

UD1 - Arquitectura e ferramentas de programación



Preparando o contorno



Índice

1.1 Introducción

1.2 Características da programación web

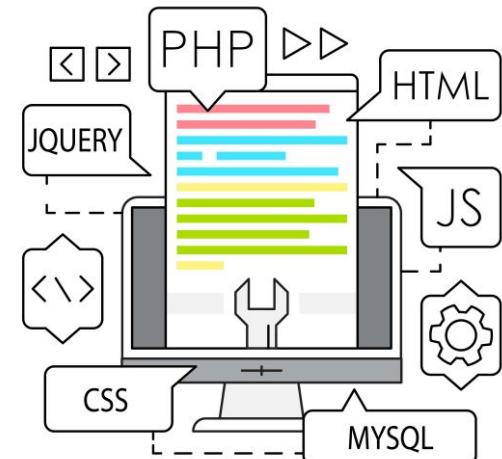
1.3 Execución de código en servidor e cliente

1.4 Tecnoloxías de programación do lado do servidor

1.5 Linguaxes de programación

1.6 Ferramentas de programación

1.7 Instalación da plataforma PHP



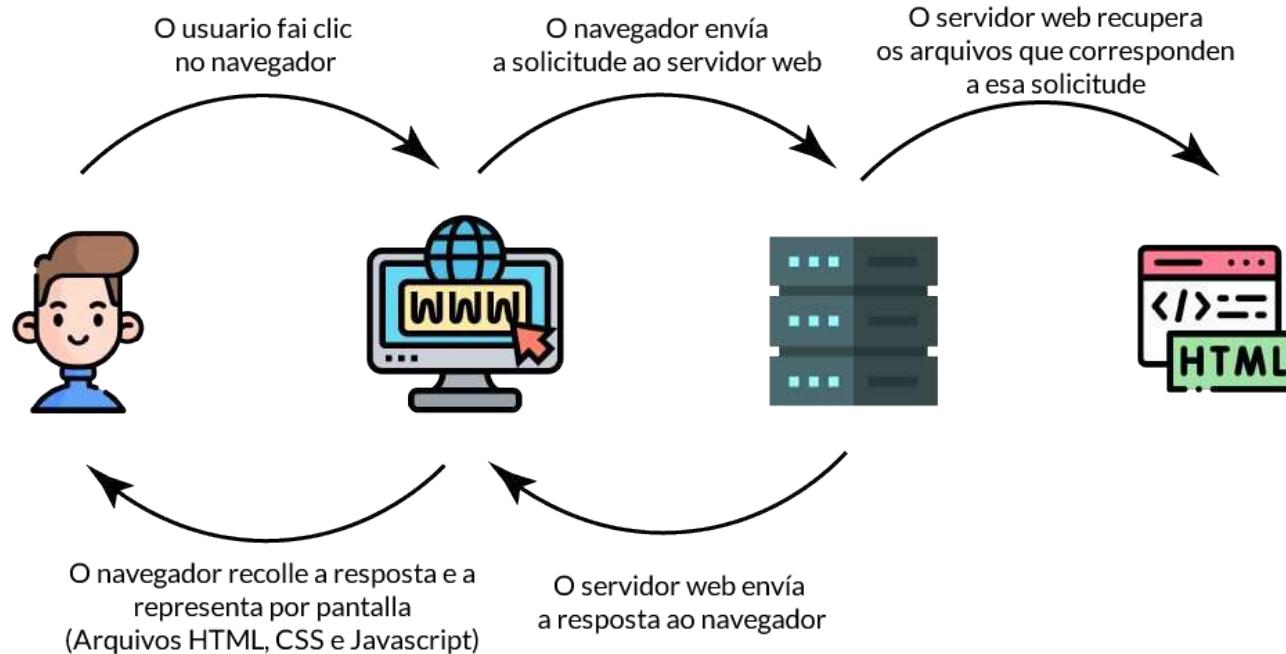
1

Introducción



1.1 Introducción

- ¿Como se accede a unha páxina web?

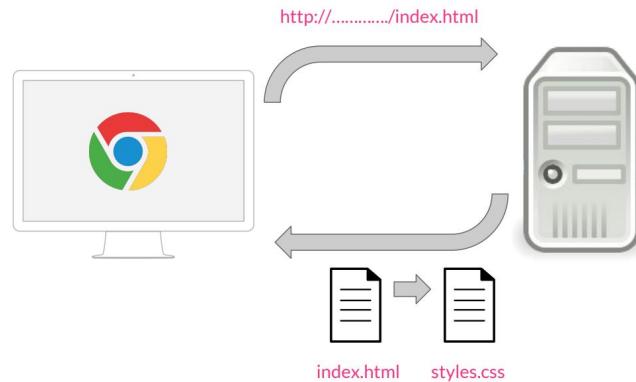


- Primeiro o navegador envía una petición HTTP ao servidor web. Despois, o servidor web busca a páxina no seu almacén de páxinas e no caso de atopala a recupera. Se non a atopa devolverá un erro HTTP (404 - Non atopada). Por último, envía ao navegador a páxina que solicitou.



1.1 Introducción

- Cando navegamos por Internet accedemos a **páxinas web**.
- As páxinas web son documentos que se descargan ao ordenador e son abertos e representados polos navegadores.
- O contido deses documentos indícalles que se debe amosar en pantalla. O contido está definido cunha linguaxe de marcas que pode ser **HTML ou XHTML** (representado coma un XML válido).
- As etiquetas indican a estrutura da páxina, separan as diferentes partes e entre elas aparece o contido a incluír.
- Como extensión a estrutura e contidos da páxina poden incorporarse outros ficheiros, as follas de estilos ou **CSS** que indican as cores, tipos de letra, aliñamento e outras accións propias do aspecto visual que terán os contidos da páxina. As follas de estilo poden indicarse nun ou máis ficheiros dependendo da complexidade da páxina web.



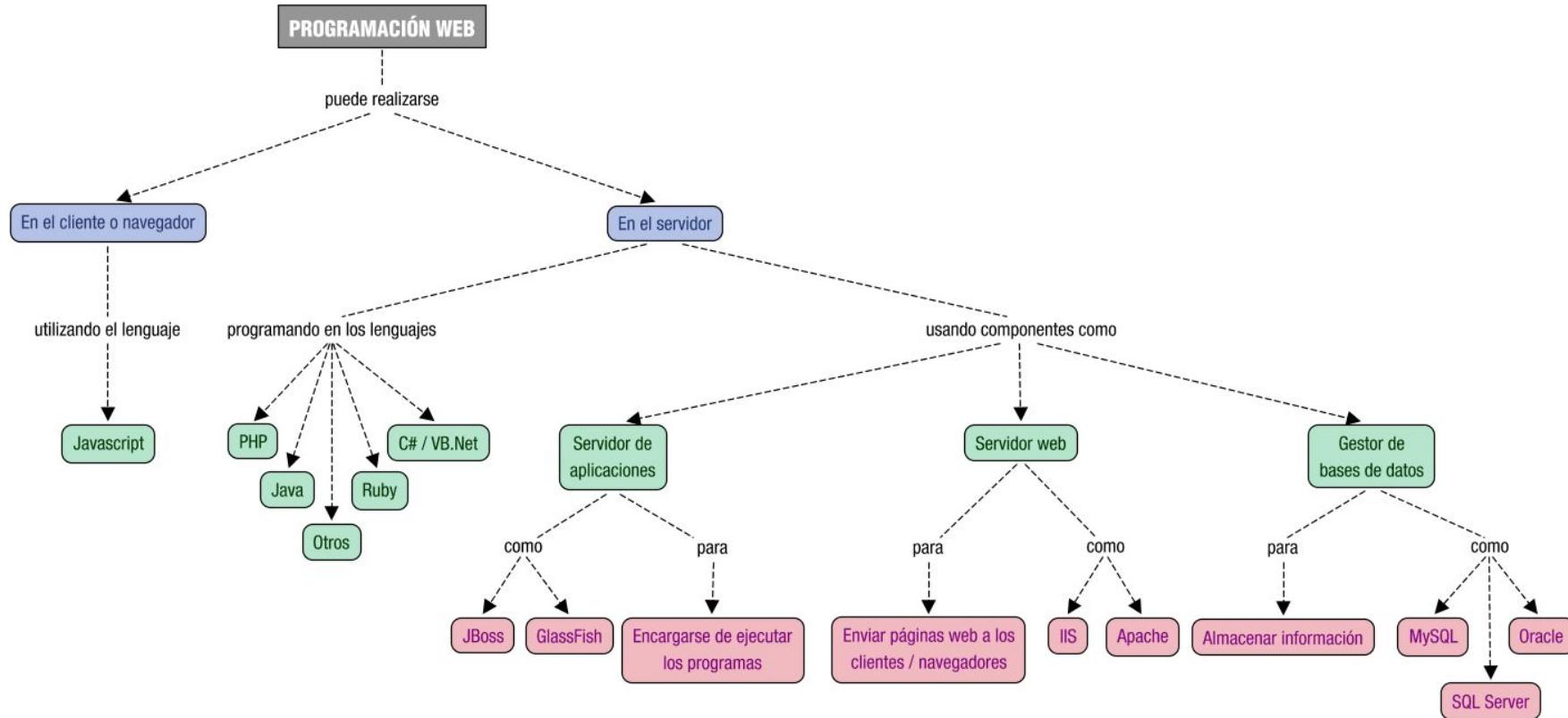


2

Características da programación web



1.2 Características da programación web





1.2 Características da programación web

Páxinas web estáticas

- O contido non pode variar, non hai elementos de programación.
- Definidas con HTML e CSS.
- Poden abrirse en local cun navegador web sen que haxa instalado un servidor.
- Son útiles para amosar unha información concreta, e amosarán esa mesma información cada vez que se carguen. A única forma en que poden cambiar é se un programador/a a modifica directamente no servidor e actualiza o seu contido editando o arquivo.
- Independentes da entrada, sempre dan a mesma saída.





