



EFA  
MORATALAZ

*2º CFGS Desarrollo de  
Aplicaciones Web*

# ***DESPLIEGUE DE APLICACIONES WEB***

***DANIEL GONZÁLEZ-CALERO  
JIMÉNEZ***

## **UT2 – IMPLANTACIÓN DE ARQUITECTURAS WEB**





EFA  
MORATALAZ

*2º CFGS Desarrollo de Aplicaciones  
Web*

## ***DESPLIEGUE DE APLICACIONES WEB***

# UT2 – IMPLANTACIÓN DE ARQUITECTURAS WEB

1. INTRODUCCIÓN
2. ARQUITECTURAS WEB
3. EVOLUCIÓN DE LA TECNOLOGÍA WEB
4. TECNOLOGÍAS USADAS EN APLICACIONES WEB
5. SERVIDORES Y APLICACIONES LIBRES Y PROPIETARIAS
6. PROTOCOLO HTTP y HTTPS
7. CLASIFICACIÓN DE SERVIDORES DE APLICACIONES DEL MERCADO ACTUAL
8. ESTRUCTURA Y RECURSOS DE UNA APLICACIÓN WEB

# INTRODUCCIÓN

1

La arquitectura web es la que define como se va a jerarquizar la información dentro de un sitio web de forma racional y lógica.

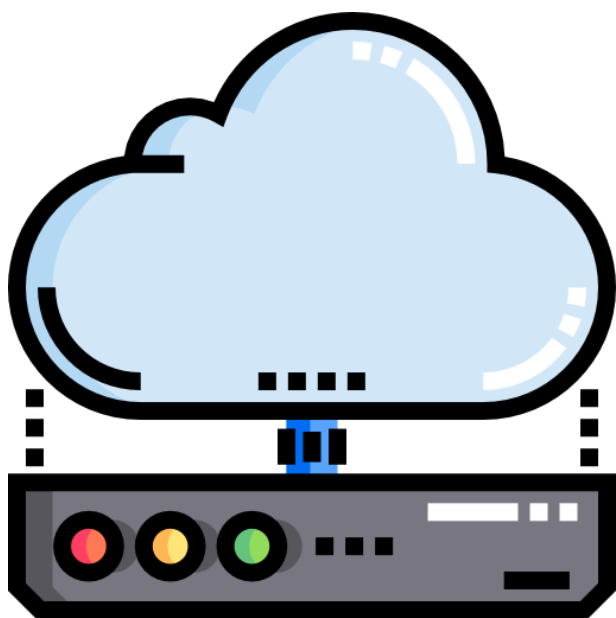
Esta jerarquización dependerá de las tecnologías que se usen y sobre todo del coste que se disponga en el proyecto.



# ARQUITECTURAS WEB



Actualmente para que una aplicación web funcione de manera correcta es necesario que disponga de los siguientes elementos:



Servidor web



Servidor web

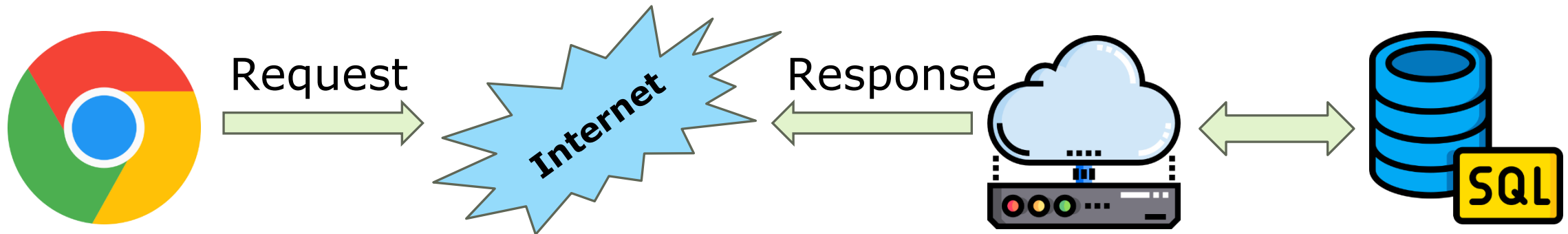


Servidor web

**Servidor web:** Es el servidor o cerebro de la arquitectura, donde está escuchando las peticiones HTTP de cliente web.

**Base de datos:** Conjunto de datos organizados jerárquicamente al servicio del servidor web.

**Cliente web:** Encargado de realizar las peticiones al servidor web, mediante un navegador.

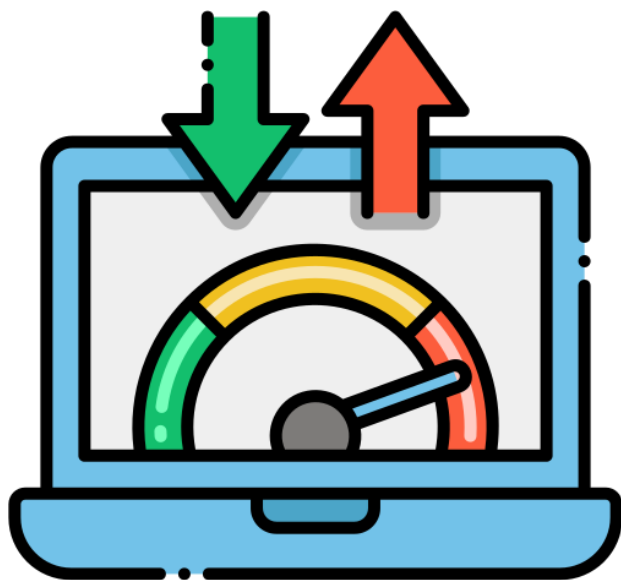


# EVOLUCIÓN DE LA TECNOLOGÍA WEB

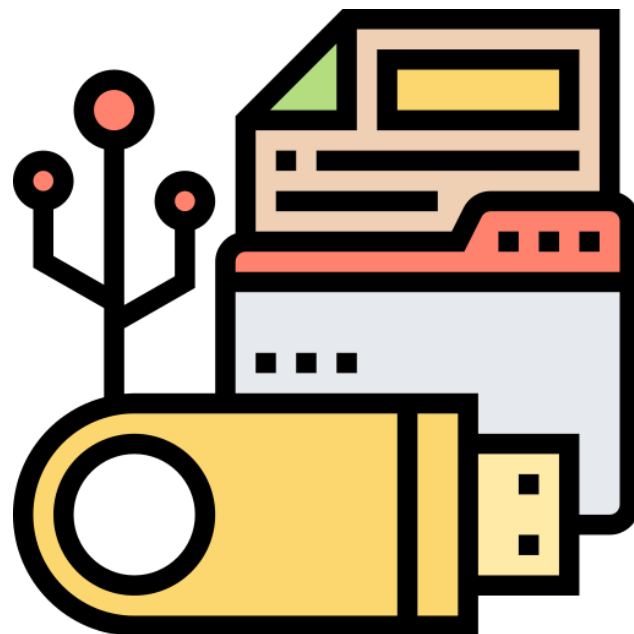




Desde que comenzó la era de la información, la tecnología ha evolucionado de una forma exponencial y la programación web lo hizo debido a avances tales como:



