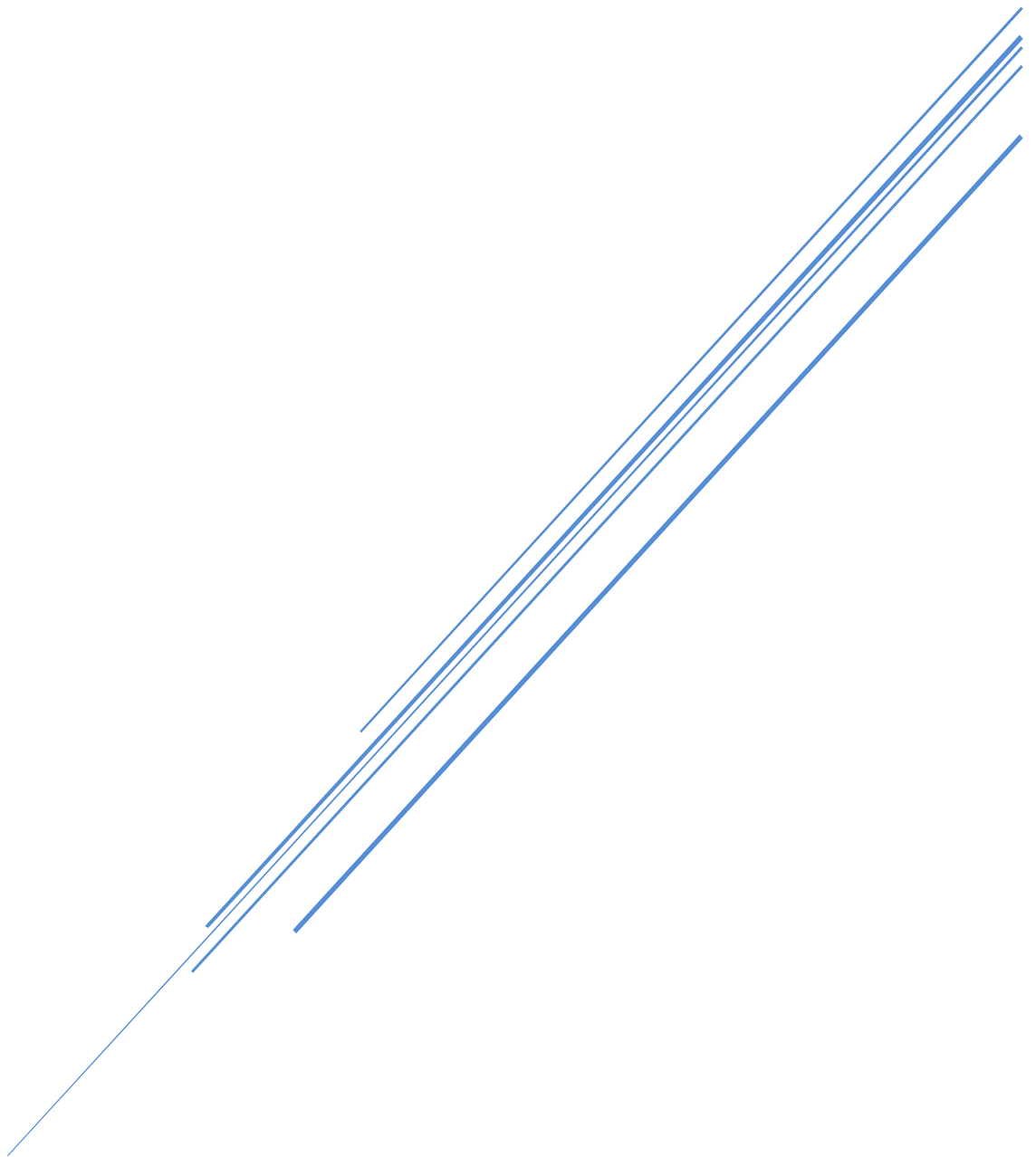


# UD 1. IMPLANTACIÓN DE ARQUITECTURAS WEB



<b>1.</b>	<b>Obxectivos .....</b>	<b>2</b>
<b>2.</b>	<b>Implantación de servizos web .....</b>	<b>2</b>
2.1	Introdución .....	2
2.1.1	Internet: Conceptos básicos .....	2
2.1.1.1	A Web .....	2
2.1.2	Aplicacións web .....	3
2.1.2.1	Vantaxes .....	3
2.1.2.2	Inconvenientes .....	4
2.1.2.3	Arquitectura web .....	4
2.1.2.4	Modelos de arquitectura .....	4
	Modelo cliente/servidor .....	5
	Modelo orientado a mensaxes .....	5
	Modelo Orientado a servizos .....	5
	Modelo orientado a recursos .....	5
	Modelo de políticas .....	5
2.1.2.5	Estrutura .....	5
2.1.2.6	Software que intervéñ .....	7
2.1.3	Protocolo HTTP .....	7
2.1.3.1	Mensaxes de petición .....	8
	Petición GET .....	8
	Petición POST .....	9
2.1.3.2	Mensaxes de resposta .....	9
	Códigos de estado .....	9
2.1.3.3	Tipos MIME .....	9
2.2	Servizos Web .....	10
2.2.1	Aspectos xerais de arquitecturas web .....	10
2.3	Evolución dos servizos web .....	11
2.3.1	As distintas versións da Web .....	11
2.3.1.1	Web 1.0 .....	11
2.3.1.2	Web 1.5 .....	12
2.3.1.3	Web 2.0 .....	12
2.3.2	Tecnoloxías empregadas en aplicacións web .....	12
2.3.2.1	No lado do servidor .....	12
2.3.2.2	No lado do cliente .....	13
2.3.2.3	En ambos lados .....	14
<b>3.</b>	<b>Servidores web .....</b>	<b>15</b>
3.1.1.1	Recursos necesarios para a instalación dun servidor web .....	15
<b>4.</b>	<b>Servidores de aplicacións .....</b>	<b>16</b>
4.1.1.1	Recursos necesarios no servidor de aplicacións .....	16
4.1.2	Tomcat .....	16
4.1.2.1	Arquitectura .....	17

# 1. Obxectivos

---

Na actividade que nos ocupa aprenderanse os seguintes conceptos e manexo de destrezas:

- Identificar as características xerais das arquitecturas web.
- Estudar os fundamentos e protocolos nos que se basea un servidor web.
- Identificar a estrutura e recursos que compoñen unha aplicación web.