1.2 Características da programación web



Páxinas web dinámicas

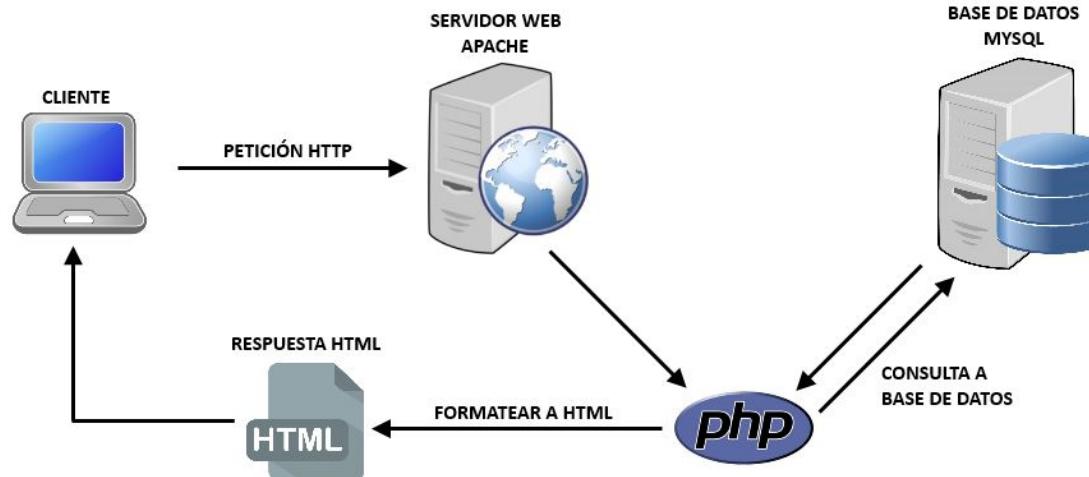
- Teñen diferentes saídas en función da entrada, inclúen código de programación.
- A entrada vén definida pola interacción do usuario, datos inseridos, premer botóns, etc.
- **Tipos:**
 - **Cliente.** Aquelas que inclúen código que se executa no navegador, sendo a linguaxe JavaScript a máis empregada. Este código aparece embebido dentro do HTML e se descarga xunto coa páxina, en ocasións en ficheiros aparte que poden ser mesmo librerías. Cando o navegador amosa a páxina, executa o código de script. Este código pode incorporar múltiples funcionalidades coma poden ser: validacións, animacións, cambios de estilos, accións a eventos, etc. Esta parte incluiríase no desenvolvemento web no contorno do cliente.
 - **Servidor.** Para determinadas páxinas web, o código HTML non se define únicamente en arquivos que se envían ao navegador senón que inclúa linguaxes de programación do lado do servidor (PHP, JSP, ASP, etc) que unha vez executados no servidor son os encargados de devolver o código HTML a representar. O código incluído funciona coma un programa que devolverá HTML e CSS en función dun conxunto de variables. Esta parte incluiríase no desenvolvemento web no contorno do servidor.



1.2 Características da programación web

Pasos:

- O cliente web (navegador) dun ordenador solicita a un servidor web una páxina web a través dunha petición HTTP(S).
- O servidor busca esa páxina e se atopa a recupera, senón devolve un erro 404.
- No caso de que se trate dunha páxina web dinámica, é dicir, con contido executable, para obter o código HTML que se devolverá, o servidor web contacta co módulo responsable de executar o código e llo envía.
- Como parte do proceso de execución, pode ser necesario obter información dalgún outro servidor, coma por exemplo consultar rexistros almacenados nunha base de datos.
- O resultado da execución será una páxina en formato HTML, similar a calquera outra páxina web estática.
- O servidor web envía o resultado obtido ao navegador, que a procesa e amosa en pantalla como calquera outro código HTML e se ten algunha linguaxe de script para o cliente, a executa o propio navegador.





1.2 Características da programación web

Aplicacións web

- As aplicacións web empregan páxinas web dinámicas para crear aplicacións que executan código nun servidor e devolven o HTML a representar en cada navegador. Exemplos: xestores de correo, tendas, xestores de contido e portais web, redes sociais, etc.
- **Vantaxes:**
 - **Instalación** → Non é necesario instalalas en cada un dos equipos nos que se vaian a utilizar. Abonda instalalas no servidor e xa terá acceso a elas calquera usuario/a dende un navegador web.
 - **Mantenemento** → Como só se instalan nun único equipo (o servidor), faise máis sinxelo xestionalas (facer copias de seguridade dos seus datos, corrixir errores, actualizalas, etc).
 - **Multiplataforma** → Pódense utilizar en todos aqueles dispositivos que dispoñan dun navegador web, independentemente das súas características (non precisan dun equipo potente, agás o servidor) ou do seu sistema operativo, resultan independentes da plataforma xa que todas teñen navegadores web.



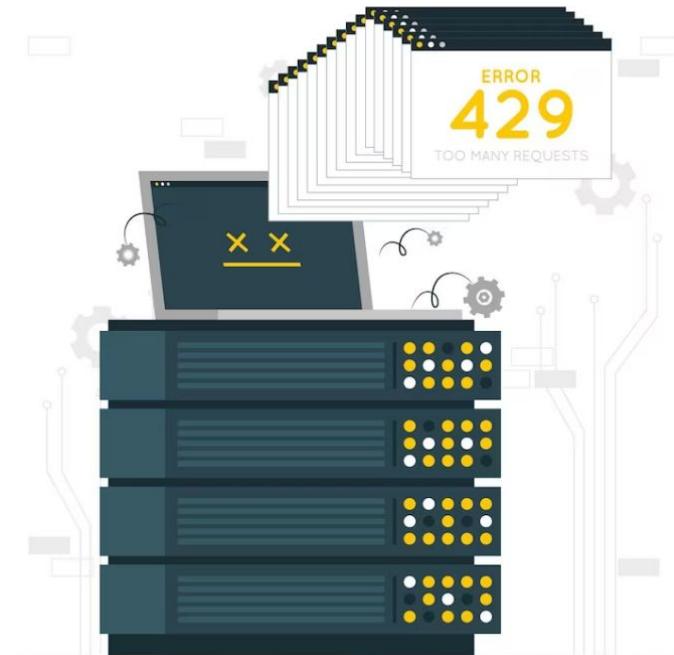


1.2 Características da programación web

Aplicacións web

- **Desvantaxes:**

- **Interface** → A interface de usuario das aplicacións virá limitado polo navegador, o que restrinxe as súas características ás dunha páxina web e dificulta o desenvolvemento.
- **Conectividade** → Para poder empregalas vaise depender sempre de que teñamos conexión co servidor (en moitos casos Internet). Tamén será afectada polo ancho de banda da conexión e sobrecargas do servidor.
- **Uso de recursos** → Determinadas aplicacións precisan de moitos recursos gráficos e de procesamento como por exemplo editores de vídeo, de gráficos 3D ou de debuxo técnico, polo que un enfoque a aplicación de escritorio permitiría aproveitar mellor os recursos ao facer uso dos da máquina cliente.
- **Tolerancia a errores** → Se cae o servidor, perden a funcionalidade todos os clientes.





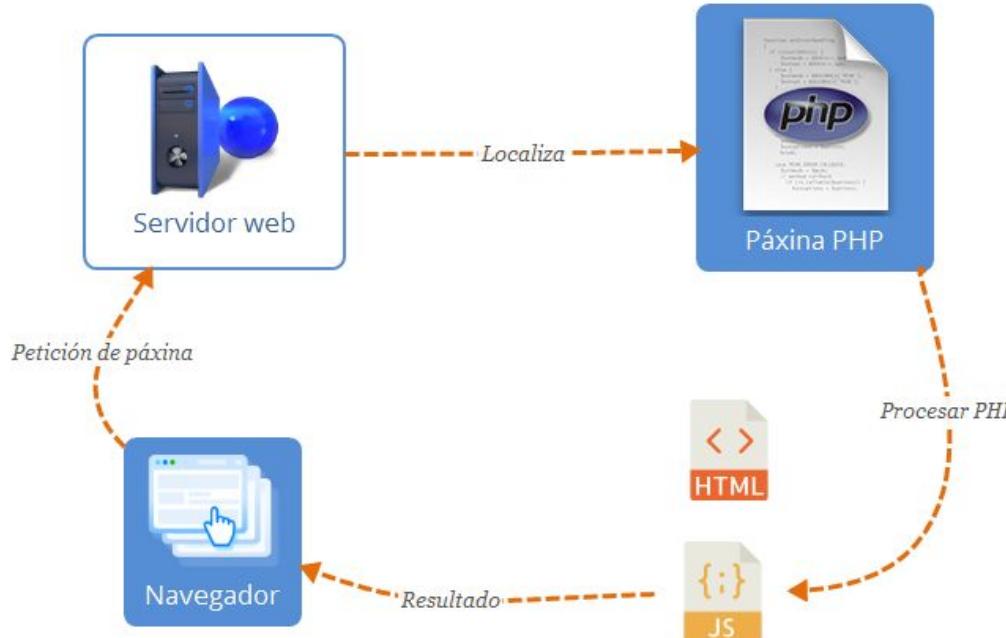
3

Execución de código en servidor e cliente



1.3 Execución de código en servidor e cliente

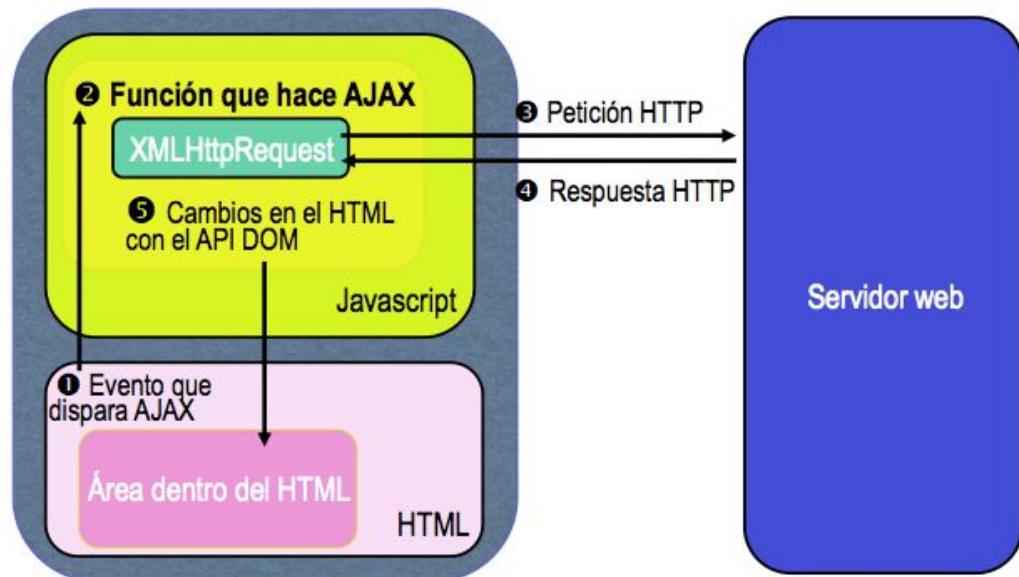
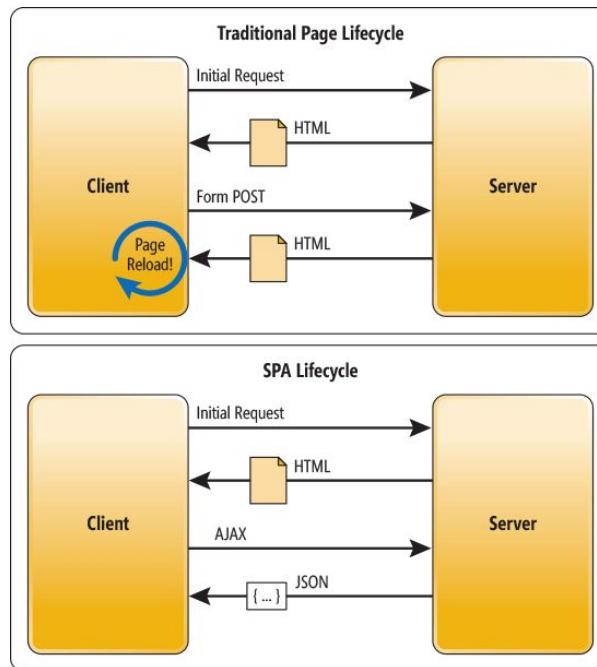
- Como se indicaba no apartado anterior, cando o navegador solicita ao servidor web unha páxina pode que antes tivera que executar, o propio servidor ou por delegación outro programa o código que permita obter a páxina. Nestes casos **o código ejecútase no servidor**.
- Ademais cando o arquivo HTML cárgase no navegador do ordenador, analízase se inclúe algún fragmento de código de script que se deba executar. Ese código de script executarase no propio navegador, é dicir, **executarase no cliente**.