Ancho de banda

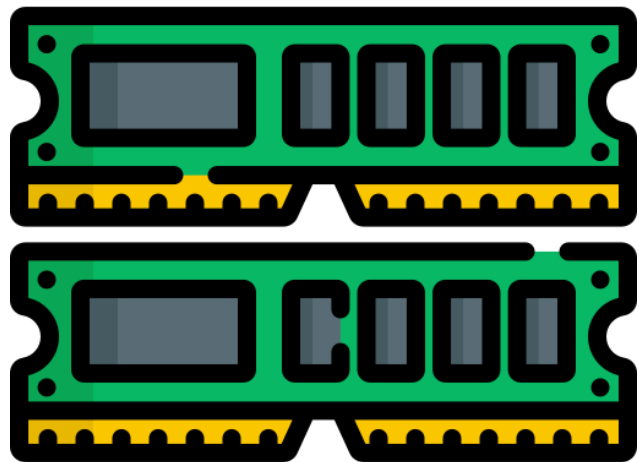


Almacenamiento



Información

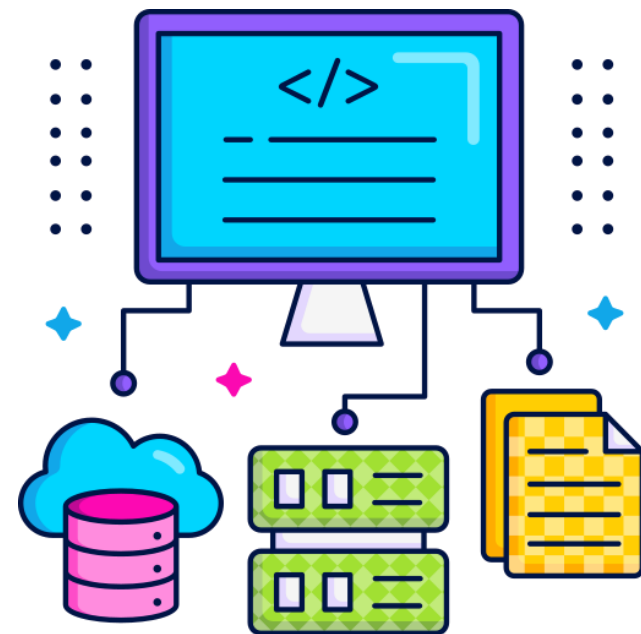
Desde que comenzó la era de la información, la tecnología ha evolucionado de una forma exponencial y la programación web lo hizo debido a avances tales como:



Computación



Tecnología



Infraestructuras

- **Ancho de banda:** Hardware de comunicaciones muy costoso y poco veloz, actualmente es más económico y de mayor velocidad.
- **Almacenamiento:** Hoy en día se usan cantidades ingentes de información las cuales se pueden almacenar en los dispositivos actuales.
- **Información:** Antes la información era estática, pero actualmente es dinámica y existe más interacción con el usuario.
- **Computación:** Los procesadores, memorias y HDD actuales hacen que las operaciones se realicen más rápido que en el pasado.
- **Tecnología:** Actualmente existen mayores tecnologías para el desarrollo de aplicaciones si lo comparamos con los existentes en el pasado.
- **Infraestructuras:** Alta disponibilidad, backup remoto, duplicidad de nodos en caso de fallo hardware y software.

Los anteriores avances hicieron que la web evolucionara de la siguiente manera.



Web 1.0



Web 1.5



Web 2.0

### 3. Evolución de la tecnología web

- **Web 1.0:** Contenido estático, sin protocolos de seguridad y estaban implementadas en su totalidad con el lenguaje de marcas HTML.
- **Web 1.5 :** Evolución de la anterior en la que las páginas web ya era más dinámicas, debido a su iteración con las bases de datos .
- **Web 2.0 :** También llamada **web social** significó una gran revolución. Este tipo de web tienen una iteración total con el usuario, algunos de los ítems de este modelo son:



Uso de CSS



Uso de JSON



Ajax



Redes sociales



Blogs

- **Web 3.0** : También llamada **web semántica**, fue el siguiente salto de la web, orientado a la inteligencia artificial, algunos de los avances de este modelo son:



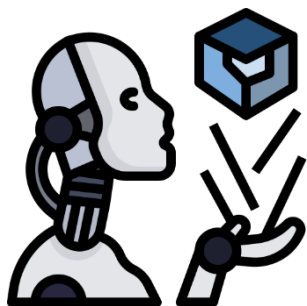
Diseño responsive



Web multimedia



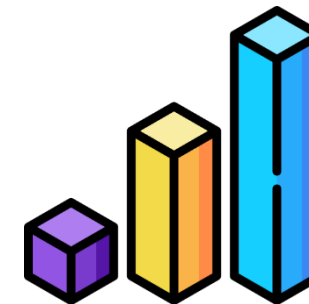
Apps inteligentes



Uso de I.A



Uso CSS3



Web 3D

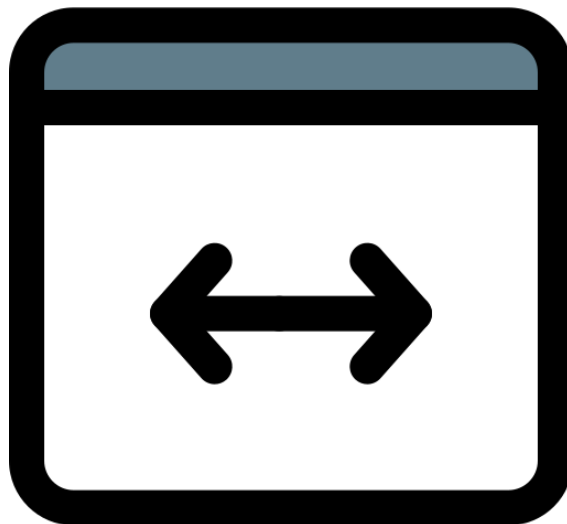
# TECNOLOGÍAS USADAS EN APLICACIONES WEB

4

Actualmente las aplicaciones web usan páginas dinámicas, que se ejecutan en el servidor y se renderizan en el cliente. A continuación, vamos a ver las distintas tecnologías que se ejecutan tanto en el cliente, como en el servidor y cuales se ejecutan en ambos.



Cliente



Ambos



Servidor



## Lado del servidor



ASP.NET



Java



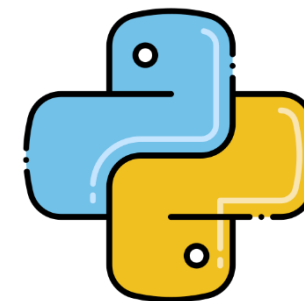
Ruby



Perl



PHP



Python

## Lado del servidor



Javascript



Spring

## Lado del servidor

- **ASP.NET:** Framework de desarrollo libre de Windows orientado a objetos.
- **Java:** La empresa de Sun Microsystems, desarrollo multitud de tecnologías para funcionar del lado del servidor, tales como, JSF y JSP, siendo esta última la más popular.
- **Ruby:** Lenguaje interpretado de propósito general dinámico y flexible.
- **Perl:** Lenguaje de programación que tiene muchas características del lenguaje C y destaca a la hora de procesar texto.

