

UD 1

# IMPLANTACIÓN DE APLICACIONES WEB

---

DESPLIEGUE DE APLICACIONES WEB

CFGS DESARROLLO DE APLICACIONES WEB

# CONTENIDOS

---

- Arquitecturas Web. Modelos. Concepto, características, ventajas e inconvenientes.
- Servidores Web. Instalación y configuración básica. Protocolos para su funcionamiento.
- Servidores de aplicaciones. Concepto y clasificación. Instalación y configuración básica.
- Estructura y recursos que componen una aplicación Web. Descriptor de despliegue.

# Índice

---

1. Aspectos generales de la aplicación web
2. Evolución de los servicios web
3. Tecnologías asociadas a las aplicaciones web
4. Tipos de aplicaciones
5. Arquitectura web. Modelos
6. Plataformas web libres y propietarias
7. Escalabilidad
8. Servidores web
9. Características deseables en un servidor web
10. IaaS, PaaS y SaaS – Modelos de servicio de Cloud
11. Aplicaciones web y servidores de aplicaciones
12. Estructura y despliegue de una aplicación web

# 1.Aspectos generales de la aplicación web

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB



# Aspectos generales de la aplicación web

---

La arquitectura **World Wide Web (WWW)** de Internet provee un modelo de programación sumamente poderoso y flexible. Las aplicaciones y los contenidos son presentados en formatos de datos estándar y son localizados por aplicaciones conocidas como "web browsers", navegadores, que envían requerimientos de objetos a un servidor y éste responde con el dato codificado según un formato estándar.

# Aspectos generales de la aplicación web

---

Los estándares WWW especifican muchos mecanismos para construir una aplicación de propósito general, por ejemplo:

- **Modelo estándar de nombre:** Todos los servidores, así como su contenido de la WWW se denominan según la URL ( Uniform Resource Locator).
- **Contenidos:** Todos los contenidos en la WWW se le especifica un determinado tipo permitiendo que los browser lo interpreten adecuadamente.
- **Formato de contenidos estándar.** Todos los navegadores soportan un conjunto de formatos , ejemplo: HTML, JavaScript, TypeScript, etc
- **Protocolos estándar:** éstos permiten que cualquier navegador pueda comunicarse con el servidor web. El más común usando en WWW es HTTP, que opera sobre un conjunto de protocolos de TCP/IP.

# Aspectos generales de la aplicación web

---

**Los aspectos generales a destacar de una arquitectura web son los siguientes:**

- Escalabilidad
- Separación de responsabilidades (premisa para la separación de capas).
- Portabilidad - Una aplicación web se adapta a distintas arquitecturas físicas posibles para el despliegue.
- Utilización de componentes en los servicios de infraestructura.
- Gestión de las sesiones del usuario.

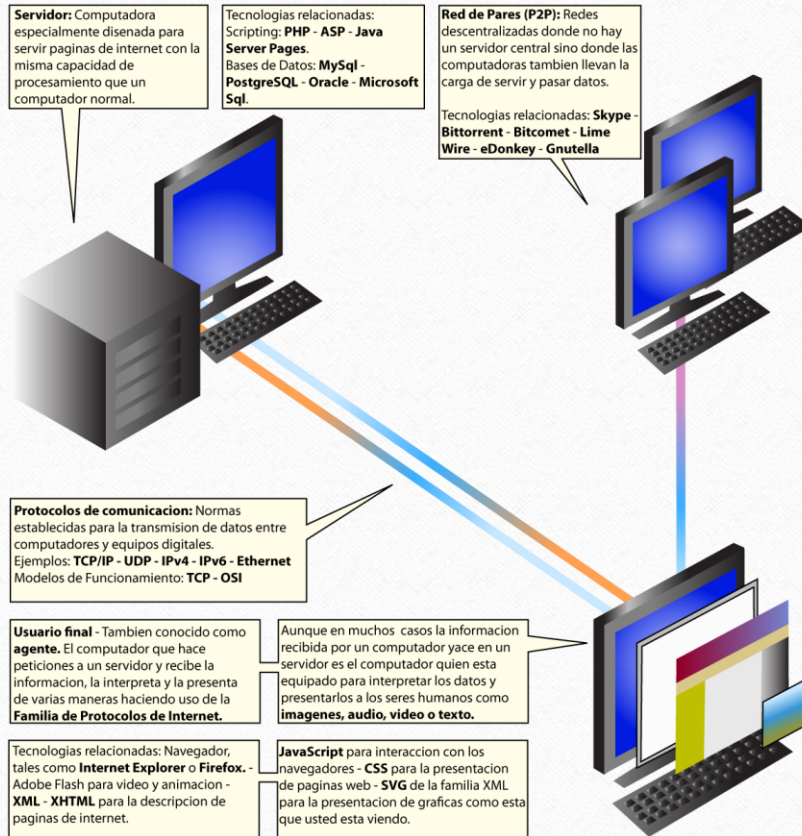
# Aspectos generales de la aplicación web

