

## **UNIDAD DIDÁCTICA 2:**

### **INTRODUCCIÓN AL LENGUAJE PHP.**

### **INSERCIÓN CON HTML**

**Módulo profesional:**

**Desarrollo Web en Entorno Servidor**

## Índice

RESUMEN INTRODUCTORIO.....	3
INTRODUCCIÓN .....	3
CASO INTRODUCTORIO .....	3
1. CÓMO FUNCIONA EL HTML.....	4
1.1 Definición de HTML.....	4
1.2 Cómo funciona .....	4
1.3 Nuestro primer HTML en local .....	6
1.4 Nuestro primer HTML en la nube .....	8
1.5 Algo de sintaxis HTML.....	8
2. NUESTRO PRIMER PHP.....	10
2.1 Primer PHP.....	10
3. PHP Y GIT .....	12
3.1 De local hacia la nube.....	12
RESUMEN FINAL .....	13

## RESUMEN INTRODUCTORIO

En esta unidad crearemos nuestro primer PHP, pero para poder comenzar a utilizar y trabajar con ficheros php necesitamos conocer aunque sea de forma general el funcionamiento y uso del lenguaje HTML. A partir de este conocimiento básico podremos incluir nuestro código php dentro de éste.

## INTRODUCCIÓN

El lenguaje PHP es un lenguaje de servidor tal y como hemos explicado, interpretado y ejecutado por un servidor en nuestro caso Apache.

Además el lenguaje PHP se encuentra embebido dentro del lenguaje HTML, base y estándar para la representación de páginas Web en cualquier Navegador.

Conociendo cómo se escribe de forma básica un fichero HTML podremos comenzar a escribir nuestro primer código PHP que podremos ejecutar en nuestro servidor una vez comprendido también el funcionamiento de una petición a través de cualquier navegador o cliente.

Por último es importante comenzar a trabajar con git, herramienta fantástica para el mantenimiento de código en proyectos.

## CASO INTRODUCTORIO

Queremos probar nuestro servidor Apache recién instalado en la empresa a través de un fichero *hola.php* que ejecutaremos en un navegador Chrome.

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

## 1. CÓMO FUNCIONA EL HTML

El primer paso previo antes de comenzar a programar con PHP es entender cómo funciona el entorno de trabajo y también saber qué está pasando cuando nosotros solicitamos (es el “palabro” técnico para decir que ponemos una dirección en nuestro navegador) una web.

### 1.1 Definición de HTML

Extraído de nuestro punto de referencia inicial de conocimiento, WikiPedia, HTML, siglas en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de **páginas web**.



#### *Para saber más sobre Navegador*

<https://es.wikipedia.org/wiki/HTML>

Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

Como ocurrirá a lo largo de todo el curso, mi recomendación es siempre acudir a la fuente origen, al proveedor de la información, y este es el consorcio WWW, <https://www.w3.org/>.

### 1.2 Cómo funciona

Una imagen tan sencilla como la que veis a continuación nos puede ayudar perfectamente para comenzar a comprender cómo funciona esto del

desarrollo web y cómo cuáles son los elementos que están interviniendo en una comunicación.

Cuando nosotros estamos solicitando una, web cuando nosotros estamos introduciendo una dirección en nuestro navegador como pueda ser Chrome.

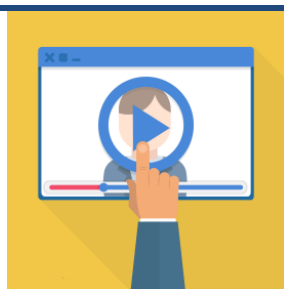


**Imagen 1: Solicitando pagina web**

Fuente de la imagen: propia

El ejemplo que os planteo donde imaginaos que Hulk introduce la dirección de una página que se llama <http://www.malos.com>, llegará a un servidor en nuestro caso ya hemos visto que vamos a utilizar Apache Server, y este servidor devolverá el contenido, devolverá el recurso que se haya solicitado. Por lo tanto los pasos van a ser los siguientes:

1. Solicitud del recursos desde un navegador, sea desktop o móvil.
2. La red redirige la petición hasta el punto final donde se encuentre el recurso
3. El servidor analiza la petición a través de la URL y devuelve el recurso.
4. El mismo navegador que solicitó dicho recurso visualiza la contestación.