## Lado del servidor

- **PHP:** Lenguaje de propósito general para el desarrollo de backend. Es muy usado y extendido y existen Frameworks como Laravel y Symfony que están basados en este lenguaje.
- **Python:** Lenguaje interpretado muy poderoso, fácil de aprender y de alto nivel. Se puede usar en multitud de aplicaciones y sobre todo en aplicaciones de seguridad, es un lenguaje orientado a objetos, que destaca por su tipificado dinámico.
- **Javascript:** Lenguaje ligero, interpretado y orientado a objetos. Para aprenderlo es preciso un conocimiento de HTML y CSS y es usado esencialmente en la parte más dinámica de la aplicación y la iteración con el usuario.

## Lado del cliente



HTML



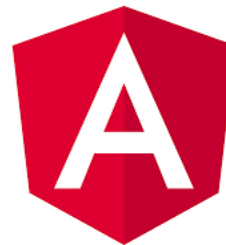
CSS



Javascript



VBScript



Angular



Node JS

## Lado del cliente



Thymeleaf



Typescript

## Lado del cliente

- **HTML y CSS:** HTML es el lenguaje de marcas que nació junto con el navegador y es el medio para transmitir información entre el cliente y el servidor. Posteriormente surgió CSS que permite diseñar gráficamente la página como tal.
- **Javascript:** Ya comentado anteriormente, se puede usar tanto en la parte servidor como en la cliente. Es soportado por la mayoría de los navegadores.
- **VBScript:** Lenguaje creado por Microsoft, que ha caído en desuso debido a que no es compatible con muchos navegadores. Está basado en el lenguaje Visual Basic.

## Ambos lados

- **DHTML:** No es un lenguaje de programación como tal sino un conjunto de todos los elementos de una página, como, por ejemplo, el fondo, posiciones de controles, cajas de contenido dinámico, etc.



DHTML



# SERVIDORES Y APLICACIONES LIBRES Y PROPIETARIAS

5

En la actualidad existe diferentes plataformas y servidores tanto libres, como privativos para poder desarrollar aplicaciones en alguno de los lenguajes visto anteriormente.

Antes de ver cuales son definiremos el como esta montada la arquitectura web para poder entender el origen de estas.



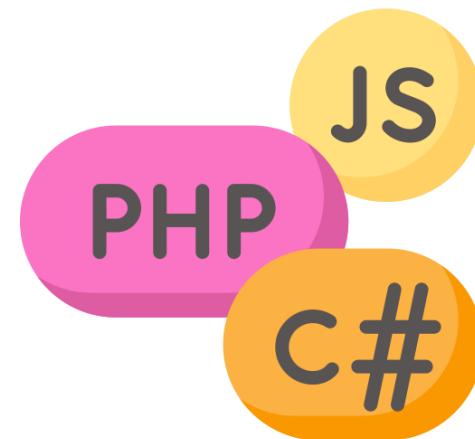
Sistema operativo



Servidor web



Base de datos



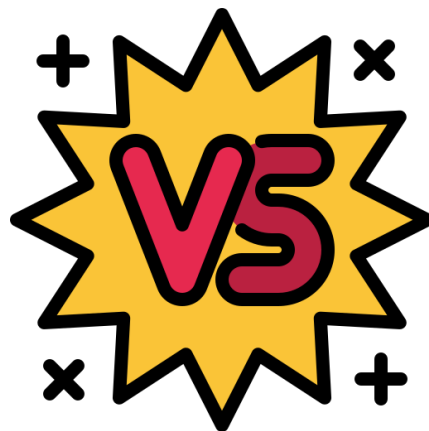
L. Programación

- **Sistema operativo:** Es fundamental, ya que será el software sobre el se sustentan los demás componentes.
- **Servidor web:** Es el servicio que atenderá las peticiones por parte del cliente y el cual deberá responder en la mayor brevedad posible.
- **Base de datos:** Es la principal fuente de información de la que se nutre el servidor web y que mostrará al cliente. Existen dos tipos **SQL** y **NoSQL**.
- **Lenguaje de programación:** Es el lenguaje que se encarga de la lógica de la aplicación. Tiene como función recibir las peticiones (**request**) del cliente, consultar la información necesario en la base de datos y posteriormente dar una respuesta al cliente (**response**)

