



INICIO GRABACIÓN



SANJOSÉ
FUNDACIÓN DE EDUCACIÓN SUPERIOR



Pueden concebirse como aquellos cuya funcionalidad se encuentra fraccionada en componentes que al trabajar sincronizada y coordinadamente otorgan la visión de un sistema único, siendo la distribución, transparente para quien hace uso del sistema.

Los sistemas distribuidos

En TÉRMINOS COMPUTACIONALES



**HADWARE Ó
SOFTWARE**

Nodos de una red

Comunican
y coordinan
acciones

ENVÍO DE MENSAJES

Ejemplo:
Bases de datos

**SERVICIO
EFICIENTE
CORRECTO
CONSISTENTE**



MOTIVACIONES

COMPARTIR RECURSOS

HADWARE
IMPRESORAS
UNIDADES DE DISCO HASTA...
MEMORIA

ENTIDADES DE
SOFTWARE
ARCHIVOS
BASES DE DATOS
SERVICIOS
OBJETOS
ELEMENTOS
MULTIMEDIA

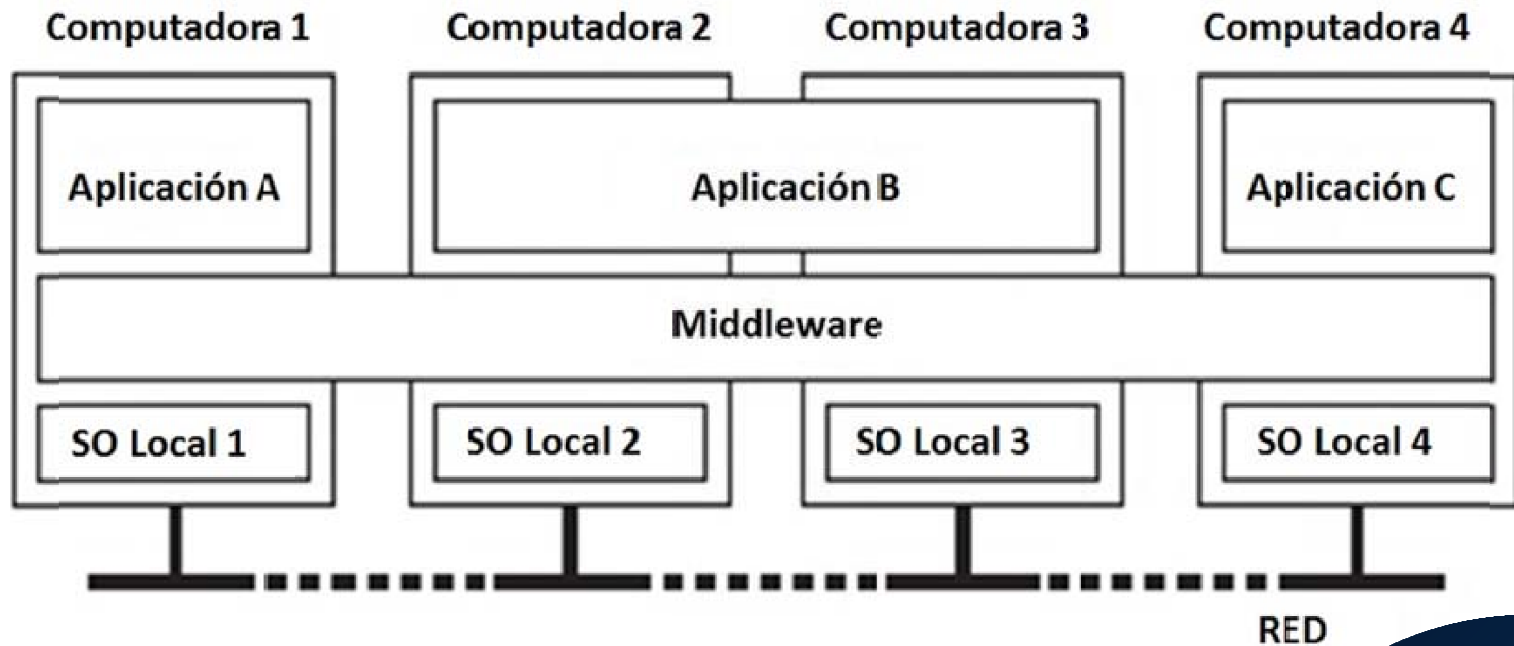
DEFINICIONES

Un sistema en el cual las componentes de hardware y software se ubican en una red de computadoras y comunican y coordinan sus acciones solo por envío mensajes.

Un sistema distribuido es una colección de computadoras independientes que se muestran al usuario del sistema como un sistema único

Un sistema que consiste de una colección de dos o más computadoras independientes que coordinan su procesamiento a través del intercambio sincrónico o asincrónico de mensajes.

DEFINICIÓN



Un sistema distribuido es una colección de computadoras autónomas enlazadas a través de una red con software diseñado para producir facilidades de cómputo integradas.

COEXISTENCIA ENTRE SISTEMA DISTRIBUIDO Y DE COMPUTADORAS

Se puede decir que mientras una red de computadoras son un conjunto de computadoras físicamente visibles y que debe ser explícita- mente direccionada, un sistema distribuido es aquel donde las múltiples computadoras autonomas son transparentes pero se basa en ellas.

RAZONES PARA DISTRIBUIR SISTEMAS

1

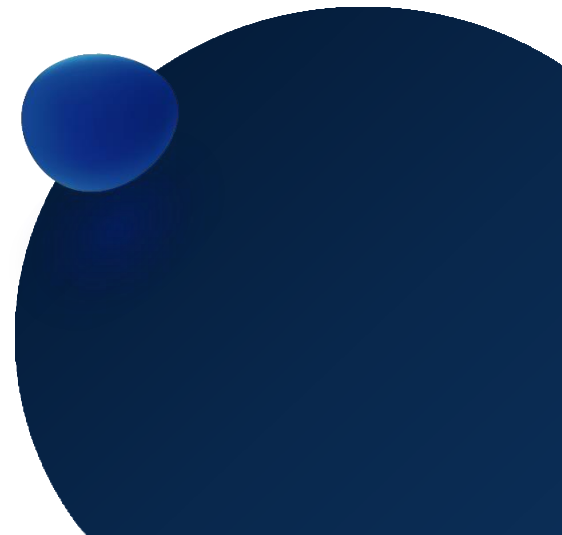
DISTRIBUCIÓN FUNCIONAL →

Capacidades diferentes, pero coordinan acciones con un objeto común



↓
CLIENTE – SERVIDOR
HOST – TERMINAL
DATOS – PROCESAMIENTO

←
DATABASE MANAGEMENT SYSTEMS



RAZONES PARA DISTRIBUIR SISTEMAS

2

DISTRIBUCIÓN INHERENTE
AL DOMINIO DE APLICACIÓN



Mismo dominio de aplicación, cuyas
componentes se identifican y
modelan dentro de este dominio