---

La arquitectura web podríamos decir que es un modelo de tres capas:

- **Capa de Base de Datos**, donde estaría toda la documentación de la información que se pretende administrar mediante un servicio web.
- **Capa de Servidores de aplicaciones web**, ejecutando aplicaciones de tipo Apache, Tomcat,...
- La tercera capa, estaría los **clientes del servicio web** al que accederán mediante un navegador.





[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA](#)

## 2. Evolución de los servicios web

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB

# Evolución de los servicios web

---

- **El contenido se está volviendo más dinámico:** contenidos instantáneos y provenientes de fuentes muy diferentes.
- **El ancho de banda es menos costoso.** A medida que crezca el ancho de banda, los servicios web deben adaptarse a nuevos tipos de contenidos.
- **El almacenamiento es más barato.** Un servicio web debe ser capaz de manejar cantidades masivas de datos y de forma inteligente.
- **El éxito de la computación extendida.** Con cientos de millones de dispositivos como teléfonos móviles, tablets, coches, etc. A medida que las plataformas se hacen más diversas, tecnologías como XML, JSON, son más importantes. Un servicio web debe servir para todo tipo de dispositivos, plataformas y navegadores, entregando contenido sobre una amplia variedad de tipos de conexión.

# Evolución de los servicios web

---

Web paso de aplicaciones tradicionales a aplicaciones web enfocadas al usuario final. Las nuevas tecnologías, tienen por objeto:

- Transformar software de escritorio hacia la web.
- Separar las hojas de estilos.
- Potenciar el trabajo colaborativo y la utilización de redes sociales.
- Dar control a los usuarios en el manejo de su información.



