

# UNIDADE 1: ARQUITECTURAS Y SOFTWARE DE PROGRAMACIÓN WEB EN CONTORNO SERVIDOR

## Índice

---

<b>1. Fundamentos das aplicacións web .....</b>	<b>2</b>
Aprenderemos a.....	2
1.1 Internet: Conceptos básicos .....	2
1.1.1 A Web .....	2
A Web 1.0.....	3
A Web 2.0.....	3
A Web 3.0.....	4
A Web 4.0, 5.0, 6.0, .....	4
1.2 Definición de aplicación web.....	4
1.3 Linguaxes empregadas polas aplicacións web .....	5
1.3.1 Linguaxes do lado do cliente .....	5
1.3.2 Linguaxes do lado do servidor .....	5
1.4 Bases de datos asociadas a un servizo web .....	6
1.5 Servidores web.....	6
1.5.1 Arquitectura dunha aplicación web .....	7

# 1. Fundamentos das aplicacións web

## Aprenderemos a...

- Describir as **orixes de Internet** e a web.
- Definir o **concepto de aplicación web**.
- Identificar as **linguaxes de cliente** e de **servidor** e diferencias entre ambas.

## 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 houbese unha catástrofe que destruíse algún deles, o resto da rede continuase funcionando.

Alguns 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.

### 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=UUpw-7\\_SWds](https://www.youtube.com/watch?v=UUpw-7_SWds)

Estas son as distintas etapas da web:

- A **web 1.0**. Caracterizada polo seu **contido estático**
- A **web 2.0**. Caracterizada polo seu **contido dinámico** e o seu carácter **colaborativo**.
- A **web 3.0**. Caracterizada polo seu carácter **semántico**.
- A **web 4.0**. Caracterizado polo seu **carácter ubicuo**, na que exista unha comunicación entre as persoas e as cousas.

### A Web 1.0

Inicialmente, as **páxinas eran estáticas** e os **usuarios non podían interactuar** con elas. As súas principais características son:

- Páxinas **escritas a man** en linguaxe HTML.
- **Contido e deseño xuntos no código HTML**.
- **Non se usaban** as **bases de datos**, polo que non existían rexistros de usuarios, todos eran visitantes.

### A Web 2.0

O termo Web 2.0 xurdiu no **ano 2004**, acuñado por Dale Dougherty, de O'Reilly Media referíndose ao tipo de sitios web que sobreviviron e foron xurdindo **trala crise das punto com**. Pouco a pouco fóronse definindo as súas **características**, das que podemos salientar:

- Os **usuarios** pasan de ser meros espectadores a contribuír e **participar activamente**.
- Uso de bases de datos que permiten o **rexistro de usuarios**, a creación de perfís, a inserción de comentarios e discusións, etc.
- **Experiencia de usuario** máis rica e **dinámica**, grazas a tecnoloxías como **Ajax** (Asynchronous JavaScript + XML).
- A importancia do **long tail**. A web non está formada por uns poucos sitios de empresas potentes, senón por moitos sitios pequenos. Ademais, procúranse modelos de mercado nos que non só se ofrezan uns poucos produtos con moitas vendas, senón moitos produtos cun número menor de vendas pero que, unidas, poden incluso superar ás primeiras.

Na Web 2.0 atopamos os seguintes tipos de **servizos**:

- **Redes sociais**: Facebook, Tuenti, etc.
- **Foros**
- Plataformas para **compartir contido** multimedia: Flickr, Youtube, Instagram, etc.
- Ofimática web, como Google Drive.
- **Blogs** e microblogs: Blogger, Wordpress, etc.

- **Wikis**, como a amplamente coñecida Wikipedia
- Sistemas de **marcadores sociais**: Diigo, Delicious, Pinterest, etc.

### A Web 3.0

A Web 2.0 está transformándose pouco a pouco na Web 3.0 ou **Web semántica** (localización de contido a través palabras chave), onde se **analiza o comportamento dos usuarios** para ofrecerlle contidos máis apropiados de acordo ás súas **preferencias** ou características. Así, son importantes o **historial de navegación** e a **compras**, as preferencias, **gustos**, etc. **Explotación e análise de datos.**

A Web 3.0 é máis intelixente e os usuarios poden facer **procuras máis próximas á linguaxe natural**. Ademais, a información terá contido semántico asociado, polo que será máis sinxelo relacionar información de distintas fontes.