1.3 Execución de código en servidor e cliente

- Estas dúas tecnoloxías, scripts de cliente e scripts de servidor, compleméntanse mutuamente. Esta división en dúas capas de scripts vén orixinalmente de que na capa do cliente non había acceso a datos, xa que se almacenan no servidor. Actualmente as tecnoloxías do lado cliente permiten o acceso a datos a través de linguaxes de script. Así por exemplo para o caso do Javascript dispense da tecnoloxía **AJAX**, que permite acceder de maneira asíncrona. Aínda así, moitas das aplicacións web actuais empregan estas dúas tecnoloxías: a execución de código en servidor e en cliente, mesmo con acceso a datos.





4

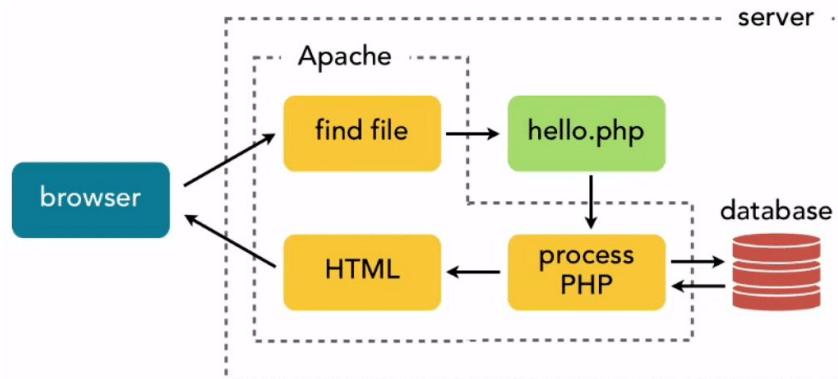
Tecnoloxías para programación web do lado do servidor



1.4 Tecnoloxías para programación web do lado do servidor

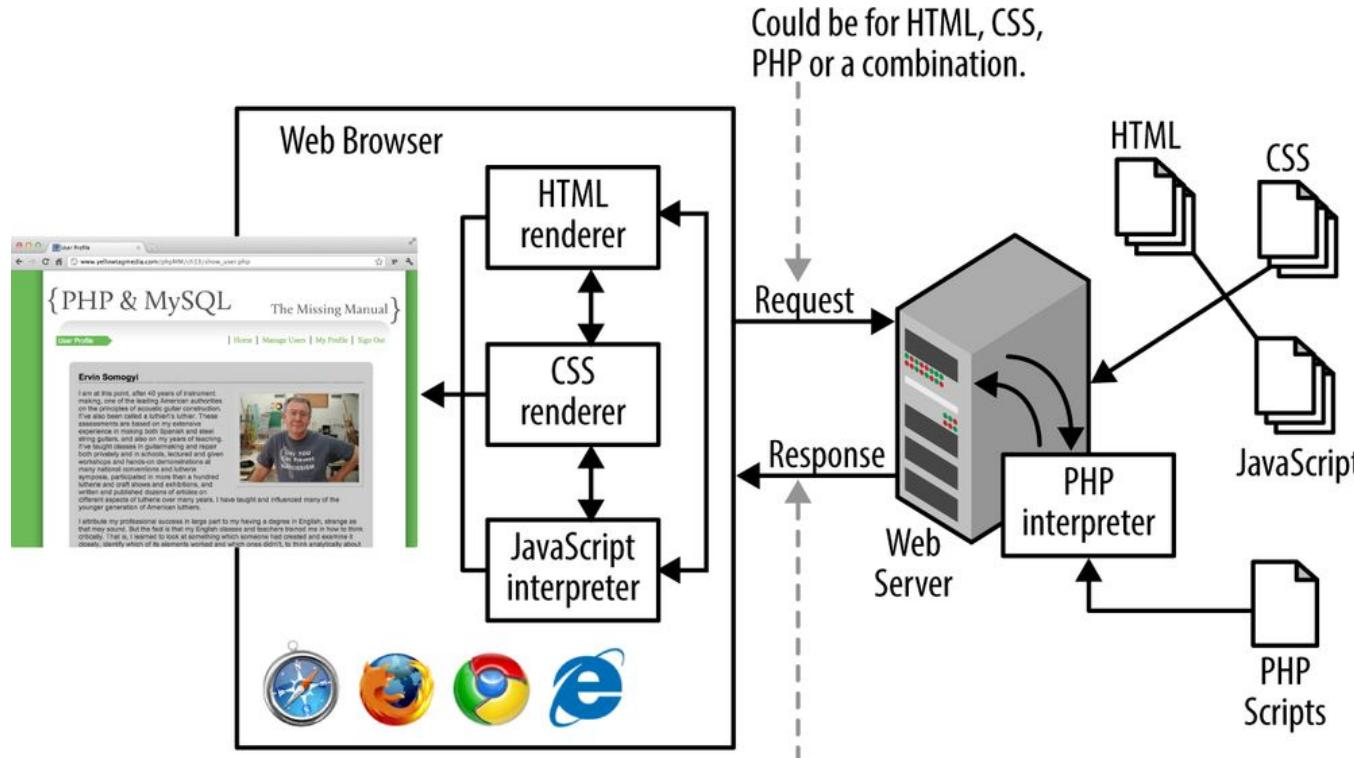
Á hora de programar unha aplicación precisaremos dos seguintes **elementos**:

- Un **servidor web** para recibir as peticións dos clientes web (normalmente navegadores) e enviarlle os documentos que solicitan (unha vez xerada posto que son páxinas web dinámicas).
- O **módulo encargado de executar o código** ou programa e xerar a páxina web resultante. Este módulo debe integrarse dalgunha forma co servidor web, e dependerá da linguaxe e tecnoloxía seleccionados para programar a aplicación web.
- Unha **aplicación ou servidor de base de datos**, que normalmente tamén será un servidor de BD. Este módulo non é estritamente necesario pero o habitual é que teña cando menos un sistema de almacenamento de datos (XML, ficheiros, servizo de directorio, etc).
- A **linguaxe de programación** que se utilizará para desenvolver a aplicación.





1.4 Tecnoloxías para programación web do lado do servidor



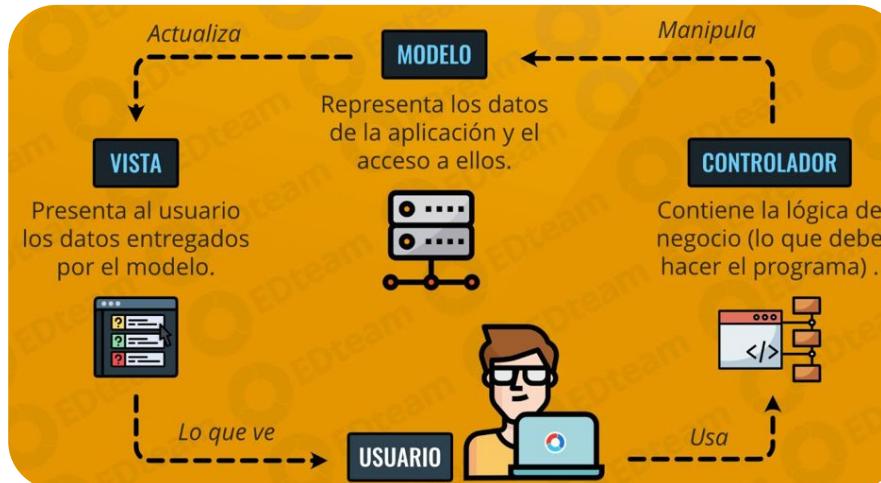
Could be for HTML, CSS, PHP or a combination.

Response is not PHP, but the result of interpreting PHP, usually more HTML and CSS.



1.4 Tecnoloxías para programación web do lado do servidor

- Ademais dos componentes que se van utilizar, tamén haberá que definir como **organizar o código** da aplicación, é dicir a súa arquitectura. As arquitecturas que se usan na programación de aplicacíons web **estruturan o código en capas o niveis**. Existen varios motivos para separar as funcións lóxicas nun deseño en capas, axuda a reutilizar código, documentar, encapsular funcionalidades comúns, ou executar unha capa nun servidor distinto ao web, por exemplo a capa de acceso a datos.
- Nunha aplicación se pode distinguir, de forma xeral, **funcións de control** ou encamiñamento de peticións, **funcións de vista ou presentación** (para dar formato aos datos de cara a presentallos ao usuario final), **de negocio ou lóxica** (utiliza os datos para executar un proceso e obter un resultado), **persistencia** (que mantén os datos almacenados de xeito organizado) e de **acceso ou datos** (que obtén e insire datos no espacio de almacenamento).





1.4 Tecnoloxías para programación web do lado do servidor

- Outro exemplo, podemos organizar aplicacións de 3 capas formadas por:
 - Unha **capa cliente**, que é onde se programará todo o relacionado coa interface de usuario, é dicir, a parte visible da aplicación coa que interactuará o usuario.
 - Unha **capa intermedia** onde se programará a funcionalidade da aplicación.
 - Unha **capa de acceso a datos**, que se encargará de almacenar a información da aplicación nunha base de datos e recuperala cando sexa necesario.