# 3. Tecnologías asociadas a las aplicaciones web

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB

# Tecnologías asociadas a las aplicaciones web

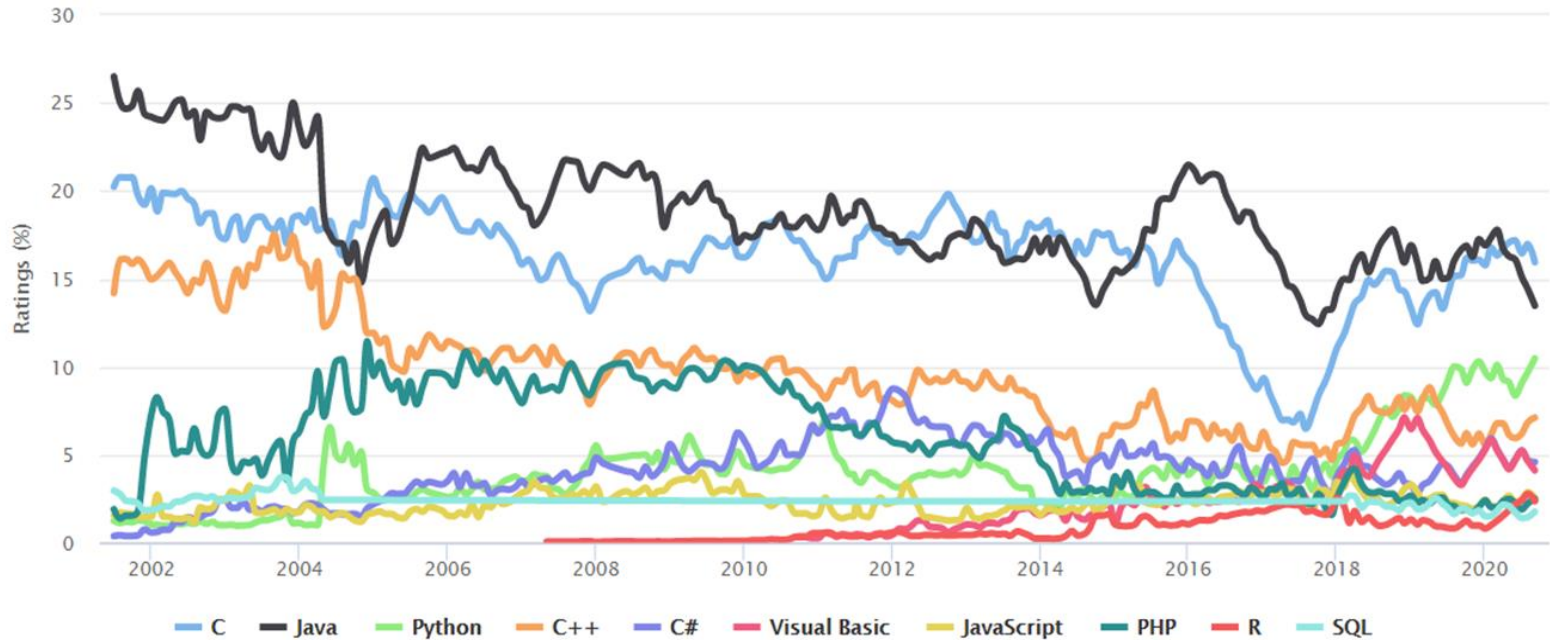
---

Las tecnologías asociadas a las aplicaciones web que se ejecutarán tanto del lado del servidor como del cliente:

- [CGI](#) (Common Gateway Interface): La "Interface Común de Entrada" es un estándar antiguo usado para trasladar información del cliente al servidor. Las rutinas CGI son habitualmente escritas en lenguajes interpretados como Perl o por lenguajes compilados como C.
- **Java.** Este es un lenguaje que trabaja en el servidor.
- [JavaScript](#) / [TypeScript](#). Lenguaje que se interpreta y se ejecuta en el cliente.
- [PHP](#) (Hypertext Preprocessor): Lenguaje del lado del servidor.
- [ASP.NET](#) Es una versión evolucionada de ASP, integrada en la tecnología .NET de Microsoft junto con el lenguaje C#.
- [Ruby on Rails](#), [Grails\(Groovy\)](#), [Django\(Phyton\)](#), [Perl](#), [ColdFusión](#) y muchas más

## TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



# 4. Tipos de aplicaciones web

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB



# Tipos de aplicaciones web

---

- Aplicaciones Web Estáticas
- Aplicaciones Web Dinámicas
- [Aplicaciones Web del Portal](#)
- Aplicaciones de comercio electrónico
- Aplicaciones Web Animadas (por ejemplo páginas de juegos basados en esta tecnología Flash)
- Aplicaciones web con gestor de contenidos (CMS)



Flash Player dejará de ser compatible a partir de diciembre del 2020.

# 5. Arquitectura web. Modelos

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB

# Arquitectura web

---



# Arquitectura web. Modelos

---

- Capa de presentación o del navegador es la encargada de diseño de la web, tanto de tipo visual (colores, tipografías, imágenes, posicionamiento bloques, etc), así como la estructuración de los contenidos en diversas secciones. Aquí entra en juego el diseño gráfico, la usabilidad, navegabilidad, validación de los datos de entrada, formateo de los datos de salida, presentación de la web, etc.; se trata de la capa que se presenta al usuario.
- Las tecnologías usadas: HTML, CSS, DOM, JavaScript, AJAX, estándares web.



# Arquitectura web. Modelos

---

- Capa de negocio es la que recibe las peticiones del usuario y desde donde se le envían las respuestas; en esta capa se verifica que las reglas establecidas se cumplen.
- Lenguajes usados:

PHP, Java, con sus tecnologías Java Servlets y JavaServer Pages (JSP), Perl, Ruby, Python, Node.js, c# y Visual Basic con sus tecnologías ASP/ ASP.NET

# Arquitectura web. Modelos

---

- Capa de acceso a datos es la formada por determinados gestores de datos que se encargan de almacenar, estructurar y recuperar los datos solicitados por la capa de negocio.