### A Web 4.0, 5.0, 6.0, ...

O desenvolvemento da Web 3.0 avanza cara á Web 4.0, 5.0, ..., na que o obxectivo será **unir as intelixencias de xeito que tanto as persoas como as cousas se comuniquen entre si para a toma de decisións**. Aplicación da **intelixencia artificial** e **integración en múltiples dispositivos**:

- Uso de **gafas** de realidade aumentada.
- Diálogo cun **axente virtual** intelixente.
- Integración nos **vehículos**.
- Implantes **neurónais** con acceso directo á web.

## 1.2 Definición de aplicación web

Na enxeñaría de software denomínase **aplicación web** a aquela **ferramenta que os usuarios poden utilizar accedendo a un servidor web a través de Internet ou dunha intranet mediante un navegador**. Noutras palabras, é unha aplicación software que se codifica nunha linguaxe soportada polos navegadores web na que se confía a execución ao 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**.

**SERVIDOR WEB + LINGUAXES DE PROGRAMACIÓN + SERVIDOR BBDD**

## 1.3 Linguaxes empregadas polas aplicacións web

Para crear unha **aplicación web** debemos servirmos dunha serie de linguaxes que nos permiten **deseñar e programar a aplicación**. A continuación expoñemos **algunhas das linguaxes máis empregadas** diferenciando as **linguaxes do lado do cliente e do servidor**.

### 1.3.1 Linguaxes do lado do cliente

As linguaxes do lado do cliente son aquelas que **poden ser interpretadas directamente polo navegador web**. Son completamente independentes do servidor, o que **permite que a páxina poida ser aloxada en calquera servidor**. O único que **fai falta é que o navegador teña instalados os plugins adecuados**. O **código é visible polo cliente**, o que pode afectar á seguridade. Estas son as máis importantes:

- **HTML** (Hypertext Markup Language). Trátase dunha linguaxe de marcas que permite crear **contido estático** nos documentos web, definindo a súa estrutura. Esta será a linguaxe que trataremos nesta parte.
- **CSS** (Cascade Style Sheet). É unha linguaxe empregada para **definir a presentación dun documento web**. Esta linguaxe será tratada a continuación de HTML.
- **JavaScript**. É unha linguaxe de programación que **permite crear scripts do lado do cliente** para, por exemplo, **controlar o navegador ou alterar o contido** que se amosa nos documentos web xerando **contido dinámico**.
- **AJAX** (Asynchronous JavaScript and XML). Trátase un conxunto de tecnoloxías web que **permite que as páxinas web se actualicen de forma asíncrona intercambiando pequenas cantidades de datos co servidor** de xeito que **sexa posible actualizar partes da páxina sen recargar a páxina completa**. As páxinas web que non utilizan AJAX deben recargar a páxina completa cando o contido cambia.

### 1.3.2 Linguaxes do lado do servidor

Son aquelas linguaxes que son **recoñecidas, executadas e interpretadas polo propio servidor** e se **envían ao cliente nun formato comprensible para el**. Son **independentes do navegador utilizado**. Ademais, o **código pode ocultarse ao cliente**, que só verá o **código HTML finalizado**. Algunhas das máis populares son:

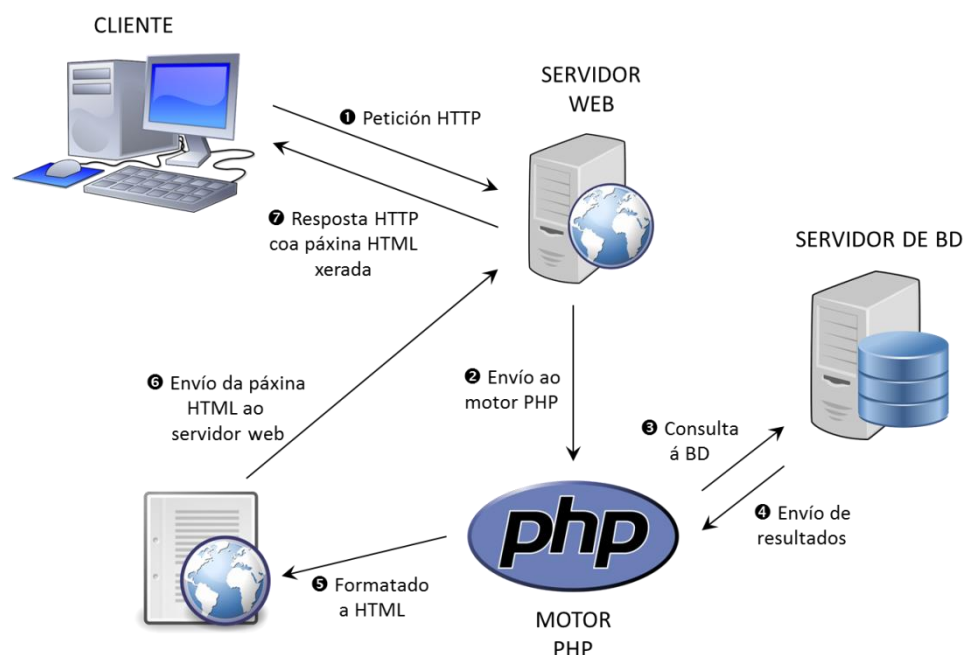
- **PHP**. Linguaxe de **script multiplataforma** creado especialmente para a **xeración de páxinas web dinámicas**, posto que **pode ser incrustado dentro do código HTML**.
- **Java: páxinas JSP e Servlets**. Orientadas a desenvolver **páxinas web en Java**. Nas páxinas **JSP** o **código Java inclúese no documento HTML** e execútase no servidor para incluír **contido** nestas **de forma dinámica**. Mentres que os **Servlets** son **aplicacións Java completas no servidor** que teñen como uso máis común a **xeración de forma dinámica de páxinas web** a partir dos parámetros que envía os navegadores web.
- **ASP.NET**. É a sucesora de **Active Server Pages (ASP)** desenvolvido e comercializado por **Microsoft**. Consiste nun conxunto de clases **.NET** utilizadas para crear páxinas web. **Utilízase en servidores Windows**.
- **Python**. É unha linguaxe **multiplataforma** e multiparadigma, xa que permite o desenvolvemento de estilos diferentes de programación. É unha **linguaxe simple, versátil e rápida** de desenvolver. Incide nunha sintaxe que favoreza un código lexible.

- **Perl**. É unha linguaxe de programación moi practica para **extraer información de arquivos de texto e xerar informes** a partires do contido dos ficheiros.

## 1.4 Bases de datos asociadas a un servizo web

Un **Sistema Xestor de Bases de Datos** permite integrar as bases de datos nas aplicacións web, o que nos permitirá o almacenamento de información nas mesmas. Para isto se precisa unha interface que consiste nunha páxina web dinámica na que se escribe o código necesario nunha linguaxe apropiada, por exemplo PHP, para realizar a conexión, extraer a información, interactuar con ela e formateala adecuadamente para a súa visualización por parte dun navegador.

O funcionamento dunha aplicación web con conexión a datos sería a que se amosa na seguinte figura:



Como Sistema Xestor de Bases de Datos, podemos atopar, entre outros, o seguinte software:

- Solucións de **código aberto**: **MySQL** (que será o que empreguemos neste módulo), **María DB**, **PostgreSQL** ou **Firebird**.
- Software **proprietario**: **Oracle RDBMS** ou **Microsoft SQL Server** de Microsoft.

## 1.5 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:

- **Escoita 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

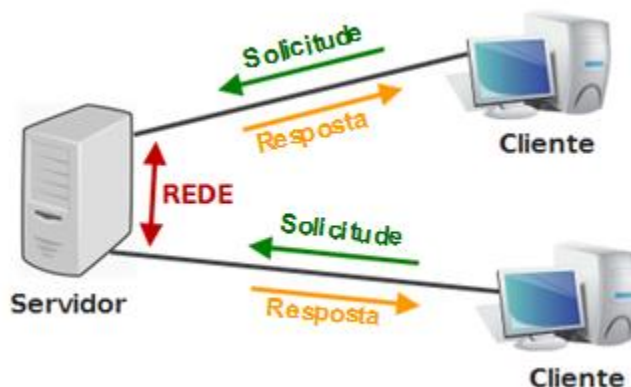
Para este propósito podemos atopar, entre outros, o seguinte software:

- **Apache** (<http://www.apache.org>), de código aberto, para o que existen versións tanto para Windows como para Linux. 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.

