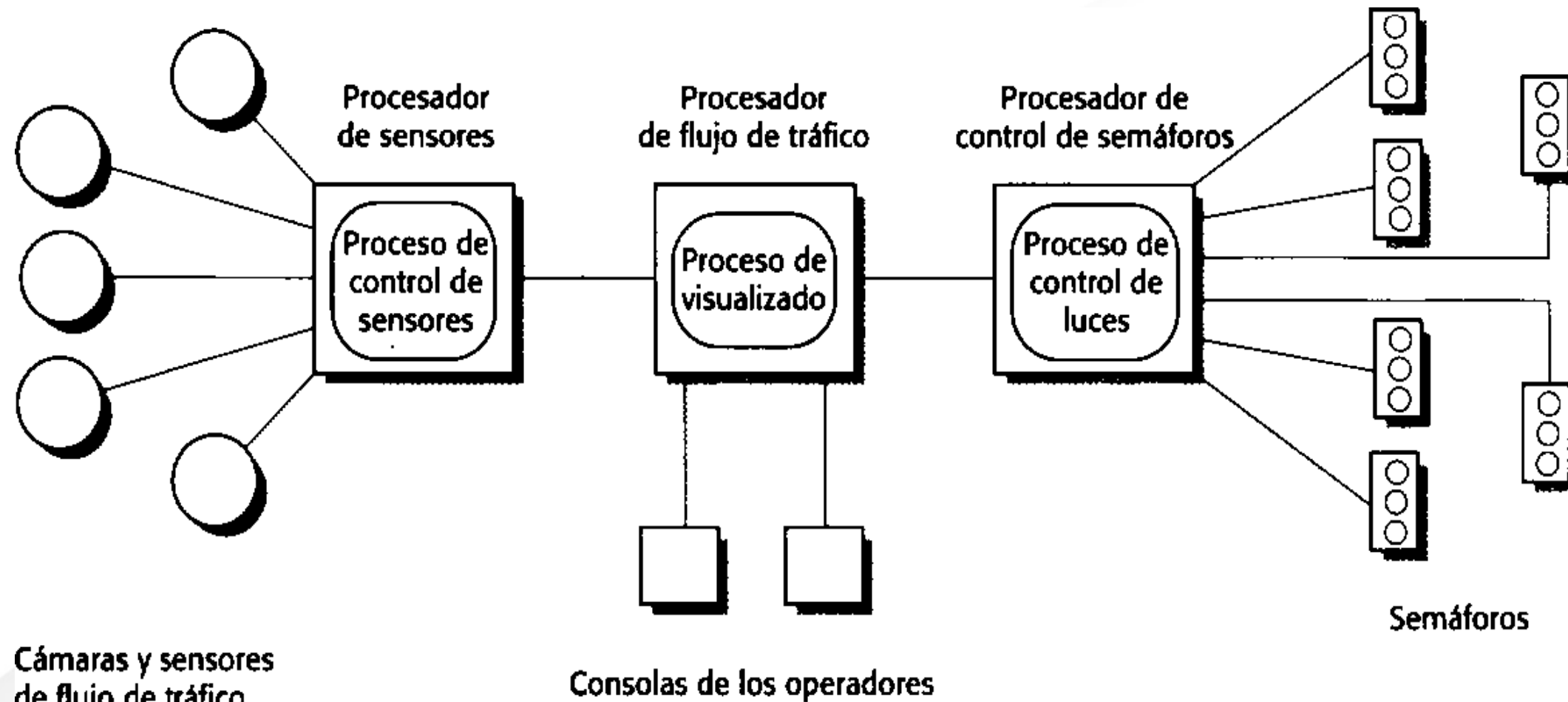
Un conjunto de sensores distribuidos recogen información sobre el flujo de tráfico y la procesan localmente, antes de enviarla a una sala de control.

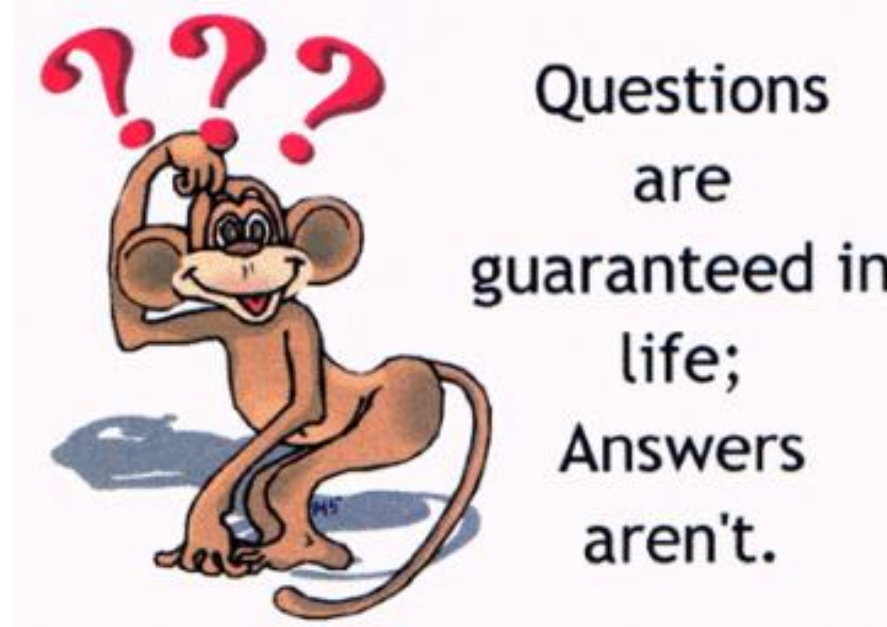
Los operadores toman decisiones, usando esta información y dan instrucciones a un proceso de control de semáforos.

- ✓ Acá hay varios procesos lógicos para gestionar los sensores, la sala de control y los semáforos.
- ✓ Estos procesos lógicos pueden individuales o un grupo de procesos.
- ✓ En este caso, se ejecutarán sobre procesadores diferentes.

Arquitecturas Multiprocesador

Sistema Multiprocesador de Control de Tránsito





¿Preguntas?

Diego Alberto Rincón Yáñez MCSc.
Twitter: @d1egoprog.



INSTITUCIÓN UNIVERSITARIA
**POLITÉCNICO
GRANCOLOMBIANO**

¡GRACIAS!

POLI.EDU.CO |   Poligran | MIEMBRO DE LA RED
ILUMNO