## 2. Implantación de servizos web

---

### 2.1 Introducción

#### 2.1.1 Internet: Conceptos básicos

Internet é un sistema global de redes de computadoras interconectadas entre si que utilizan a familia de protocolos TCP/IP para compartir recursos e intercambiar información entre miles de millóns de dispositivos en todo o mundo. Trátase dunha rede de redes descentralizada que, grazas a dita familia de protocolos, funciona como unha única rede a pesares da heteroxeneidade das redes físicas que a compoñen.

Ten a súa orixe na rede ARPANET (Advanced Research Projects Agency Network), creada por encargo do Departamento de Defensa dos EEUU en 1969, coa finalidade de comunicar organismos oficiais, de xeito que, se houboese unha catástrofe que destruíse algún deles, o resto da rede continuase funcionando.

Algúns dos servizos de Internet aínda amplamente empregados hoxe en día son:

- **Correo electrónico.** Servizo de correspondencia que permite enviar ou recibir mensaxes textuais e outros documentos desde un emisor a un ou varios destinatarios.
- **FTP (File Transfer Protocol).** Servizo que permite a transferencia de ficheiros entre máquinas.
- **WWW (World Wide Web).** Este servizo é un sistema de documentos de hipertexto (texto enriquecido con ligazóns) interconectados entre si. É a ferramenta máis utilizada de Internet e permite aos usuarios acceder, mediante un navegador, a páxinas web que poden conter texto, imaxes, vídeos e outro contido multimedia e navegar entre eles mediante hiperligazóns.

##### 2.1.1.1 A Web

A World Wide Web foi creada en 1989 por Tim Berners-Lee mentres traballaba no CERN (Organización Europea para a Investigación Nuclear). Consiste nun sistema de organizar a información mediante páxinas conectadas a través de hiperligazóns, que utiliza para a súa transferencia a Internet como medio físico e o protocolo HTTP (Hypertext Transfer Protocol), para que os navegadores web poidan facer as peticións de ditas páxinas aos servidores web e recibir as respostas sobre as mesmas. Defínese tamén a linguaxe HTML (HyperText Markup Language) para a edición de páxinas web.



Nos primeiros 3 minutos e medio do seguinte vídeo, Tim Berners-Lee fala de como inventou a web. [https://www.youtube.com/watch?v=OM6XIIcm\\_qo](https://www.youtube.com/watch?v=OM6XIIcm_qo) Podemos atopalo subtítulado na seguinte ligazón: [https://www.youtube.com/watch?v=UUpw-7\\_SWds](https://www.youtube.com/watch?v=UUpw-7_SWds)

## 2.1.2 Aplicacións web

Na enxeñaría de software denomínase aplicación web a aquela ferramenta que os usuarios poden empregar accedendo a un servidor web a través de Internet ou dunha intranet mediante un navegador.

As aplicacións web son populares debido ao práctico que resulta o navegador web como cliente lixeiro, á independencia do sistema operativo, así como á facilidade para actualizar e manter aplicacións web sen distribuír e instalar software a miles de usuarios potenciais. Existen aplicacións como wikis, weblogs, tendas en liña, etc., que son exemplos bastante coñecidos de aplicacións web.

### 2.1.2.1 Vantaxes

- Aforro de tempo: pódense realizar tarefas sinxelas sen necesidade de descargar nin instalar ningún programa no noso ordenador.
- Non hai problemas de compatibilidade: xa que simplemente se precisa ter un navegador web actualizado para poder utilizalas.
- Non ocupan espazo no noso disco duro: xa que non se necesita gardar nin o propio programa nin os datos que se necesitan para operar con el.
- Actualizacións inmediatas: como o software o xestiona o propio desenvolvedor de software, cando nos conectamos á aplicación sempre estamos a usar a última versión que lanzou.
- Consumo de recursos baixo: dado que todo (ou gran parte) da aplicación non se atopa no noso ordenador, moitas das tarefas que realiza o software non consumen recursos nosos porque se realizan desde outra computadora, á cal estamos conectados vía Internet.
- Multiplataforma: pódense usar desde calquera sistema operativo porque soamente é necesario ter un navegador para usalas.
- Portables: é independente do ordenador onde se utilice (PC sobremesa, portátil) porque se accede a través dunha páxina web, polo que só é necesario un navegador e dispoñer de acceso a Internet.
- Disponibilidade alta: porque o servizo, aínda que hai ocasións no que se pode ver interrompido, ofrécese desde múltiples localizacións para asegurar a continuidade do mesmo.
- Os virus que poidan afectar ao noso ordenador non danan os datos porque estes están almacenados no servidor da aplicación.
- Colaboración: grazas a que o acceso ao servizo se realiza desde unha única ubicación, é sinxelo o acceso e a compartición de datos por parte de varios usuarios. Ten moito sentido, por exemplo, en aplicacións en liña de calendarios ou oficina.