## Arquitectura web. Modelos:

### Modelo 1

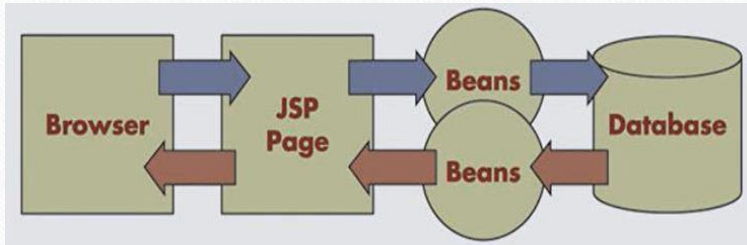
---



- Interfaz de entrada común (CGI) es una tecnología de World Wide Web que permite al cliente (navegador) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el servidor web. Normalmente se programan en C o en Perl.
- Modelo presentación, negocio y acceso se confundían. El estado se almacena en el cliente y cada petición supone una ejecución completa independiente del estado.

## Arquitectura web. Modelos: Modelo 1.5

---

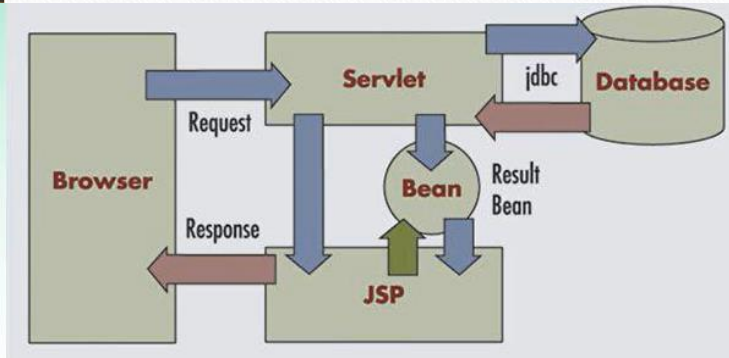


- Aplicado a la tecnología java, se da con la aparición de las JSP y los servlets. En este modelo, las responsabilidades de presentación recaen en las páginas JSP, mientras que los beans incrustados en las mismas son los responsables del modelo de negocio y acceso a datos.



## Arquitectura web. Modelos: Modelo 2

---



- Como evolución del modelo anterior, con la incorporación del patrón MVC en este tipo de aplicaciones, se aprecia la incorporación de un elemento controlador de la navegación de la aplicación. El modelo de negocio queda encapsulado en los javabeans que se incrustan en las páginas JSP.

## Arquitectura web. Modelos: Modelo 2X Aplicaciones multicanal

---

- Aparecen con el objetivo de dar respuesta a la necesidad, cada vez más habitual, de desarrollar aplicaciones multicanal, es decir, aplicaciones web que pueden ser atacadas desde distintos tipos de clientes remotos.
- Así, una aplicación web multicanal podrá ejecutarse desde una tablet, desde un terminal de telefonía móvil, o desde cualquier navegador HTML estándar. Un medio para lograr publicar la misma aplicación para distintos dispositivos es emplear plantillas XSL para transformar los datos XML.

# 6. Plataformas web libres y propietarias

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB

## 6. Plataformas web libres y propietarias

---

- El diseño de aplicaciones web requiere de un entorno funcional, que permita el desarrollo de cada uno de los componentes que forman la aplicación web, junto con un entorno complejo de herramientas que ofrecen al cliente los servicios de las aplicaciones web.
- En el mercado pone a disposición de los programadores un abanico de herramientas software, que puede ser de dos tipos: software libre o software propietario.



## 6. Plataformas web libres :LAMP

---

- La plataforma LAMP trabaja enteramente con componentes de software libre y no está sujeta a restricciones propietarias. El nombre LAMP surge de las iniciales de los componentes de software que la integran:
- Linux: Sistema operativo.
- Apache: Servidor web.
- MySQL: Gestor de bases de datos.
- PHP: Lenguaje interpretado PHP, aunque a veces se sustituye por Perl o Python

## 6. Plataformas web libres :WISA

---

- La plataforma WISA está basada en tecnologías desarrolladas por la compañía Microsoft; se trata, por lo tanto, de software propietario. La componen los siguientes elementos:
- Windows: Sistema operativo.
- Internet Information Services: servidor web.
- SQL Server: gestor de bases de datos.
- ASP o ASP.NET: como lenguaje para scripting del lado del servidor.

