

UNIDAD DIDÁCTICA 1:

CONFIGURACIÓN DEL ENTORNO DE TRABAJO

Módulo profesional:

Desarrollo Web en Entorno Servidor

Índice

RESUMEN INTRODUCTORIO.....	3
INTRODUCCIÓN	3
CASO INTRODUCTORIO	3
1. EL ENTORNO DE TRABAJO.....	4
2. CLIENTE	5
2.1. Navegador Web.....	6
2.2. Instalando Google Chrome.....	7
2.3. IDE de desarrollo.....	8
2.4. Instalando Atom.io	8
2.5. Control de versiones git.....	9
2.6. Instalando git	10
2.7. Cuenta en github.....	11
3. SERVIDOR	12
3.1. Tipos de instalaciones	13
3.2. Sistemas locales.....	15
3.3. Instalando Xampp	16
3.4. Comprobando la instalación de Xampp	17
3.5. Instalando LAMP.....	18
3.6. Instalando MAMP.....	19
3.7. Sistemas en la nube	19
RESUMEN FINAL	21

RESUMEN INTRODUCTORIO

En esta unidad prepararemos el entorno de trabajo necesario para poder desarrollar nuestras aplicaciones dentro del mundo del servidor. La parte de servidor necesita de una infraestructura, sistemas y herramientas para poder comenzar a desarrollar, por ese motivo es necesario conocer tanto la arquitectura que ya la iniciamos en la Unidad Didáctica anterior, como las tecnologías propias del desarrollo.

INTRODUCCIÓN

El desarrollo en la parte de servidor ha evolucionado tanto y tan rápido como lo han hecho los sistemas y las tecnologías de las cuales depende.

Mientras no hace mucho unos 15 años, sólo podíamos contar con 3 o 4 arquitecturas de desarrollo en backend potentes y la mitad eran propietarias, ahora mismo las alternativas que tenemos muy muy profesionales son muy amplias y variadas. Es el caso de Node o de Django.

Nuestros esfuerzos se centrarán en instalar y configurar un entorno para desarrollar sobre PHP, y para ello configuraremos e instalaremos dos de los servidores más utilizados para estos menesteres como son Apache y MySQL.

Por otro lado el interfaz de desarrollo IDE, también ha evolucionado de la misma manera, encontrándonos con múltiples opciones para el desarrollo que dependerán mucho más del gusto del programador que de las capacidades del IDE que son muchas.

CASO INTRODUCTORIO

Una pequeña empresa quiere desarrollar una Intranet, para ello el responsable de tecnología y desarrollo de la misma necesita instalar en el servidor de la misma que tiene un sistema operativo Windows, los servidores necesarios para poder desarrollar dicha Intranet.

Cuando finalices esta Unidad serás capaz de tener ese sistema preparado para poder desarrollar dicha Intranet.

1. EL ENTORNO DE TRABAJO

En la primera unidad hemos realizado una introducción al entorno de funcionamiento de las aplicaciones dentro del mundo web. Una de las imágenes que mejor resume la forma de trabajo de nuestras aplicaciones es la que podemos ver a continuación:



Imagen 1: Lenguajes clientes-servidor

Fuente de la imagen: propia

En este diagrama/imagen podemos ver claramente los dos entornos en los que vamos a impactar al realizar programación orientada a servidor:

- Parte de servidor, en el que se ejecutará nuestro código. En esta parte necesitaremos de varios softwares que se ejecutan en modo servidor (ver definición en Glosario) para poder realizar todas las tareas necesarias para atender una petición de un cliente y devolverle una respuesta a dicha petición.
- Parte en cliente, en este caso necesitaremos tanto un navegador como un entorno de desarrollo que denominaremos IDE.

