



Tabla de contenido

Descripción material del programa	1
Mapa conceptual	3
1. Definición de PHP	4
¿Qué significa PHP?	4
Un poco de historia	4
2. ¿Cómo funciona?	6
3. Arquitectura de PHP	8
4. ¿Por qué PHP?	9
5. ¿Qué se necesita?	11
Referencias	13





Descripción material del programa

Este material está diseñado para facilitar el proceso de aprendizaje, por esta razón, los contenidos buscan que el aprendiz se apropie del conocimiento que realmente necesita para desarrollar sus habilidades y que lo haga de una forma sencilla y organizada; además de la lectura general cuenta con algunos apartes que contienen: frases o datos para recordar, segmentos de código y consejos, estos elementos se destacan por las siguientes convenciones gráficas:

Ícono	Elemento importante
	Frases o datos para recordar: son extraídas de la lectura previa.
	Segmentos de código: pueden tomarse como base para los ejercicios propuestos.
	Consejos: buenas prácticas para el proceso de desarrollo.

Fuente de imágenes: SENA

La mayor parte de los segmentos de código que aparecen en este material de formación los encuentra como archivos .php que se pueden descargar del material complementario en la siguiente ruta: Materiales del programa / Materiales de apoyo / Documentos complementarios / Documentos complementarios: Actividad de aprendizaje 1

Para usar estos segmentos de código solo se necesita copiarlos y pegarlos en el editor o entorno que esté usando para el desarrollo, o en el caso de que se





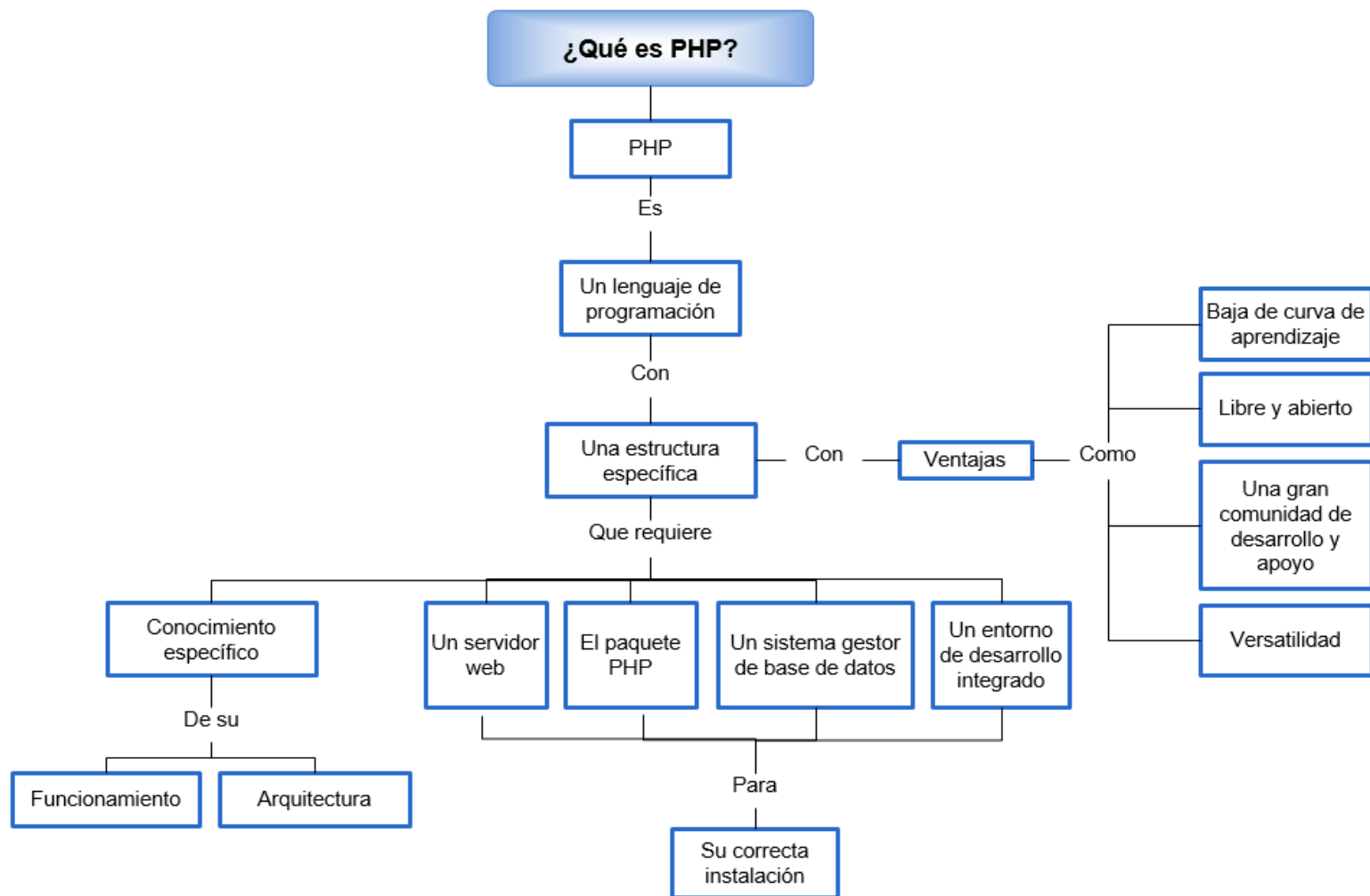
encuentren etiquetados con la ruta del archivo puede abrirse directamente desde el editor o entorno. Los segmentos de código están comentados (usando los comentarios de cada lenguaje: HTML y PHP) para facilitar su comprensión y uso, dichos comentarios pueden ser modificados o retirados de ser necesario.