1.4 Tecnoloxías para programación web do lado do servidor

Linguaxes e plataformas

- **Java EE (Enterprise Edition).** Plataforma orientada á programación de aplicacións Java. Portable, multisistema e con librerías e especificacións para facilitar os desenvolvimentos. Está soportada por empresas como Sun ou Oracle, aínda que dispón de todos os compoñentes necesarios como software libre. Integra tecnoloxías como JSP, Servlets para a capa de vista e EJB e outros frameworks de desenvolvemento para a parte da lóxica de negocio. Como resulta algo complexa e a infraestrutura consume bastantes recursos, acostuma a empregarse en solucións de mediano ou gran tamaño.
- **AMP (Apache, Mysql/MaríaDB e PHP).** O nome desta arquitectura fai referencia ás tecnoloxías que a integran: Apache como servidor web (en competencia con Nginx), MySQL como servidor de base de datos e PHP como linguaxe de programación máis empregado, aínda que existen outras linguaxes de script equivalentes coma Perl ou Python. Dependendo do Sistema Operativo sobre o que se implante pode denominarse LAMP (Linux), WAMP (Windows) ou MAMP (Para MacOS). Tamén se poden substituír outros compoñentes como o SXBD MySQL/MariaDB ou PostgreSQL. Todos os compoñentes esenciais (agás os sistemas operativos Windows e MacOS) son de código aberto (*open source*). Xa que resulta moi lixeira e sinxela de implementar acostuma a empregarse en solucións de pequeno ou mediano tamaño. O seu principal punto forte vén da grande comunidade de usuarios que a soporta o grande número de aplicacións e frameworks xa implantados como Wordpress, Drupal, Prestashop, ou Moodle.



1.4 Tecnoloxías para programación web do lado do servidor

Linguaxes e plataformas

- **CGI/Perl.** Combina dous componentes: Perl como linguaxe de programación de código aberto e CGI un estándar para que os servidores web poidan executar directamente programas xenéricos e devolver HTML. É a plataforma máis antiga das expostas e a execución de CGI pode resultar algo lenta comparada con outras plataformas. Actualmente atópase en desuso.
- **ASP/.NET.** Plataforma comercial de Microsoft orixinalmente baseada na linguaxe ASP que medrou a .NET, incorporando linguaxes coma Visual Basic .NET ou C# entre outros. O servidor web sería Microsoft IIS (Internet Information Server) e permite diferentes SXBD sendo o habitual Microsoft SQL Server, pola súa facilidade de integración nesta plataforma.
- **Outros:** Python, Ruby, Go, Node.js (Javascript), etc.

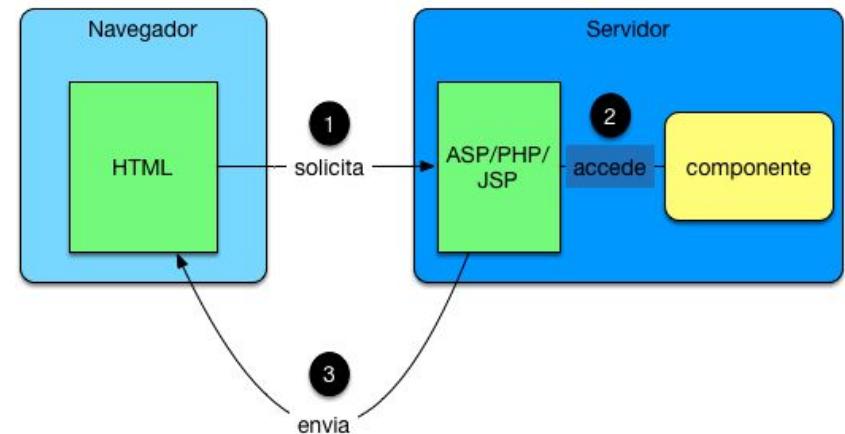
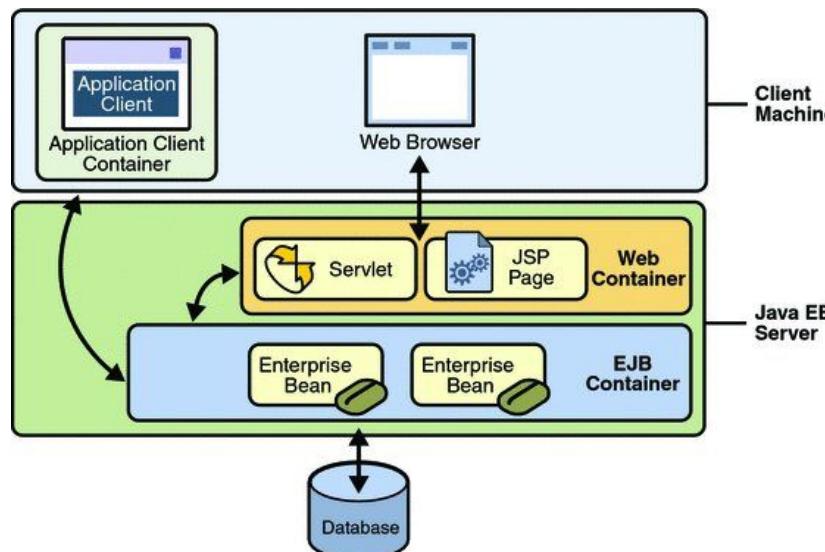




1.4 Tecnoloxías para programación web do lado do servidor

Servidores de aplicacións

Tecnoloxías como JEE e .NET dispoñen de servidores de aplicacións como IIS, WebLogic, WebSphere ou Apache Gerónimo. Estes servidores amplían as funcionalidades dos servidores web como Apache e Nginx. Estas funcionalidades inclúen: escalabilidade e balanceo de carga, melloras no rendemento, caché, seguridade e despregamento das aplicacións, así coma no uso de recursos, en especial compoñentes que se executan no servidor como librarías ou obxectos de programación compartidos. Coma desvantaxes requiren de máis configuración, é máis complexo o mantemento, aumentan o custe do proxecto e requieren de persoal especializado.





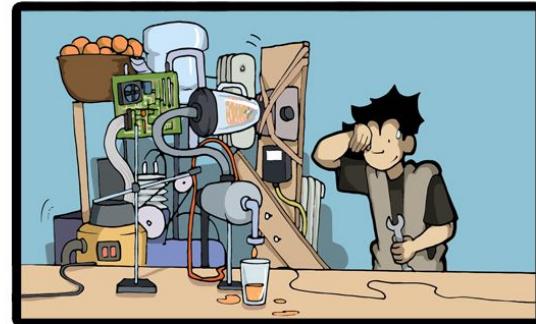
1.4 Tecnoloxías para programación web do lado do servidor

¿Como escoller a plataforma axeitada?

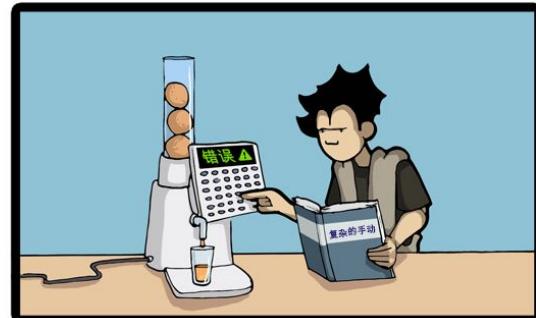
Cada proxecto software terá as súas particularidades e as diferentes plataformas ofrecen características moi similares pero tamén teñen diferenzas que se poden axustar mellor ou peor a cada problema. Por outra banda a nosa experiencia persoal con cada tecnoloxía tamén será un factor determinante á hora de decidir. ¿Que factores poden terse en conta?

- O tamaño do proxecto.
- Se xa coñezo unha plataforma, ¿compensa aprender outra e súa linguaxe de programación para este problema?
- Ferramentas de código aberto ou comercial, ¿que custos engadidos habería?
- ¿Teño completa liberdade de elección ou xa teño compoñentes impostaos pola infraestrutura do sitio onde se vai integrar?

Empregar unha tecnoloxía que coñeces ben pero non se adapta ao proxecto.



Empregar unha tecnoloxía que se adapta perfectamente pero da que non sabes nada.





5

Linguaxes de programación



1.5 Linguaxes de programación

Ademais da sintaxe, unha das diferenzas máis destacables entre unha linguaxe de programación web e outra, e que serve para clasificalas, é **segundo a maneira en que se executan no servidor web**, deste xeito habería:

- **Linguaxes de guións (scripting) ou interpretados.** Son aquelas nas que os programas se executan directamente a partir do seu código fonte orixinal, que se almacena nun ficheiro de texto. Cando o servidor web necesita executar código programado nunha destas linguaxes, pásalle a petición a un intérprete, que procesa as liñas do programa e xera como resultado unha páxina web.
Exemplos de linguaxes deste tipo serían: **Perl, Ruby, Python, PHP e ASP**.
- **Linguaxes compiladas a código nativo.** Son aquelas nas que o código fonte tradúcese a código binario, dependente do procesador, antes de ser executado. O servidor web almacena os programas en modo binario, que executa directamente. O método principal para executar programas binarios nun servidor web é o **CGI**, deste xeito podemos facer que se execute código doutras linguaxes de propósito xeral como pode ser **C** ou **C#**. **Go** tamén segue este modelo.
- **Linguaxes compiladas a código intermedio.** Son linguaxes nas que o código fonte orixinal tradúcese a un código intermedio (**bytecode** ou **CIL**), independente do procesador, antes de ser executado. É a forma na que se executan por exemplo as aplicacións programadas en **Java** e **.NET**, e o que permite que poidan executarse en varias plataformas distintas.





1.5 Linguaxes de programación

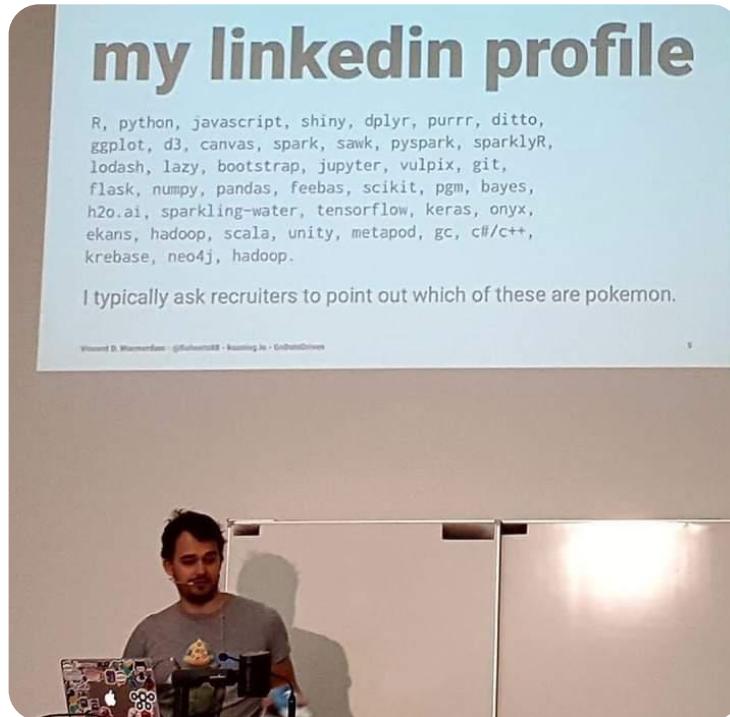




1.5 Linguaxes de programación

Pregunta de entrevista de traballo

¿Cales son linguaxes de programación e cales Pokemon?