Hoy en día esta, donde los proyectos han ido creciendo y la complejidad de los mismos también no sólo necesitaremos tener solucionado estos dos ámbitos, la del servidor y la de cliente, sino que necesitaremos otras herramientas para la realización de nuestros proyectos, individualmente y por supuesto en un equipo de trabajo:

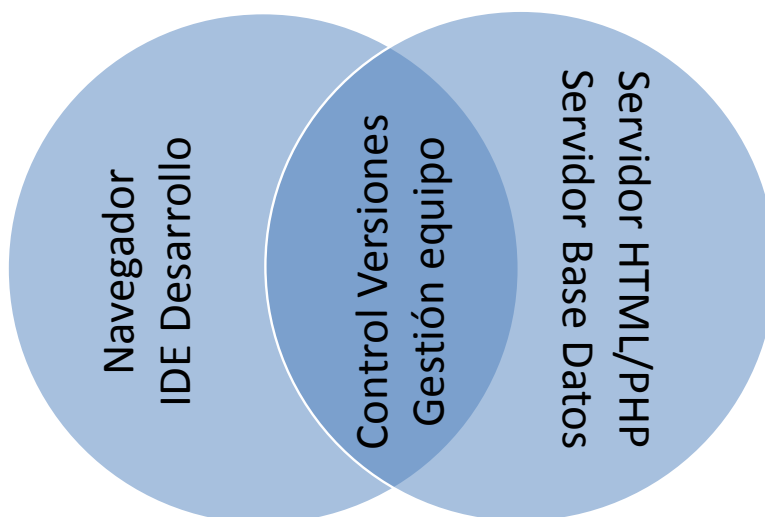


Imagen 2: Herramientas necesarias en el desarrollo

Fuente de la imagen: propia

Como descubrimos en el diagrama tenemos una serie de herramientas que nos van a servir en ambos mundos, tanto en la parte de cliente como en la de servidor:

- Control de versiones, herramientas que nos permitirán no solo mantener nuestro código salvado y guardado, sino que además nos permitirá realizar un seguimiento de las diferentes versiones, actualizaciones y planificar y llevar a cabo ramas de trabajo.
- Gestión de equipos, la gestión del trabajo, planificación de las tareas, gestión de los recursos y gestión del tiempo son herramientas muy muy necesarias hoy en día para llevar a cabo nuestra tarea de desarrollo.

2.CLIENTE

Comenzamos preparando la parte de cliente, ya que es la más sencilla. En este caso y en un principio sólo necesitaremos 3 softwares para poder comenzar a desarrollar nuestro código:

- Navegador web
- IDE de desarrollo

- Control de versiones git

2.1. Navegador Web

Continuamos preparando la parte de navegador. Mientras que en el desarrollo para servidor, el navegador nos realiza las funciones de comprobación final, en la parte de cliente, el navegador es pieza fundamental y además deberemos tener varios navegadores para poder comprobar el correcto funcionamiento en ellos.



RECUERDA...

Un navegador web se define como

- Componente software que se utiliza en el cliente y que permite acceder al contenido ofrecido por los servidores de Internet sin la necesidad de que el usuario instale un nuevo programa.

En este módulo, Desarrollo de Aplicaciones en Servidor, el Navegador es un elemento final, una herramienta en la que visualizar el resultado de nuestro código programado. Es necesario e imprescindible, pero no va a ser el objetivo como lo es en el desarrollo de cliente.

En este punto viene la pregunta clave, ¿qué navegador debo tener instalado? La respuesta no es única, ya que por un lado debemos tener en cuenta nuestras necesidades como desarrollador, y por otro lado nuestras necesidades como "testers" ya que necesitamos comprobar el buen funcionamiento de nuestras aplicaciones en múltiples dispositivos.

Veamos unas estadísticas que nos permiten tener una visión general de los posibles navegadores y su uso:

2017	<u>Chrome</u>	<u>IE/Edge</u>	<u>Firefox</u>	<u>Safari</u>	<u>Opera</u>
March	75.1 %	4.8 %	14.1 %	3.6 %	1.0 %
February	74.1 %	4.8 %	15.0 %	3.6 %	1.0 %
January	73.7 %	4.9 %	15.4 %	3.6 %	1.0 %

Imagen 3: Estadísticas de los navegadores

Fuente de la imagen: <https://www.w3schools.com/browsers/default.asp>

Como observamos el navegador más utilizado actualmente es Chrome de la empresa Google, tanto por su rapidez y configurabilidad, como por su uso como desarrollador.

En segundo lugar, dependerá del sistema operativo en el que nos encontremos, siendo Firefox muy utilizado en sistemas Linux y Windows, y Safari para entornos OS.