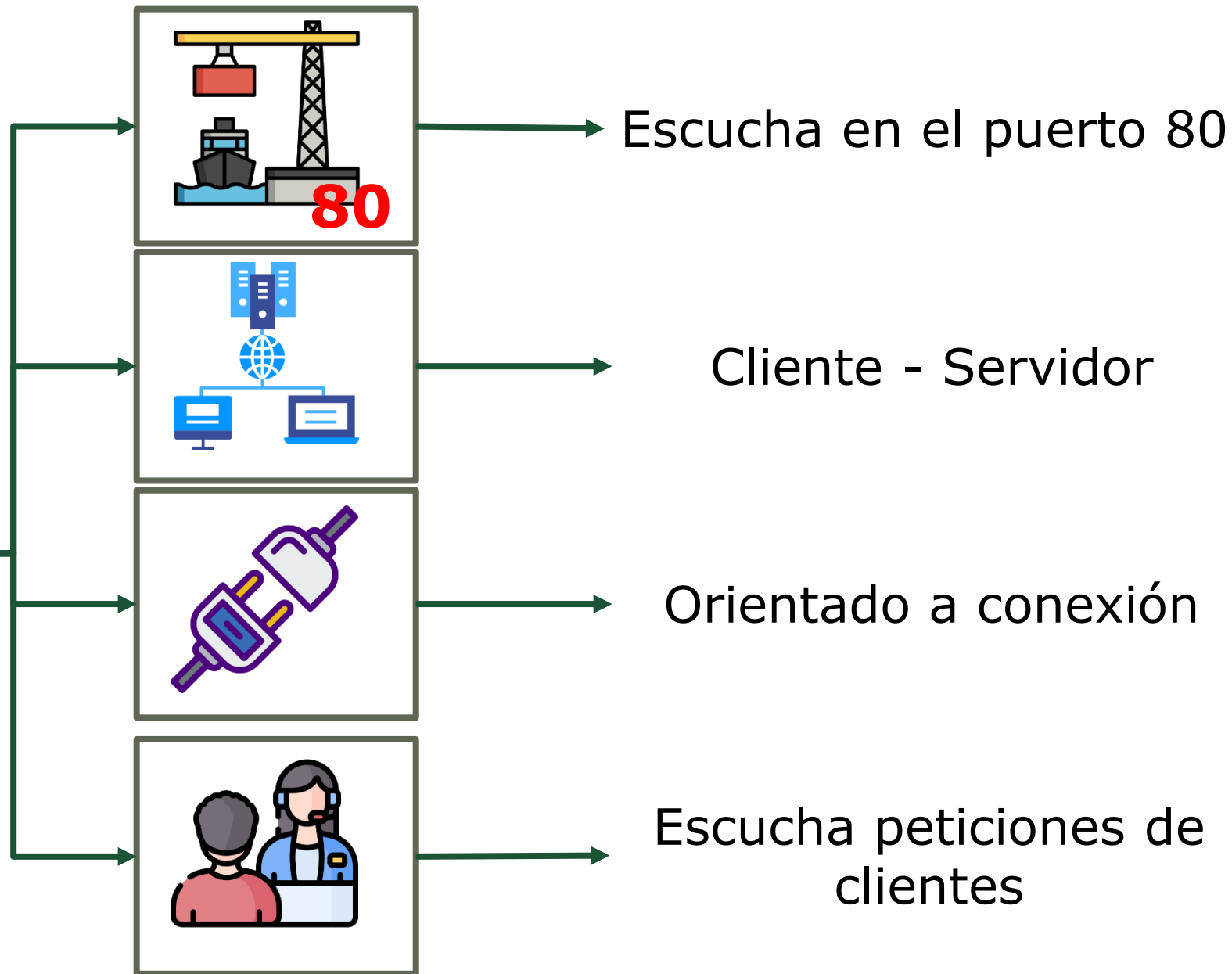
# PROTOCOLO HTTP Y HTTPS



Categoría Aplicación



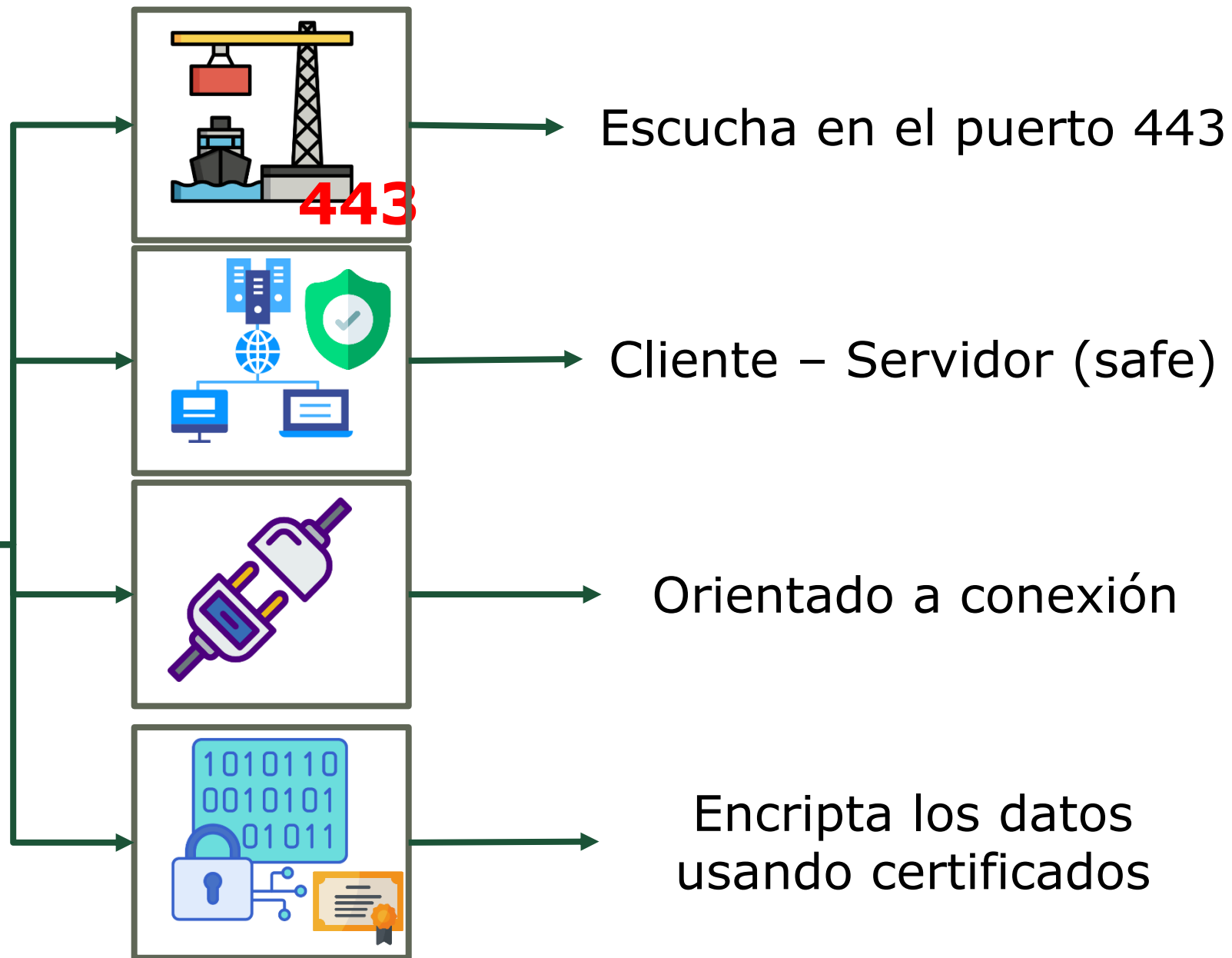




### HTTP:

Es un protocolo de la capa de aplicación, que escucha en el puerto 80 basado en el modelo cliente-servidor. Se basa en TCP, por lo tanto, es orientado a conexión y escucha las peticiones de los clientes. Una vez aceptada la conexión, se encarga de mantenerla y garantizar la transferencia sin errores.





### HTTPS:

Es un protocolo de la capa de aplicación que escucha en el puerto 443, se basa en el modelo cliente-servidor, pero en modo seguro. Se basa en TCP, es orientado a conexión y encripta los datos para asegurar la transmisión de extremo a extremo. Para realizar la encriptación hace uso de certificados.

Este protocolo usa encriptación SSL (Secure Socket Layer) que encripta los datos sensibles (actualmente denominado TLS).

A continuación se explica como funciona este protocolo cuando se teclea en el navegador:

- 1- El usuario teclea una dirección web en el navegador https://ww.....
- 2- Se traduce el dominio DNS a la IP correspondiente.
- 3- Se busca en el servidor web la IP de la página solicitada por el puerto TCP 443
- 4- Antes de transferir la información al navegador, se negocia mediante TLS, que consiste en el envío del certificado al cliente, y el cliente debe aceptar este certificado. Una vez aceptado, se usará un canal cifrado con la información.
- 5- Por último, se realiza la petición HTTP y se envía la respuesta al cliente.

# CLASIFICACIÓN DE SERVIDORES DE APLICACIONES DEL MERCADO ACTUAL



## 7. Clasificación de servidores de aplicaciones del mercado actual

En el mercado actual tan cambiante, existen multitud de servidores preparados para ser instalados en cualquier plataforma. A continuación, daremos algunas pinceladas de algunos de ellos.