1.5 Linguaxes de programación

Cada unha destas formas de execución do código polo servidor web ten as súas **vantaxes e inconvenientes**:

- As **linguaxes de guións ou interpretadas** teñen a vantaxe de que non é necesario traducir o código fonte orixinal para ser executados, o que aumenta a súa **portabilidade**. No caso de precisar realizar algúna modificación a un programa, pode facerse no momento. Polo contrario o proceso de interpretación ofrece un **peor rendemento** que as outras alternativas (dez veces máis lento).
- As linguaxes **compilados a código nativo** son os de maior velocidade de execución, pero teñen **problemas no relativo á súa integración co servidor web**. Son programas de propósito xeral que non están pensados para executarse no contorno dun servidor web. Por exemplo, non se reutilizan os procesos para atender a varias peticionés e por cada petición que se faga ao servidor web, debe executarse un novo proceso. Ademais os programas **non son portables** entre distintas plataformas.
- As linguaxes **compiladas a código intermedio** ofrecen un **equilibrio** entre as dúas opcións anteriores. O seu rendemento é bo e poden portarse entre distintas plataformas nas que exista unha implementación da arquitectura (coma un contedor de *servlets* ou un servidor de aplicacións Java EE). Por contra as aplicacións resultan **máis complexas de desenvolver** que coas linguaxes de guión, por exemplo.





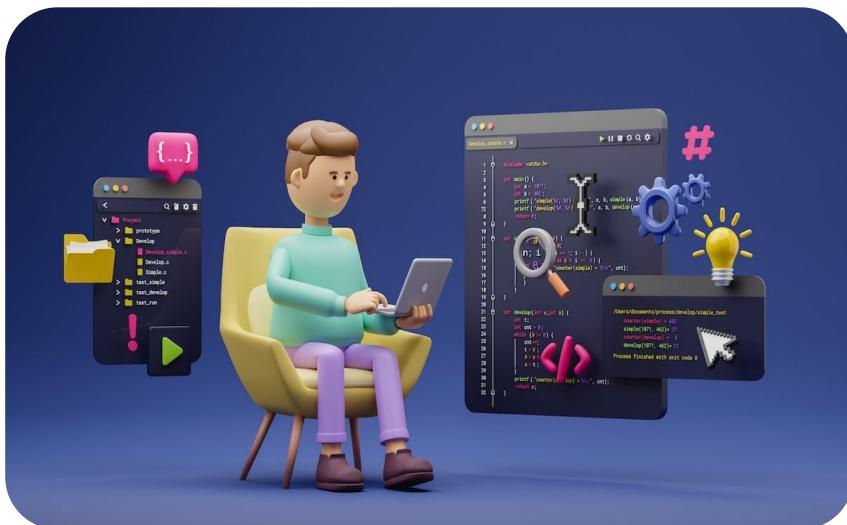
6

Ferramentas de programación



1.6 Ferramentas de programación

- Os **contornos integrados de desenvolvemento (IDE)** agrupan nun único programa unha serie completa de ferramentas que axudan á programación. Algúns deles son específicos dunha plataforma ou linguaxe de programación como Visual Studio, de Microsoft, así coma outros como Eclipse ou NetBeans.
- Non é imprescindible empregar un IDE para programar, realmente abonda cun editor de texto. Algúns coma Gedit, Mousepad ou Notepad++ incluso permiten incluír funcións específicas que praticamente os converten en IDE ultralixeiros, xa que consumen moi poucos recursos. Incluso hai alternativas intermedias como Visual Studio Code, Atom ou Brackets.



Integrated development environment (IDE)

¿Qué es un IDE?



Entorno de desarrollo integrado



NetBeans

eclipse





1.6 Ferramentas de programación



¿Que nos **aportan** os IDE á hora de desenvolver aplicacións web?

- **Resaltado de texto.** Emprega diferentes cores ou tipos de letra para amosar os elementos da linguaxe: comentarios, variables, operadores, errores, etc.
- **Completado automático.** Detecta que se está escribindo e suxire opcións para completalo, se xeito similar aos procesadores de texto.
- **Xeración automática de código.** Certas estruturas como por exemplo as sentenzas de peche pueden completarse automaticamente ao poñer o inicio das mesmas. Isto pode ir moito máis alá con extensións como Copilot, Blackbox ou outras extensións para ChatGPT.
- **Navegación no código.** Permite buscar de xeito sinxelo elementos dentro do texto ou en estruturas xerarquizadas.
- **Comprobación de errores.** Recoñece elementos da sintaxe da linguaxe de programación e indica errores no código, de xeito similar aos correctores ortográficos dos procesadores de texto.
- **Integración con sistemas de control de versións.** Permite administrar as diferentes copias de seguridade e recuperar versións anteriores a cambios, ou fusionar diferentes versións de desenvolvemento dun proxecto.
- **Execución e depuración.** Permite executar un programa para poder probar o seu funcionamento, dando valores ás variables e visualizando os resultados ou os puntos de erro.
- **Instalación de extensións específicas.** Ampliación da funcionalidade con todo tipo de plugins.



7

Instalación da plataforma PHP



1.7 Instalación da plataforma PHP

As plataformas **LAMP** son as máis empregadas na actualidade para desenvolver aplicáisons web. O acrónimo de LAMP indica os compoñentes da arquitectura:

- **L** de **Linux**, como sistema operativo.
- **A** de **Apache**, como servidor web
(ou Nginx).
- **M** de **MySQL** ou **MariaDB** como SXBD.
- **P** de **PHP** como linguaxe de programación
(tamén pode representar Perl ou Python).





1.7 Instalación da plataforma PHP

Servizos de hosting LAMP

Moita competencia e variabilidade nos servizos ofrecidos. Parámetros para elixir un **básico**:

- Prezo (De media 5-10€/mes os plans básicos, cada dominio aparte 15€/ano).
- VPS e Dedicados → Suben 100-500€ más ao mes, depende das prestacións.
- Finalidade (Web → Wordpress, Tenda → Prestashop)
- Capacidad do servidor (15-50 Gb discos NVMe, 1-2Gb RAM, Mbps “ilimitados”)
- Soporte técnico e referencias a caídas do servizo.
- Outros servizos: correo, máis dun hosting/base de datos, backups, SSL, etc.
- País de orixe, a LOPDGDD require que sigan a lexislación europea.





1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

O servidor HTTP Apache é un servidor web de **código aberto e multiplataforma** (Linux, Windows, MacOS e BSD). Posúe unha **arquitectura modular**, de feito que se lle poden ir instalando os módulos segundo se precisen (cantos menos, menos recursos consume). Estes módulos lle permiten interpretar diferentes linguaxes de servidor (PHP, Ruby, Perl, Python, ...), autentificación baseada en LDAP, cifrado SSL e TLS, soporte IPv6, balanceo de carga e moitas más.



Nota: O seu nome débese que se buscaba que tivera a connotación de algo firme e enérgico, pero non agresivo e a tribo Apache foi a última en renderse (ano 1886).

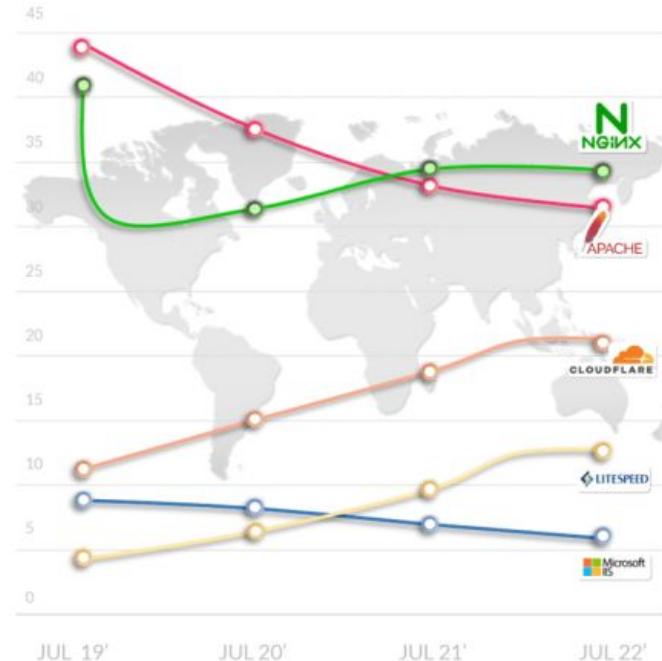


1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

- Durante moitos anos foi o servidor web máis utilizado, aínda que actualmente está a ser igualado por Nginx que está obtendo un mellor rendemento.



A porcentaxe de sitios web que empregan Nginx é do 34.3% fronte ao 31.2% de Apache.

(Datos obtidos en setembro do 2023)

Fuente: w3tech.com



1.7 Instalación da plataforma PHP



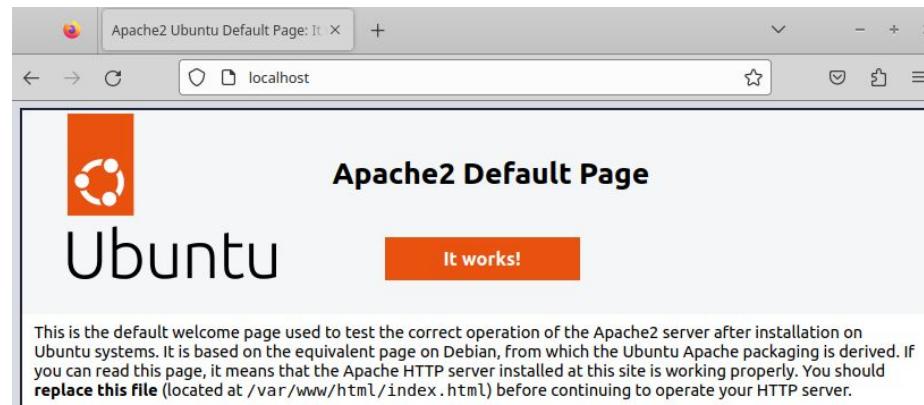
1.7.1 Instalación do servidor web Apache

No caso de Linux, temos **varias opcións de instalación**: por liña de comandos dos **repositorios**, cos **centros de software**, **descargando o software** de cada compoñente (.deb), cun **paquete de instalación** ou cun paquete completo como XAMPP. Alternativamente poderían empregarse **contedores** como Docker ou LXD.

A instalación por liña de comandos é moi sinxela, xa que está incluído nos repositorios oficiais de todas as distribucións:

```
$ sudo apt-get update  
$ sudo apt-get install apache2
```

A continuación procédese a comprobar a instalación accedendo dende un navegador do propio equipo á ruta local <http://localhost> ou <http://127.0.0.1>, co que debería aparecer unha páxina similar a:



This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.



