

CAPÍTULO 1.

CONCEPTOS GENERALES DE LA ARQUITECTURA DE APLICACIONES WEB

OBJETIVOS

- Preparar el entorno de desarrollo y los servidores de aplicaciones web instalando e integrando las funcionalidades necesarias.
- Conocer las diferencias entre aplicaciones web y aplicaciones de escritorio.
- Identificar las diferencias entre los modelos de arquitectura cliente-servidor frente a modelos de arquitectura de tres capas.

ÍNDICE

1.1 INTRODUCCIÓN

1.2 APLICACIONES WEB VS. APLICACIONES DE ESCRITORIO

1.3 ARQUITECTURA CLIENTE SERVIDOR. ELEMENTOS

1.4 ARQUITECTURA DE TRES NIVELES

1.5 PROTOCOLOS DE APLICACIÓN MÁS UTILIZADOS

1.1.INTRODUCCIÓN

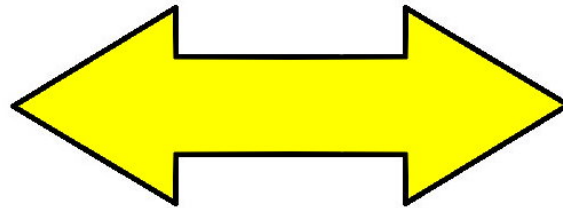
1.1.1. CONCEPTO

- Existen dos lados: uno es el **cliente**, donde se encuentra el usuario final utilizando la aplicación por medio de un navegador web. A través de este cliente web, el usuario interactúa con la aplicación localizada al otro lado, en el **servidor**, que es donde residen realmente los datos, reglas y lógica de la aplicación.

1.1. CONCEPTO



CLIENTE



SERVIDOR

1.1.2. ¿POR QUÉ ESTE CONCEPTO HA TOMADO TANTA RELEVANCIA?

- El servidor debe realizar las operaciones importantes: almacenamiento de datos, transacciones, reglas del negocio y la lógica del programa.
- El concepto de aplicación web ha tomado una mayor relevancia con el auge de las redes locales y la popularidad de Internet.
- Internet ha elevado y extendido aún más el concepto de aplicación web para servir a usuarios ubicados en cualquier sitio donde se tenga acceso a Internet.

1.1.3. PROBLEMAS CON LAS APLICACIONES DE ESCRITORIO

- Duplicidad de datos por la falta de unificación de los mismos.
- Diseminación de la información y lógica en muchas partes (cada computador que la use).
- Falta de portabilidad de la aplicación a diferentes sistemas operativos.
- Traumas a la hora de realizar actualizaciones o correcciones al programa ya que las instalaciones están diseminadas.
- La administración de la seguridad, controlando el acceso a los usuarios a información no relevante o privada es un caos.
- Dificultad para configurar cada una de las instalaciones (*deployments*) dependiendo de las necesidades de cada usuario.

1.1.4. ¿QUÉ PASA CON LAS APLICACIONES DE CONSOLA O MODO TEXTO?

- El concepto de las aplicaciones de consola es similar al de una aplicación web con una arquitectura del tipo cliente-servidor en la cual el cliente también se puede considerar ligero.
- Aunque existen algunas diferencias como son:
 - Protocolos de comunicación propios y no estándar, como ocurre en la Web con el protocolo HTTP y el concepto de URL.
 - Formatos de intercambio propios y no estándar, como ocurre en la Web con el formato HTML o XML.

1.5. LA WEB

- La Web se puede considerar como una plataforma o “sistema operativo” en el cual los recursos están distribuidos en la Red y están siendo extendidos en todo momento con posibilidades ilimitadas.