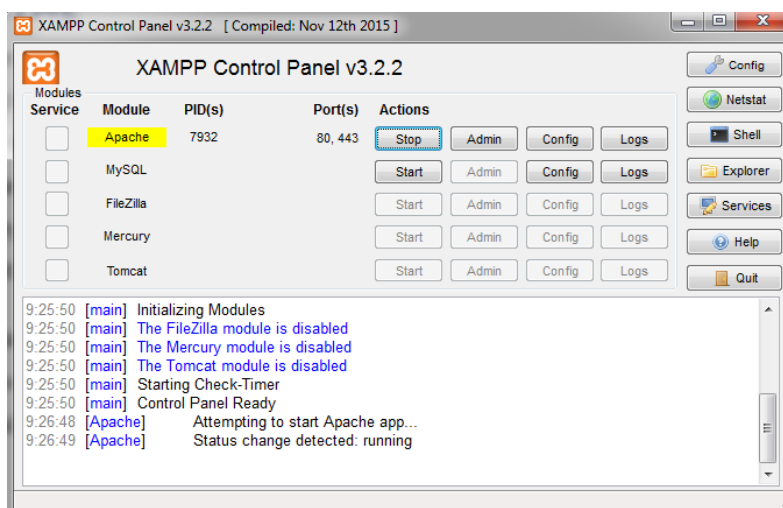
En el vídeo que encontrarás la explicación completa del funcionamiento de una petición de un fichero HTML

 <https://youtu.be/8znGllvW8jE>

## 1.3 Nuestro primer HTML en local

Una vez que hemos entendido el funcionamiento de las páginas html, llega el momento de crear nuestra primera página. Para ello utilizaremos los dos entornos que tenemos creados, el local y el de la nube. En el caso de la edición de ficheros y proyecto tanto en PHP-HTML-JS-CSS3 vamos a necesitar un IDE de desarrollo, un interfaz para poder manejar dichos documentos y a la vez ayudarnos en el proceso de desarrollo. Ya explicamos en la unidad anterior diferentes alternativas de IDEs, a partir de aquí el IDE usado será Atom.io en mi caso.

Recordando también de la unidad anterior, debemos tener instalado y funcionando el servidor Apache, para ello a través del panel de XAMPP activaremos el mismo (para sistemas diferentes a Windows, realizaremos los mismo pasos en los respectivos paneles de control).

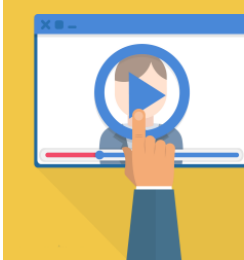





**Imagen 2: Apache encendido**

Fuente de la imagen: propia

Como va a ocurrir a lo largo de todo nuestro curso, os presentaremos tanto la explicación extendida a través de un vídeo que te permitirá

perfectamente realizar el paso a paso de vuestro aprendizaje como el código fuente como el resumen de los pasos escritos.

	En el vídeo que encontrarás la explicación completa de la creación de nuestro primer HTML  <a href="https://youtu.be/hn5AxP9kqP8">https://youtu.be/hn5AxP9kqP8</a>
	En el siguiente enlace tienes el código utilizado:  <a href="https://github.com/pacogomezarnal/codigoFuentePHP/tree/master/html">https://github.com/pacogomezarnal/codigoFuentePHP/tree/master/html</a>

Teniendo por lo tanto estas premisas ya realizadas pasaremos a realizar nuestro primer HTML y su comprobación:

1. Nos colocaremos en el directorio dentro de Apache que nos permita después visualizar el contenido de nuestro HTML realizado. En el caso de Windows, C:\xampp\htdocs
2. Creamos una nueva carpeta denominada **html**
3. Creamos el código con la extensión html o htm siguiendo los pasos del vídeo antes marcado. El fichero se denomina ***hola.html***