2.2. Instalando Google Chrome

La instalación de Google Chrome es muy muy sencilla y por lo tanto no requiere de grandes explicaciones. Como ocurrirá a lo largo del módulo, utilizaremos 3 vídeos como base para realizar dicha instalación:

	<p>En el vídeo que encontrarás en el siguiente enlace podrás visualizar un ejemplo explicativo de instalación de Chrome en Windows 10</p> <p> https://youtu.be/4x8oRVMUQXg</p>
	<p>En el vídeo que encontrarás en el siguiente enlace podrás visualizar un ejemplo explicativo de instalación de Chrome en MAC</p> <p> https://youtu.be/y1wM7tZEJUs</p>
	<p>En el vídeo que encontrarás en el siguiente enlace podrás visualizar un ejemplo explicativo de instalación de Chrome en Ubuntu</p> <p> https://youtu.be/pt4TujfVf9E</p>

2.3. IDE de desarrollo

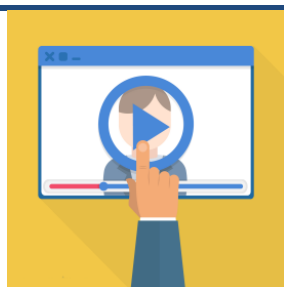
El mundo de los editores o IDE ha evolucionado de una forma tan rápida y evolucionada como el resto de tecnología relacionada con el desarrollo. Puede ser que todo comenzara con Sublime hace un par de años: ligero, flexible y hackeable. Otro editor que también está creciendo en este “mundillo” es phpStorm, muy completo pero excesivamente pesado para mi gusto. Por último Atom.io de GitHub, que al igual Sublime es ligero, flexible y hackeable.



El blog www.genbetadev.com es un referente para todos aquellos desarrolladores que quieran estar al tanto de las novedades, eventos y tecnología. Un blog muy profesional con artículos muy bien documentados como el siguiente sobre los IDEs: <https://www.genbetadev.com/herramientas/las-herramientas-del-programador-y-tu-cual-usas>

2.4. Instalando Atom.io

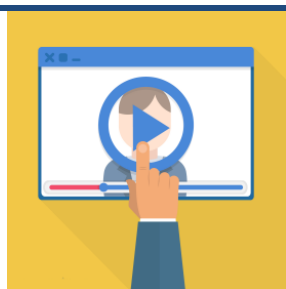
La instalación de Atom.io es muy sencilla y por lo tanto no requiere de grandes explicaciones, pero lo mejor es verlo a través de un vídeo:



En el vídeo que encontrarás en el siguiente enlace podrás visualizar un ejemplo explicativo de instalación de Atom.io

 <https://youtu.be/uj5hXILyIAw>

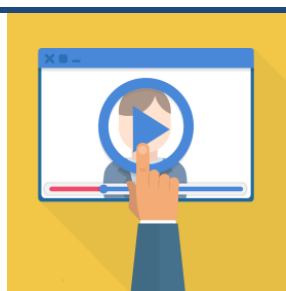
Son muchos los aspectos que hacen muy muy interesante Atom.io, entre ellos algo tan sencillo como utilizar plugins y aplicaciones, un ejemplo lo vemos en el siguiente vídeo:



En el vídeo que encontrarás en el siguiente enlace podrás visualizar un ejemplo explicativo para la instalación del plugin emmet:

 <https://youtu.be/uj5hXILyIAw>

Por último, Atom permite instalar y configurar diferentes skins o temas que permiten modificar la forma en la que se presenta Atom:



En el vídeo que encontrarás en el siguiente enlace podrás visualizar un ejemplo explicativo para la instalación del plugin emmet:

 <https://youtu.be/uj5hXILyIAw>


Tarea 1
Título: INSTALACIÓN DE PLUGIN SOBRE ATOM.IO

2.5. Control de versiones git


Hoy en día la programación no se concibe de una forma aislada e individual. Hoy en día con la cantidad de mecanismos y herramientas que disponemos, compartir código, dividir las tareas del código, trabajar de forma distribuida y en general mantener dicho código, sus versiones y pruebas es algo esencial en el día a día de cualquier desarrollador. Para ayudarnos en esta tarea de trabajar en equipo tenemos los sistemas de control de versiones y en concreto Git, un sistema de control de versiones distribuido que utilizan actualmente casi todas las empresas a nivel profesional.