1.6. INTEGRACIÓN

- Las aplicaciones de escritorio se han usado y se seguirán usando y tienen un campo enorme.
- No todo está en la Web, hay cosas que necesitan ejecutarse estrictamente en su máquina para aprovechar el poder que tiene a su alcance.
- Pero la fusión e integración de servicios de los computadores, las aplicaciones de escritorio y la extensión de las facultades de comunicación con las aplicaciones web que hacen posible Internet son la plataforma óptima que sirve de infraestructura para todos tipo de usuarios.

1.2 APLICACIONES WEB VS. APLICACIONES DE ESCRITORIO

1.2.1 VENTAJAS DEL SOFTWARE WEB

- No requiere instalar software especial (en los clientes).
- Bajo coste en actualizar los equipos con una nueva versión.
- Acceso a la última y mejor versión.

1.2.1 VENTAJAS DEL SOFTWARE WEB

- Información centralizada.
- Seguridad y copias de seguridad.
- Movilidad.
- Reducción de costes en los puestos cliente (mayor longevidad).

1.3 ARQUITECTURA CLIENTE SERVIDOR. ELEMENTOS

- La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados **servidores**, y los demandantes, llamados **clientes**.
- Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

Características del CLIENTE:

- Es el que inicia solicitudes o peticiones. Tiene, por tanto, un papel activo en la comunicación.
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente, interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.
- Al contratar un servicio de red, se debe de tener en cuenta la velocidad de conexión que se le otorga al cliente y el tipo de cable que utiliza.

Características del SERVIDOR:

- Al iniciarse espera a que le lleguen las solicitudes de los clientes. Desempeñan entonces un papel pasivo en la comunicación.
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

Ventajas:

- Centralización del control.
- Escalabilidad.
- Fácil mantenimiento.
- Tecnologías.

Desventajas:

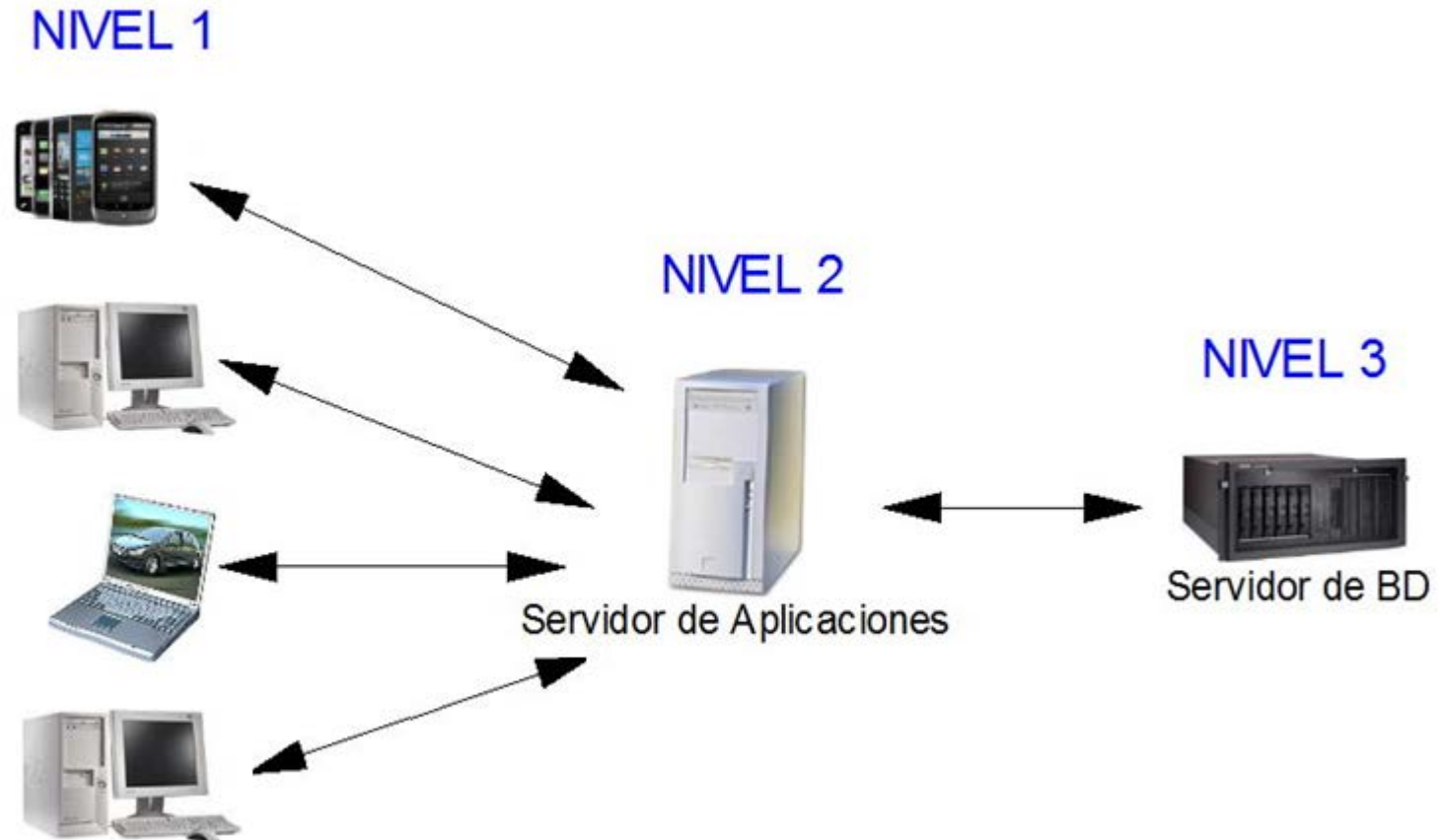
- Congestión del tráfico
- Cuando un servidor está caído las peticiones de los clientes no pueden ser satisfechas
- El software y el hardware de un servidor son generalmente muy determinantes
- El cliente no dispone de los recursos que puedan existir en el servidor.