Como veremos nos seguintes apartados, as aplicacións web precisan doutros servizos ademais dun servidor web, polo que, habitualmente, se instalan solucións software que inclúen eses servizos, o que se coñece como servidores **WAMP** (Windows Apache MySQL PHP) ou **LAMP** (Linux Apache MySQL PHP). Neste módulo instalaremos e empregaremos XAMPP. O nome provén do acrónimo de **X** (para calquera dos diferentes sistemas operativos), **A**ppache, **M**ySQL, **P**HP, **P**erl.

### 1.5.1 Arquitectura dunha aplicación web

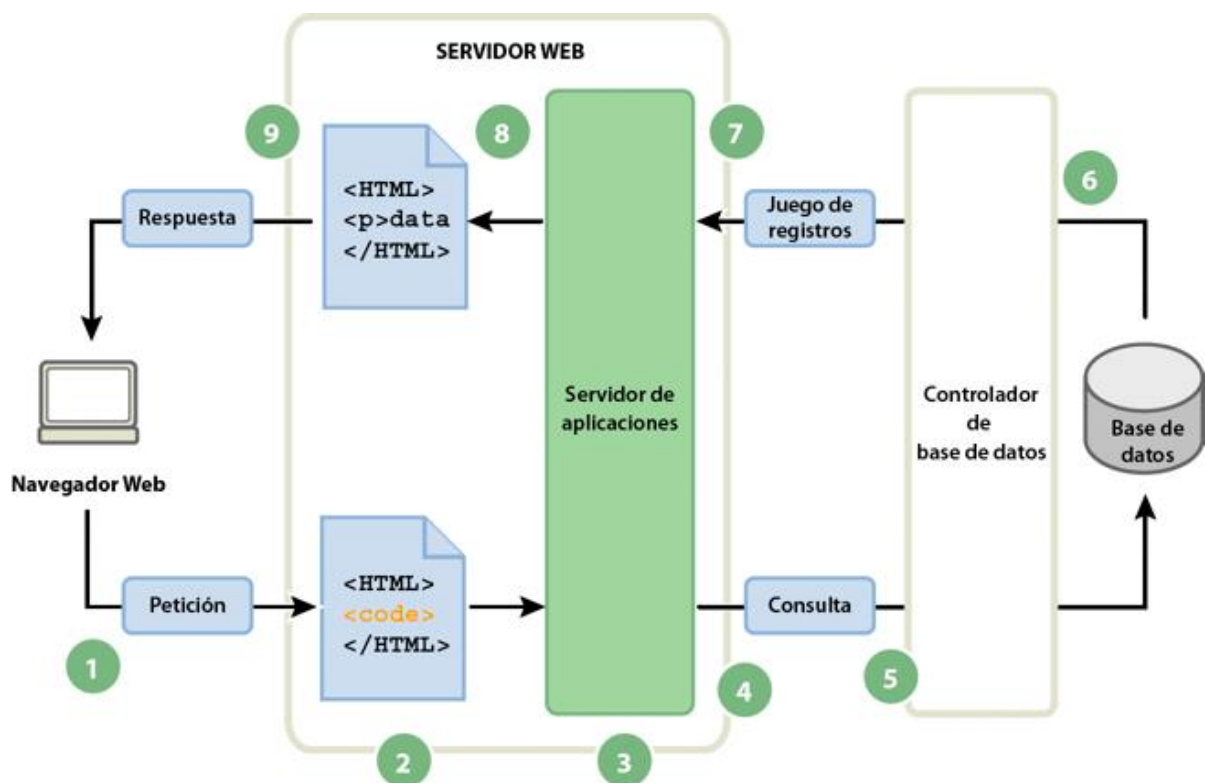
As aplicacións web baséanse nunha arquitectura cliente/servidor, onde por un lado está o cliente, que será o navegador web, e polo outro lado está o servidor, que é o servidor web.



Aínda que existen moitas variacións posibles, unha aplicación web está normalmente estruturada como unha aplicación multicapa, máis habitualmente como unha aplicación de tres capas, onde:

- **Primeira capa:** navegador web.
- **Segunda capa** ou capa intermedia: motor que sexa capaz de usar algunha tecnoloxía web dinámica, como PHP, Python, etc.
- **Terceira capa:** a base de datos.





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 envíalla 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 necesitará 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 está completa, pásalla ao servidor web.
- 9. O servidor web envía a páxina finalizada ao navegador solicitante.



O funcionamento dunha aplicación web programada en PHP con conexión a datos sería a que se amosa na seguinte figura:

