Proyecto, diseño e implementación de sistemas: Trabajo Practico PHP

Integrantes: Galeano Ciro

Mastrangelo Tamara

Pérez Agustina

Curso: 7mo Informática “B”

Instituto Leonardo Murialdo

Introducción:

En este trabajo vamos a hacer un repaso general de la historia de PHP, programa con el que estuvimos trabajando las ultimas clases.

PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web favoreciendo la conexión entre los servidores y la interfaz de usuario. De este lenguaje podemos destacar que es de código abierto, es decir que el usuario puede programar cualquier proyecto y comercializarlo sin problemas. Otra característica que podemos resaltar es que está en constante perfeccionamiento.

A. Respondan las siguientes preguntas

1. ¿En qué año surge el lenguaje PHP? ¿Qué agrega a los lenguajes que existían hasta el momento?

PHP surge en el año 1994 gracias a Rasmus Lerdorf, un programador nacido en Groenlandia.  El lenguaje fue diseñado para asemejarse a C en estructura, haciéndolo una adopción sencilla para desarrolladores familiarizados con C, Perl, y lenguajes similares. Incluía soporte interno para DBM, mSQL, y bases de datos Postgres95, cookies, soporte para funciones definidas por el usuario, y mucho más.

1. ¿Qué significa PHP?

PHP significa PHP Hypertext Preprocessor.

1. ¿Cuáles son las diferencias más notables entre JavaScript y PHP?

Las diferencias más notables son las siguientes:

PHP se inicia en el servidor, lo que genera un código HTML que se envía como respuesta del navegador. JavaScript se inicia en el navegador, no en el servidor por lo que a menudo los oirá referirse a ellos como lenguajes del lado del servidor o del lado del cliente.

PHP es de código abierto, tiene un buen soporte y está bien mantenido. JavaScript no es de código abierto sino de estándar abierto, lo que significa que es una implementación de ECMAScript. Debido a que las descargas de JavaScript son gratuitas, no tendrá que pagar nada por los scripts producidos con JavaScript.

PHP es un lenguaje asincrónico de subprocesos múltiples. Por el contrario, JavaScript es un lenguaje de programación asincrónico de un solo subproceso.

PHP es más seguro que JavaScript ya que el código PHP no está visible en el navegador. La desventaja de JavaScript es que puede estar sujeto a riesgos de seguridad. Sin embargo, esto puede superarse haciendo uso de herramientas de análisis de seguridad y cumpliendo con las prácticas de desarrollo recomendadas.

PHP facilita el acceso a la base de datos porque utiliza lenguaje del lado del servidor. JavaScript, como ya sabemos, usa lenguaje front-end. Esto le dificulta el acceso a la base de datos.

1. ¿Qué necesitamos para trabajar con PHP? Explica cada recurso.

Se precisa Instalar el siguiente grupo de aplicaciones:

Servidor Web: Apache.

Módulo o intérprete de PHP.

Servidor de Base de Datos: MySQL.

Un valor tipo resource es una variable especial, que contiene una referencia a un recurso externo. Los recursos son creados y usados por funciones especiales.

Dado que las variables resource contienen gestores especiales a archivos abiertos, conexiones con bases de datos, áreas de pintura de imágenes y cosas por el estilo, la conversión a tipo resource carece de sentido.

Gracias al sistema de conteo de referencias introducido con el Motor Zend, un recurso que ya no es referenciado es detectado automáticamente, y es liberado por el recolector de basura.

Por esta razón, rara vez se necesita liberar la memoria manualmente.

Canvas – Un entorno de desarrollo para PHP5 con la facilidad de uso como principal atractivo.

CakePHP – Un entorno de desarrollo PHP orientado a funcionar al estilo de Ruby on Rails.

CodeIgniter – Un entorno de desarrollo PHP pensando especialmente para páginas alojadas en servidores compartidos.

Coders4fun – Un blog sobre programación, incluye un montón de pequeños fragmentos de código PHP para ayudarte a aprender nuevos trucos.

Script para formularios de DagonDesign – Un script en PHP para ayudarte a crear formularios de cualquier longitud.

DIY Framework – Un entorno de desarrollo PHP minimalista y sin florituras, tal como implica su

nombre.

GoodPHPTutorials.com – Una extensa colección de tutoriales para programadores PHP de todos los niveles.

Horde.org – Un entorno de desarrollo especializado en aplicaciones PHP y con facilidades para otros idiomas que no sean el inglés.