CAJAS DE SUPERMERCADO
INVENTARIO

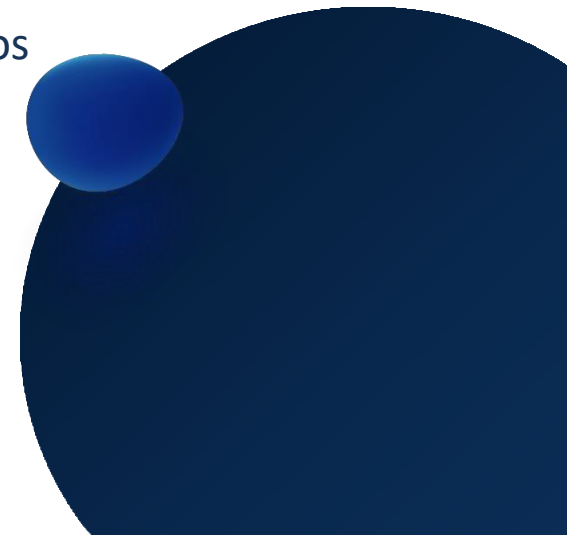


3

BALANCEO Y CARGA DE
DISTRIBUCIÓN



Asignar tareas a distintos
procesadores
Mejorar rendimiento
general del sistema





RAZONES PARA DISTRIBUIR SISTEMAS

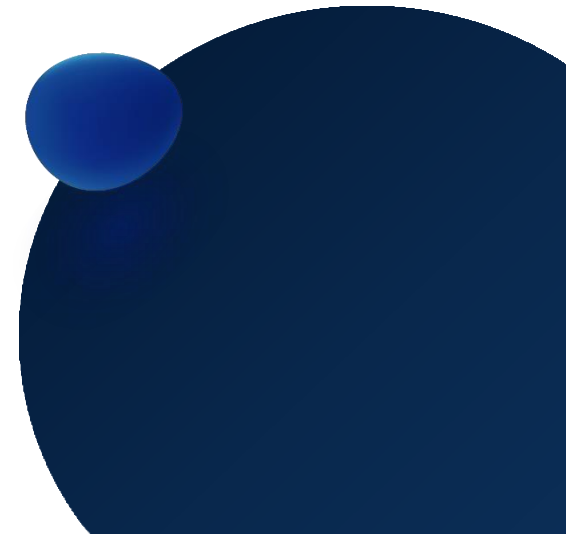
PRECIO - RENDIMIENTO



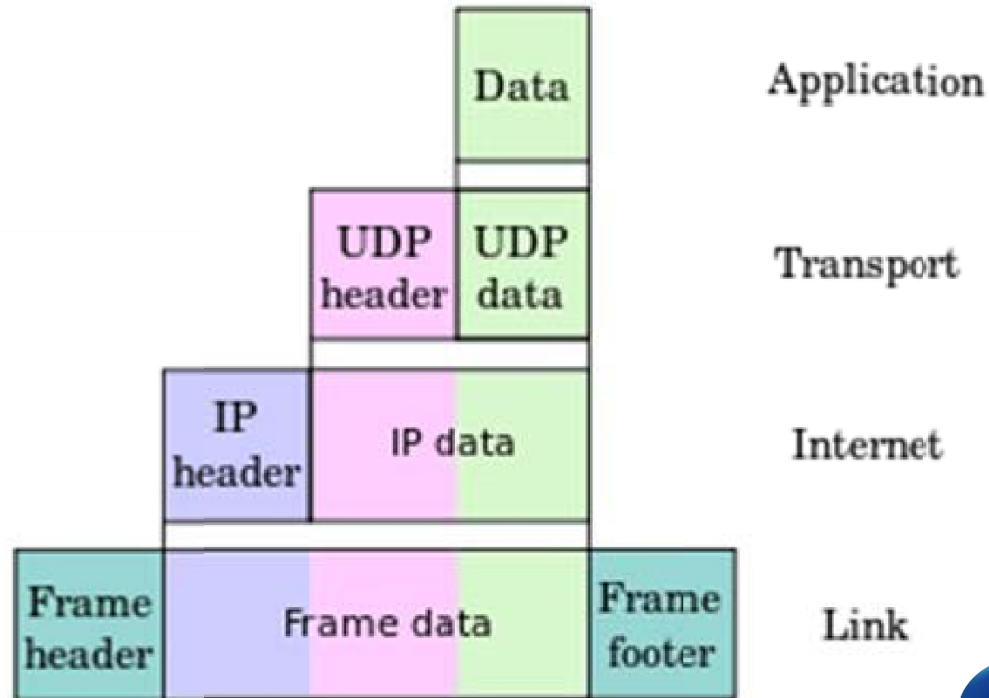
Un conjunto de de microprocesadores ofrecen una mejor relación precio/rendimiento que un gran mainframe¹ que es 10 veces más rápido pero 1000 veces más caro.

¿Por qué usar sistemas distribuidos y no hardware o software aislado? para compartir re- cursos, para incrementar la comunicación entre personas, para potenciar la capacidad de cada componente (de hardware y de software) y otorgar flexibilidad al sistema

¿Qué problemas aparecen en los sistemas distribuidos y conectados? el diseño de software se complejiza porque intervienen más componentes, pueden surgir dependencias de la infraestructura de ejecución subyacente, la facilidad de acceso a datos compartidos puede afectar la seguridad.