### 2.1.2.2 Inconvenientes

- Habitualmente ofrecen menos funcionalidade que as aplicacións de escritorio, debido a que as funcionalidades que se poden realizar desde un navegador poden ser máis limitadas que as que se poden realizar desde os sistema operativo.
- A dispoñibilidade depende dun terceiro, o provedor da conexión a Internet ou o que prové o enlace entre o servidor da aplicación e o cliente. Así que, a dispoñibilidade do servizo está supeditada ao provedor.
- Non todos os navegadores web teñen as mesmas capacidades, polo que pode ser preciso facer adaptacións e moitas probas para o seu correcto funcionamento na maioría deles.
- Dependemos dunha conexión co servidor para poder utilizarlas; se falla a conexión non poderemos acceder á aplicación web, e implica tráfico de rede (e consumo de datos en dispositivos móbiles).
- A información que se amosa no navegador debe transmitirse desde el servidor o que non as fai axeitadas para tratamento de datos moi pesados en tempo real (por exemplo as aplicacións que manexan contido multimedia como as de edición de vídeo).
- 

### 2.1.2.3 Arquitectura web

A arquitectura web define cómo se vai a xerarquizar a información dentro dun sitio web de forma racional e lóxica.

O seu objetivo é a agrupación visual de información nun sitio web que permitirá o acceso intuitivo e a navegabilidade.

Existen unha serie de tecnoloxías a nivel de cliente e de servidor que debemos coñecer para implantar as nosas solucións web; a elección dependerá da dimensión, requisitos e custe do proxecto.

Hai tres elementos principais nunha aplicación que permiten a conexión e o acceso a datos por parte de calquera petición dun cliente:

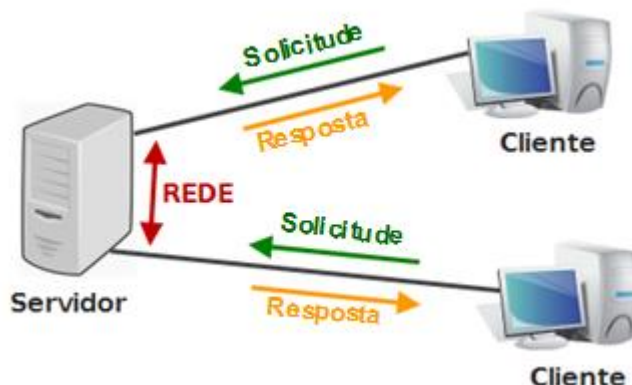
- **Servidor web:** é o servidor ou cerebro da arquitectura escoitando as peticións HTTP desde o navegador; realiza tamén consultas á base de datos para responder a esas peticións.
- **Base de datos:** é onde se almacenan un conxunto de datos organizados xerárquicamente. O servidor web require esa información ou a actualización/eliminación da mesma, e o sistema xestor de base de datos fai esa operación.
- **Cliente web:** é o que realiza as peticións o servidor web mediante un navegador e un sistema operativo concreto que é independente da arquitectura.

### 2.1.2.4 Modelos de arquitectura

O W3C (World Wide Web Consortium) define catro modelos de arquitectura para as aplicacións web: modelo cliente/servidor, modelo orientado a mensaxes, modelo orientado a servizos e modelo orientado a recursos. Neste apartado se explicarán todos eles pero, posto que o modelo cliente/servidor é a arquitectura máis estendida hoxe en día nas aplicacións web, será na que nos centremos a partires de agora.

## Modelo cliente/servidor

Nesta arquitectura, por un lado está o cliente, que será o navegador web, e polo outro lado está o servidor, que é o servidor web. A lóxica da aplicación é compartida por ambos.



## Modelo orientado a mensaxes

O Modelo Orientado a Mensaxes (MOM) céntrase en definir as mensaxes, a súa estrutura, a forma de transportalos, etc., sen importar o significado semántico de cada mensaxe ou a súa relación con outras mensaxes.

## Modelo Orientado a servizos

O Modelo Orientado a Servizos (SOM) ocúpase dos aspectos da arquitectura relacionados co servizo a a acción.

O seu obxectivo principal é o de explicar as relacións entre un axente e os servizos que ofrece e solicita. O SOM baséase no MOM, pero centrado na acción, en lugar da mensaxe.

## Modelo orientado a recursos

O Modelo Orientado a Recursos (ROM) céntrase nos aspectos da arquitectura relacionados cos recursos e as súas características principais (propietario, políticas asociadas, etc.), independentemente do papel que o recurso xogue no contexto dos servizos web.

## Modelo de políticas

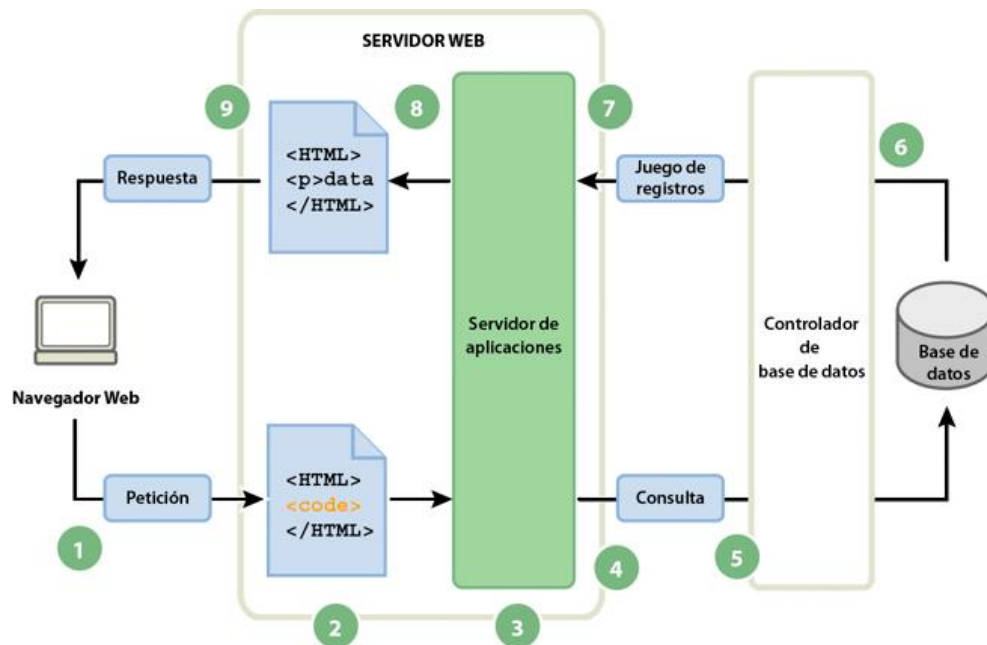
Este modelo baséase en definir os comportamentos dos axentes que empregan os servizos, definindo como accederán aos recursos.