Fichero: hola.html

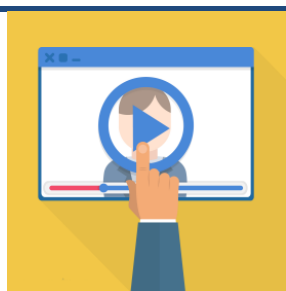
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Nuestro primer html</title>
  </head>
  <body>
    Paco es un campeón!!!!
  </body>
</html>
```

4. Abrimos el navegador Chrome y escribimos,  
***http://localhost/html/hola.html***

## 1.4 Nuestro primer HTML en la nube

Comenzaremos diciendo que este apartado es totalmente voluntario, y sólo para aquellos que tengáis curiosidad de cómo funciona un servicio en la nube como C9 (<https://c9.io/>) del cual hablamos en la anterior unidad. Nosotros durante toda la formación nos vamos a centrar en el desarrollo utilizando un servidor local.

El proceso va a ser muy parecido al visto anteriormente, los pasos diferentes son los de encendido de la máquina Apache puesto que ya viene preinstalado y los de la utilización del editor Atom.io, puesto que utilizaremos el editor propio de c9.



En el vídeo que encontrarás la explicación completa de cómo utilizar c9 en la nube

 <https://youtu.be/rfkZg4oQVZ8>

## 1.5 Algo de sintaxis HTML

No es el objetivo ni repasar ni formar en el lenguaje HTML, ya que de esto se encargan otros módulos pero sí que es importante conocer al menos algo de sintaxis importante de HTML para poder comenzar a trabajar con PHP.

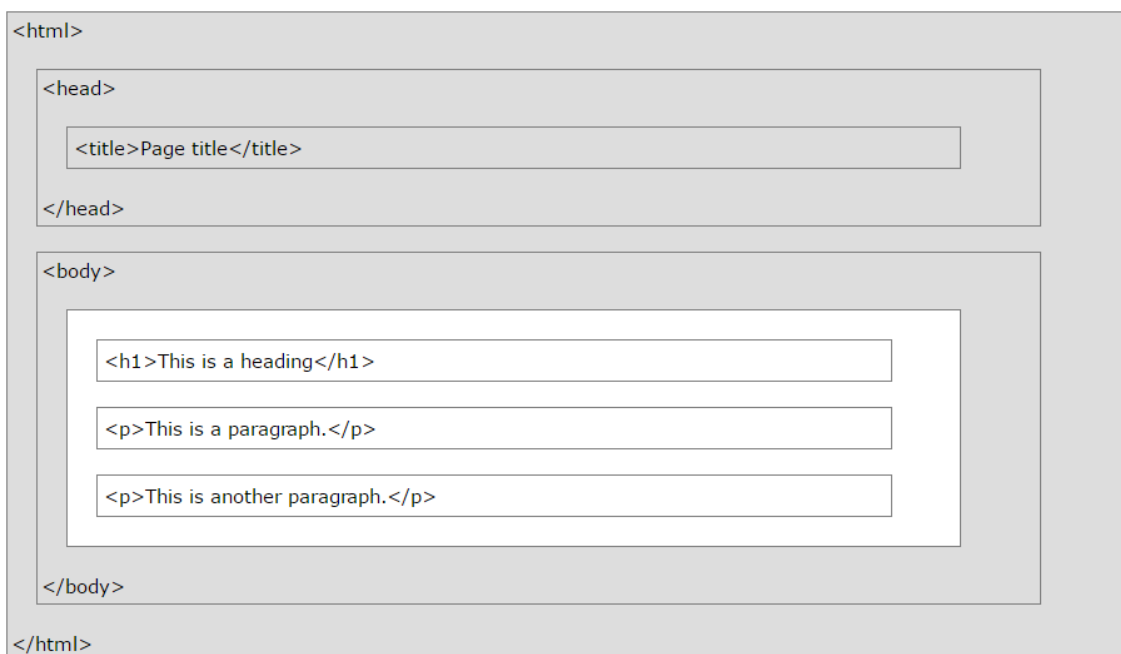
Un excelente tutorial y guía de documentación es W3 Schools (<https://www.w3schools.com/html/>), donde podéis encontrar mucho código y muy bien documentado.

El html es un lenguaje de marcas basado en lo que se denomina "tags". Los tags son elementos que describirán una determinada funcionalidad. Extraído de w3 schools:

`<tagname>content goes here... </tagname>`

Una página web escrita con html deberá tener como mínimo la siguiente estructura:





**Imagen 3: Estructura HTML**

Fuente de la imagen: [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)

En esta estructura observamos las siguientes partes:

- Definición del tipo de fichero mediante la etiqueta <html>
- Definición de la cabecera mediante la etiqueta <head>
  - Definición del título de la página mediante la etiqueta <title>
- Definición del cuerpo del fichero mediante la etiqueta <body>

Observamos también otras premisas que van a ser muy muy importantes:

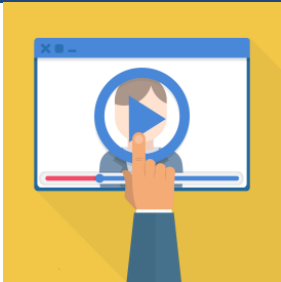

- Todo documento html TIENE la estructura mínima antes descrita
- Toda etiqueta tiene su inicio y finalización (salvo contadas excepciones)
