

***Despliegue de aplicaciones WEB***

## TEMA 0

# Introducción a las Aplicaciones WEB

Juan Antonio López Mera  
IES VELÁZQUEZ

# 0. Índice

## 1. *Aplicaciones Web*

- i. ¿Qué puede y debe hacer una aplicación web?*
- ii. Diferencias entre una aplicación web y una aplicación de escritorio*
- iii. Tipos particulares de aplicaciones web*
- iv. Arquitectura de una aplicación web*

## 2. *Servidores web*

## 3. *Servidores de aplicaciones*

## 4. *El futuro...*

## 5. *Bibliografía*

# 1. Aplicaciones web

- Es una aplicación que está disponible en la red.
- Usa el protocolo HTTP de transporte.
- Produce contenido DINÁMICO (generando contenido en tiempo de ejecución)

Ejemplos: cesta de la compra, cuenta bancaria...

La principal diferencia con un SITIO WEB es que el contenido de éstos es estático, no se puede modificar en ningún caso en tiempo de ejecución.

# 1. Aplicaciones web

## i. *¿Qué puede y debe hacer una aplicación web?*

- **DISTRIBUIR INFORMACIÓN**

**Blog:** nos permite compartir información alojándola en un servidor web. Esta información puede ser actualizada en cualquier momento.

- **GESTIONAR LA CONCURRENCIA**

**Blog:** el acceso debe poder ser simultáneo.

**Tienda on-line:** Hay que controlar que el stock se vaya actualizando a medida que se van realizando compras.

- **GENERAR CONTENIDO DINÁMICO**

**Wiki:** Pueden ser editados por los usuarios añadiendo, modificando o eliminando información del mismo.

# 1. Aplicaciones web

## i. ¿Qué puede y debe hacer una aplicación web?

- **SEGURIDAD BASADA EN ROLES**

Tienda online: no tendrá los mismos accesos un cliente, el vendedor o el propio administrador de la aplicación.

- **FUNCIONALIDAD y BUEN USO DE LAS BBDD**

Wiki: Si nuestra wiki alcanza un tamaño considerable y son muchos los usuarios que acceden, consultan y modifican sus contenidos, se debe contar con una BBDD distribuida y con contenidos en varios servidores, además de la posibilidad de ampliar las posibilidades de la misma.

- **BBDD FIABLES Y CONSISTENTES**

Controlar las transacciones para, por ejemplo, el hecho que dos usuarios modifiquen contenidos a la vez en una wiki.

- **ESCALABILIDAD DEL SISTEMA**

Tener en cuenta el crecimiento del sistema, a nivel hardware y software para la ampliación en caso que sea necesario.

# 1. Aplicaciones web

## ii. Diferencias con una aplicación de nativa

- Versatilidad a la hora de poder ejecutarse en distintos dispositivos.
- Se verán afectadas por problemas de comunicaciones y de acceso múltiple a los datos.
- Parte de su código se ejecutarán en la parte del servidor y otras se ejecutarán en la parte del cliente

# 1. Aplicaciones web

## *iii. Algunos tipos de aplicaciones web*

- **Blogs:** Aplicaciones donde se comparte información y se permite a otros usuarios comentar en las publicaciones.
- **Wikis:** También sirven para compartir información didáctica y se permite a otros usuarios modificar o añadir los contenidos.
- **Redes sociales:** También se permite compartir información, pero en este caso personal, aunque su uso se está ampliando a anunciar empresas y productos también.
- **Tiendas on-line:** Comprar y vender productos. Su seguridad en las transacciones debe ser alta.
- **Web bancarias:** Permite ver la información de cuentas, realizar transacciones, etc...

# 1. Aplicaciones web

## *iv.Arquitectura de las aplicaciones web*

- Las aplicaciones web usan una **ARQUITECTURA POR CAPAS**: cada una de estas capas tiene un rol perfectamente diferenciado de las otras.
  - **PRESENTACIÓN**: Relacionada con la interfaz de usuario.
  - **APLICACIÓN**: Relacionada con los procesos de negocio en las comunicaciones.
  - **GESTIÓN DE DATOS**: Relacionada con el manejo de los datos de la BBDD de la aplicación.