Apache



Mac OS X Server



Nginx



LiteSpeed



Microsoft IIS

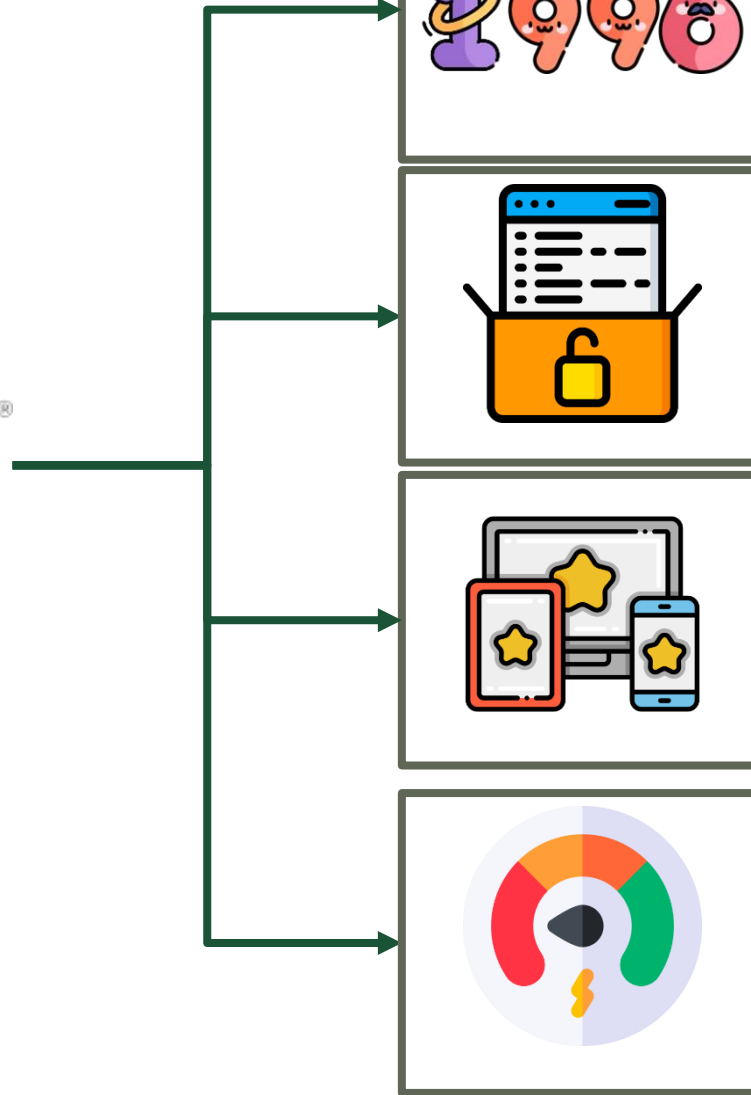


GWS



Sun Java System  
Web server

## 7. Clasificación de servidores de aplicaciones del mercado actual

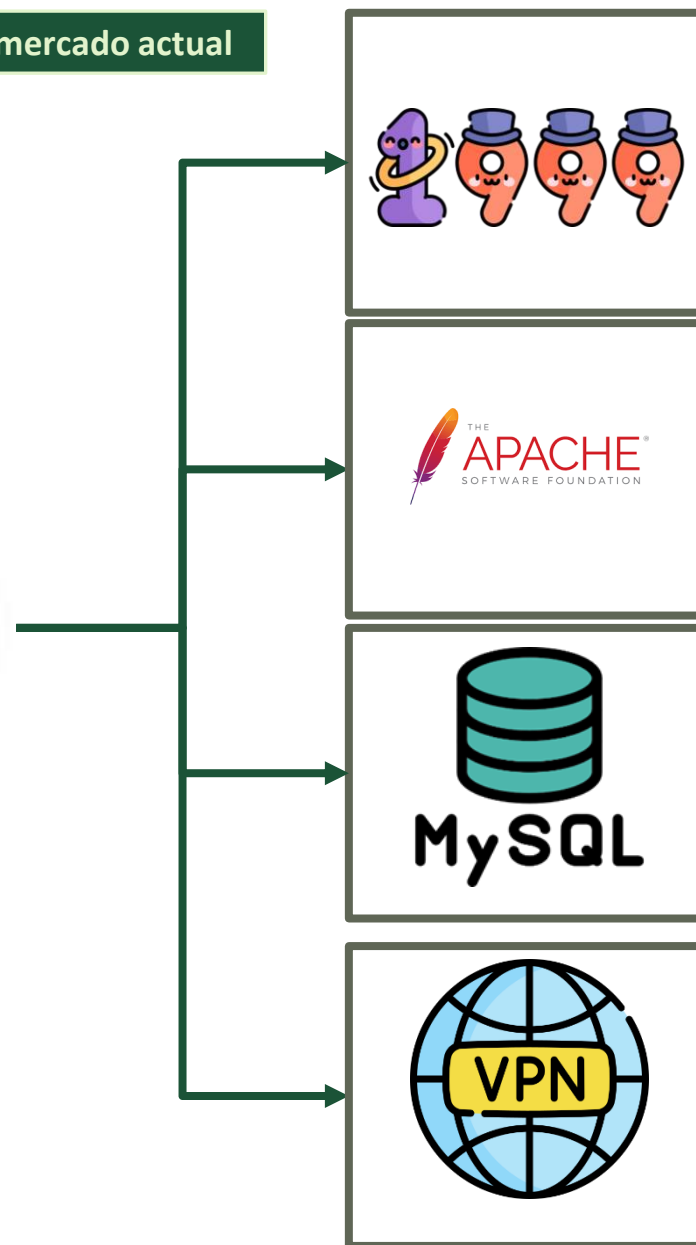


- La primera versión de este servidor fue en 1996.
- Tiene como grandes competidores a Nginx y Microsoft IIS.
- Tiene como ventaja que es de código abierto y multiplataforma.
- Algunas de sus desventajas es que tiene bajo rendimiento cuando se realizan muchas peticiones.