- La disposición de las etiquetas ocupan una estructura jerárquica. <body> se encuentra dentro de <html> y después de <head>

Caso práctico 1
Título: Reforzando HTML

## 2. NUESTRO PRIMER PHP

El funcionamiento de las páginas php es muy similar al de las páginas html, al menos en lo que concierne a tecnología, ya que los servidores usados son los mismos, Apache, y los mecanismos de petición y devolución de la respuesta también, HTTP. Pero la gran diferencia es DÓNDE SE EJECUTA Y SE ANALIZA NUESTRO CÓDIGO PHP.

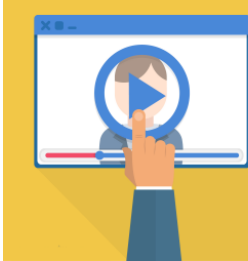



Esto estuvimos trabajándolo ya en la unidad didáctica anterior, por lo que simplemente lo vamos a reforzar con un vídeo:

	<p>En el vídeo se refuerza el funcionamiento de php</p> <p> <a href="https://youtu.be/rfkZg4oQVZ8">https://youtu.be/rfkZg4oQVZ8</a></p>
--	--

### 2.1 Primer PHP

Para escribir código php necesitaremos todo lo visto hasta el momento, un entorno de trabajo, en local, o en la nube, y a partir de ahí podremos escribir nuestro primer programa. Muy recomendable tener siempre a mano la documentación y referencia de php.net, y en este caso seguiremos la documentación de qué es el php.

De igual forma al código HTML, os presentaremos tanto la explicación extendida a través de un vídeo que te permitirá perfectamente realizar el paso a paso de vuestro aprendizaje como el código fuente como el resumen de los pasos escritos.

	<p>En el vídeo que encontrarás la explicación completa de la creación de nuestro primer PHP</p> <p> <a href="https://youtu.be/wQo-QXdCfX0">https://youtu.be/wQo-QXdCfX0</a></p>
	<p>En el siguiente enlace tienes el código utilizado:</p> <p> <a href="https://github.com/pacogomezarnal/codigoFuentePHP/tree/master/php">https://github.com/pacogomezarnal/codigoFuentePHP/tree/master/php</a></p>

Teniendo por lo tanto estas premisas ya realizadas pasaremos a realizar nuestro primer PHP y su comprobación:

1. Nos colocaremos en el directorio dentro de Apache que nos permita después visualizar el contenido de nuestro HTML realizado. En el caso de Windows, C:\xampp\htdocs
2. Creamos una nueva carpeta denominada **php**
3. Creamos el código con la extensión **php** siguiendo los pasos del vídeo antes marcado. El fichero se denomina **hola.php**

#### Fichero: hola.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Primer php</title>
</head>
<body>
  <?php echo "Hola Paco campeón!!!!" ?>
</body>
</html>
```