Normalmente, se usa un **MODELO DISTRIBUIDO**:

- La capa de presentación se ejecuta en los dispositivos de los usuarios (tablets, móviles, PC's).
- La capa de aplicación se ejecuta en una máquina (o máquinas) que hace las veces de servidor (o servidores) de contenidos.
- La capa de gestión de datos, dependiendo del tamaño, estará junto a la anterior en el mismo servidor (centralizadas) o distribuidas en otras máquinas.



# 1. Aplicaciones web

## iv. Arquitectura de las aplicaciones web

### ● VENTAJAS DE LA ARQUITECTURA DE 3 CAPAS:

- Clientes ligeros: la principal herramienta de uso en la parte del cliente es un simple navegador.
- Capacidad de migración de las BBDD: Éstas, al realizarse en el lado del servidor, no afectan en el lado del cliente.
- Fácil conexión: El protocolo es standard para que cualquier usuario desde cualquier dispositivo que tenga conectividad.
- Fácil escalabilidad: Permite una ampliación en el lado del servidor. A medida que aumente el número de conexiones por parte del cliente, se debe ampliar la capacidad del servidor, pero esto no afecta en ningún caso al cliente.
- Independencia de la plataforma: No importa el SSOO que use el cliente, todos deben poder contar con una herramienta para el acceso a la aplicación web.
- Facilidad de implementación de personalizaciones: la aplicación se adapta a cada cliente, ya que ésta parte se realiza en el lado del cliente (cookies).
- Facilidad de actualizaciones: se realizan sin que afecte directamente al cliente.

## 2. Servidores web

- **Son máquinas remotas donde se albergan las páginas y aplicaciones web.**
- **Los clientes se comunican con ellos mediante peticiones de recursos a partir de direcciones IP, URL, URI.**
- **El administrador del servidor debe encargarse de configurar:**
  - **Rutas y direcciones desde las que serán accesibles los recursos.**
  - **Control de acceso a éstos.**
- **Inicialmente, los servidores sólo se encargaban de servir páginas estáticas en HTML. La apertura a las aplicaciones web a ampliado las capacidades de estos mediante tecnologías como ASP, JSP, PHP...**
- **El software más popular usado por los servidores para gestionar los contenidos son Apache e IIS. Existen otros como NGINX, Sun Java, Lighttp.**

### 3. Servidores de aplicaciones

- **Proporciona contenido dinámico usando el procesamiento en el lado del servidor y presentándolo, principalmente, mediante páginas HTML.**
- **Las tecnologías que permiten esto son:**
  - **Lenguajes de scripts de Servidor: Perl, Python, PHP.**
  - **Java Servlet: clases de Java que se ejecutan en la parte del servidor y sirven para responder peticiones HTTP. Apache Tomcat actúa como contenedor y gestor de estas clases.**
  - **CGI (Common Gateway Interface): ya en desuso, pero fue la primera tecnología que permitió el contenido dinámico.**
- **Se necesitan una serie de aplicaciones para montar un servidor web, algunas son:**
  - **LAMP (Linux, Apache, MySQL, PHP)**
  - **WAMP (Windows, Apache, MySQL, PHP).**
  - **XAMPP (x representa cualquier SO, Apache, MySQL, PHP, Perl).**

## 4. El futuro (o más bien el presente)...

- **Aplicaciones web híbridadas:** Aprovecharía lo mejor del desarrollo web multiplataforma (el desarrollo sería válido para distintos dispositivos) y lo mejor del desarrollo nativo o sobre un solo sistema operativo (aprovechando al máximo los recursos del SO sobre el que se ejecuta la aplicación, haciendo uso de las API del SO donde se ejecuta).
  - <https://www.ceac.es/blog/el-desarrollo-de-aplicaciones-en-webs-hibridas>
- **Aplicaciones web progresivas o PWA:** Siendo una aplicación web, utiliza los recursos para que la aplicación se ejecute como si fuera nativa, con el ahorro de recursos que eso implica (permite trabajar offline, se puede anclar en el escritorio, se puede ejecutar en segundo plano...).
  - <https://www.ceac.es/blog/que-son-y-para-que-sirven-las-aplicaciones-web-progresivas>

## **4. Bibliografía**

*Servicios de Red e Internet. Álvaro García Sánchez, Luis Enamorado Sarmiento, Javier Sanz Rodríguez. Editorial Garceta*

***Www.Wikipedia.org***

***Www.Edu4Java.com***

***Www.internetLab.es***

***Www.ceac.es***