Mapa conceptual

En el mapa conceptual que se comparte a continuación, se evidencia la interrelación temática del contenido que se plantea en este material de formación:





1. Definición de PHP

Seguramente muchos de quienes están iniciando este proceso de formación tienen ya algunos conocimientos sobre el lenguaje PHP y tal vez incluso algo sobre su historia, pero no está de más tener suficiente información sobre el tema que quiere aprender, es por eso que al iniciar este programa de formación se hace una breve introducción al proceso de creación de este lenguaje.

“PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML” (The PHP Group, s.f.).

Como puede ver se ha tomado la definición literal que aparece en la página web oficial de PHP (www.php.net), puesto que esta fuente es el referente en cuanto a la documentación del lenguaje.

¿Qué significa PHP?

Es un lenguaje de programación cuyas instrucciones pueden ser escritas dentro del código de una página web escrita en HTML, esto hará que las páginas del sitio web a desarrollar sean dinámicas y no estáticas; esto último se logra gracias a la posibilidad de gestionar la información que se obtiene del usuario a través de formularios, al pasar información entre páginas del sitio e interactuar con repositorios de datos tales como los archivos y las bases de datos.



Fuente: SENA

Básicamente PHP es un lenguaje de programación cuyas instrucciones pueden ser escritas dentro del código de una página web en HTML para que las páginas del sitio web sean dinámicas y no estáticas.

Un poco de historia

PHP fue diseñado en 1994 por Rasmus Lerdorf (programador informático nacido en Groenlandia), quien necesitaba tener un contador de las visitas a su currículum en línea, por lo tanto creó su propio lenguaje para desarrollar la aplicación en cuestión, la sigla significaba Personal Home Page Tools (PHP Tools), Lerdorf creó






uno de los primeros lenguajes interpretados del lado del servidor que podía incrustarse directamente en el código HTML y no ser llamado como un archivo aparte que procesara los datos, pero además su creación se convertiría en uno de los hitos de trabajo comunitario para el desarrollo de una herramienta, ya que al publicar su código en 1995 abrió las posibilidades para que muchos desarrolladores a nivel mundial colaboraran con el proyecto.

