**Tema 1**

**Hosting Compartido**

Comparte los recursos del servidor (CPU, RAM…) con otras webs en el mismo servidor.

**Ventajas:**

* Económico, fácil de gestionar a través de paneles ‘Plesk’ o ‘cPanel’.
* Para principiantes y proyectos pequeños

**Desventajas**:

* Problemas de rendimiento y poco control de las configuraciones.

**VPS (Virtual Private Server)**

Servidor físico se divide en servidores virtuales, con asignación de recursos cada uno.

**Ventajas:**

* Mas control en la configuración.
* Mejor rendimiento que el hosting compartido.

**Desventajas:**

* Mayor coste que el compartido
* Necesario algunos conocimientos básicos de gestión de servidores.

**Servidor Dedicado**

Tiene servidor físico donde solo estará tu web.

**Ventajas:**

* Mejor rendimiento ya que no compartimos recursos.
* Para aplicaciones grandes.

**Desventajas:**

* Opción más cara que los anteriores.
* Necesitas de conocimientos avanzados.

**CLOUD COMPUTING**

**IaaS (Infraestructure as a Service)**

Da recursos de infraestructura como servidores virtuales, redes y almacenamiento. El usuario lo gestiona todo (Confg, SO, Soft).

**Ventajas:**

* Mas flexibilidad, control y escalabilidad.

**Desventajas:**

* Conocimientos técnicos avanzados y más coste dependiendo de los recursos.

**CaaS (Containers as a Service)**

Contenedores como Docker, permiten empaquetar aplicaciones y dependencias para ejecutar en cualquier entorno.

**Ventajas:**

* Facilitan la portabilidad
* Ideal para microservicios

**Desventajas:**

* Conocimientos técnicos avanzados

**KaaS (Kubernetes as a Service)**

Plataforma para orquestar y gestionar contenedores a gran escala.

**Ventajas:**

* Facil gestión de aplicaciones distribuidas y su escalado

**Desventajas:**

* Conocimientos técnicos avanzados

**Hosting estático con CDNs (Content Delivery Network)**

Para desplegar webs estáticas (HTML, CSS y Javascript), se usa para distribuir este contenido por el mundo.

**Ventajas:**

* Eficiente para webs estáticas y veloz.
* Proveedores free y económicos.

**Desventajas:**

* Limitado para webs estáticas

**Arquitectura web**

**Cliente Servidor**

Cliente se solicita (request) y solicita servicios y servidor devuelve (response), el servidor se ejecuta ininterrumpidamente en espera.

**Capa de transporte**

Se encuentran los protocolos de la web, los que usan los navegadores y servidores web para comunicarse.

**Puertos**

Identifican procesos de un equipo y envían y reciben información

* Bien conocidos: 0 – 1023
* Registrados: 1024 – 49151
* Dinámicos: 49152 – 65535

La asignación de estos puede ser ‘dinámica' el SO lo elige o ‘estatico’ definidos en la configuración.

**Protocolos**

**UDP:** Protocolo no orientado a conexión es poco fiable pero también es mas rápido, tiene datagramas.

**TCP:** Protocolo orientado a conexión, control de errores, flujo y congestión, tiene segmentos.

**Capa de aplicación**

Funcionamiento P2P

**HTTP**

Protocolo para la transferencia de paginas web (hipertexto), solicita mediante request HTTP al servidor web.

Se contesta con una response HTTP del servidor, este protocolo no tiene estado, esto significa que el servidor no almacena ninguna información sobre los clientes.

**SSL(Secure Sockets Layer) y TLS (Transport Layer Security)**

Son protocolos de cifrado, cifran las comunicación, TLS es el sucesor de SSL, estos protocolos sirven de soporte a HTTP que pasaría a HTTPS.

**SSH (Secure Shell)**

Permite la comunicación basada en texto entre dos maquinas remotas, **comunicación cifrada**, el uso mas habitual es la gestión remota de maquina UNIX/LINUX.

**SCP (Secure Copy)**

Basado en SSH, permite copiar ficheros entre dos equipos.

**FTP (File Transfer Protocol)**

Protocolo para la transferencia de ficheros entre dos maquinas, con SFTP tenemos comunicación cifrada gracias a SSH.

**Tipos de arquitectos**

* En sistemas: Configuran distintos elementos hardware con elementos software.
* En datos: Estructuran la información manejada por aplicaciones.
* En almacenamiento: Diseñan redes de almacenamiento.
* En redes: Planean y diseñan redes de comunicación.
* En software: Diseñan y construyen aplicaciones para ofrecer servicios.
* En web: Diseñan y construyen aplicaciones pero web

Los clientes web son los navegadores

**Servidor de aplicaciones**

Programa en una red distribuida que proporciona la lógica de negocio para una aplicación.

**Sistemas Gestores (BBDD)**

Gestor de bases de datos, conjunto de programas que permiten la administración y gestión de los datos.

**Frameworks**

Herramientas y bibliotecas para simplificar operaciones comunes de desarrollo.

**Paginas estáticas y aplicaciones web**

**Aplicaciones web**

Ventajas:

* Ahorra tiempo, y no hay problemas de compatibilidad,
* No ocupan, recursos bajos, portables, los virus no dañan…

Inconvenientes:

* Menos funcionalidades
* Depende de un tercero

**CGI**

Método estándar usado por servidores para delegar la generación del contenido a archivos ejecutables.

**Servlets (Java)**

Para backend, reciben peticiones del cliente y ofrece unas respuestas html al servidor web se ejecutan en un contenedor de Servlets.