En este punto de nuestro desarrollo no es imprescindible comprender al 100% el funcionamiento de git, todos sus comandos y el ciclo esencial de

trabajo, pero sí es esencial tener preparada la herramienta para comenzar a utilizarla. Sirva como introducción el siguiente vídeo improductivo:




En el vídeo que encontrarás en el siguiente enlace podrás visualizar una introducción sobre git:


 https://youtu.be/TuUZpb_ZjnY

2.6. Instalando git

Instalar git consiste en descargar el paquete actualizado que el proveedor nos facilita en la web, <https://git-scm.com/downloads>, y seguir las instrucciones dependiendo del sistema operativo. Sirva como base el sistema operativo Windows para dicha instalación:



En el vídeo que encontrarás en el siguiente enlace podrás visualizar cómo realizar la instalación sobre Windows:

 <https://youtu.be/KjKouicjfXc>

Para el caso de Linux o de Mac OS, seguiremos las instrucciones que nos marca el proveedor:

Sistema Operativo	Pasos
Linux	Debian/Ubuntu
	\$ apt-get install git
	Fedora
	\$ yum install git (up to Fedora 21)
	\$ dnf install git (Fedora 22 and later)
	Gentoo
	\$ emerge --ask --verbose dev-vcs/git

	FreeBSD \$ pkg install git Otros https://git-scm.com/download/linux
MacOs	https://git-scm.com/download/mac

2.7. Cuenta en github

Antes, cuando hablábamos de Atom.io, hemos nombrado GitHub, una compañía que se ha hecho muy famosa por poner al alcance de todos los programadores la posibilidad de tener un entorno de control de versiones colaborativo en la nube y totalmente gratuito. Otra de las opciones que podemos utilizar es BitBucket.

Sea cual sea la opción que utilicemos para almacenar nuestro código en la nube debemos saber que ambas soluciones utilizan git, para su interacción, y que son un mero almacenaje en la nube.

Crear una cuenta gratuita en GitHub es tan sencillo como ir a la página de <https://github.com/> y seguir los pasos de registros comunes a cientos de aplicaciones en Internet.

3.SERVIDOR

Es importante recordar de la unidad introductoria el funcionamiento de una aplicación en Internet, los pasos que siguen una petición, y los elementos que intervienen.

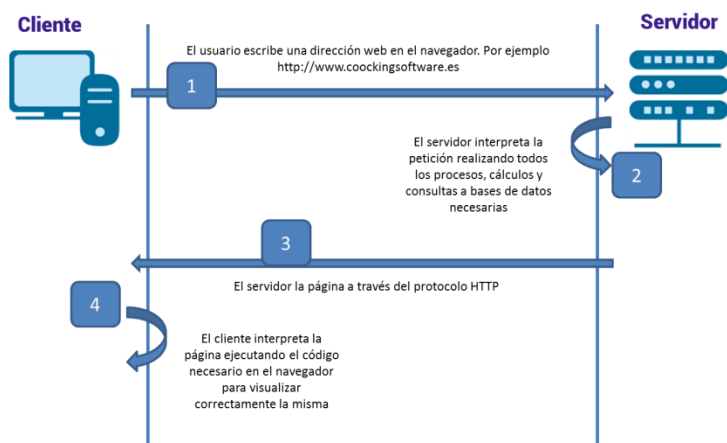


Imagen 4: Petición de página web

Fuente de la imagen: propia

A partir de este diagrama nos vamos a centrar en el punto 2 para dar paso a la instalación de todos los elementos necesarios para poder realizar la instalación.