Pero quienes dieron el siguiente paso en el proceso fueron los estudiantes israelíes Andi Gutmans y Zeev Suraski, quienes en 1997 se dieron a la tarea de reescribir totalmente el analizador sintáctico (parser) del lenguaje, creando PHP3, que es la primera versión del lenguaje como se conoce hoy en día y la que le dio el nombre al actual PHP: Hypertext Preprocessor, puesto que para ellos el nombre original limitaba las posibilidades del lenguaje por estar orientado solo a las páginas personales; posteriormente en 1999 reescribieron todo el código produciendo la versión 4 del lenguaje que venían implementando a lo que ellos llamaron el motor Zend (que proviene de sus nombres: Zeev y Andi).

Gutmans y Suraski son además los fundadores de la que se considera la empresa de PHP, Zend Technologies, la cual se ha encargado de generar herramientas para el trabajo con PHP, tales como el Entorno de Desarrollo Integrado (IDE por sus iniciales en inglés), Zend Studio la plataforma de servicio de aplicaciones PHP Zend Server, el framework de desarrollo Zend Framework y los estándares y buenas prácticas de desarrollo que son considerados por la mayoría de los programadores como los referentes a seguir. Todas las herramientas de la compañía son de tipo propietario, es decir que deben ser compradas para usarlas más allá de la versión de prueba, pero el lenguaje completo se mantiene libre, debido a la filosofía original de su creador Rasmus Lerdorf. (The PHP Group, s.f.)

Al momento de elaborar este documento PHP se encuentra en la versión 5, esta ha incorporado una gran cantidad de cambios que incluyen mejoras de seguridad y programación, orientada a objetos bien soportados, entre otros.

 <p data-bbox="284 1764 487 1795">Fuente: SENA</p>	<p>Aunque PHP fue diseñado originalmente por Rasmus Lerdorf, fueron Andi Gutmans y Zeev Suraski quienes llevaron el lenguaje a su nivel actual, iniciando con la creación de PHP 3.0 y sus posteriores versiones; creando la empresa de PHP, Zend Technologies. A pesar de que las herramientas de desarrollo y producción creadas por Zend Technologies son propietarias, el lenguaje en sí mismo se mantiene libre.</p>
--	---






2. ¿Cómo funciona?

Para comprender mejor cómo se programa en un lenguaje específico, es conveniente que se conozca la forma en que funciona. Como ya se había dicho antes, todo programa escrito en PHP puede estar escrito dentro del código HTML de una página web, esto facilita el trabajo del desarrollador puesto que todo lo que no necesite ningún tipo de procesamiento de datos puede escribirse como HTML puro y solo se escribirán instrucciones PHP cuando se requiera realmente.

El siguiente segmento de código es un ejemplo simple de un archivo con código PHP.

Ejemplo 1:

	<pre><!DOCTYPE html> <html> <!-- En los archivos .php el código HTML se escribe normalmente--> <head> <!-- como puedes ver toda la estructura de la página está escrita en HTML--> <title>Ejemplo 1</title> <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" /> </head> <body> <p> Esto es HTML puro</p> <!-- Esto está escrito en HTML--> <?php echo "Esto es código PHP"; /* Esto está escrito en PHP*/ ?> </body> </html></pre>
--	--

Fuente: SENA

Descargue el segmento anterior del código como archivo .php del material complementario de este programa de formación en la siguiente ruta: Materiales del programa / Materiales de apoyo / Documentos complementarios / Documentos complementarios: Actividad de aprendizaje 1 / Ejemplo 1

Las nueve primeras líneas del segmento de código pertenecen a la estructura básica de cualquier página escrita en HTML y como se puede ver no se requiere de ningún tipo especial de lenguaje diferente al HTML para escribirlas (es necesario recordar que para este programa de formación es indispensable tener conocimientos de HTML, de no ser así se recomienda investigar sobre el tema), en la décima línea se encuentra un objeto `<p>` que pertenece a una salida que pueden verse en el área de visualización del navegador, el resultado será una





línea de texto que dice “Esto es HTML puro”, luego en la undécima línea se ve la primera instrucción de PHP a la que tiene acceso en este material y es la etiqueta `<?php` que indica que allí empieza el código PHP, en la línea doce hay una sentencia PHP que se encarga de generar una salida de texto al área de visualización del navegador, en la línea trece esta la etiqueta de cierre de un segmento de código PHP y lo que sigue son más instrucciones HTML de la estructura básica de la página.