## 7. Clasificación de servidores de aplicaciones del mercado actual



Mac OS X Server

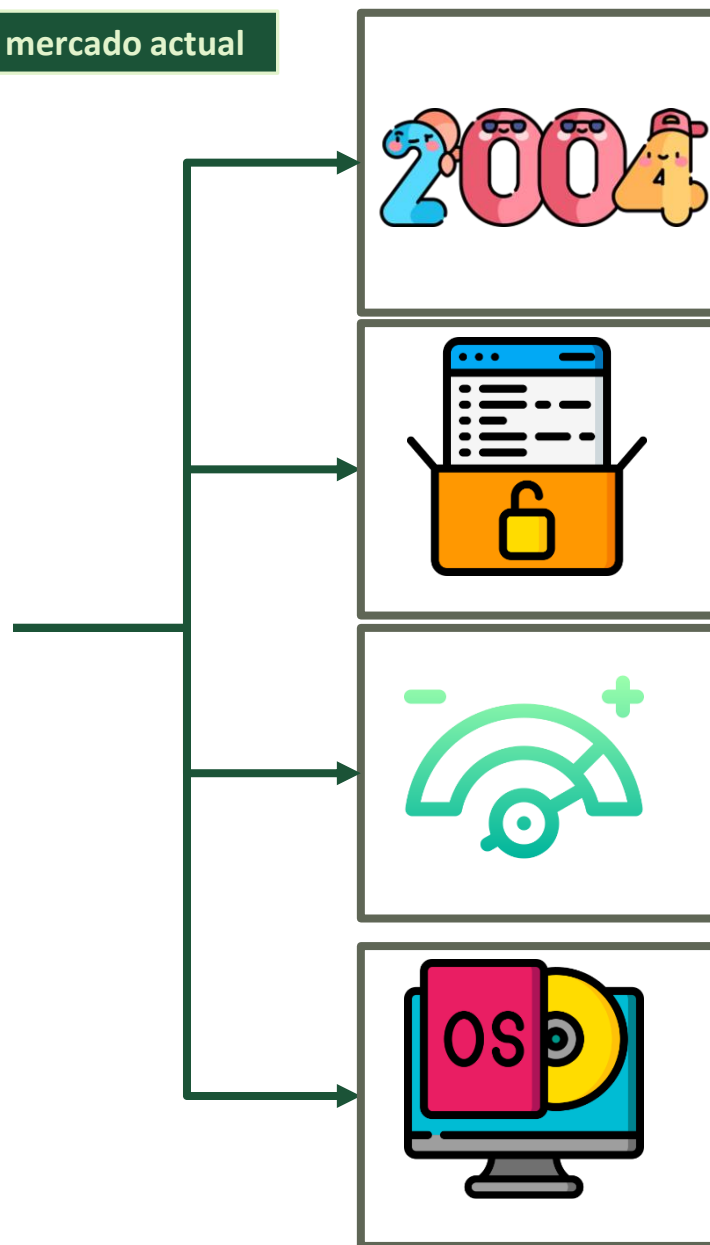


- La primera versión de este servidor fue en 1999.
- Incluye entre otros servicios Apache para atender conexiones de clientes.
- Incluye como BBDD a MySQL.
- Incluye servicio NAT, servidor DHCP, servidor VPN, PHP y alguno más.

## 7. Clasificación de servidores de aplicaciones del mercado actual



Nginx



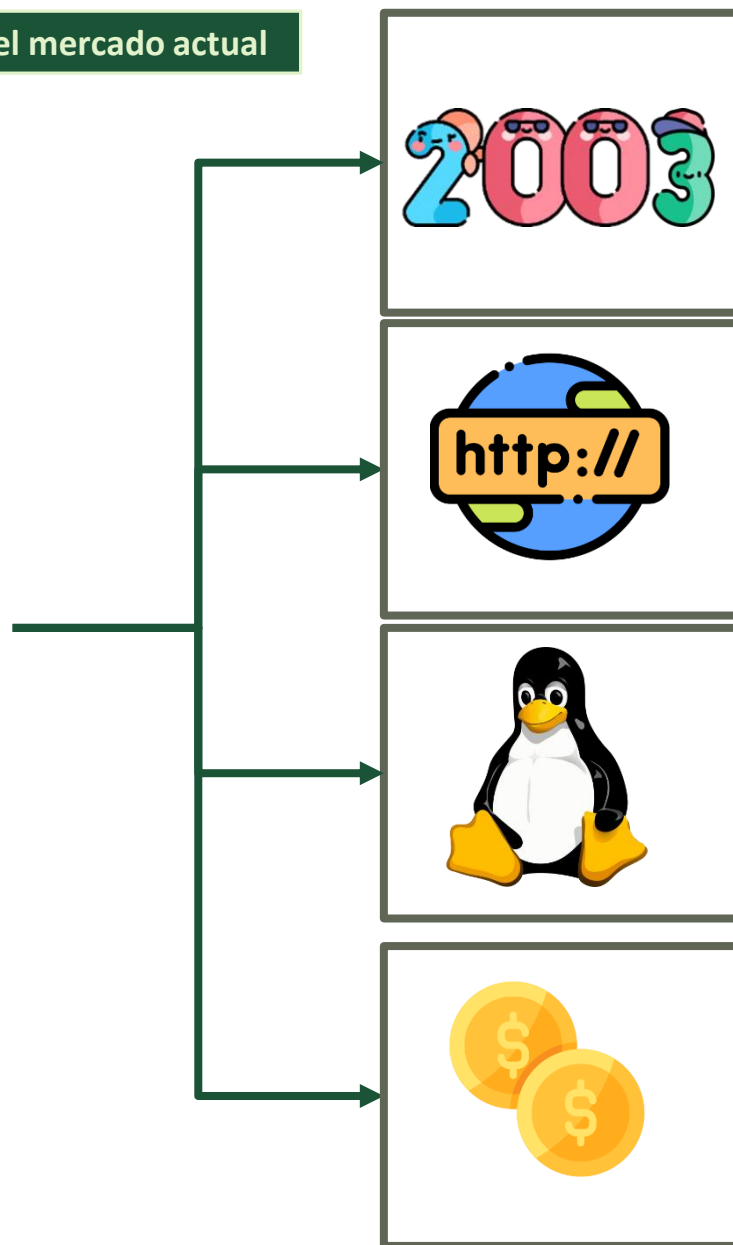
- La primera versión de este servidor nace en 2004.
- Es de código abierto y gratuito.
- Destaca por su alto rendimiento en funciones como servidor de proxy inverso HTTP, balanceador, POP3 e IMAP.
- Se puede instalar en múltiples SSOO.