### 2.1.2.5 Estrutura

Unha aplicación web que emprega o modelo cliente/servidor está estruturada habitualmente como unha aplicación en tres capas:

- Capa de presentación. É a que ve o usuario, presenta o sistema ao usuario, comunícalle a información e captura a información do usuario. Nunha aplicación web reside no cliente e as tecnoloxías empregadas habitualmente son HTML, CSS e Javascript, entre outras.

- Capa de negocio. É onde residen os programas que se executan, recíbense as peticións do usuario e envíanse as respostas tras o proceso. Nunha aplicación web reside no servidor e algunhas das tecnoloxías empregadas son PHP, Java Servlets o ASP, ASP.NET, CGI, ColdFusion, embPerl, Python ou Ruby on Rails.
- Capa de datos. É onde residen os datos e é a encargada de acceder aos mesmos. Nunha aplicación web reside no servidor, que pode ser fisicamente o mesmo que o que se ocupa da capa de negocio ou outro. Algunhas das tecnoloxías posibles para implementar esa capa son PostgreSQL, MySQL, Oracle DBMS, MongoDB ou Cassandra.



Na imaxe anterior pode verse claramente cal é o funcionamento dunha arquitectura web de 3 capas:

- 1. O usuario interactúa co navegador web que é o que realiza a solicitude de páxinas, que poden ser páxinas almacenadas (estáticas) ou creadas dinamicamente, con información aos servidores web.
- 2. O servidor web localiza a páxina e lla envía ao servidor de aplicacións para executar as instrucións necesarias.
- 3. O servidor de aplicacións busca as instrucións na páxina e execútaas.
- 4. O servidor de aplicacións precisará datos para poder executar as sentenzas, co cal envía a consulta ao controlador da base de datos.
- 5. O controlador executa a consulta na base de datos.
- 6. Unha vez resolta a consulta, envíase ao controlador o conxunto de rexistros resultantes da consulta.
- 7. O controlador pasa o conxunto de rexistros ao servidor de aplicacións.
- 8. O servidor de aplicacións insire os datos nunha páxina. Unha vez que estea completa, pásalla ao servidor web.
- 9. O servidor web envía a páxina finalizada ao navegador solicitante.



No documento Programación de aplicacións web, publicado por Sergio Luján Mora, pódese ver claramente como son as distintos tipos de arquitecturas que se poden dar

#### 2.1.2.6 Software que intervén

O tipo de software que rodea o uso e funcionamento dunha aplicación web no modelo cliente/servidor é o seguinte:

- Navegador web: é o software, aplicación ou programa que permite o acceso á web. A funcionalidade básica dun navegador é permitir a visualización de documentos, comunemente denominados páxinas web. Correspóndese coa capa de presentación dunha aplicación web. Co navegador interactúan os usuarios para poder facer uso dunha aplicación web.
- Servidor web: é un programa que procesa unha aplicación do lado do servidor. O servidor e o cliente comunícanse xeralmente utilizando o protocolo HTTP para as comunicacións. Verase máis en detalle o servidor web na seguinte actividade.
- Linguaxes de script: son un tipo de linguaxes de programación que non precisan ser compilados, senón que xeralmente son interpretados, de forma que as súas sentenzas se executan directamente, sen unha previa compilación. O conxunto de instrucións se denomina script, que é o que adoita acompañar a un documento HTML ou estar contido no seu interior. As instrucións do script execútanse antes de enviar o documento ao navegador, cando se carga o documento ou cando se produce algunha circunstancia, é dicir, cando ocorre un evento.
- Xestor de base de datos: é un conxunto de programas que permiten o almacenamento, modificación e extracción da información nunha base de datos, ademais de proporcionar ferramentas para engadir, borrar, modificar e analizar os datos.



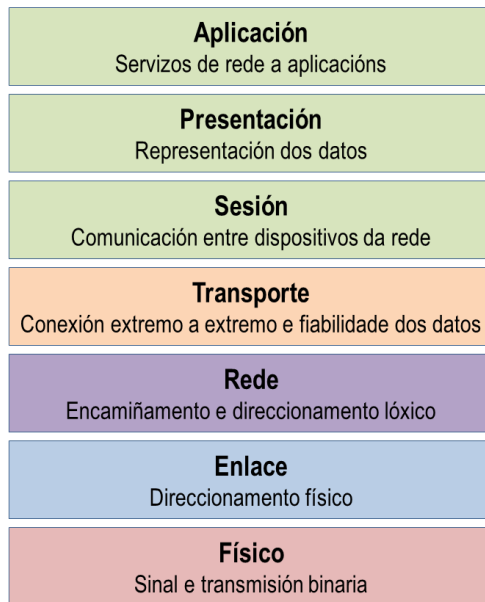
Realizar a tarefa 1.1 “Selección de software para unha aplicación web”, na que se creará unha táboa con aplicacións software clasificadas segundo o tipo de software dunha aplicación web.