En la Figura 1 que se muestra a continuación se observa el resultado de ejecutar el segmento de código anterior, básicamente lo que el usuario verá en el área de visualización del navegador son dos líneas de texto; la primera resultado de una instrucción HTML y la segunda derivada de una instrucción PHP, pero para el usuario básico el hecho de que la página tenga código PHP es totalmente transparente, ya que lo que se ve en el navegador es simplemente texto, se puede deducir la existencia de este lenguaje en la página porque la extensión del archivo que aparece en la URL se lo indica.

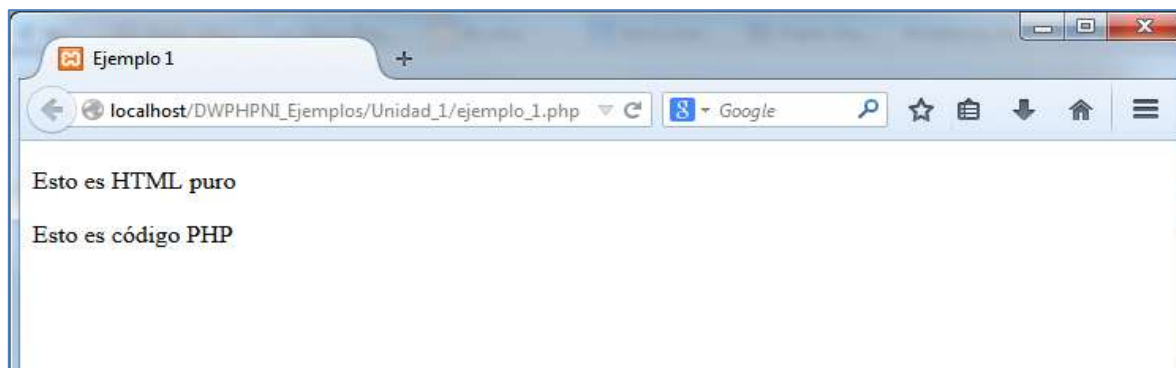


Figura 1. Ejecución ejemplo 1
Fuente: SENA

Los usuarios de las aplicaciones PHP no solo tienen acceso a lo que pueden ver en el área de visualización del navegador, también desde el mismo navegador pueden acceder a la codificación de la página, con solo hacer uso de una función del menú o de una combinación de teclas (normalmente es **Ctrl + U** en la mayoría de los navegadores).

Para el caso de los lenguajes interpretados del lado del cliente (como por ejemplo JavaScript) el código fuente puede verse con esta funcionalidad si está incrustado en el mismo archivo, lo cual no sucede con PHP, puesto que su código fuente es interpretado por el paquete del lenguaje y luego de su ejecución lo que devuelve es solo código HTML, por lo tanto, si el usuario hace uso de la opción que le





permite ver el código fuente del ejemplo ejecutado, únicamente verá lo que se presenta en la Figura 2.

Figura 2. Visualización del código fuente que recibe el navegador al ejecutar el ejemplo 1
Fuente: SENA

3. Arquitectura de PHP

Cuando un usuario explora en internet, lo que hace desde su navegador es solicitar un documento web que puede encontrarse en un servidor en cualquier parte del mundo del cual conoce previamente su dirección electrónica o la ha recibido como respuesta a una consulta en un motor de búsqueda, el servidor (que es básicamente un computador especializado que está diseñado para estar encendido 24 horas al día, 365 días al año y además está conectado a la internet la misma cantidad de tiempo) utiliza un programa llamado servidor web (Apache o Internet Information Server) que se encarga de recibir la solicitud, buscar el documento solicitado en los dispositivos de almacenamiento y finalmente devolver el código HTML del documento al navegador del usuario, esto es lo que pasa cuando se solicita cualquier documento con extensión .html.