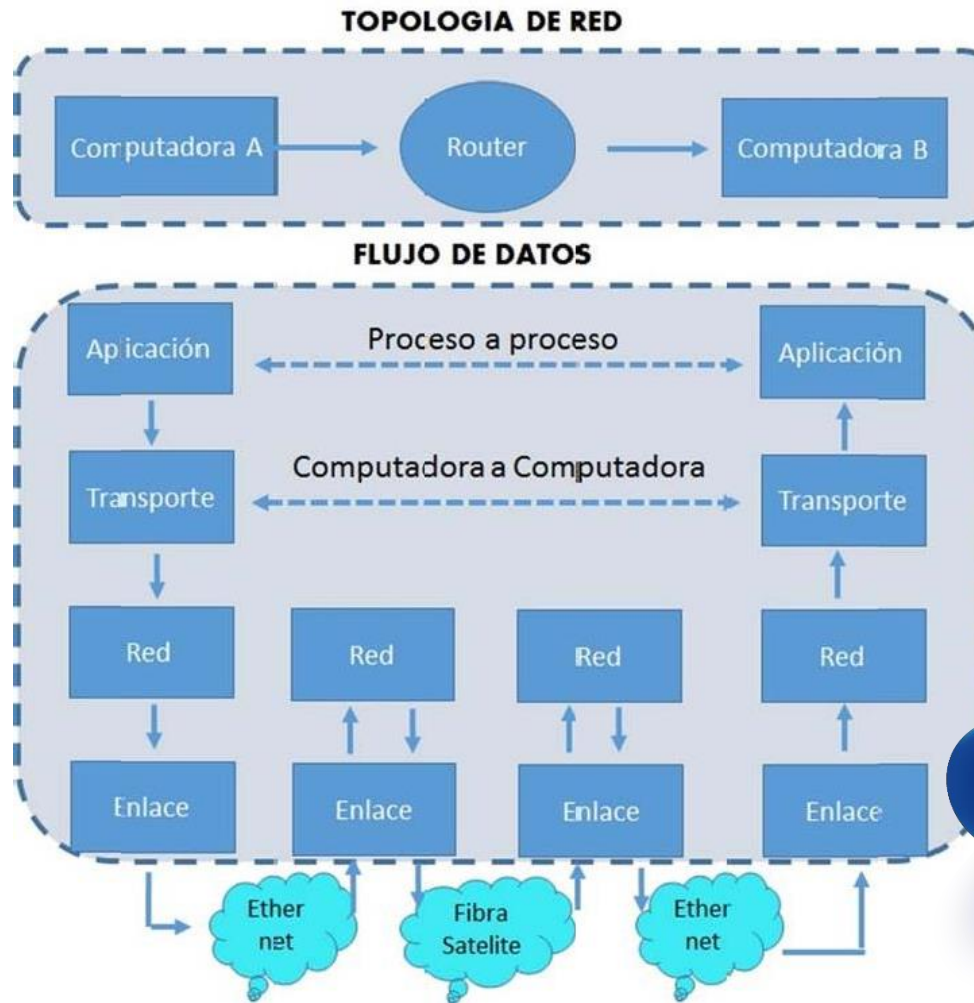
EJEMPLOS INTRANET



Comunicación entre capas y protocolos y cual es el flujo de datos dentro de cada capa.