## 7. Clasificación de servidores de aplicaciones del mercado actual



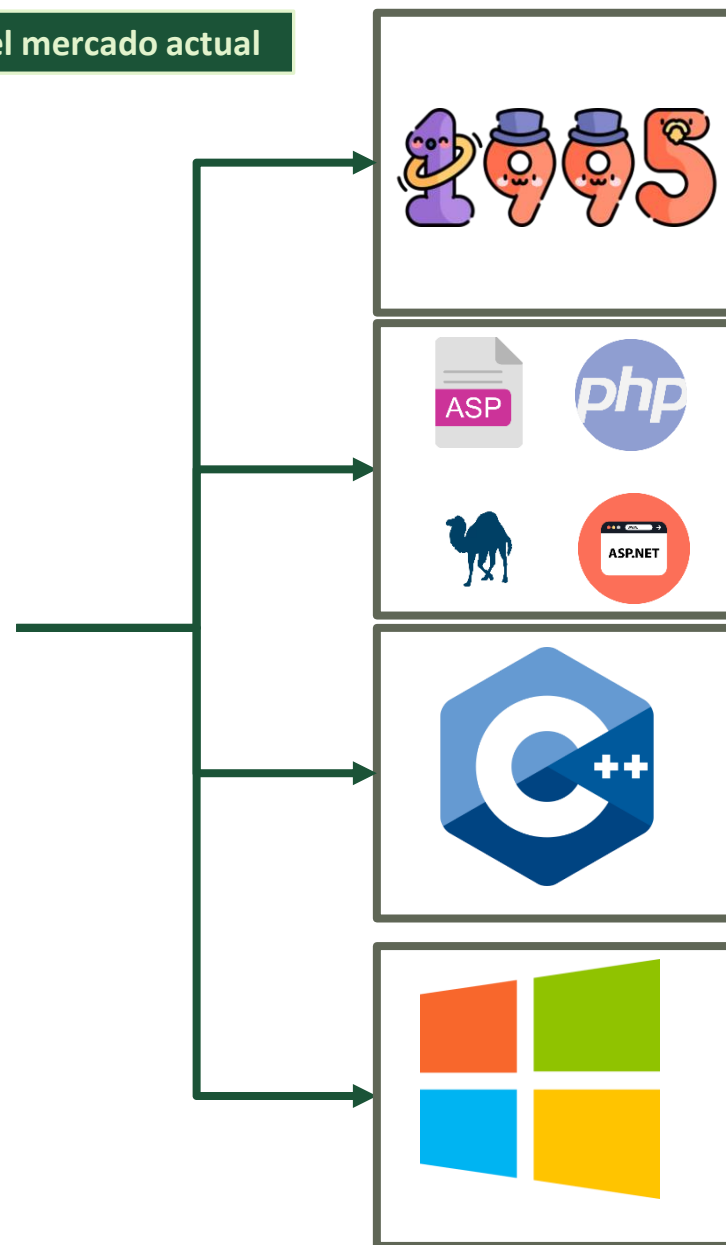
LiteSpeed



- La primera versión de este servidor fue en 2003.
- Servidor que acepta peticiones HTTP y orientado a software libre.
- Acepta muchas conexiones simultaneas.
- La versión completa es de pago.



Microsoft IIS

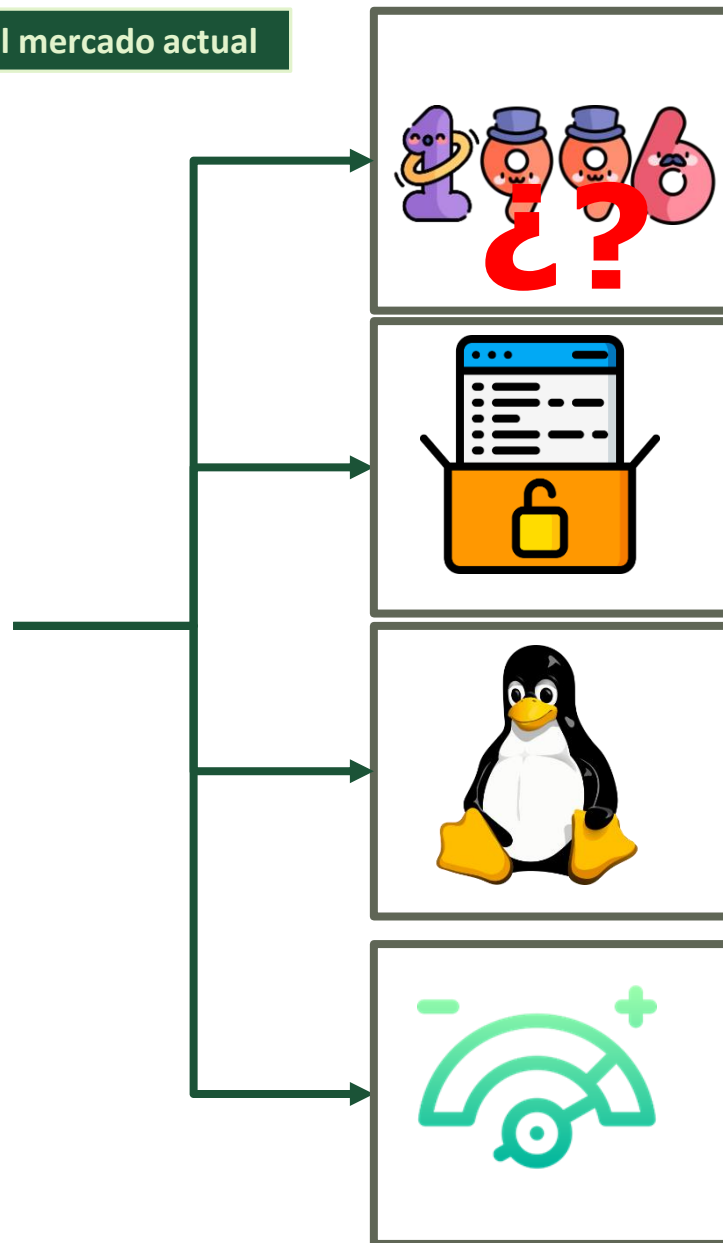


- La primera versión de este servidor fue en 1995.
- Permite el procesamiento de páginas ASP, ASP.NET, PHP o Perl.
- Está programado en C++.
- Solo está disponible para Windows, con lo que conlleva un gran riesgo en la seguridad.

## 7. Clasificación de servidores de aplicaciones del mercado actual

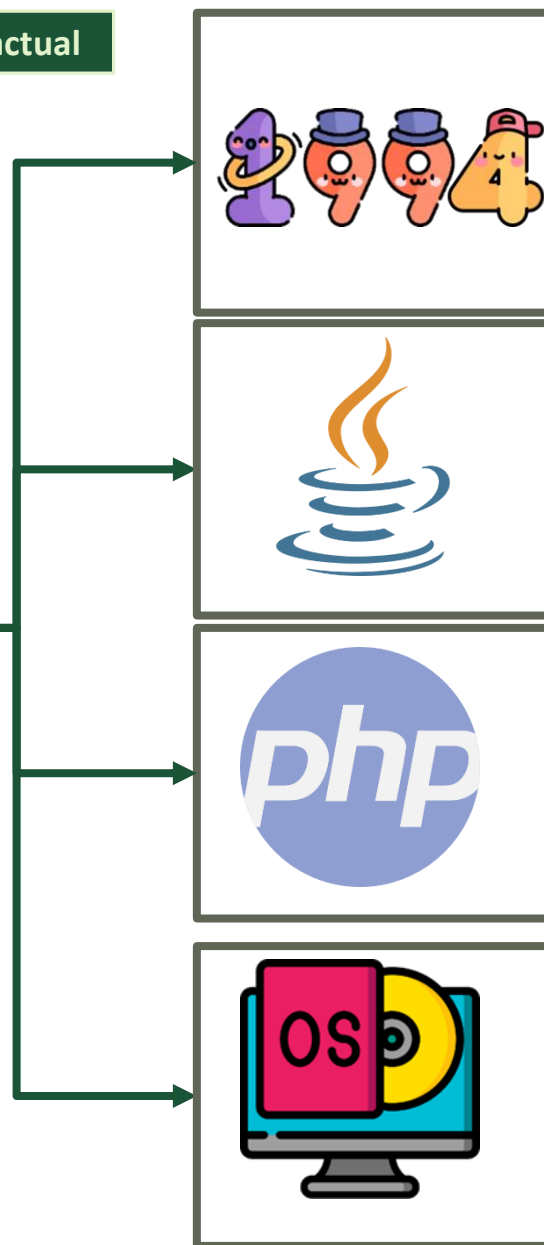


GWS



- Se ejecuta en sistemas operativos como GNU/Linux, por tanto, es software libre.
- Tiene un excelente rendimiento.