Pero cuando se hace la solicitud de un documento con extensión .php el servidor web no está en capacidad de procesar su código, por lo tanto necesita tener instalado el paquete de PHP (de no ser así lo que haría es enviar el archivo .php en forma de descarga, lo que sería contraproducente pues entregaría el código fuente de la aplicación al usuario) al cual llama como subservicio, solicitándole que interprete el código PHP que se encuentra en el documento, una vez que termina su trabajo devuelve al servidor web solamente código HTML conteniendo el





resultado de las rutinas, si además las rutinas de PHP hacen llamados a datos contenidos en un Sistema Gestor de Bases de Datos (como el MySQL) es el intérprete el que se encarga de enviar la solicitud a dicho gestor y este último le devuelve los datos para que el intérprete los incorpore en el código HTML que va a devolver al servidor web. Es aquí donde se explica porque a pesar de que el usuario está solicitando un documento con extensión .php finalmente solo recibe un documento que contiene código HTML, puede verse la explicación gráfica de esta arquitectura en la Figura 3.

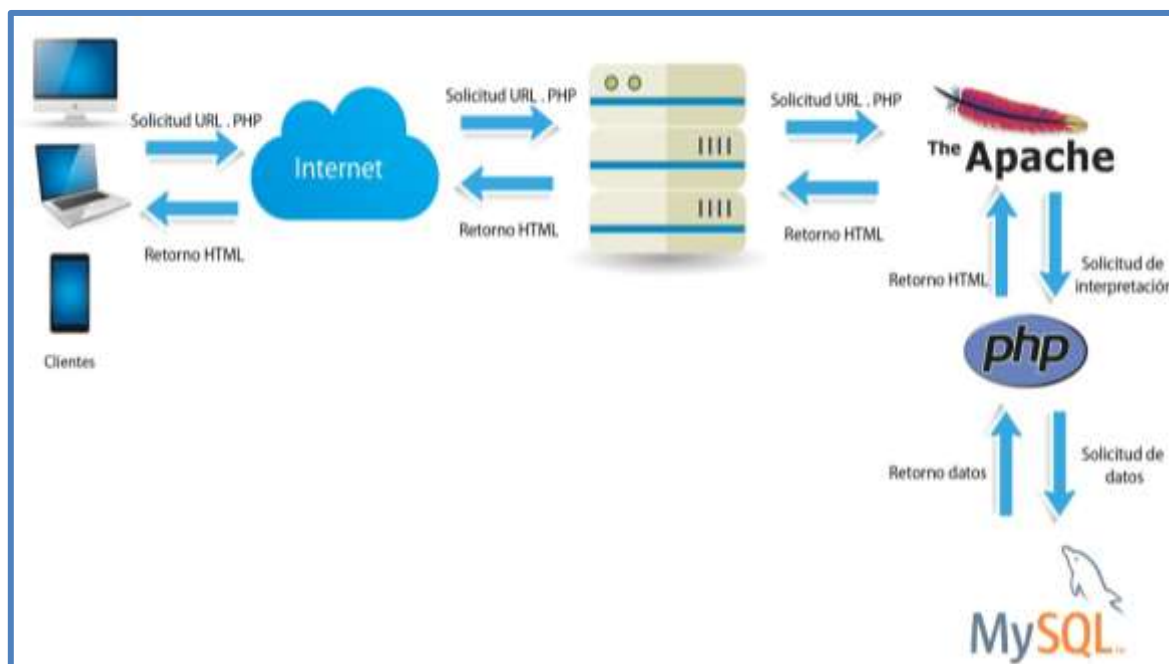


Figura 3. Diagrama de arquitectura del funcionamiento de PHP
Fuente: SENA

4. ¿Por qué PHP?