INTRANET

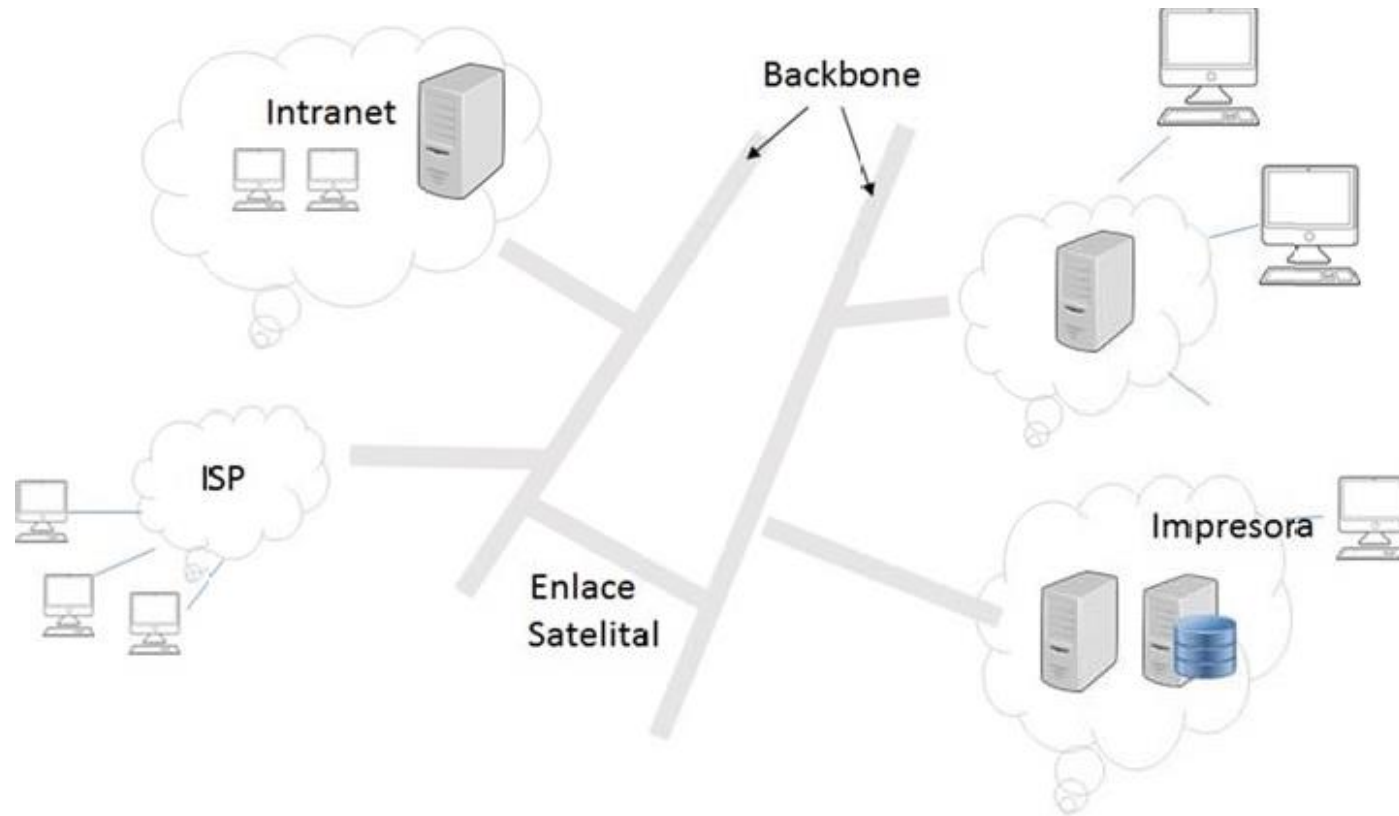
Capa de enlace



Topología de red y flujo de datos entre capas

INTRANET

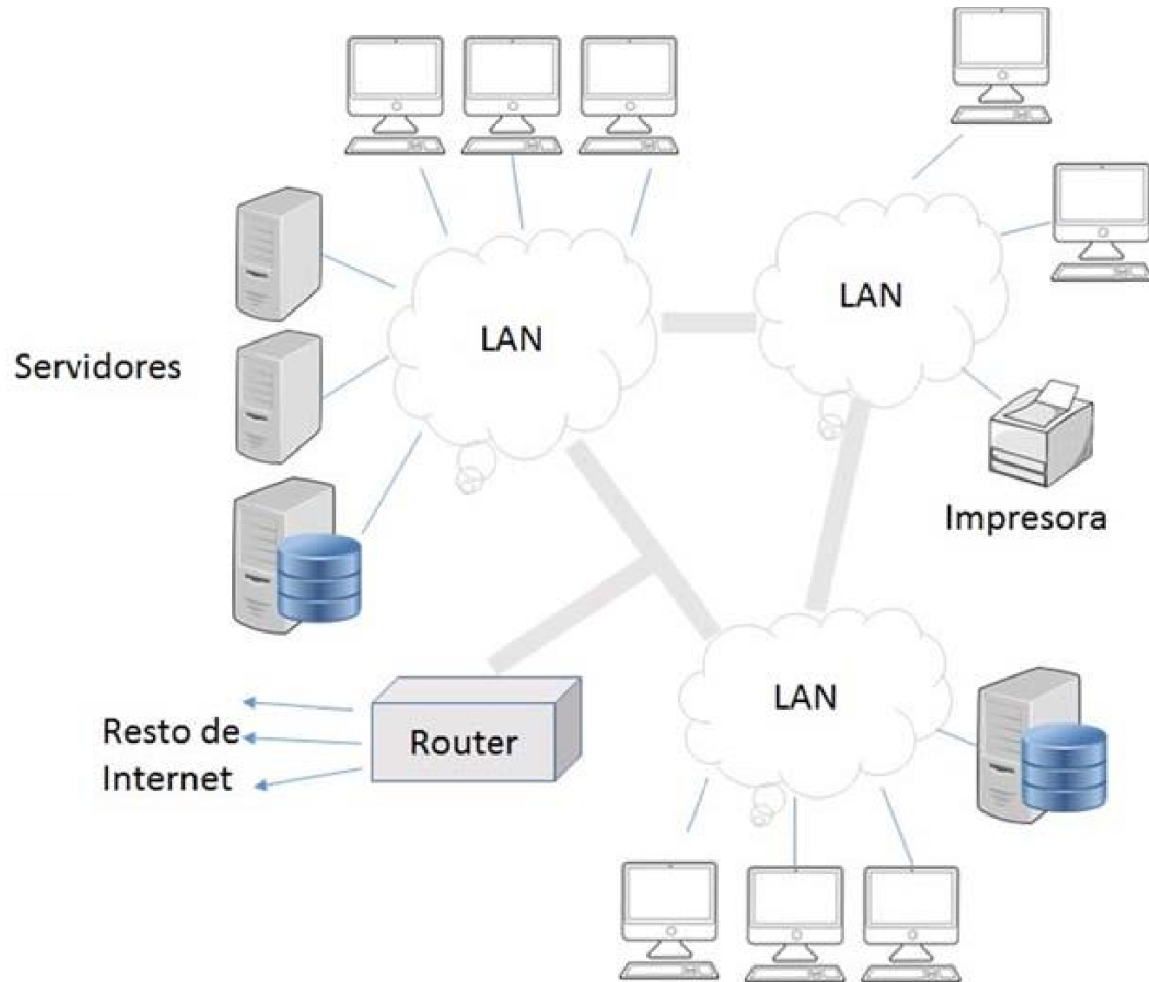
Capa de enlace



Configuración global típica de internet

LAS INTRANETS

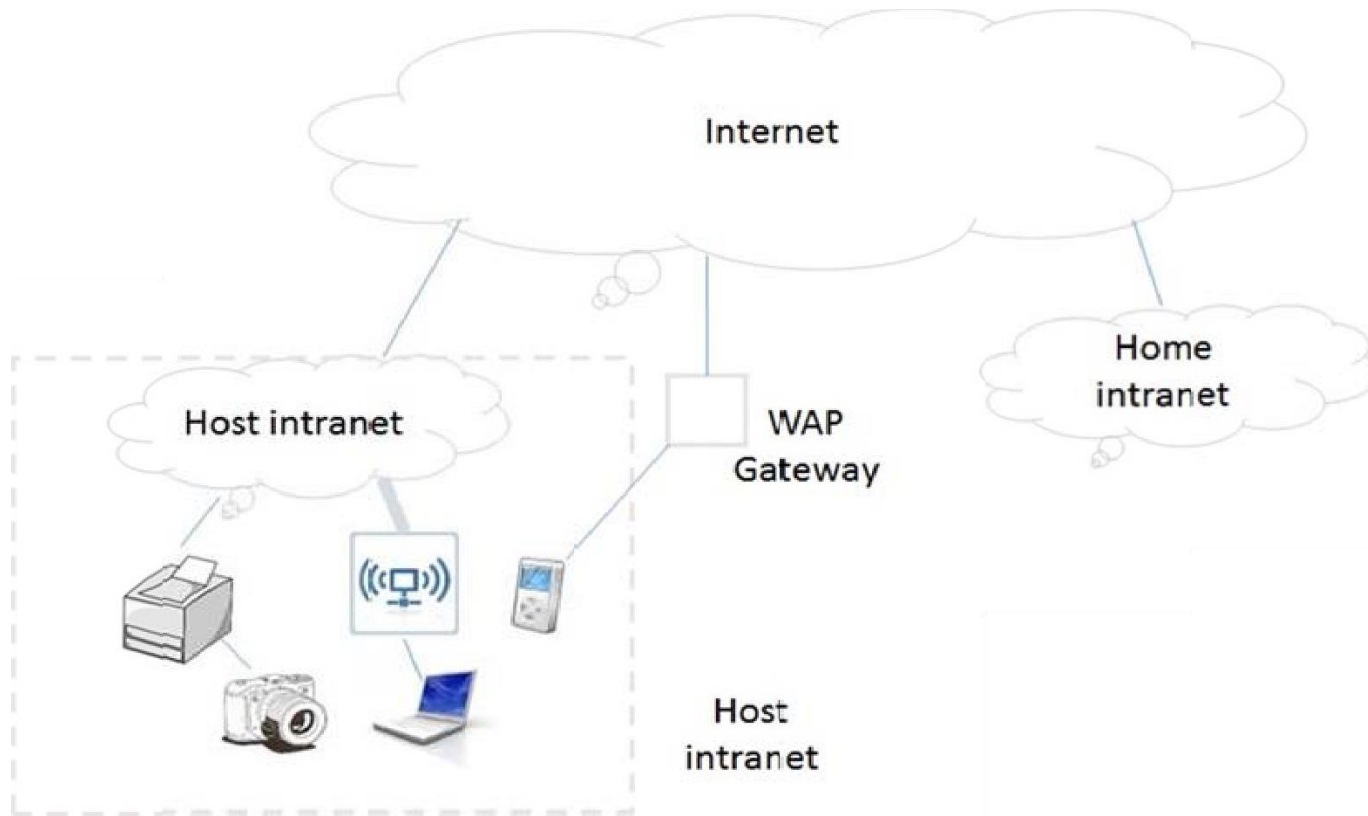
Capa de enlace



Red del campus



SISTEMAS DE COMPUTO MOVILES





DESAFIOS

DESAFIOS

CONCURRENCIA

Los recursos de un sistema distribuido se caracterizan por estar expuestos al acceso concu- rrente de uno o más usuarios, debiendo ser capaz de planificar los accesos evitando deadlocks.

SEGURIDAD

Además de garantizar la seguridad a nivel de transporte de datos, debido al alto grado de promiscuidad de la red como medio de comunicación, un sistema distribuido debe abordar la seguridad en términos de autenticación de usuarios, permisos de acceso sobre los recursos y auditoría del uso de los mismos.

HETEROGENEIDAD

Un sistema distribuido es heterogéneo en cuanto es capaz de permitir que los usuarios ac- cedan a servicios y ejecuten aplicaciones bajo cualquier plataforma (esto es diferentes redes, hardware, sistemas operativos y lenguajes de programación)

HETEROGENEIDAD

Un sistema distribuido es heterogéneo en cuanto es capaz de permitir que los usuarios accedan a servicios y ejecuten aplicaciones bajo cualquier plataforma (esto es diferentes redes, hardware, sistemas operativos y lenguajes de programación)