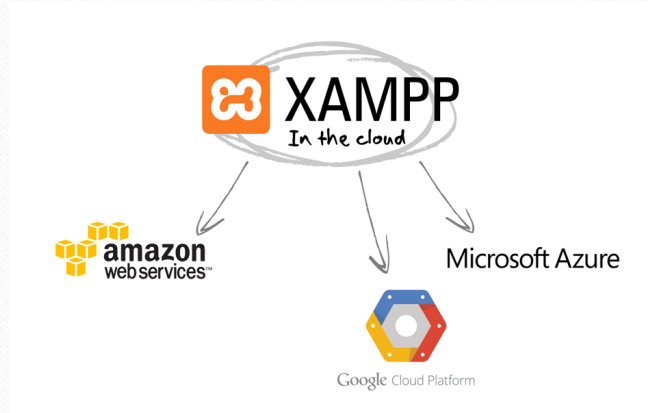
## 6. Plataformas web libres :XAMPP

---

¿Qué es XAMPP?

XAMPP Apache + MariaDB + PHP  
+ Perl

Alojamiento



# 7. Escalabilidad

---

Aspectos generales de la arquitectura web

UD 1 IMPLANTACIÓN DE APLICACIONES WEB



# Escalabilidad

---

- Se entiende por escalabilidad a la capacidad de adaptación y respuesta de un sistema con respecto al rendimiento del mismo a medida que aumentan de forma significativa el número de usuarios, transacciones, . . .
- Si un sistema tiene como objetivo crecer en el número de usuarios manteniendo su rendimiento actual, tiene que evaluar dos opciones:
  - Con hardware de mayor potencia
  - Con una mejor combinación de hardware y software

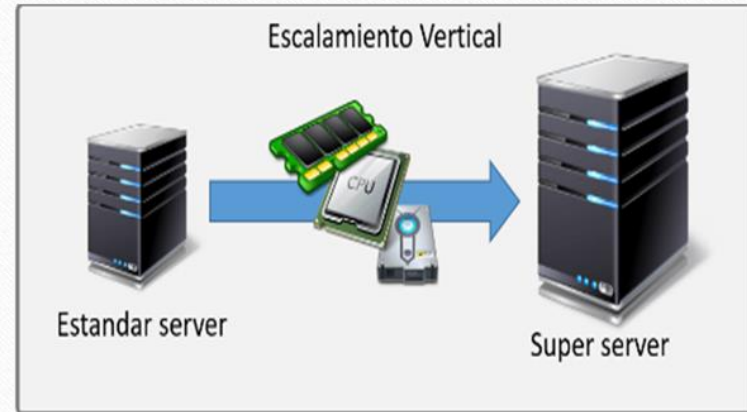
# Escalabilidad: Tipos

---

- Escalabilidad vertical
- Escalabilidad horizontal

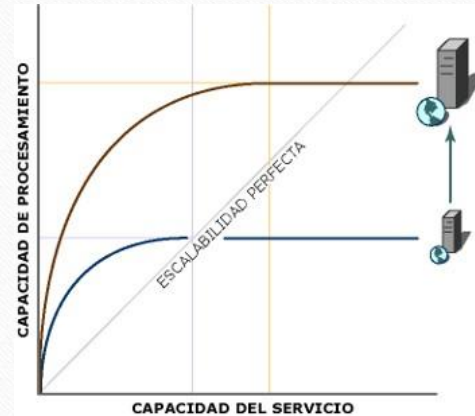
# Escalabilidad Vertical

- El escalar verticalmente o escalar hacia arriba, significa añadir más recursos a un solo nodo en particular dentro de un sistema, tal como añadir memoria , disco duro, procesador, etc. pero también puede ser la migración completa del hardware por uno más potente.