1.4 ARQUITECTURA DE TRES NIVELES

Partes de la arquitectura de 3 niveles:

- Un **cliente**, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador web) para la presentación.
- El **servidor de aplicaciones** (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El **servidor de datos**, que proporciona al servidor de aplicaciones los datos que este le solicitó.

Partes de la arquitectura de 3 niveles:



La arquitectura en tres niveles permite:

- Un mayor grado de flexibilidad.
- Mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel.
- Mejor rendimiento, ya que las tareas se comparten entre servidores.

1.5 PROTOCOLOS DE APLICACIÓN MÁS UTILIZADOS

1.5.1 EL PROTOCOLO HTTP

- El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato HTML) entre un navegador (el cliente) y un servidor web localizado mediante una cadena de caracteres denominada dirección URL (Uniform Resource Locator, localizador uniforme de recursos).
- La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:
 - El navegador realiza una solicitud HTTP.
 - El servidor procesa la solicitud y después envía una respuesta HTTP.

1.5.2 EL PROTOCOLO HTTPS

- El protocolo seguro de Transferencia de hipertexto (HTTPS: Hiper Text Transfer Protocol Secure) es la versión segura del protocolo HTTP. La diferencia es que HTTPS permite realizar transacciones de forma segura. Por lo tanto, podremos desarrollar actividades de tipo *e-commerce*, acceso a cuentas bancarias *on line*, tramites con la administración pública, etc.

1.5.3 EL PROTOCOLO FTP

- El protocolo FTP (File Transfer Protocol, Protocolo de transferencia de archivos) es, como su propio nombre indica, un protocolo para transferir archivos.
- El objetivo del protocolo FTP es:
 - Permitir que equipos remotos puedan compartir archivos.
 - Permitir la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor.
 - Permitir una transferencia de datos eficaz.

1.5.3 EL PROTOCOLO SMTP

- El protocolo SMTP (Simple Mail Transfer Protocol, Protocolo simple de transferencia de correo) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.