En general el desarrollo web presenta grandes ventajas frente al desarrollo de aplicaciones de escritorio, por ejemplo: una aplicación web es poco vulnerable a la piratería ya que los usuarios no tienen acceso a las distribuciones sino a la ejecución de los paquetes desarrollados (tanto es así que incluso empresas como Microsoft están orientando sus esfuerzos a poner sus herramientas online, Office con 365 es un claro ejemplo de ello), así mismo puesto que las aplicaciones están centralizadas cuando se quieren publicar actualizaciones no es necesario que el usuario obtenga los nuevos paquetes para instalarlos en su equipo (ya sea en medios físicos o descargándolos de internet), sino que simplemente se instalan las





nuevas versiones en el servidor y el usuario automáticamente estará trabajando en ellas, lo cual hace menos traumáticos los avances de las herramientas y evita tener usuarios trabajando con versiones desactualizadas con posibles vulnerabilidades y sin ningún tipo de soporte.

PHP cuenta con otras ventajas particulares que se explican a continuación:

- **Baja curva de aprendizaje:** PHP es considerado como uno de los lenguajes de más fácil aprendizaje debido a su sintaxis simple y a la gran cantidad de funciones incorporadas que facilitan el desarrollo, incluso se recomienda para el aprendizaje de programación para quienes no han desarrollado aplicaciones nunca antes.
- **PHP es libre y abierto:** todo el código del lenguaje en sí mismo se mantiene abierto y libre para su uso, modificación y redistribución (siempre que se respeten los mandatos de las licencias de software libre), esto hace que los proyectos desarrollados con este lenguaje sean mucho más baratos. Algunas críticas contra PHP aseguran que su condición de software libre lo hacen vulnerable e inseguro, pues al tener acceso al código fuente los posibles atacantes pueden encontrar puertas de acceso a los sistemas desarrollados con este, pero es algo subjetivo ya que así mismo la comunidad que trabaja para hacer PHP es inmensa y por lo tanto se tienen miles de auditores encontrando errores y colaborando con su solución, lo cual lo hace realmente mucho más seguro que los lenguajes propietarios que solo tienen un pequeño número de personas que acceden a su código y pueden auditarlo y mejorarlo, dando paso a que muchos errores queden en las versiones de producción. Además existen herramientas libres que funcionan de maravilla en conjunto con PHP como Apache (servidor web), MySQL (Sistema de Gestión de Bases de Datos) y NetBeans (Entorno de Desarrollo Integrado).
- **Gran comunidad de desarrollo y apoyo:** como ya se dijo en el punto anterior, la condición de software libre del PHP hace que tenga una de las comunidades más grandes de desarrollo usando el lenguaje, apoyando a quienes empiezan a usarlo y aportando para su mejoramiento. Se pueden encontrar en internet una gran cantidad de manuales, foros y comunidades locales en las que se solucionan todas las dudas y en las que se comparte valiosas rutinas que pueden ser aplicadas en proyectos propios. Una gran cantidad de compañías se han dedicado al desarrollo de Frameworks (básicamente son librerías de funciones y objetos que pueden ser usados en los proyectos propios) y Sistemas de Gestión de Contenidos que facilitan y agilizan la creación de nuevas aplicaciones.





- **Versatilidad:** PHP es multiplataforma, puede ser instalado en casi todos los sistemas operativos, es soportado por la mayoría de servidores web, esto hace que encontrar un servicio de hosting (el hosting o alojamiento es el servicio que prestan diferentes empresas alquilando espacio en sus servidores para almacenar y publicar un sitio web) para soportar las aplicaciones desarrolladas en este lenguaje sea relativamente fácil y muy económico, comparado con otras tecnologías como JAVA (se tiene muy poca oferta de hosting que soporten este lenguaje).

5. ¿Qué se necesita?

En lo referente a la plataforma de trabajo como se observó en la Figura 3, se requiere cuando menos tener un servidor web, el paquete PHP (última versión estable que se encuentre disponible) y un Sistema Gestor de Bases de Datos (para cuando se requiera trabajar en aplicaciones con repositorios de datos).