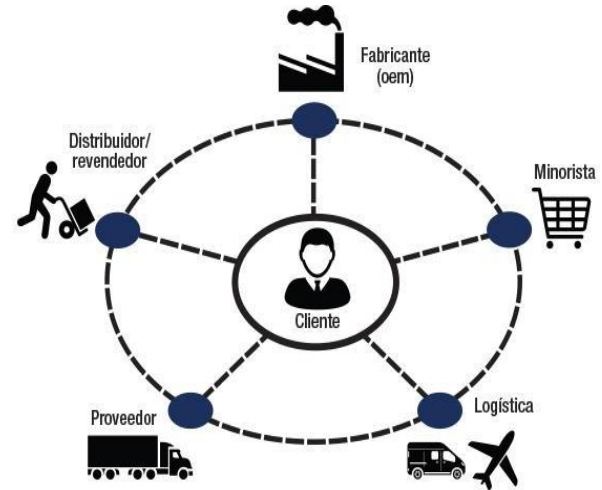


OBJETIVOS

OBJETIVOS	APERTURA	Es abierto cuando ofrece extensibilidad y mantenibilidad mediante la adhesión a estándares que describen la sintaxis y semántica de los servicios que ofrece.
	ESCALABILIDAD	Ofrece capacidad de crecimiento en términos de cantidad y tipo de recursos que administra y cantidad de usuarios que lo acceden.
	ACCESIBILIDAD	Cuando garantiza el acceso a los recursos distribuidos que lo componen a todos los usuarios y en todo momento.
	TRANSPARENCIA	En la medida que oculta su heterogeneidad y su distribución a los usuarios, quienes conservan la visión de un sistema único.

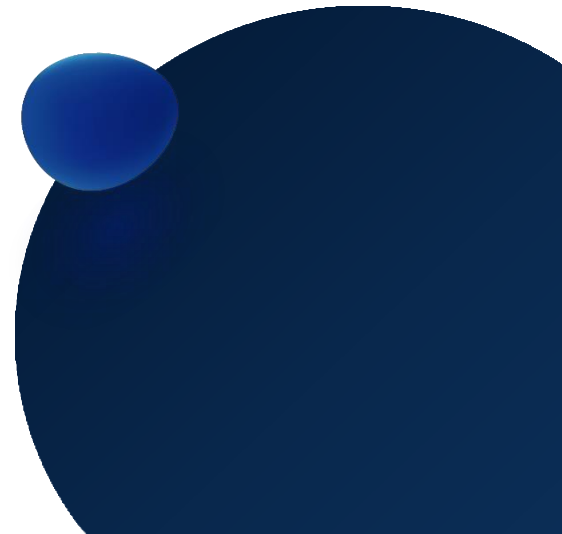
MODELOS DE DISTRIBUCIÓN

Los modelos de distribución pueden responder a aspectos arquitectónicos como a aspectos fundamentales o semánticos



El modelo arquitectónico de un sistema distribuido determina su estructura en función de cada una de sus componentes por separado

El modelo fundamental o semántico revela los problemas claves a los que se enfrenta un desarrollador en un sistema distribuido y que condiciona la metodología de trabajo para alcanzar sistemas confiables en términos de seguridad, correctitud y fiabilidad.