1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

Iniciar e deter o servidor

Método 1: A través do cartafol de servizos/servidores **etc/init.d** poden iniciarse, deterse, reiniciarse ou comprobar o estado [start | stop | restart | status] os servizos ou servidores instalados no equipo, incluíndo o servidor web.

```
$ sudo /etc/init.d/apache2 start | stop | status | restart
```

Método 2: A través do comando **systemctl** poden iniciarse, deterse, reiniciarse ou comprobar o estado [start | stop | restart | status] servizos e unidades do sistema.

```
$ sudo systemctl start | stop | status | restart apache2
```

Método 3: A través do comando **service** poden iniciarse, deterse, reiniciarse ou comprobar o estado [start | stop | restart | status] servizos e unidades do sistema.

```
$ sudo service apache2 start | stop | status | restart
```



1.7 Instalación da plataforma PHP



```
Terminal - xubu@xubu-VirtualBox:~  
Ficheiro Editar Ver Terminal Lapelas Axuda  
xubu@xubu-VirtualBox:~$ sudo service apache2 status  
● apache2.service - The Apache HTTP Server  
    Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)  
    Active: active (running) since Thu 2023-09-07 19:21:13 CEST; 7min ago  
      Docs: https://httpd.apache.org/docs/2.4/  
        Main PID: 2066 (apache2)  
          Tasks: 55 (limit: 9431)  
            Memory: 5.1M  
              CPU: 96ms  
            CGroup: /system.slice/apache2.service  
                  ├─2066 /usr/sbin/apache2 -k start  
                  ├─2071 /usr/sbin/apache2 -k start  
                  └─2072 /usr/sbin/apache2 -k start  
  
Set 07 19:21:13 xubu-VirtualBox systemd[1]: Starting The Apache HTTP Server...  
Set 07 19:21:13 xubu-VirtualBox apachectl[2059]: AH00558: apache2: Could not reliably determine t>  
Set 07 19:21:13 xubu-VirtualBox systemd[1]: Started The Apache HTTP Server.  
lines 1-16/16 (END)
```



1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

Principais arquivos e cartafolos de configuración

Unha vez instalado o servidor Apache, hai que ter en conta que determinados arquivos e cartafolos terán información de configuración relevante. Estes son os más destacados:

- **/var/www/html**. Cartafol que conterá os arquivos que serve Apache (.html, .php, .css, js, ...)
- **/etc/apache2/**. Cartafol de configuración de Apache.
- **/etc/apache2/apache2.conf**. Arquivo de configuración principal de Apache.

Nos equipos Linux en xeral o ficheiro de configuración non soe ser único. En Ubuntu/Debian, todos os ficheiros de configuración están dentro do directorio **/etc/apache2**.

O ficheiro de configuración principal é **/etc/apache2/apache2.conf**. Ten unha estrutura de ficheiro moi ben documentada, e nel atopamos un gráfico en modo texto, que nos explica como se inclúen os ficheiros de configuración.

```
# /etc/apache2/
# |-- apache2.conf
# | `-- ports.conf
# |-- mods-enabled
# | |-- *.load
# | `-- *.conf
# |-- conf-enabled
# | `-- *.conf
# '-- sites-enabled
#   '-- *.conf
```



1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

Principais arquivos e cartafoles de configuración

A explicación de cada ficheiro incluído no directorio **/etc/apache2** é a seguinte:

- **envvars**: ficheiro con variables de contorna e os seus valores.
- **mods-available**: este directorio contén ficheiros de configuración para cargar módulos e configuralos, áínda que non todos os módulos teñen ficheiros de configuración específicos. Actívanse e desactívanse cos **comandos: a2enmod e a2dismod**.
- **mods-enabled**: mantén ligazóns simbólicas aos ficheiros en */etc/apache2/mods-available*. Cando un ficheiro de configuración de módulo é enlazado mediante un ligazón simbólica será activado despois do seguinte reinicio de Apache.
- **ports.conf**: Indica as directivas que determinan os portos TCP nos que Apache escoita e atende peticións (80 e 443 por defecto). Con isto modifícaríase a URL de acceso. Se por exemplo escoitamos no porto 8080 teríamos que indicar <http://localhost:8080/proba.php>
- **sites-available**: Este directorio ten os ficheiros de configuración dos *Host Virtuais* de Apache. Os Host virtuais permiten que Apache sexa configurado para múltiples sitios web con configuracións separadas.
- **sites-enabled**: Ao igual que *mods-enabled*, *sites-enabled* contén ligazóns simbólicas a ficheiros dentro de */etc/apache2/sites-available*. De xeito similar, cando se crea unha ligazón simbólica o sitio virtual estará activo unha vez se reinicie Apache. Actívanse e desactívanse cos **comandos: a2ensite e a2dissite**.



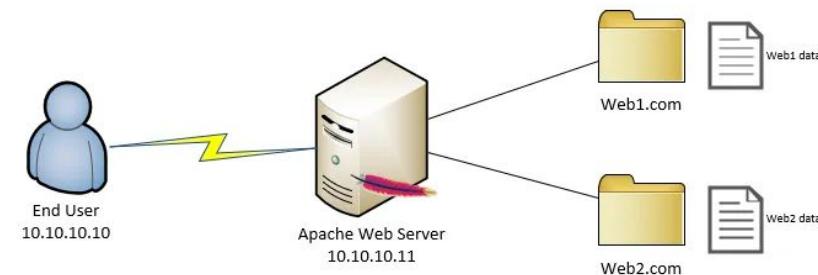
1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

Hosts virtuais

- Os **hosts virtuais baseados en enderezos IP** empregan o endereço IP da conexión para determinar o host virtual axeitado a servir. Desa maneira necesítase un endereço IP dedicado para cada host virtual.
- Os **hosts virtuais baseados en nomes**, onde o servidor depende de que o cliente reporte o nome do host como parte das súas cabeceiras HTTP. Desta maneira varios host diferentes poden compartir o mesmo endereço IP. Para configurar host virtuais baseados en nomes só é necesario configurar o servidor DNS para que cada nome apunte ao endereço correcto, e logo configurar o servidor web Apache para que recoñeza os diferentes nomes.
- Funcionamento: Cando chega unha petición o servidor busca a directiva `<VirtualHost>` que mellor encaixe (a máis específica) baseado no endereço IP e no porto especificado na petición. Se hai máis de un virtual host que encaixe coa combinación de endereço IP e porto, Apache2 compara as directivas `ServerName` e `ServerAlias` co nome de servidor presente na petición. Se non se atopa ningunha directiva `ServerName` ou `ServerAlias` no conxunto de host virtuais que conteñen a combinación de endereço IP e número de porto que mellor encaixa, entón o primeiro host virtual en ser listado que encaixe será empregado.





1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

Hosts virtuais

Para que o nome que imos engadir se redirixe á IP real do servidor, temos que engadir esa regra no servidor DNS na rede, ou no seu defecto, se estamos traballando en local no arquivo hosts correspondente:

```
$ sudo mousepad /etc/hosts
```

127.0.0.1 www.exemplo.com

Para a continuación copiar o arquivo de configuración dese Host segundo o modelo que hai en sites-available:

```
/etc/apache2/sites-available$ sudo cp 000-default.conf exemplo.com.conf
```

Editamos o documento e o primeiro paso para crear un bloque `<VirtualHost>` para cada host diferente que se quere servir. Dentro de cada bloque `<VirtualHost>` necesítase polo menos unha directiva `ServerName` para designar que host virtual se serve, e tamén unha directiva `DocumentRoot` que indique en que lugar do sistema de ficheiros reside o contido para ese virtual host.



1.7 Instalación da plataforma PHP



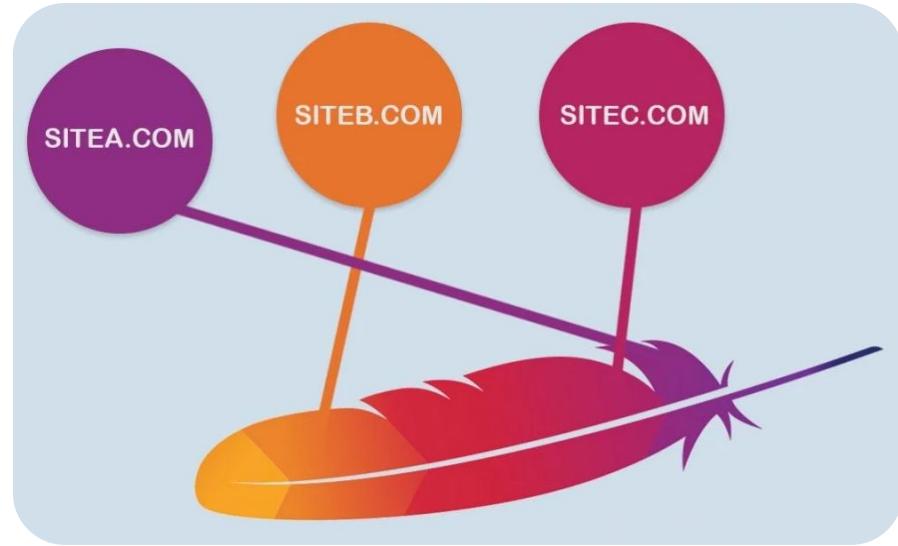
1.7.1 Instalación do servidor web Apache

Hosts virtuais

```
<VirtualHost *:80>
    ServerName www.exemplo.com
    ServerAlias exemplo.com
    DocumentRoot "/var/www/html/exemplo"
    <Directory "/">
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Só queda activar o novo sitio web coa ferramenta `a2ensite` e reiniciar Apache:

```
$ sudo a2ensite exemplo1.com.conf
$ sudo /etc/init.d/apache2 restart
```



Outras directivas de interese:

```
ErrorDocument 404 "/404.html"
```

```
DirectoryIndex "inicio.php"
```

```
ErrorLog "/var/www/html/exemplo.log"
```



1.7 Instalación da plataforma PHP



Compartido



Dedicado

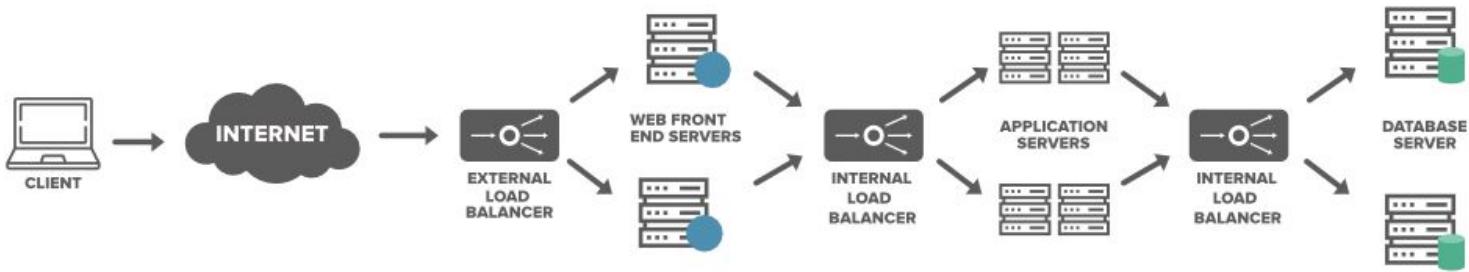
VPS



Na nube



1.7 Instalación da plataforma PHP





1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

Logs

- Para manexar de xeito efectivo un servidor web, é necesario ter información sobre a actividade e o rendemento do servidor, así coma dos problemas que poden ocorrer.
- O servidor HTTP Apache, fornece unha variedade de diferentes mecanismos para rexistrar todo o que ocorre no servidor, desde a petición inicial pasando polo mapeo da URL ata a resolución final da conexión incluíndo os errores que poden ocorrer no proceso. Adicionalmente módulos de terceiros, poden proporcionar capacidade de rexistro, ou inxectar entidades nos ficheiros de log, e tamén aplicacións coma programas CGI, scripts PHP ou outros poden mandar mensaxes ao log do servidor.
- O rexistro de log de errores do servidor, cuxo nome e localización se establece coa directiva `ErrorLog` é o ficheiro de log máis importante. Aquí é onde Apache manda información de diagnóstico e rexistra calquera erro que se atope no procesamento das peticións. É o primeiro lugar onde se debe mirar cando se produce un erro co inicio do servidor, ou coa operación do servidor, xa que poden ofrecer detalles do que está mal e como poder solucionalo.
- O log de errores, habitualmente escríbese nun ficheiro (tipicamente `error.log` en sistemas Unix/Linux). En Ubuntu/Debian o ficheiro máis típico é `/var/log/apache2/error.log`. É moi importante lembralo!



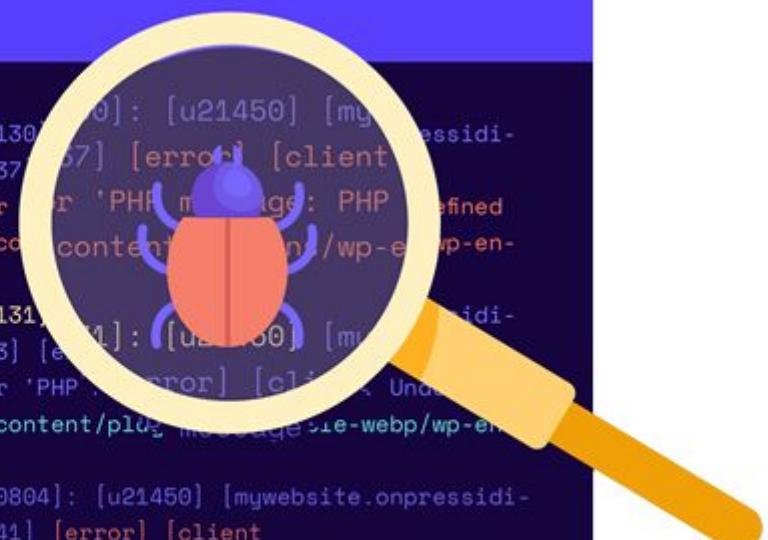
1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

Logs

● LOGS FILE



```
Oct 7 06:25:54 192.168.128.60 apache2[24130]: [u21450] [mywebsite.onpressidi-  
um.com] [cc99409988accfc5] [207.148.31.137:67] [error] [client  
192.168.171.173:47286] AH01071: Got error 'PHP message: PHP undefined  
index: HTTP_ACCEPT in /mywebsite-www/wp-content/plugins/pre-webp/wp-en-  
able-webp.php on line 26  
Oct 7 08:27:20 192.168.128.60 apache2[24131]: [u21450] [mywebsite.onpressidi-  
um.com] [d2aa451ca5461e86] [94.102.49.193:61] [error] [client  
192.168.171.173:60206] AH01071: Got error 'PHP message: PHP Notice: Undefined  
index: HTTP_ACCEPT in /mywebsite-www/wp-content/plugins/pre-webp/wp-en-  
able-webp.php on line 26  
Oct 7 08:48:52 192.168.175.156 apache2[10804]: [u21450] [mywebsite.onpressidi-  
um.com] [ae7440d29262f05e] [185.191.171.41] [error] [client  
192.168.134.96:15764] AH01071: Got error 'PHP message: PHP Notice: Array to  
string conversion in /mywebsite-www/wp-includes/class-wp-query.php on line  
4099  
Oct 7 09:24:25 192.168.175.156 apache2[10803]: [u21450] [mywebsite.onpressidi-
```



1.7 Instalación da plataforma PHP



1.7.1 Instalación do servidor web Apache

Control de acceso

Existen varias formas de protexer os cartafios de Apache. Como requisito previo, acostuman requerir que o módulo *auth_basic* estea cargado e habilitado en *mods-enabled*. Unha vez comprobado podemos crear un ficheiro **.htaccess** no directorio a protexer con este contido:

```
AuthName "Acceso Restrinxido a este recurso"  
AuthType Basic  
AuthUserFile /var/www/.htpasswd  
Require valid-user
```

Engadimos un contrasinal para o usuario/a de exemplo *admin* (a continuación solicitaran o seu contrasinal):

```
$ sudo htpasswd -c /var/www/.htpasswd admin
```

Ou editamos o arquivo como texto cunha ferramenta tipo: htaccesstools.com



1.7 Instalación da plataforma PHP

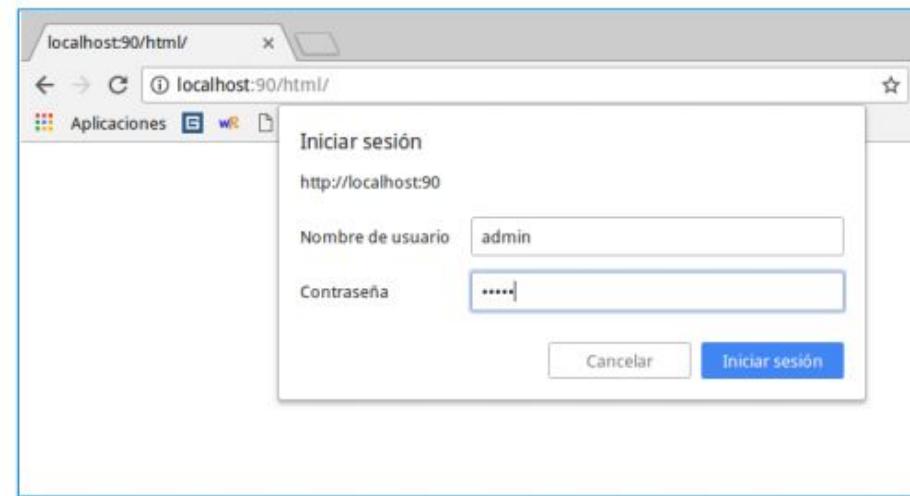


1.7.1 Instalación do servidor web Apache

Control de acceso

Editamos a configuración do cartafol en **/etc/apache2/apache2.conf** indicando na directiva **AllowOverride** que poden redefinir todas as directivas nos arquivos **.htaccess**:

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    #AllowOverride None
    Require all granted
</Directory>
```



Por último, cómpre **reiniciar Apache** e ao volver a acceder ao sitio:

```
$ sudo /etc/init.d/apache2 restart
```

Ao acceder ao recurso aparecerá unha ventá solicitando a autorización...



1.7 Instalación da plataforma PHP

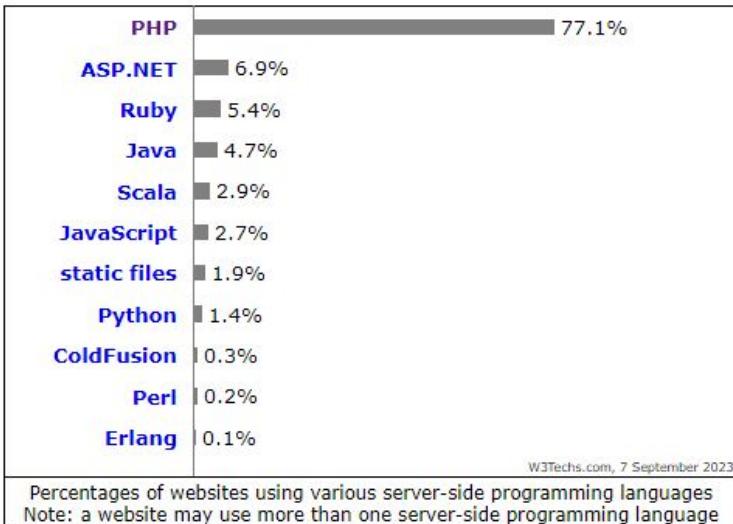
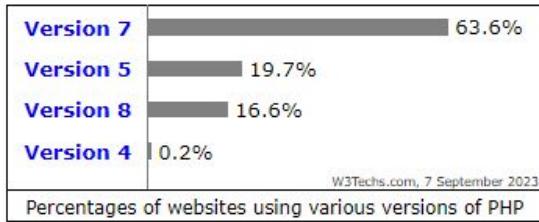


1.7.2 Instalación de PHP

- PHP era orixinalmente o acrónimo de (Pre-procesador de Hipertexto PHP). PHP é unha **linguaxe de programación web interpretada**, libre e moi utilizada para xerar aplicáns, en especial de software libre. Trátase dunha linguaxe de guíóns de propósito xeral que **emprega código embebido** dentro de linguaxe de marcas ou permite indicar a saída HTML para cada páxina. Actualmente á linguaxe de programación máis empregada no mundo con moita diferenza (77,1 %).
- A súa sintaxe está baseada na de C / C++, e polo tanto é moi similar á de Java.
- Instalación por liña de comandos en máquinas Ubuntu/Debian:

```
$ sudo apt-get update  
$ sudo apt-get install php
```

- ¿Que versión





1.7 Instalación da plataforma PHP



1.7.2 Instalación de PHP

- O paquete base de php collerá a versión estable máis recente ou actual subida aos repositorios (por exemplo a 8.1). Así coma unha serie de dependencias para instalar a funcionalidade básica.

<input checked="" type="checkbox"/> libapache2-mod-php8.1	8.1.2-1ubuntu2.14	8.1.2-1ubuntu2.14	Linguaxe de scripts do lado do servidor incorporada en HTML (módulo para Apache 2)
<input checked="" type="checkbox"/> libgtksourceview-4-common	4.8.3-1	4.8.3-1	Ficheiros comúns do trebello de realizdo da sintaxe de GTK+ server-side, HTML-embedded scripting language (default)
<input checked="" type="checkbox"/> php	2.8.1+92ubuntu1	2.8.1+92ubuntu1	Common files for PHP packages
<input checked="" type="checkbox"/> php-common	2.92ubuntu1	2.92ubuntu1	Linguaxe de scripts do lado do servidor incorporada en HTML (metapaquete)
<input checked="" type="checkbox"/> php8.1	8.1.2-1ubuntu2.14	8.1.2-1ubuntu2.14	intérprete para a liña de ordes para a linguaxe de scripts PHP
<input checked="" type="checkbox"/> php8.1-cli	8.1.2-1ubuntu2.14	8.1.2-1ubuntu2.14	documentation, examples and common module for PHP
<input checked="" type="checkbox"/> php8.1-common	8.1.2-1ubuntu2.14	8.1.2-1ubuntu2.14	Zend OpCache module for PHP
<input checked="" type="checkbox"/> php8.1-opcache	8.1.2-1ubuntu2.14	8.1.2-1ubuntu2.14	Módulo de readline para PHP
<input checked="" type="checkbox"/> php8.1-readline	8.1.2-1ubuntu2.14	8.1.2-1ubuntu2.14	Controlador de impresoras para as Color LaserJet 35xx/36xx da HP
<input checked="" type="checkbox"/> printer-driver-pxlqr	1.4+repack0-6build1	1.4+repack0-6build1	

- A maiores poden instalarse outros módulos máis específicos segundo se precisen. Por exemplo un módulo de integración de Apache con PHP ou de PHP con MySQL:

```
$ sudo apt-get install libapache2-mod-php
```

```
$ sudo apt-get install php-mysql
```

Nota: Neste exemplo os dous módulos indican “php” sen o número de versión o que implica que se buscarán os paquetes da versión instalada.