#### 2.1.3 Protocolo HTTP

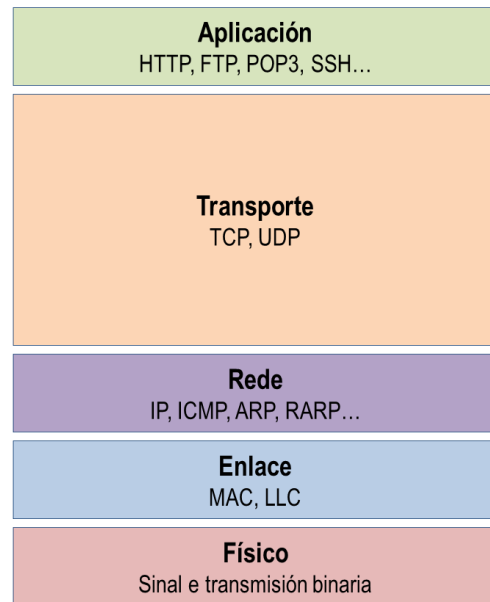
Un servidor web opera mediante o protocolo HTTP, da capa de aplicación do Modelo OSI. HTTP é un protocolo simple, textual e sen estado para o intercambio de datos tipificados (obxectos) entre un cliente e un servidor web baseado en comandos e respostas.



## PILA OSI



## PILA TCP/IP



Cada servidor ten un proceso que permanece á escoita nun porto TCP (por defecto, o 80), esperando conexións entrantes dos clientes. Cando se establece a conexión, os navegadores solicitan información ao servidor enviándolles un URL. O servidor responde cunha mensaxe que contén o estado da operación e o seu posible resultado. Todas as operacións poden achegar un obxecto ou recurso sobre o que actúan. Despois da transmisión libérase a conexión TCP.

### 2.1.3.1 Mensaxes de petición

Os mensaxes de petición están formados por tres partes:

- Liña inicial de petición: método, URL, versión.
- Liña(s) de cabeceira.
- Corpo da mensaxe (opcional). Parámetros ou ficheiros a enviar ao servidor.

Os posibles métodos de petición son GET, POST, OPTIONS, HEAD, PUT, DELETE, TRACE, CONNECT e PATH.

### Petición GET

Emprégase para obter calquera tipo de información do servidor. As peticións GET non teñen corpo da mensaxe. Permite enviar parámetros ao servidor na URI (ou URL) (coñecidos como Query String). Por exemplo:

`http://sitiodeproba.com/novousuarioGET.php?nome=Emilio&apelido=Dapena`

Nesta URL podemos distinguir varias partes:

- `http://sitiodeproba.com/novousuarioGET.php` é o enderezo web onde se atopa o script que procesará os datos.
- O símbolo `?` indica onde empezan os parámetros que se reciben desde o formulario que enviou os datos á páxina.
- As parellas `dato1=valor1`, `dato2=valor2`, etc. reflicten o nome e o valor dos campos enviados polo formulario.

O tamaño da información enviada estará limitada. Non se pode empregar para enviar

arquivos ou realizar outras operacións que requiran enviar unha gran cantidade de datos ao servidor.

## Petición POST

Emprégase para solicitar ao servidor que acepte información que se envía axunta nunha petición. As peticións POST envíanse no corpo da mensaxe. Os parámetros, polo tanto, non son visibles na URL.

### 2.1.3.2 Mensaxes de resposta

Os mensaxes de petición están formados por tres partes:

- Liña inicial de resposta (liña de estado): Versión HTTP, código de estado e texto explicativo.
- Liña(s) de cabeceira.
- Corpo da mensaxe (opcional). Determinado polo tipo de recurso solicitado.

## Códigos de estado

Os códigos de estado son códigos que envían os servidores nas respostas HTTP. Informan ao cliente de cómo foi procesada a petición. Consisten nun código de 3 díxitos e se clasifican en función do primeiro, van acompañados dun texto explicativo. A continuación expoñemos os principais grupos de códigos de estado con algúns dos exemplos máis salientables.

- 100 - 199 (Informativo, Informational).
- 200 - 299 (Éxito, Successful):
  - 200 - OK (Resposta estándar para peticións correctas).
- 300 - 399 (Redirección, Redirection):
  - 304 - Not Modified (indica que a petición á URL non foi modificada desde que foi requerida por última vez).
- 400 - 499 (Erros do cliente, Client Error).
  - 404 - Not found (Non atopado).
  - 403 - Forbidden (A solicitude foi legal, pero o servidor rexeita respondela porque o cliente non ten os privilexios para facela).
  - 406 - Not Acceptable (O servidor non é capaz de devolver os datos en ningún dos formatos aceptados polo cliente).
- 500 - 599 (Erros no servidor, Server Error)
  - 503 - Service Unavailable (O servidor non pode responder á petición do navegador porque está conxestionado ou realizando tarefas de mantemento).



Para un listado de todos os códigos de estado das respostas HTTP pódese consultar o RFC-1945 <https://tools.ietf.org/html/rfc1945> no seu apartado 9: “Status Code Definitions”.

### 2.1.3.3 Tipos MIME

Inicialmente as páxinas web contiñan texto e ligazóns (hipertexto), polo que HTTP era un protocolo que unicamente enviaba texto. Agora, HTTP permite enviar todo tipo de documentos, para o que emprega os tipos MIME (tipos de medios de Internet). Os tipos MIME

son entón unhas especificacións definidas pola IANA para dar formato a mensaxes non ASCII.

Exemplos de tipos MIME habituais son `text/html`, `text/css` ou `video/mpeg`.