# Escalabilidad Vertical

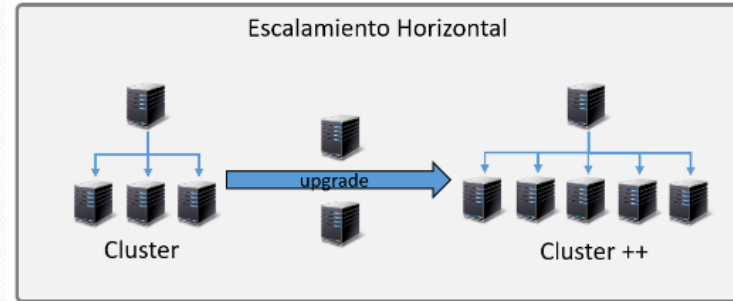
- Al escalar hacia arriba viene a significar una migración de todo el sistema a un nuevo hardware que es más potente y eficaz que el actual. Una vez configurado el sistema futuro, se realizan una serie de validaciones y copias de seguridad y se pone en funcionamiento.
- Aspecto negativo: Limitación hardware, los costos se disparan.





# Escalabilidad Horizontal

- La escalabilidad horizontal es el modelo más potente, significa agregar más servidores (conocidos como Nodos) trabajando como un solo nodo. Se crea una red de servidores conocida como **Clúster**, con la finalidad de repartirse el trabajo entre todos los nodos del clúster.



# 8. Servidores web

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB

## 8. Servidores web

---

- Un [servidor web](#) es un programa que se ejecuta de forma continua en un ordenador, se mantiene a la espera de peticiones por parte de un cliente (un navegador de Internet) y contesta a estas peticiones de forma adecuada, sirviendo una página web que será mostrada en el navegador o mostrando el mensaje correspondiente si se detectó algún error.

## 8. Servidores web





# 9. Características deseables de un servidor web

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB

## 9. Características deseables de un servidor web

---

- Que tenga soporte multitarea, que pueda ser capaz de servir más de una petición a la vez.
- Que sea rápido, y a ser posible que lo haga con los mínimos requisitos de hardware.
- Que aporte mecanismos de autenticación.
- Que se puedan personalizar los mensajes de error, para dar respuestas con sentido a los clientes en caso de fallos.
- Que soporte cuantos más formatos posibles: JPEG, GIF,...
- Que pueda funcionar como [servidor proxy](#).
- Que sea seguro ante ataques.
- Que sea fácil de administrar.

# 10. IaaS, PaaS y SaaS – Modelos de servicio de Cloud

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB

# IaaS Infrastructure as a Service

---

- **IaaS (Infrastructure-as-a-Service)** es el sistema idóneo para desarrolladores que deseen encargarse de la gestión y administración de su infraestructura. Ofrece un mayor control que otras alternativas como PaaS, de modo que el desarrollador es el responsable de todo lo relacionado con el mantenimiento de la infraestructura, incluso de escalar sus aplicaciones en función de cuáles sean sus necesidades.



# PaaS Platform as a Service

---

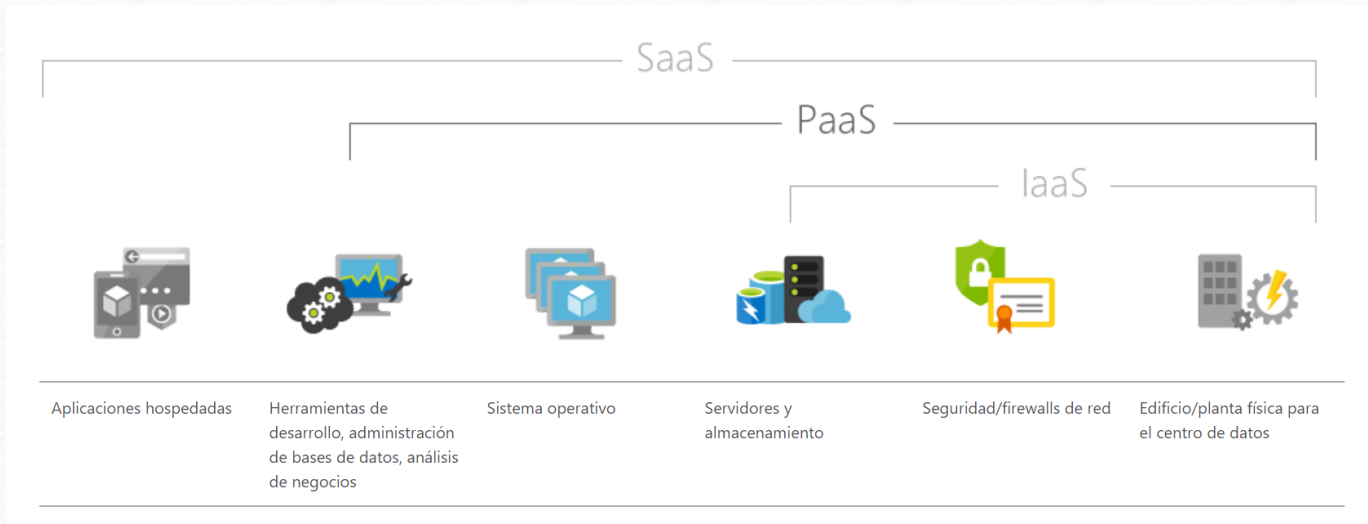
- **PaaS PaaS (Platform-as-a-Service)** se presenta como la alternativa idónea para aquellos desarrolladores de aplicaciones que únicamente quieren preocuparse de construir la app. La infraestructura la proporciona la plataforma y se ocupa tanto de su gestión como de su mantenimiento.