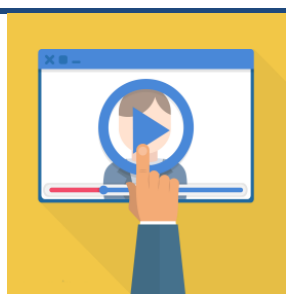
4. Abrimos el navegador Chrome y escribimos, ***http://localhost/php/hola.php***

Caso práctico 2

Título: Reforzando PHP

### 3. PHP Y GIT

¿Cómo unimos el trabajo que realizamos en un entorno local a un entorno en la nube? ¿Cómo comenzamos a trabajar colaborativamente? ¿Cómo comenzamos a compartir nuestros proyectos y desarrollos web? Son preguntas que cuando comenzamos a desarrollar, cuando comenzamos a aprender un determinado lenguaje de programación no nos asaltan ya que no tenemos dicha necesidad, pero que cuando comenzamos a necesitar trabajar en equipo se convierten en indispensables.

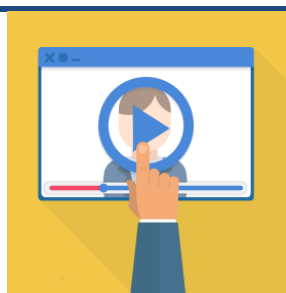


En el vídeo podemos ver el por qué utilizamos git en nuestros proyectos

 <https://youtu.be/GI7b3g9xp0o>

#### 3.1 De local hacia la nube

El primer paso, y sin explicar en detalle cómo funciona git, vamos a generar un proyecto en local y lo vamos a subir a un repositorio remoto.



En el vídeo podemos ver de forma detallada cómo creamos y subimos un repositorio local a la nube

 [https://youtu.be/\\_yzNoenLOik](https://youtu.be/_yzNoenLOik)

Los pasos a realizar son:

1. Abrimos el git bash o consola en el directorio raíz que queramos mantener el repositorio.
2. Inicializaremos el repositorio mediante el comando *git init*
3. Añadimos el fichero, el directorio o todos los ficheros mediante *git add -A*

4. Añadimos el contenido al repositorio local mediante `git commit -m "Mensaje a introducir"`
5. Creamos un repositorio en github.com
6. Copiamos el repositorio creado
7. Añadimos el repositorio remoto mediante el comando `git remote -v origin http://github.com/XXXXXXXXXX`
8. Subimos a github el contenido mediante el comando `git push origin master`

Una vez que hemos realizado todos estos pasos, mantener o añadir código a nuestro repositorio será simplemente:

1. Añadimos el fichero, el directorio o todos los ficheros mediante `git add -A`
2. Añadimos el contenido al repositorio local mediante `git commit -m "Mensaje a introducir"`
3. Subimos a github el contenido mediante el comando `git push origin master`

Como podéis observar:

- NO VOLVEMOS A CREAR UN NUEVO REPOSITORIO MEDIANTE GIT INIT
- NO VOLVEMOS A CREAR UN NUEVO REPOSITORIO REMOTO

Caso práctico 3
Título: Subiendo a github

## RESUMEN FINAL

Hemos realizado un primer fichero php, para ello hemos necesitado conocer:

- la arquitectura de funcionamiento de una petición cliente servidor
- tener en funcionamiento el servidor Apache

- conocer la sintaxis básica de HTML
- conocer el funcionamiento de un editor como Atom.io
- conocer la sintaxis básica de PHP y su inclusión dentro del código HTML

Además de esto hemos profundizado dentro del uso de git para el mantenimiento de nuestro código.