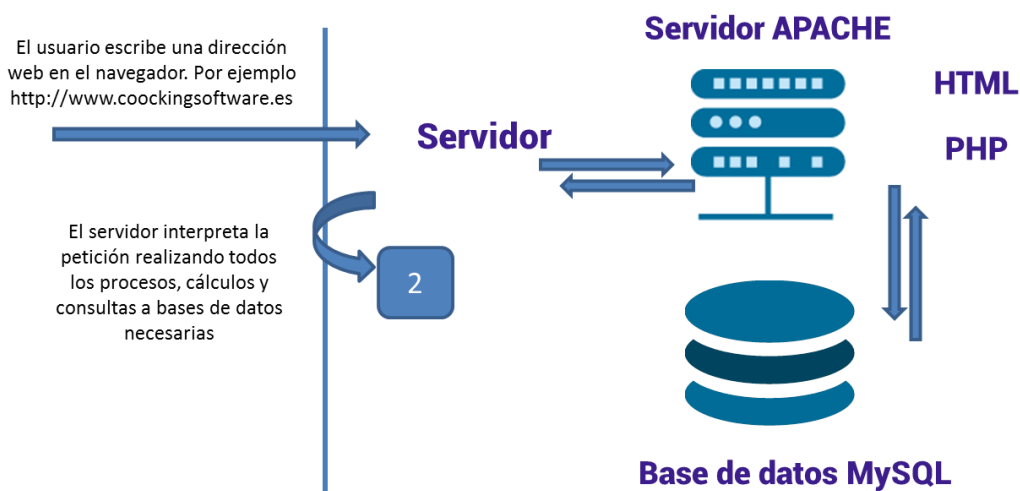


Imagen 5: Petición de página web. Parte Servidor

Fuente de la imagen: propia

Como podemos observar en la anterior imagen, una vez que la petición realizada por el cliente alcanza el servidor, este realiza una serie de acciones, pasos, que dependiendo de la complejidad de nuestra aplicación, dependiendo de la complejidad de la petición y de las tecnologías implicadas e instaladas, se necesitará de una serie de servidores que tendremos que tener instalados en nuestro entorno.

En el caso que mostramos, comenzamos con un esquema de arquitectura muy sencillo:

1. Servidor Apache que permite analizar las peticiones HTML y ejecutar el código PHP
2. Servidor de base de datos (en nuestro caso MySQL), que permite almacenar información persistente en un sistema servidor de base de datos estructurada.

3.1. Tipos de instalaciones

Este módulo no se explicará en detalle la arquitectura y configuración de los servidores necesarios para poder programar mediante el lenguaje PHP y por lo tanto tener aplicaciones de Internet.

En este módulo nos centraremos en los sistemas, paquetes y elementos que de una forma rápida y sencilla nos preparen un entorno de desarrollo para poder desarrollar.

Los tipos de instalaciones que nos podemos encontrar son:

- Sistemas en la nube, a su vez nos podemos encontrar diferentes modelos:
 - Platform as a Service (PaaS), nos permite tener una serie de recursos, normalmente virtualizados, como espacio en disco duro, memoria RAM, procesador, ... A partir de estos recursos la instalación requiere de un Sistema Operativo y el resto de elementos que vayamos a utilizar para la instalar tanto el servidor Apache como el MySQL.

- Software as a Service (SaaS), en este no solo se nos ofrece una serie de recursos hardware en la nube sino una serie de software en la nube, en nuestro caso los dos servidores necesarios. Este sistema es super interesante para nosotros como desarrolladores ya que nos despreocupamos de la instalación y mantenimiento del sistema y nos centramos en lo que realmente nos importa, el desarrollo software.
- Instalaciones locales mediante software empaquetado como XAMPP, WAMPServer, LAMP, MAMP, ... En este caso, y mediante una instalación lo más desatendida posible, un software nos guía a la instalación de todos los servidores y software necesarios para pasar al desarrollo en local.