# SaaS Software as a Service

---

- **SaaS (Software-as-a-Service).** Se define como tal a cualquier servicio que esté basado en la web, como por ejemplo el Webmail de Gmail. En este caso los usuarios acceden al servicio sin prestar la más mínima atención al software. Tanto el desarrollo como el mantenimiento y resto de gestiones son responsabilidad única del proveedor.
- Así, los usuarios tienen un control mínimo sobre el servicio en cuestión. Ellos se sitúan en la capa más superficial del mismo. Ejemplos de SaaS son Dropbox o Google Drive.

# – Modelos de



# 11. Aplicaciones web y servidores de aplicaciones

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB



# 11. Aplicaciones web y servidores de aplicaciones

---

- Se define una **aplicación web** como una aplicación informática que se ejecuta en un entorno web, de forma que se trata de una aplicación cliente-servidor junto con un protocolo de comunicación previamente establecido:
  - Cliente: navegador.
  - Servidor: servidor web
  - Comunicación: protocolo HTTP

# 11. Aplicaciones web y servidores de aplicaciones

---

Se denomina **servidor de aplicaciones** a un servidor en una red de computadores que ejecuta ciertas aplicaciones.

Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. ([wikipedia](#))

# 11. Aplicaciones web y servidores de aplicaciones.

## Tomcat

---

**Mucha gente confunde Tomcat con un servidor de aplicaciones; sin embargo, es solamente un contenedor de servlets.**

Java EE provee estándares que permiten a un servidor de aplicaciones servir como "contenedor" de los componentes que conforman dichas aplicaciones. Estos componentes, escritos en lenguaje Java, usualmente se conocen como Servlets, Java Server Pages (JSPs) y Enterprise JavaBeans (EJBs) y permiten implementar diferentes capas de la aplicación, como la interfaz de usuario, la lógica de negocio, la gestión de sesiones de usuario o el acceso a bases de datos remotas

# 11. Aplicaciones web y servidores de aplicaciones.

## Tomcat

---

Apache Tomcat (también llamado Jakarta Tomcat o simplemente Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation (aunque creado por Sun Microsystems).

Tomcat es desarrollado y actualizado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software License. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 8.x, que implementan las especificaciones de Servlet 3.0 y de JSP 2.2. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.



# 12. Estructura y despliegue de una aplicación web

---

UD 1 IMPLANTACIÓN DE APLICACIONES WEB

## 12. Estructura y despliegue de una aplicación web

---

- Una **aplicación web** Java está compuesta de una serie de servlets, páginas jsp, ficheros html, ficheros de imágenes, ficheros de sonidos, texto, clases, etc.; de forma que todos estos recursos se pueden empaquetar y ejecutar en varios contenedores distintos.
- Un **servlet** es una aplicación java encargada de realizar un servicio específico dentro de un servidor web.

## 12. Estructura y despliegue de una aplicación web

---

- La especificación Servlet 2.2 define la estructura de directorios para los ficheros de una aplicación web. El directorio raíz debería tener el nombre de la aplicación y define la raíz de documentos para la aplicación web. Todos los ficheros debajo de esta raíz pueden servirse al cliente excepto aquellos ficheros que están bajo los directorios especiales META-INF y WEB-INF en el directorio raíz. Todos los ficheros privados, al igual que los ficheros class de los servlets, deberían almacenarse bajo el directorio WEB-INF.