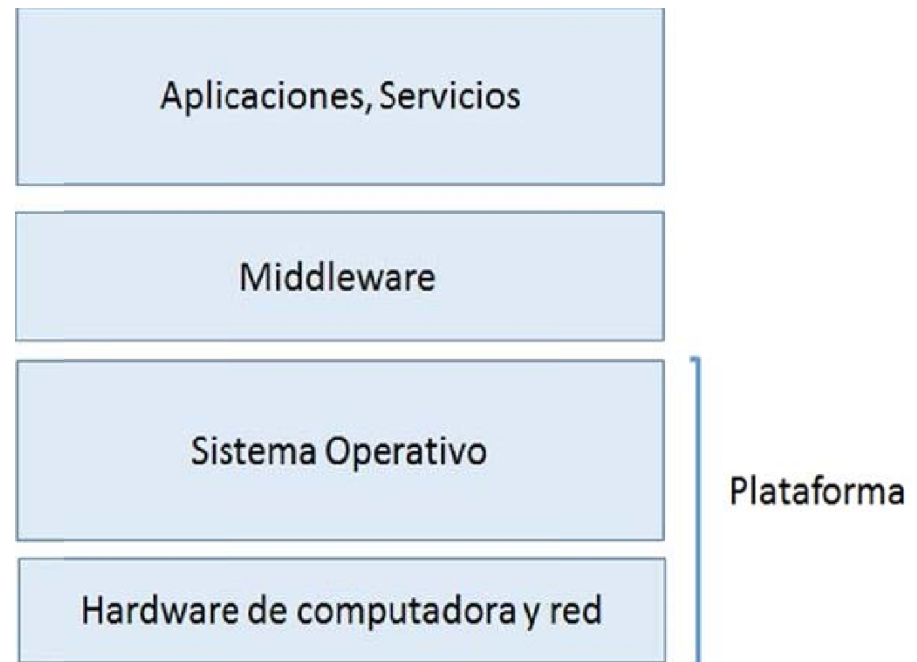
ARQUITECTURAS Y MODELO ARQUITECTÓNICO



La arquitectura de un sistema distribuido define la estructura completa de un sistema según sus componentes específicos, determinado el rol y funciones que cumple cada componente, la ubicación dentro de la red y la interrelación entre las componentes lo que determina su rol o interfaz de comunicación.

FUNCIONES

- PROCESOS SERVIDORES (requerimientos de otros)
- PROCESOS CLIENTES (comunicación-requerimiento-rta)
- PROCESOS PARES (cooperan y comunican para realizar tarea)



ARQUITECTURAS Y MODELO ARQUITECTÓNICO



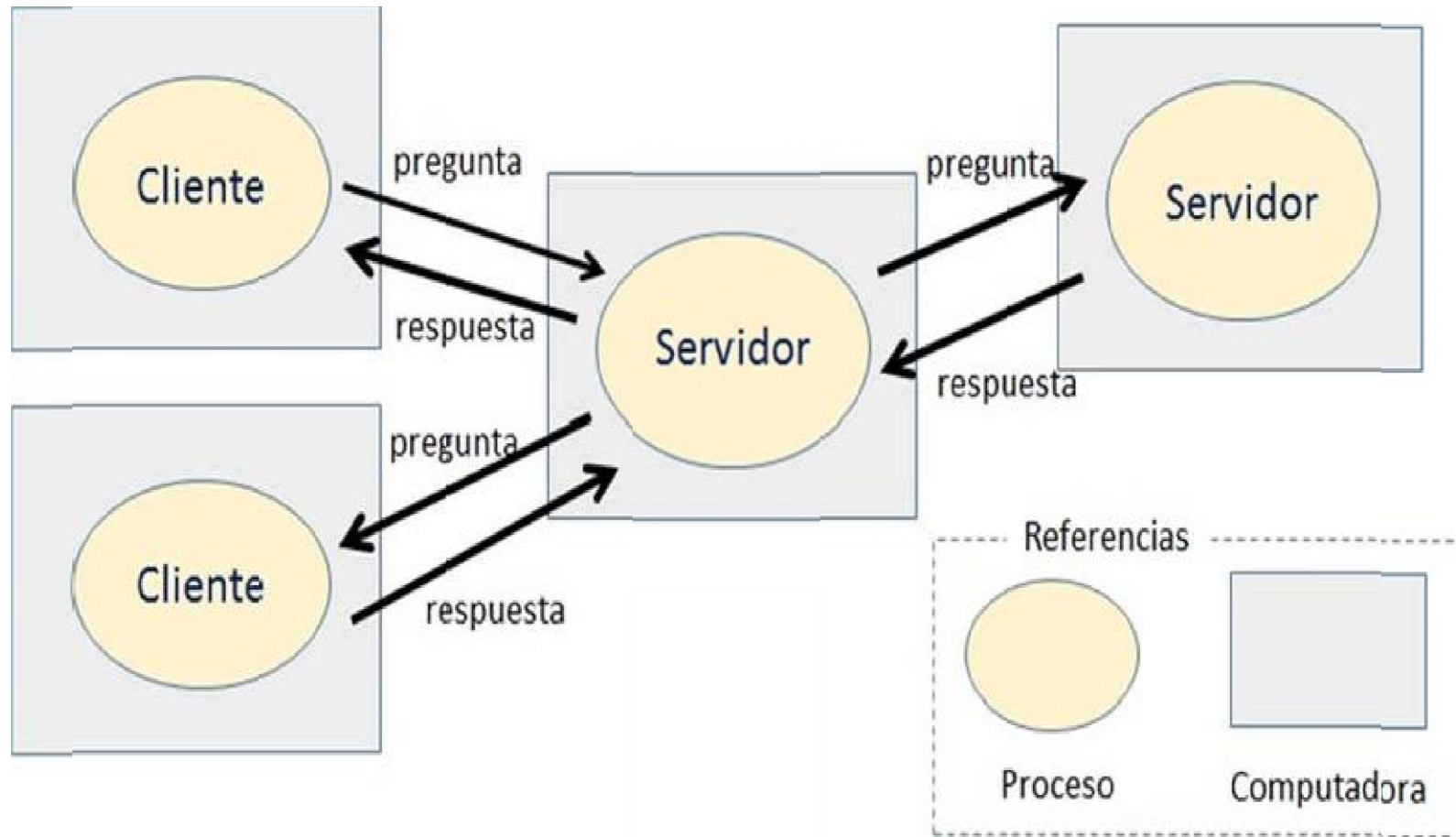
La plataforma está conformada por el software y hardware de bajo nivel que establece el entorno de ejecución de cada componente, sin dicho entorno, no es viable ejecutar aplicaciones siendo esto válido tanto para componentes distribuidas como para componentes que se ejecutan en un sistema único.