Pódense consultar todos os tipos mime na especificación oficial do IANA <http://www.iana.org/assignments/media-types/media-types.xhtml>

## 2.2 Servizos Web

### 2.2.1 Aspectos generales de arquitecturas 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", que envían requerimientos de objetos a un servidor y éste responde con el dato codificado según un formato estándar.

Los estándares WWW especifican muchos de los mecanismos necesarios para construir un ambiente de aplicación de propósito general, por ejemplo:

- ✓ Modelo estándar de nombres: todos los servidores, así como el contenido de la WWW se denominan según un Localizador Uniforme de Recursos (Uniform Resource Locator: URL).
- ✓ Contenido: a todos los contenidos en la WWW se les especifica un determinado tipo permitiendo de esta forma que los browsers (navegadores) los interpreten correctamente.
- ✓ Formatos de contenidos estándar: todos los navegadores soportan un conjunto de formatos estándar, por ejemplo HTML, ECMA, JavaScript, etc.
- ✓ Protocolos estándar: éstos permiten que cualquier navegador pueda comunicarse con cualquier servidor web. El más comúnmente usado en WWW es HTML (Protocolo de Transporte de Hiper-Texto), que opera sobre el conjunto de protocolos TCP/IP.

Esta infraestructura permite a los usuarios acceder a una gran cantidad de aplicaciones y servicios de terceros. También permite a los desarrolladores crear aplicaciones y servicios para una gran comunidad de clientes.

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

- ✓ Escalabilidad.
- ✓ Separación de responsabilidades.
- ✓ Portabilidad.
- ✓ Utilización de componentes en los servicios de infraestructura.
- ✓ Gestión de las sesiones del usuario.
- ✓ Aplicación de patrones de diseño.

El esquema de funcionamiento de los servicios web requiere de tres elementos fundamentales:

1. Proveedor del servicio web, que es quien lo diseña, desarrolla e implementa y lo pone disponible para su uso, ya sea dentro de la misma organización o en público.
2. Consumidor del servicio, que es quien accede al componente para utilizar los servicios que éste presta.

3. Agente del servicio, que sirve como enlace entre proveedor y consumidor para efectos de publicación, búsqueda y localización del servicio.

De forma genérica podríamos decir que la arquitectura web es un modelo compuesto de tres capas:

1. Capa de Base de Datos, donde estaría toda la documentación de la información que se pretende administrar mediante el servicio web y emplearía una plataforma del tipo MySQL, PostgreSQL, etc.
2. En una segunda capa estarían los servidores de aplicaciones web, ejecutando aplicaciones de tipo Apache, Tomcat, Resin, etc.
3. En una tercera capa estarían los clientes del servicio web al que accederían mediante un navegador web como Firefox, Internet Explorer, Opera, etc.

## 2.3 Evolución dos servizos web.

La evolución del uso de Servicios web en las organizaciones está fuertemente ligada al desarrollo de Internet como red prestadora de servicios.

A tecnoloxía evolucionou de forma exponencial desde que comezou a era da informática ata os nosos días. No contexto da web ligado a tecnoloxía de Internet son moitos os compoñentes da web que evolucionaron. Entre outros:

- Ancho de banda: o hardware de comunicacións era moi costoso e lento; é agora máis económico e veloz.
- Almacenamento: manexamos cantidades inxentes de información que se poden almacenar e explotar gracias ás investigacións e os dispositivos de almacenamento actuais, que non teñen nada que ver en capacidade, fiabilidade e rendemento cos pasados.
- Información: dunha información moito máis estática pasamos a unha moito máis dinámica con máis interacción co usuario.
- Computación: os procesadores, memoria e disco duro na actulidade permiten operacións moito máis rápidas.
- Tecnoloxía: existe un amplo abanico de posibilidades que se poden implementar tanto no cliente coma no servidor axiliando o desenvolvemento dunha aplicación e aumentando a súa calidae; xa non é preciso partir de cero nin reinventar a roda.
- Infraestructuras: engloba todo o mencionado.

*Na actualidade existen unha variedade de posibilidades que antes non existían polo avance da tecnoloxía: existe alta disponibilidad, backup remoto, duplicidade de nodos en caso de fallo hardware e software, uso de intelixencia artificial e algoritmos adaptativos, explotación da información mediante a xestión de big data*

### 2.3.1 As distintas versións da Web

#### 2.3.1.1 Web 1.0

Nos inicios da informática o contido da web era estático e non existían protocolos de seguridade que permitiran a fiabilidade das conexións, cousa impensable na actualidade en calquera organización.

Esta primeira fase na que non existía interacción co usuario e as páxinas web eran estáticas

baseadas na linguaxe de marcas HTML denominouse Web 1.0. Tim Berners-Lee foi no ano 1991 o creador da primeira páxina web con contido estático, onde existían marcas e botóns gif. A interacción dos usuarios era soamente mediante email, foros e chats.

### 2.3.1.2 Web 1.5

Os desenvolvedores investigaron sobre as posibilidades de ampliar esta primeira web e incluír máis funcionalidades; xurdiron as primeiras aplicacións web dinámicas que interactuaban con bases de datos no que algúns autores deron en denominar Web 1.5.

### 2.3.1.3 Web 2.0

A verdadeira evolución viu coa Web 2.0 ou web social na que se pasou á interacción total co usuario e un compoñente activo no modelo web.

Algúns ítems importantes foron:

- Follas de estilo CSS que dan vistosidade as webs.
- Uso de JSON.
- Desenvolvemento en Ajax.
- Soporte para os blogs.
- Comezo das redes sociais.
- Control total dos usuarios no manexo da información.