1.7 Instalación da plataforma PHP



1.7.2 Instalación de PHP

Configuración

O arquivo de configuración principal de php, chámase **php.ini** e estaría (por exemplo para a versión 8.1):

`/etc/php/8.1/apache2/php.ini`

Os parámetros más destacables a configurar dese arquivo son:

- **Display errors = On** → Amosar errores. Amosa os errores nas mesmas páxinas do sitio. Cando se está desenvolvendo e hai errores nos scripts, é máis doado atopalos se se amosan xa nas páxinas. Actívase só en local, non é recomendable en produción.
- **max_execution_time=30** → Tempo máximo en segundos, de execución dun script. Se deixaramos que un script se executara indefinidamente, podería tirar ese servidor.
- **post_max_size=8M** → Tamaño máximo de datos que se poden enviar ao servidor mediante o método POST.
- **upload_max_filesize = 8M** → Tamaño máximo de arquivo que se pode subir ao servidor.
- **extension=mysqli** → Activa o módulo correspondente, neste caso o módulo de integración coas bases de datos MySQL dende PHP.



1.7 Instalación da plataforma PHP



1.7.2 Instalación de PHP

Probando a execución de PHP

Coa configuración predeterminada, a carpeta raíz de onde o servidor web Apache le os documentos que vai a publicar será </var/www/html/>. Por tanto, para probar o correcto funcionamento da plataforma podes crear con calquera editor de texto (mousepad, gedit, nano) nese directorio un ficheiro “[phpinfo.php](#)” co seguinte contido:

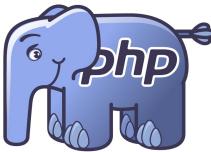
```
<?php  
    phpinfo();  
?>
```

O que fai é chamar a unha función da linguaxe PHP que xera por si mesma unha páxina web completa con información sobre a instalación de PHP. Se todo vai ben, ao abrir esa páxina cun navegador web (dende a propia máquina de Linux a URL sería <http://localhost/phpinfo.php>) deberíase ver algo como o seguinte:

PHP Version 8.1.2-1ubuntu2.14	
System	Linux xubu-VirtualBox 6.2.0-32-generic #32~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Aug 18 10:40:13 UTC 2023 x86_64
Build Date	Aug 18 2023 11:41:11
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902



1.7 Instalación da plataforma PHP



1.7.2 Instalación de PHP

Probando a execución de PHP por liña de comandos

O intérprete PHP-CLI permite executarse por liña de comandos de forma similar a Python e outras linguaxes, polo que poderíamos empregalo para facer scripts que realicen operacións na propia máquina, sen darlle un enfoque web.

```
!<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Proba</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
  <body>
    <?php echo "Boas, mundo!"; ?>
  </body>
</html>
```

```
xubu@xubu-VirtualBox:~/var/www/html$ php -version
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
xubu@xubu-VirtualBox:~/var/www/html$ php proba.php
!<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Proba</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
  <body>
    Boas, mundo!    </body>
</html>xubu@xubu-VirtualBox:~/var/www/html$ █
```



1.7 Instalación da plataforma PHP



1.7.2 Instalación de PHP

Así mesmo podemos crear archivos executables (para máquinas que teñan instalado PHP toxicamente).



```
#!/usr/bin/php -q
<?php
    echo "Boas!! \n";
?>
```

```
xubu@xubu-VirtualBox:/var/www/html$ chmod 755 outraproba.php
xubu@xubu-VirtualBox:/var/www/html$ ./outraproba.php
Boas!!
xubu@xubu-VirtualBox:/var/www/html$
```

Nota: A opción `-q` (`--quiet`) activa o modo silencioso, sen amosar advertencias.

```
xubu@xubu-VirtualBox:/var/www/html$ php -h
Usage: php [options] [-f] <file> [--] [args...]
php [options] -r <code> [--] [args...]
php [options] [-B <begin_code>] -R <code> [-E <end_code>] [--] [args...]
php [options] [-F <file>] -F <file> [-E <end_code>] [--] [args...]
php [options] -S <addr>:<port> [-t docroot] [router]
php [options] -- [args...]
php [options] -a

-a           Run as interactive shell (requires readline extension)
-c <path>|<file> Look for php.ini file in this directory
-n           No configuration (ini) files will be used
-d foo[=bar] Define INI entry foo with value 'bar'
-e           Generate extended information for debugger/profiler
-f <file>   Parse and execute <file>.
-h           This help
-i           PHP information
-l           Syntax check only (lint)
-m           Show compiled in modules
-r <code>   Run PHP <code> without using script tags <?..?>
-B <begin_code> Run PHP <begin_code> before processing input lines
-R <code>   Run PHP <code> for every input line
-F <file>   Parse and execute <file> for every input line
-E <end_code> Run PHP <end_code> after processing all input lines
-H           Hide any passed arguments from external tools.
-S <addr>:<port> Run with built-in web server.
-t <docroot> Specify document root <docroot> for built-in web server.
-s           Output HTML syntax highlighted source.
-v           Version number
-w           Output source with stripped comments and whitespace.
-z <file>   Load Zend extension <file>.

args...      Arguments passed to script. Use -- args when first argument
            starts with - or script is read from stdin

--ini        Show configuration file names

--rf <name>  Show information about function <name>.
--rc <name>  Show information about class <name>.
--re <name>  Show information about extension <name>.
--rz <name>  Show information about Zend extension <name>.
--ri <name>  Show configuration for extension <name>.
```



1.7 Instalación da plataforma PHP



1.7.2 Instalación de PHP

Permisos

O usuario propietario do arquivo debería ser por seguridade e sobre todo nun servidor de producción tería que ser o usuario propio do Apache: [www-data](#). Para facilitar a edición podemos agregar o usuario do equipo ao grupo de Apache e permitir que os usuarios do grupo poidan escribir alí:

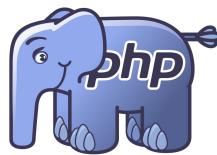
```
$ sudo chown -R www-data:www-data /var/www/html  
$ sudo usermod -a -G www-data [usuario]  
$ sudo chmod -R 777 /var/www/html
```

Octal	Decimal	Permission	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx

Se o usuario indicado é o da máquina (por exemplo “xubu”) dende o terminal poderemos abrir e modificar arquivos na ruta pública do servidor con calquera editor de texto ou código. Por suposto esta auténtica barbaridade de permisos só se poden asignar deste xeito en local. En producción o usuario da máquina nunca debería estar no grupo do Apache, nin moito menos ter os permisos totais de 777 (RWX / RWX / RWX) para o usuario, o grupo e outros. En producción deberían asignarse os permisos 644 (RW / R / R) para arquivos php e 600 (RW / - / -) para arquivos de configuración.



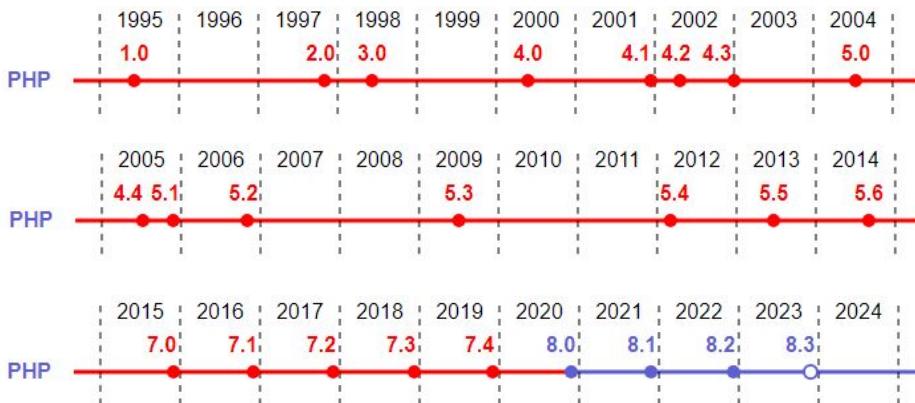
1.7 Instalación da plataforma PHP



1.7.2 Instalación de PHP

Evolución de PHP

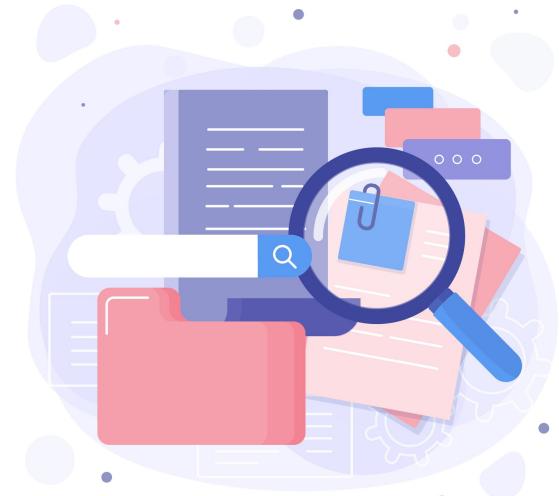
- Foi creado por Rasmus Lerdorf en xuño de 1995.
- PHP4 → Engádese as sesións.
- PHP5 → Engádese funcionalidades de POO.
- PHP6 → Fracaso no soporte para Unicode (descartouse).
- PHP7 → Mellora no rendemento.
- PHP8 → Aínda mellor rendemento.
 - Compilador JIT en tempo de execución.
 - Desactivado por defecto.
 - Pasaría a ser unha linguaxe intermedia.





Documentación complementaria

- [Documentación oficial do servidor HTTP Apache.](#)
- [XAMPP: Contorno de programación instalable.](#)
- [Apache: Contido dinámico con CGI/Perl.](#)
- [Documentación de Hosts virtuais en Apache.](#)
- [Autenticación e autorización en Apache.](#)
- [Lista completa das directivas para PHP.](#)
- [PHP Command Line Interface \(CLI\).](#)





Fin!

¿Algunha pregunta?

Lembra que tamén podes preguntar:

- ◉ Nos foros da aula virtual.
- ◉ Pola mensaxería da aula virtual.