EL MIDDLEWARE

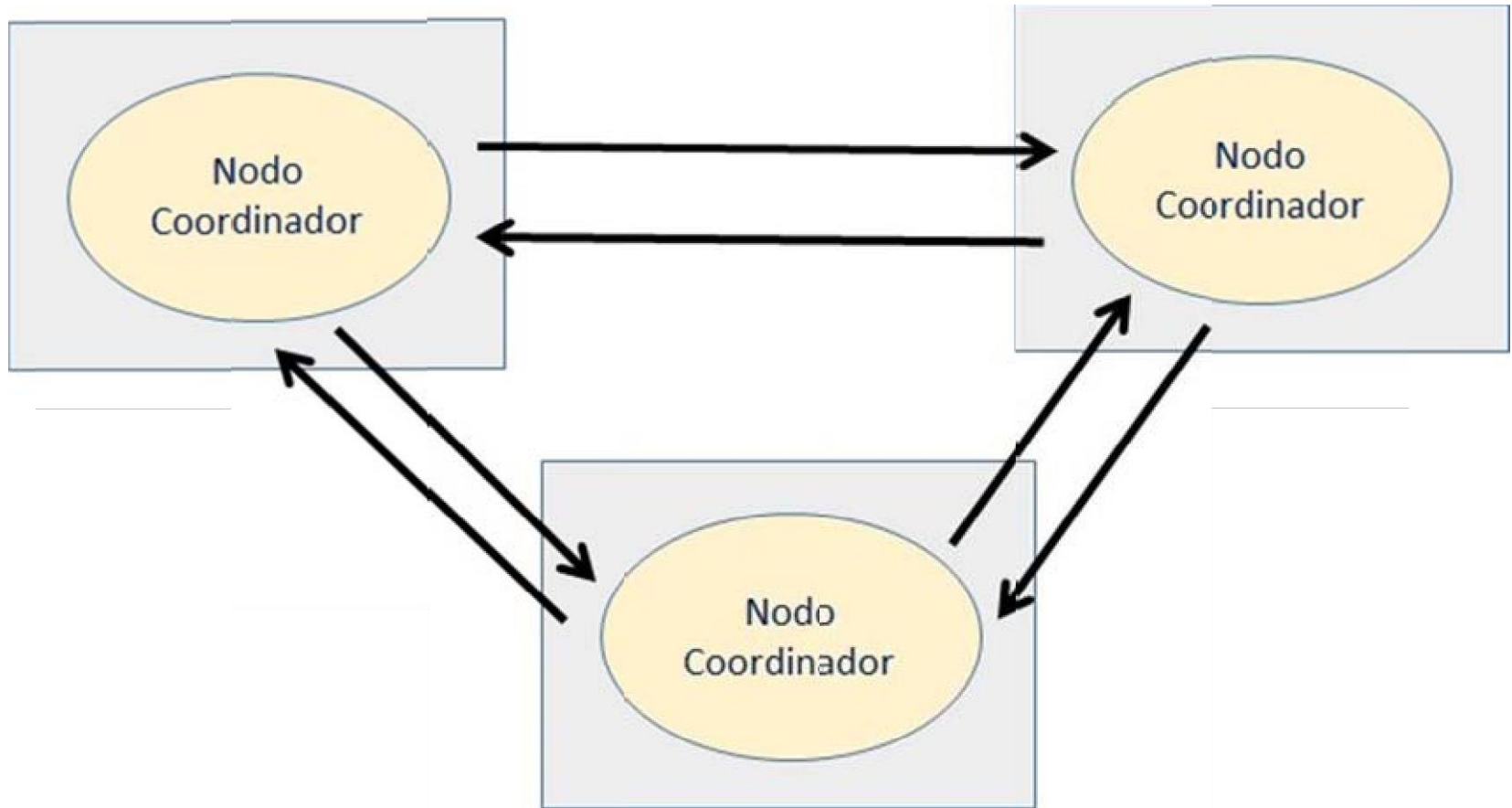
Es el nivel de software que otorga abstracción al programador y a las aplicaciones así como también enmascara la heterogeneidad de la plataforma subyacente



MODELO CLIENTE SERVIDOR



MODELO PEER-TO-PEER



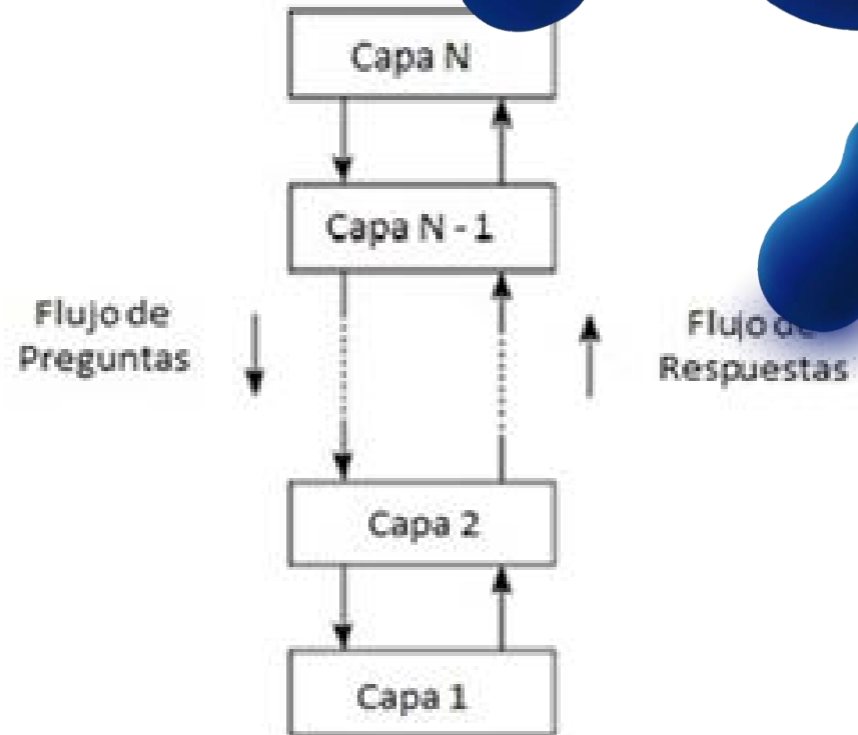
https://www.youtube.com/watch?v=ABDRT7mXs70&ab_channel=ComputerHoy.com

https://www.youtube.com/watch?v=481sJFMooNE&ab_channel=JosiasCastro

ARQUITECTURAS DE SOFTWARE

Se fundamentan en la arquitectura de software subyacente que determina la organización lógica (no física) de los componentes entre varias máquinas.