## 12. Estructura y despliegue de una aplicación web

---

- Durante la **etapa de desarrollo** de una aplicación web se emplea la estructura de directorios, a pesar de que luego en la **etapa de producción**, toda la estructura de la aplicación se empaqueta en un archivo **.war**
- El código necesario para ejecutar correctamente una aplicación web se encuentra distribuido en una estructura de directorios, agrupándose ficheros según su funcionalidad



# 12. Estructura y despliegue de una aplicación web

---

**Ejemplo de estructura de directorios de una aplicación web:**

/index.jsp

/WebContent/jsp/welcome.jsp

/WebContent/css/estilo.css

/WebContent/js/utills.js

/WebContent/img/welcome.jpg

/WEB-INF/web.xml

/WEB-INF/struts-config.xml

/WEB-INF/lib/struts.jar

/WEB-INF/src/com/empresa/proyecto/action/welcomeAction.java

/WEB-INF/classess/com/empresa/proyecto/action/welcomeAction.class

## 12. Estructura y despliegue de una aplicación web: Archivos war

---

Su nombre procede de Web Application Archive (Archivo de Aplicación Web); permiten empaquetar en una sola unidad aplicaciones web de Java completas, es decir que su contenido:

- Servlets y JSP.
- Contenido estático: HTML, imágenes, etc.
- Otros recursos web.

Aportan como ventaja, la simplificación del despliegue de aplicaciones web, debido a que su instalación es sencilla y solamente es necesario un fichero para cada servidor .

## 12. Estructura y despliegue de una aplicación web: Archivos war

---

Su estructura de una aplicación web es la siguiente:

- **/:** En la carpeta raíz del proyecto se almacenan elementos empleados en los sitios web, tipo documentos html, CSS y los elementos JSP (\*.html \*.jsp \*.css).
- **/WEB-INF/:** Aquí se encuentran los elementos de configuración del archivo .WAR como pueden ser: la página de inicio, la ubicación de los servlets, parámetros adicionales para otros componentes. El más importante de éstos es el archivo web.xml.
- **/WEB-INF/classes/:** Contiene las clases Java empleadas en el archivo .WAR y, normalmente, en esta carpeta se encuentran los servlets.
- **/WEB-INF/lib/:** Contiene los archivos JAR utilizados por la aplicación y que normalmente son las clases empleadas para conectarse con la base de datos o las empleadas por librerías de JSP.

## 12. Estructura y despliegue de una aplicación web: Despliegue de aplicaciones con Tomcat

---

Una aplicación web puede ser desplegada empleando uno de los siguientes métodos:

- Por medio de archivos WAR (Web Archive).
- Editando los archivos web.xml y server.xml.



## 12. Estructura y despliegue de una aplicación web: Descriptor de despliegue

---

Un Descriptor de Despliegue es un documento XML que describe las características de despliegue de una aplicación, un módulo o un componente. Por esto, la información del descriptor de despliegue es declarativa, y esta puede ser cambiada sin la necesidad de modificar el código fuente.

Cualquier aplicación web tiene que aportar un descriptor de despliegue situado en WEB-INF/web.xml; en el caso concreto de Tomcat el descriptor `<TOMCAT_HOME>/conf/web.xml` es un descriptor por defecto que se ejecuta siempre antes del descriptor de la aplicación y, solamente, debería contener información general y no específica de la aplicación.

## 12. Estructura y despliegue de una aplicación web: Descriptor de despliegue

---

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!DOCTYPE web-app PUBLIC  
    "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"  
    "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">  
<web-app>  
    <!-- Tus definiciones van aquí -->  
</web-app>
```

## 12. Estructura y despliegue de una aplicación web: Descriptor de despliegue

---

Situadas entre las etiquetas `<web-app>` y `/<web-app>` estarían los descriptores de despliegue de servlets, los cuales deben contener las siguientes etiquetas en el siguiente orden:

`<servlet>`

`<servlet-name>nombre</servlet-name>`

`<servlet-class>package.nombre.MiClass</servlet-class>`

`</servlet>`

## 12. Estructura y despliegue de una aplicación web: Descriptor de despliegue

---

Para probar el servlet, una vez arrancado el servidor Tomcat, abrimos un navegador web, en el cual escribiríamos una URL con el siguiente formato:

```
http://{address}:{port}/{servletName}
```