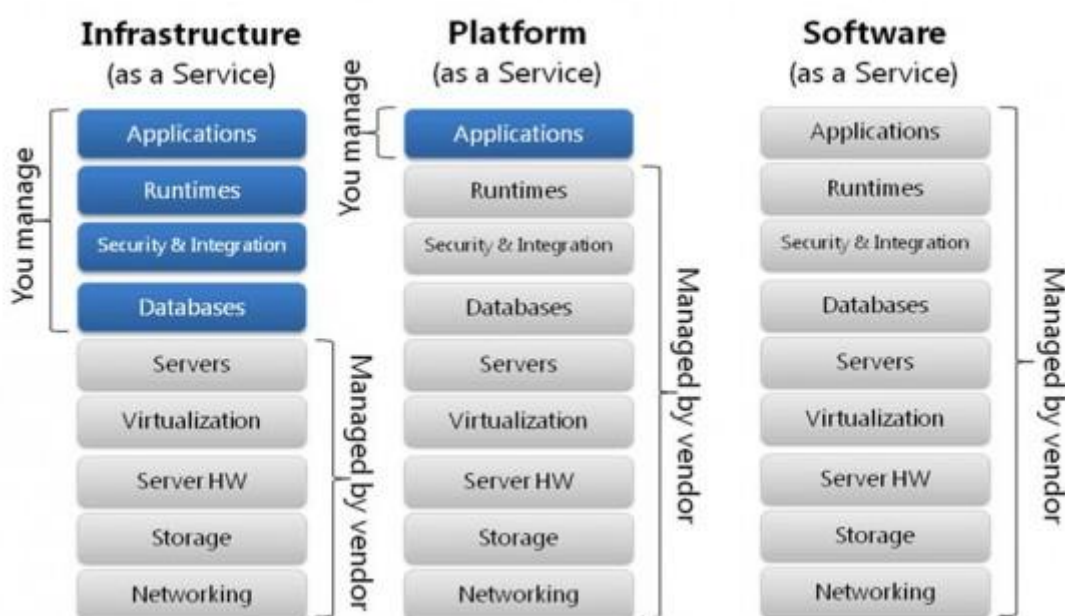


Imagen 6: SaaS frente a PaaS

Fuente de la imagen: <https://www.smartfile.com/blog/wp-content/uploads/2013/06/UnderstandingCloudComputing-e12911886773681.jpg>



En el siguiente enlace encontrarás una comparativa entre SaaS y PaaS:

<https://www.genbetadev.com/programacion-en-la-nube/entendiendo-la-nube-el-significado-de-saas-paas-y-iaas>

Nosotros nos centraremos en dos de los sistemas antes vistos, ya que los sistemas PaaS se exceden de nuestros objetivos:

- Configuración y uso de un sistema **SaaS**
- Instalación, configuración y uso de un **sistema local**

3.2. Sistemas locales

Empecemos con la arquitectura que vamos a utilizar para comenzar a trabajar y programar. Esta arquitectura tiene sus pros y sus contras:

Ventajas	Desventajas
<ul style="list-style-type: none"> • Fácil de instalar y configurar. • Total control del sistema. • Fácil de mantener y realizar backups. 	<ul style="list-style-type: none"> • Complicado el trabajo colaborativo • Sistema de debug, pero no se puede utilizar como sistema en producción.

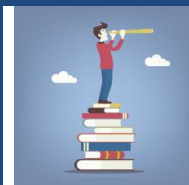
Aunque poco a poco los entornos de trabajo para desarrollo web a nivel local ya no son tan utilizados como antiguamente lo eran, no deja de ser interesante conocerlos y tenerlos configurados para utilizarlos a nivel de prueba o incluso a nivel de desarrollo siempre y cuando los configuremos adecuadamente para un desarrollo remoto y en producción.

Existen muchos entornos, y además son dependientes del sistema operativo con el que estemos trabajando. En nuestro caso haremos la configuración y los ejemplos con ejemplos preparados para ser instalados con sistemas Windows. Para sistemas Linux o Mac existen sus homólogos cuya configuración es muy similar y veremos más adelante.

Libro recomendado, PHP7 Programming cookbook

(Bierer, 2016)

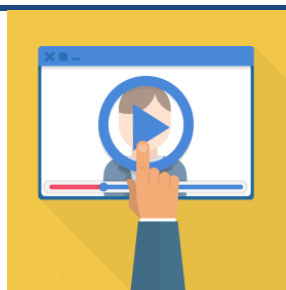
Libro muy completo sobre programación PHP donde en el capítulo 1 se habla sobre el servidor propio de PHP.



3.3. Instalando Xampp

Dos de los entornos más conocidos son xampp y wamp. Instalaremos y arrancaremos Xampp de una forma sencilla.

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB (un sistema de gestión de bases de datos derivado de MySQL con licencia GPL) PHP y Perl.