O seguinte avance significativo foi a aparición da **Web 3.0**, data web ou web semántica: un gran salto no avance tecnolóxico hacia a intelixencia artificial.

Algúns dos avances son:

- Diseño reponsive.
- Web multimedia.
- Aplicacións intelixentes.
- Web semántica máis intelixencia artificial.
- Impulso á Web 3D.
- Participación máis activa na rede.
- CSS3.

Por último, trala incorporación no cine da tecnoloxía 3D, en CSS3 xa é posible traballar en terceira dimensión, e HTML5 comeza a incorporar avances neste sentido.

## 2.3.2 Tecnoloxías empregadas en aplicacións web

Actualmente, a maioría das aplicacións web do mercado usan páxinas dinámicas que se executan no servidor web e se visualizan no cliente (navegador).

Existen páxinas de contido estático e dinámico; normalmente, cando existe contido dinámico excútase código tanto no cliente coma no servidor.

Podes atopar información máis detallada de linguaxes no [lado servidor](#) e [no lado cliente](#). A continuación indícanse algúñas das alternativas.

### 2.3.2.1 No lado do servidor

- **CGI** (Common Gateway Interface)

Ó principio de Internet o servidor soamente podía executar programas de tipo C, Perl e liñas de comando Powershell. Estas instrucións eran executadas polo sistema operativo e se transmitían ó navegador mediante o CGI.

- **ASP.NET** (Active Server Pages)

Framework de desenvolvemento libre de Windows orientado a obxectos; existen versións para Linux ye Unix.

- **Java**

Aínda que tamén se pode utilizar no lado cliente, destaca no seu uso no lado servidor polo grupo de tecnoloxías asociadas: JSF (JavaServer Faces), JSP (JavaServer Pages) e os servlets.

- **Ruby**

Linguaxe interpretada de propósito xeral, dinámica e flexible; é de alto nivel, software libre e multiplataforma.

- **Perl**

Toma moitas características da linguaxe C e caracterízase pola súa destreza á hora de procesar texto. Apache conta cun módulo que permite executar programas deste tipo.

- **PHP**

Linguaxe de propósito xeral dos máis usados no desenvolvemento backend. É motor de sistemas coma Wordpress, Drupal, Magento, Joomla, etc. Existen frameworks moi potentes (Laravel, Symfony, codeIgnition) que facilitan a súa programación. Permite acceso a bases de datos relacionais coma NoSQL (por exemplo MongoDB).

- **Python**

Linguaxe interpretada moi poderosa, fácil de aprender e de alto nivel; é unha linguaxe orientada a obxectos que destaca polo seu tipado dinámico.

- **Javascript**

Linguaxe lixeira, interpretada e orientada a obxectos. Para aprendelo é preciso un coñecemento básico de HTML e CSS. Intervén sobre todo no contido dinámico e na interacción co usuario. Permite controlar ficheiros multimedia, creación de imaxes animadas, animación en 3D.

A linguaxe R é moi usada en tecnoloxía web debido o crecemento exponencial do Big Data.

Podes atopar máis información sobre algunhas destas linguaxes neste artigo

#### 2.3.2.2 No lado do cliente

- **HTML e CSS**

HTML é a linguaxe de marcas que xurdíu xunto cos navegadores e o medio para transmitir información entre o cliente e o servidor. Posteriormente CSS (Cascading Style Sheets) permitiu deseñar gráficamente e maquetar a páxina HTML. É unha especificación que parte do W3C da que na actualidade está publicada a versión 3. Non son propiamente linguaxes de programación, pero si linguaxes informáticas.

- **Javascript**

Pódese usar tanto no cliente coma no servidor. É unha linguaxe moi extendida porque é soportado pola maioría de navegadores.

- **VBScript**

É a competencia de Java que fixo Microsoft. Xa moi en desuso por non ser compatible coa maioría dos navegadores (soamente con Internet Explorer). É unha linguaxe interpretada por Windows Scripting Host de Microsoft baseada na linguaxe de programación de Visual Basic

### 2.3.2.3 En ambos lados

- **DHTML**

non é unha linguaxe de programación sino unha linguaxe informática; abrangue o conxunto de todos os elementos dunha páxina, coma por exemplo son o fondo, as posicións de controis, caixas, contido dinámico, etc.

- **XML**

é unha tecnoloxía relacionada coas linguaxes de marca que permite compartir datos e incluso formar parte dunha base de datos. Estes ficheiros .xml útilízanse tamén para a configuración da moitas das aplicacións e dos servidores web e de aplicacións.

URL de axuda:

[https://ikastaroak.birt.eus/edu/argitalpen/ba-ckupa/20200331/1920k/es/DAW/DEAW/DEAW01/es\\_DAW\\_DEAW01\\_Contenidos/website\\_index.html](https://ikastaroak.birt.eus/edu/argitalpen/ba-ckupa/20200331/1920k/es/DAW/DEAW/DEAW01/es_DAW_DEAW01_Contenidos/website_index.html)

## 3. Servidores web

Un servidor web é un sistema que procesa peticións a través do protocolo HTTP. O termo pode referirse tanto ao sistema completo como ao software que acepta e supervisa ditas peticións.

Un servidor web básico ten un esquema de funcionamento moi simple, baseado en executar indefinidamente as seguinte secuencia de operacións:

- Escolta peticións no porto TCP indicado (por defecto, para HTTP é o 80).
- Recibe unha petición.
- Busca o recurso solicitado.
- Envía o recurso utilizando a mesma conexión na que recibiu a petición ou devolve o erro 404 se non o atopa.