En el caso de PHP existe lo que podría llamarse el equipo de trabajo ideal: Apache + PHP + MySQL, las tres son herramientas libres con un gran respaldo y estabilidad, son la combinación más utilizada y la mayoría de los hosting que ofrecen soporte para PHP la usan. Para trabajar en un ambiente de desarrollo se requiere de estas herramientas, pero no serán instaladas en un servidor, sino como servicios locales en el equipo en el que se va a ejecutar PHP.

Anteriormente era necesario descargar cada una de las herramientas por separado, instalarlas y configurarlas para que funcionaran mancomunadamente, pero hoy en día se cuenta con diferentes paquetes que contienen las tres herramientas y adicionan un paquete de instalación que configura y deja perfectamente funcional todo el entorno de trabajo, como es el caso del paquete XAMPP que se utilizará en este programa de formación, su proceso de instalación se explica en los videos tutoriales:

- ¿Cómo instalar el paquete XAMPP?
- ¿Cómo instalar el Entorno de Desarrollo Integrado (IDE) NetBeans?
- Mi primer proyecto PHP.

La ruta para acceder a estos videos es la siguiente: Materiales del programa / Materiales de formación / Material: Actividad de aprendizaje 1

Quienes usen sistemas operativos como GNU/Linux o Mac identificarán algunas diferencias en el proceso de instalación del entorno de trabajo que se indica en este programa de formación, el resto de material e indicaciones les servirá sin





ninguna modificación, ya que al ser un lenguaje interpretado, la programación con PHP es la misma bajo cualquier sistema operativo.

Por otra parte, requieren de un entorno de desarrollo para escribir y depurar las rutinas de las aplicaciones que se van a crear, la buena noticia es que PHP puede escribirse incluso en el bloc de notas (se recomienda que las personas que están iniciando usen este editor de texto o uno de similar simplicidad para que memoricen mejor las estructuras básicas del lenguaje), en el mercado existen un sinnúmero de Entornos Integrados de Desarrollo tanto libres como propietarios, entre estos se destacan: NetBeans, Eclipse, Aptana Studio, phpDesigner, entre otros.

Para este programa de formación en particular se aconseja el uso de NetBeans por su gran soporte al código PHP (se puede descargar un módulo especialmente orientado a este lenguaje), pero además porque es totalmente gratuito y tiene una versión en español.



Fuente: SENA

Se puede usar el bloc de notas para codificar en PHP, esto implica recordar todas las instrucciones de memoria, sin las ayudas de un IDE, lo que le permite ser más autónomo en la programación.



Fuente: SENA

Aunque es buena idea aprender como instalar todas las herramientas de forma manual para el proceso de desarrollo, en este proceso de formación se sugiere la instalación de un paquete como XAMPP, ya que integra y configura todas las herramientas de forma automática, incluido el IDE NetBeans.

Con el propósito de poner en práctica los conocimientos adquiridos a través de este material de formación, consulte la guía de aprendizaje y realice todas las evidencias propuestas en ella.

Para acceder a la guía y a las evidencias diríjase al botón: Actividades / Actividad de aprendizaje 1





Referencias

- The PHP Group. (s.f.) *¿Qué es PHP?* Consultado el 30 de junio de 2015, en <http://php.net/manual/es/intro-what-is.php>
- The PHP Group. (s.f.). *Historia de PHP*. Consultado el 30 de junio de 2015, en <http://php.net/manual/es/history.php.php>
- The PHP Group. (s.f.). *PHP 5.6.9 is available*. Consultado el 30 de junio de 2015, en www.php.net
- The PHP Group. (s.f.). *Manual de PHP*. Consultado el 30 de junio de 2015, en <http://www.php.net/manual/es/index.php>

Control del documento

	Nombre	Cargo	Dependencia	Fecha
Autor	Jorge Luis Ballesteros Vargas	Instructor	Centro Metalmeccánico Regional Distrito Capital	Diciembre de 2014
Adaptación	Paola Andrea Bobadilla Gutiérrez	Guionista - Línea de producción	Centro Agroindustrial Regional Quindío	Junio de 2015