En el vídeo que encontrarás en el siguiente enlace podrás visualizar cómo realizar la instalación de XAMPP

 <https://youtu.be/oqNcknWcteE>



Si prefieres ver el sistema de instalación de forma textual:

<https://devcode.la/tutoriales/instalar-xampp-en-windows-7/>

Una vez instalado, tal y como nos muestra toda la documentación facilitada tendremos los siguientes directorios y recursos:

- Directorio con todos los software, servicios y recursos por debajo de la carpeta <directorio de instalación>/xampp (Normalmente esta carpeta está por debajo de c:)
- Xampp control Panel desde el cual podremos:

- Iniciar o parar los servicios Apache Web Server, esencial para poder ejecutar nuestro código PHP
- Iniciar o para el servicio MySQL, esencial para utilizar la base de datos
- Configurar o revisar los logs de Apache Web Server y MySQL database

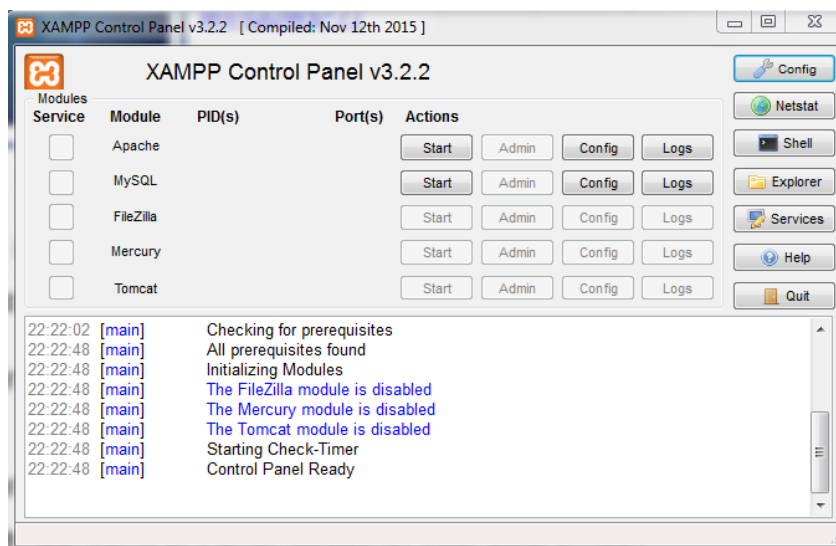


Imagen 7: XAMPP Control Panel

Fuente de la imagen: Propia

3.4. Comprobando la instalación de Xampp

Una vez que tenemos instalado Xampp podemos comprobar de una forma rápida y sencilla su correcta instalación realizando los dos pasos siguientes:

1. Abrir el servidor Apache HTML
2. Escribir en el navegador <http://localhost>. Si aparece el mensaje que podéis ver en la imagen 9, entonces la instalación ha sido correcta.

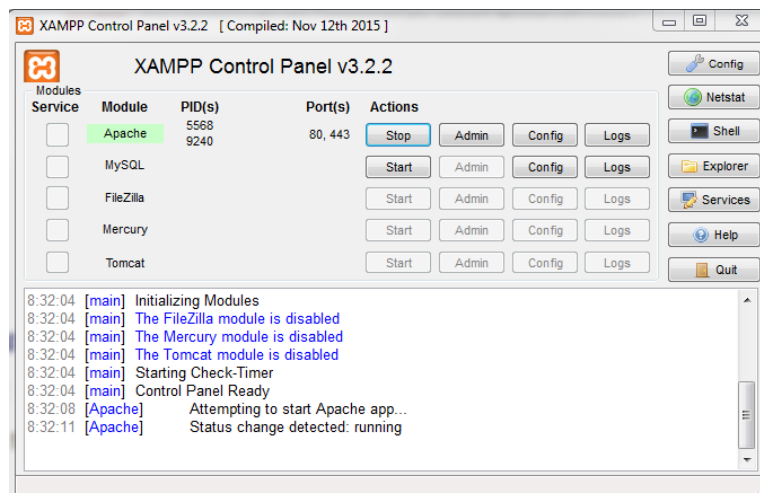


Imagen 8: Abrir Apache HTML

Fuente de la imagen: Propia



Imagen 9: Welcome XAMPP

Fuente de la imagen: Propia

3.5. Instalando LAMP

LAMP es el acrónimo de Linux, Apache, Mysql, PHP, y el objetivo al igual que ocurría con XAMPP, es tener instaladas de una forma sencilla y rápida todas esos servidores y tecnologías para comenzar a desarrollar.