Para este propósito podemos atopar, entre outros, o seguinte software:

- **Apache** (<http://www.apache.org>), de código aberto, para o que existen versións tanto para Windows como para Linux. Na actualidade é o servidor máis empregado polos sitios activos, servido case un 50% dos mesmos. Este será o que instalemos neste módulo.
- **IIS** (Internet Information Services), software propietario que ven integrado nas versións de servidor de Windows.
- **Nginx** (<http://nginx.org>), tamén de código aberto, que resulta moi apropiado para aqueles sitios web que reciben un gran número de peticións nun curto espazo de tempo, e que está gañando terreo, sendo o segundo servidor web máis utilizado nos sitios activos na actualidade.



Na seguinte ligazón podemos consultar datos sobre o uso dos distintos servidores web na actualidade: <https://news.netcraft.com/archives/2016/02/22/february-2016-web-server-survey.html>

### 3.1.1.1 Recursos necesarios para a instalación dun servidor web

Para poder instalar o noso propio servidor web precisamos, como mínimo, os seguintes recursos:

- Máquina cunha potencia capaz de atender as peticións que teña que procesar. É aconsellable que se trate dun servidor dedicado ou, como moito, compartido pero con funcións relacionadas, por exemplo, un servidor de transferencia de arquivos (FTP).
- Sistema operativo estable.
- Enderezo IP estático.
- Enderezo simbólico configurado nun servidor DNS para que dirixa as peticións ao enderezo IP correspondente.
- Conexión a Internet 24h.
- Software do servidor.



## 4. Servidores de aplicacións

---

Un servidor de aplicacións é un paquete software que proporciona servizos ás aplicacións, como seguridade, soporte para transaccións, balanceo de carga e xestión de sistemas distribuídos.

Usualmente trátase dun dispositivo software que proporciona servizos de aplicación ás computadoras cliente. Un servidor de aplicacións xeralmente xestiona a maior parte (ou a totalidade) das funcións de lóxica de negocio e de acceso aos datos da aplicación. Os principais beneficios da aplicación da tecnoloxía de servidores de aplicación son a centralización e a diminución da complexidade no desenvolvemento de aplicacións.

O termo foi inicialmente empregado para servidores da plataforma Java Enterprise Edition (J2EE), sendo o máis popular Apache Tomcat, pero existindo tamén outros coma WebLogic, de Oracle, pero na actualidade tamén se estende a outras tecnoloxías. Por exemplo, co aumento da popularidade de .NET, Microsoft fala do seu produto Internet Information Server como un servidor de aplicacións. E tamén se poden atopar servidores de aplicación de código aberto e comerciais doutros provedores, por exemplo, Base4 Server (aplicacións .NET) e Zope (aplicacións escritas en Python).

### 4.1.1.1 Recursos necesarios no servidor de aplicacións

Para poder instalar o noso propio servidor web precisamos, como mínimo, os seguintes recursos:

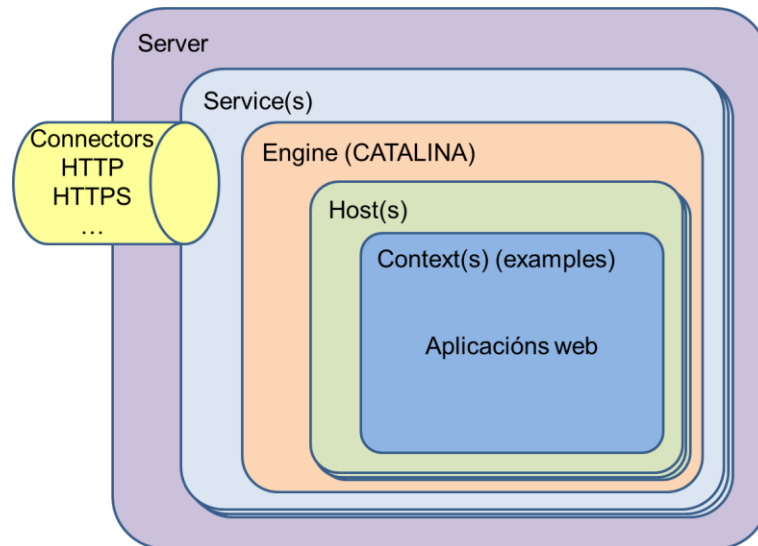
- Máquina cunha potencia capaz de atender as peticións que teña que procesar. É aconsellable que se trate dun servidor dedicado ou, como moito, compartido pero con funcións relacionadas, por exemplo, un servidor de transferencia de arquivos (FTP).
- Sistema operativo estable.
- Enderezo IP estático.
- Enderezo simbólico configurado nun servidor DNS para que dirixa as peticións ao enderezo IP correspondente.
- Conexión a Internet 24h.
- Software do servidor.

### 4.1.2 Tomcat

Apache Tomcat (tamén chamado Jakarta Tomcat ou simplemente Tomcat) funciona como un colector de servlets desenvolvido baixo o proxecto Jakarta na Apache Software Foundation. Tomcat implementa as especificacións dos servlets e de JavaServer Pages (JSP).

Xestiona as solicitudes e respostas http e, ademais, é servidor de aplicacións ou colector de Servlets e JSP. Inclúe o compilador Jasper, que compila JSP convertíndoas en servlets.

#### 4.1.2.1 Arquitectura



- O Server é o primeiro elemento contenedor.
- Un Conector (Connector) é unha asociación con porto IP para manexar as peticións e as respostas cos clientes.
- Un Servizo (Service) asocia un ou mais Connectors cun único Engine.
- Engine é un colector dun ou mais Hosts. É posible configurar Virtual Hosts. Recibe as peticións dos Conectores e as traslada ao host correspondente.
- Un Host define un servidor virtual (Virtual Host). Pode conter unha ou mais aplicacións web. Cada unha delas represéntase por un Context.
- Cada Context representa unha aplicación web (HTML, CSS, Servlets, JSP...) executándose dentro dun Host.