MODELO DE CAPAS

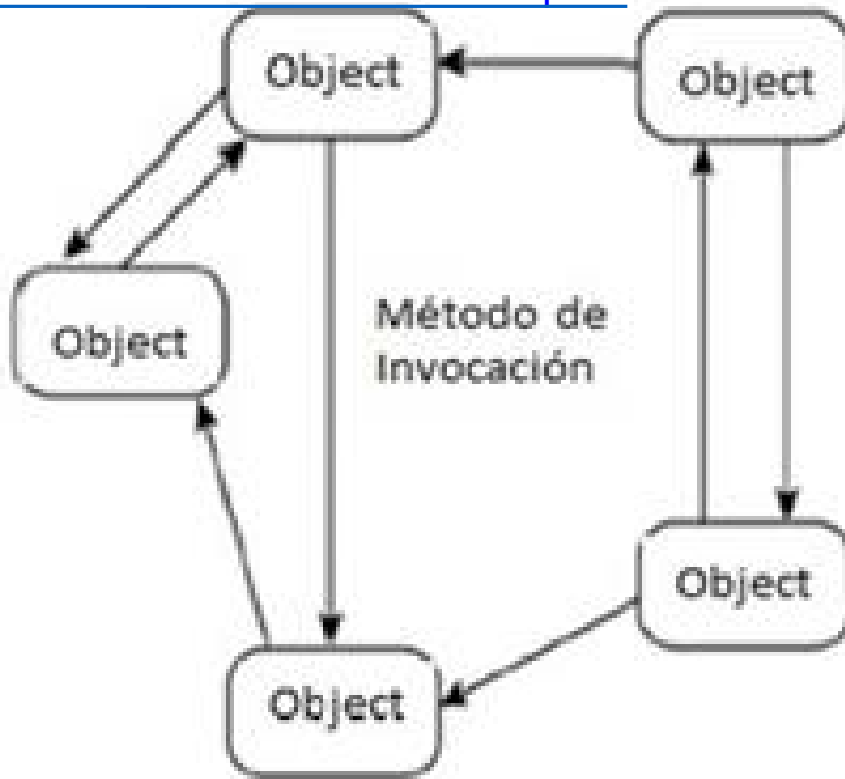


https://www.youtube.com/watch?v=uFDT-NHPe-s&ab_channel=ManuelZapata

Capas determina un flujo de comunicación dentro de cada componente que deberá recorrer el camino inverso cuando llegue a destino y además el destino debe estar disponible.

ARQUITECTURAS DE SOFTWARE

https://www.youtube.com/watch?v=5FxRTjnQbn8&ab_channel=ManuelZapata



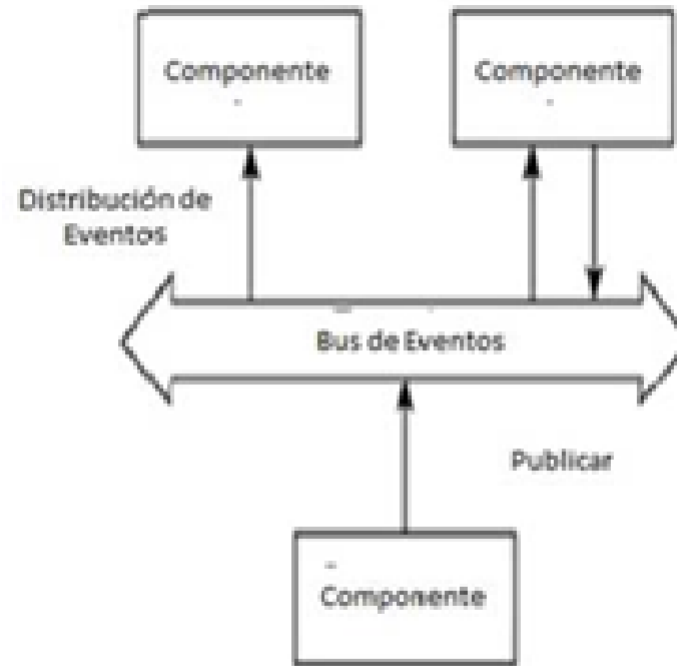
Sigue el paradigma de la orientación a objetos donde cada objeto posee un estado interno y un comportamiento que será puesto en ejecución ante la llegada de un mensaje que lo requiera. En este caso no hay niveles o capas pero sí se requiere la disponibilidad del objeto que recibe el mensaje y de la computadora que lo contiene.

MODELO DE OBJETOS DISTRIBUIDOS

ARQUITECTURAS DE SOFTWARE

Hace parte del Modelo de objetos distribuidos

No requiere sincronismo (coincidencia en el tiempo) debido a que las componentes publican sus funcionalidades en un bus de eventos que se ocupa de dar respuesta a las componentes que se suscriben a cada una de las funcionalidades publicadas



(a)

MODELO DE PUBLICACIÓN/SUSCRIPCIÓN

ARQUITECTURAS DE SOFTWARE



Permite que la comunicación entre las componentes se encuentre absolutamente desacoplada en el tiempo y favorece el acceso a los datos usando una descripción más que una referencia explícita.

MODELO DE ESPACIOS DE DATOS
COMPARTIDO



FUNDACIÓN DE EDUCACIÓN SUPERIOR

SAN JOSÉ

INSTITUCIÓN TECNOLÓGICA

FIN DE
GRABACIÓN