El método sencillo y rápido lo podríamos realizar con dos comandos:

- `sudo apt-get update`
- `sudo apt-get install lamp-server^`

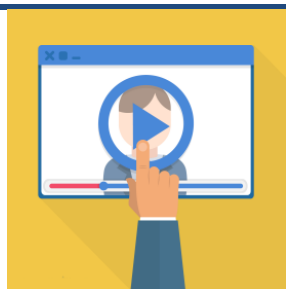


La forma completa y totalmente controlada la puedes ver en el siguiente manual

- <https://www.digitalocean.com/community/tutorials/como-instalar-linux-apache-mysql-php-lamp-en-ubuntu-16-04-es>

3.6. Instalando MAMP

Encontraréis mucha información y documentación al respecto. Os dejo una vídeo que os dará el paso a paso de la instalación de MAMP:



En el vídeo que encontrarás en el siguiente enlace podrás visualizar cómo realizar la instalación de MAMP

 <https://youtu.be/LjNKHGFgAXU>

O bien esta otra alternativa en castellano

 <https://youtu.be/iZA5sbXri8>

Caso práctico 1

Título: Tendencias Node.js

Foro 1

Título: Node contra PHP

3.7. Sistemas en la nube

En este apartado vamos a explicar el SaaS de Cloud9, aunque encontraremos otros como CodeEnvoy, IceCoders u otros.

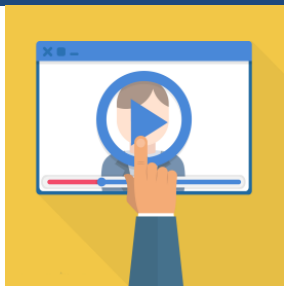


En el siguiente enlace encontrarás una entrada sobre alternativas a editores en la nube:

<https://www.genbeta.com/web/los-7-mejores-editores-de-codigo-online-para-programar>

Al ser desarrollos de terceros en la nube, adelanto que desde se escribe esta documentación hasta que vosotros podáis utilizarlo seguramente puede haber cambiado la instalación o el formato de las pantallas. En este procedimiento lo más importante será testar o comprender un sistema totalmente en la nube como este.

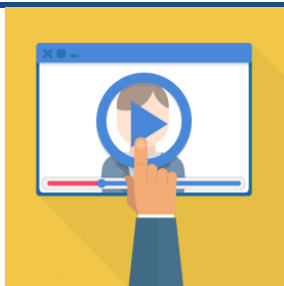
El primer apartado y punto será el registro y vista general del entorno, que podremos seguir las instrucciones en el siguiente vídeo:



En el vídeo que encontrarás en el siguiente enlace podrás visualizar cómo realizar el registro y una vista general del entorno

 <https://youtu.be/gpXlm0CLVt8>

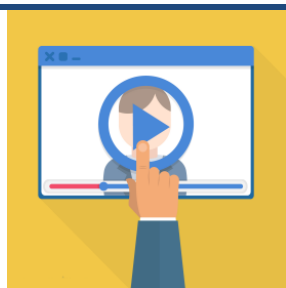
Una vez que hemos registrado una cuenta, el siguiente paso sería crear una primera máquina para comenzar a desarrollar. Crear una una máquina o workspace significará elegir sus características generales de hardware y el sistema final de desarrollo:



En el vídeo que encontrarás en el siguiente enlace podrás visualizar cómo crear una nueva máquina

 <https://youtu.be/xmXxzo1COOg>

Una de las cuestiones más importantes, y es el caso de Cloud 9, es el trabajo colaborativo, en el que una máquina creada o workspace puede ser compartida con otros desarrolladores con el objeto de poder analizar código, resolver problemáticas o simplemente trabajar en equipo.



En el vídeo que encontrarás en el siguiente enlace podrás visualizar cómo realizar trabajo colaborativo entre usuarios



<https://youtu.be/7-J1PK6yxLs>

RESUMEN FINAL

Para poder realizar un desarrollo sobre PHP, vamos a necesitar un entorno que tenga en cuenta el sistema de servidor y el sistema de cliente.

Parte de servidor, en el que se ejecutará nuestro código. En esta parte necesitaremos de varios softwares que se ejecutan en modo servidor (ver definición en Glosario) para poder realizar todas las tareas necesarias para atender una petición de un cliente y devolverle una respuesta a dicha petición.

Parte en cliente, en este caso necesitaremos tanto un navegador como un entorno de desarrollo que denominaremos IDE.