## 7. Clasificación de servidores de aplicaciones del mercado actual

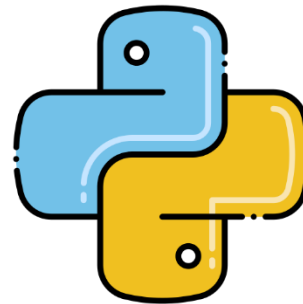


- La primera versión de este servidor fue en 1994, actualmente el servidor se llama Oracle Iplanet Web Server.
- Es un servidor orientado a la programación con Java con licencia BSD.
- Da soporte webDA, JDBC, PHP....
- Es multiplataforma, se puede instalar en todos los SSOO, menos en Mac.

# ESTRUCTURA Y RECURSOS DE UNA APLICACIÓN WEB

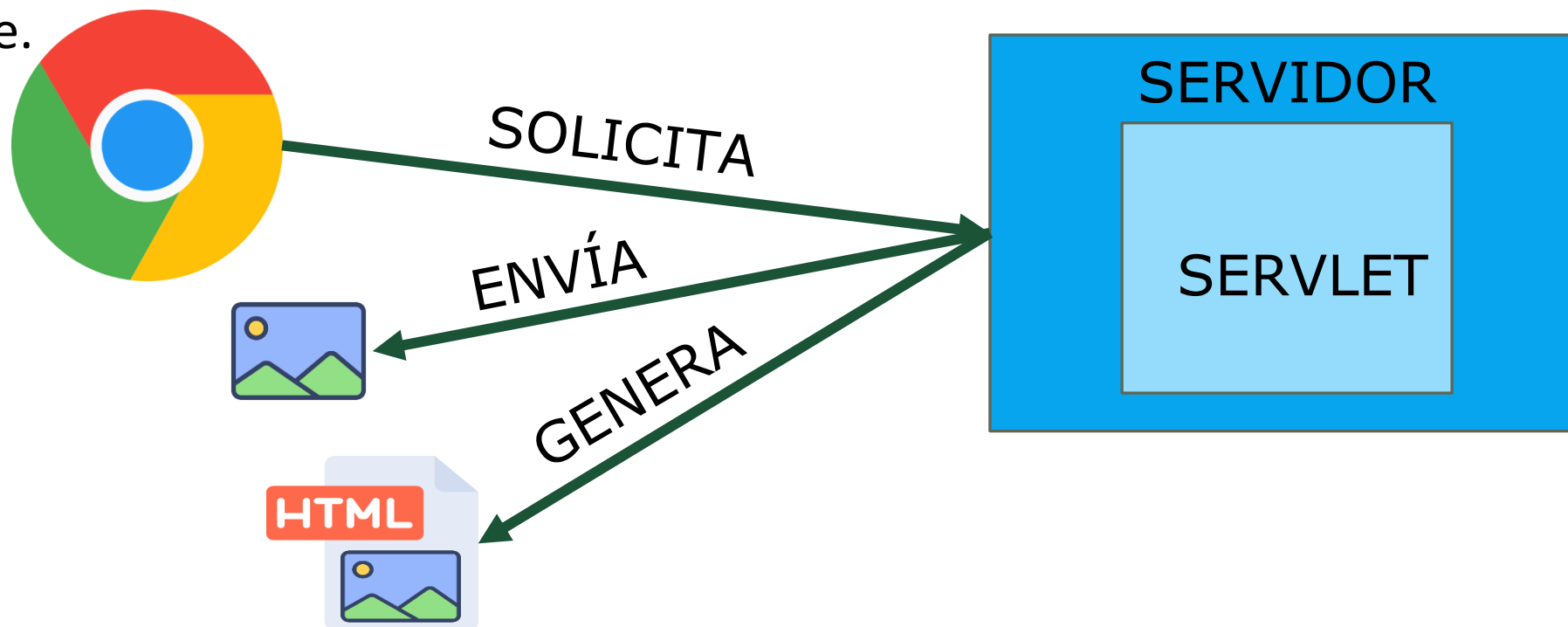


Las aplicaciones web se basan en la versión Java EE 8 Web y en un servidor de aplicaciones que deben especificar de alguna forma como se van a estructurar los ficheros para que la aplicación se despliegue de forma óptima. Para ello se usa un fichero denominado **descriptor de despliegue**. Una aplicación web está compuesta físicamente por:



De los indicados anteriormente el más importante es el servlet, y estos se encargan de una función muy específica dentro de la aplicación web.

La última versión de estos, permiten el uso de HTTP 2.0 que permite operaciones push. Esto es que el servidor puede enviar recursos al cliente sin que este realmente los necesite.



Cuando se crea un proyecto de una aplicación web se crea el directorio raíz y a partir de este se crean todos los directorios y ficheros importantes de la aplicación, los cuales detallamos a continuación en la siguiente tabla.

| Directorio       | Descripción   |
|------------------|---|
| WEB-INF          | Es un área privada de la aplicación donde no se puede acceder directamente desde el navegador. Aquí se almacenan los componentes de la configuración del archivo .war. El fichero más importante de este directorio es el fichero web.xml o descriptor de despliegue. |
| WEB-INF/classes/ | Tiene como componentes las clases Java usadas en el archivo .war y usualmente se encuentran los servlets codificados.   |
| WEB-INF/lib/     | En este directorio se ubicarán los ficheros .jar usados por la aplicación, por ejemplo, para conectarse a una base de datos, o para usar Hibernate o cualquier otra librería que requiera la aplicación.  |
| META-INF         | Es un directorio privado de la aplicación y contiene normalmente un fichero denominado <b>context.xml</b> que permite definir el contexto de la aplicación.   |



- **Archivo war (Web Application Archive):**

Este archivo es una de las partes más importantes de la aplicación web, ya que es el método de encapsular una aplicación y prepararla para la distribución en los servidores compatibles con servlet y páginas jsp. Este fichero se puede generar de diversas formas, como un IDE de programación o con la herramienta jar.

El contenido de este fichero puede ser el siguiente:

- ✓ Servlets, JSP o cualquier lenguaje de programación elegido para tal fin.
- ✓ Información web: XML, HTML, sonidos, vídeos, etc
- ✓ Recursos web.

- **Descriptor de despliegue:**

El descriptor de despliegue es un fichero XML que describe como ha de desplegarse una aplicación web, un modulo o cualquier componente, en caso de no existir la aplicación no funcionará. Este fichero suele estar ubicado en la carpeta WEB-INF/ y su nombre suele ser web.xml.

```
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1">
  <servlet>
    <servlet-name>comingsoon</servlet-name>
    <servlet-class>mysite.server.ComingSoonServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>comingsoon</servlet-name>
    <url-pattern>/*</url-pattern>
  </servlet-mapping>
</web-app>
```

- **Descriptor de despliegue:**

Como punto final indicar que los servlet deberán estar dentro de las etiquetas **<web-app>** y **</web-app>**, tal y como se puede apreciar en la captura de la diapositiva anterior.