KillerPHP.com – Cerca de dos docenas de vídeo tutoriales de PHP que te enseñan desde los pasos básicos de PHP hasta las técnicas más avanzadas.

PHPBuddy.com – Un sitio lleno de recursos para el desarrollador principiante de PHP, incluye

numerosos scripts para ayudarte a salvar los obstáculos típicos del principiante.

PHPClasses.org – Fuente de un montón de scripts PHP para ahorrarte el trabajo de programarlas tareas más simples.

PHPFreaks – Un extenso sitio de ayuda para usuarios PHP de todos los niveles, con más de 400

fragmentos de código para ayudarte con diferentes aspectos de un proyecto.

PHPOpenBiz – Un entorno de desarrollo PHP centrado en aplicaciones para empresas más que

en otra cosa.

PHPVideoTutorials.com – 13 vídeo tutoriales sobre PHP desde 6 a 22 minutos de duración.

QCodo.com – Un entorno de desarrollo de código abierto que funciona con PHP5 centrado principalmente en hacer formularios de código PHP de fácil lectura.

Seagull Project – Un entorno de desarrollo PHP con un sistema de plantillas integrado.

Symfony Project – Un entorno de desarrollo PHP de código abierto con implementación de AJAX, generador del panel de administración y mucho más.

The PHP Resource Index – Tiene numerosos recursos, pero la estrella es su archivo de más de 3600 scripts de PHP para cada función que puedas imaginar.

Xajax Project – Un proyecto de sourceforge para añadir AJAX a cualquier página de PHP.

Zend Framework – Otro entorno de desarrollo PHP de código abierto con más de 300 colaboradores.

Zoop Framework – Entorno de desarrollo PHP capaz de producir formularios estilo web 2.0 y manejar AJAX.

1. Investiga sobre las diferentes versiones de PHP existentes y sus características.

PHP 1.0

Su lanzamiento fue en el año 1995. llamada “Personal Home Page Tools (PHP Tools)”. Es la primera versión en usar el nombre “PHP”. Hoy en día ya no tiene soporte

PHP 2.0

Su lanzamiento fue en el año 1997. Considerada por sus creadores la “herramienta más rápida y sencilla” para crear páginas web dinámicas. Hoy en día se desconoce el soporte de esta versión.

PHP 3.0

Su lanzamiento fue en el año 1998. Zeev suraski y Andi Gutmans reescribieron la base para esta versión. Llego a su fin en el año 2000, siendo el fin del soporte técnico para esta versión.

Una de las mejoras de PHP 3.0 era su gran extensibilidad. Esta es la existencia de un núcleo ocore al que se le pueden ir agregando líneas de código que permiten hacer más cosas.

PHP 4.0

Su lanzamiento fue en el año 2000, el fin del soporte se dio en 2001, su principal cambio fue agregar un sistema de parsing de dos fases, el motor Zend. Además del aumento en el rendimiento, se introdujeron novedades relacionadas con el soporte de objetos en PHP. La programación orientada a objetos es una forma avanzada de programación y gracias a estas características PHP se encuentra entre los lenguajes más destacados del medio.

PHP 5.0

Su lanzamiento fue en el año 2004, el fin del soporte se dio en 2005, las características importantes de esta versión son: Motor Zend II con un nuevo modelo de objetos.

PHP 5.6

Su lanzamiento fue en el año 2014, su soporte está dada hasta 2018, sus mejoras son: Constantes con expresiones escalares, listas de argumentos de longitud variable y exponencial mediante el operador.

PHP 7.0

Su lanzamiento fue en el año 2015, el soporte será hasta 2018, principales mejoras son mejoras de rendimiento, declaración de tipos de retorno en funciones. El más importante cambio es que no sólo se mejoró la velocidad, el uso de memoria en esta nueva versión se ha visto reducido drásticamente. Otra mejora aparte de duplicar la velocidad es que el uso de memoria se redujo un 40% con respecto a las versiones anteriores.

Conclusión:

Nos pareció un buen trabajo debido a que aprendimos un poco más sobre las bases de PHP. También algo muy interesante es que las páginas que programemos a futuro no van a ser archivos planos programados solo con HTML, sino que las programaremos junto a este lenguaje nuevo para nosotros.

Bibliografía:

<https://disenowebakus.net/presentacion-php.php>

<https://www.php.net/manual/es/history.php.php>

<https://desarrolloweb.com/articulos/436.php>

<https://yapame.com.bo/cual-es-la-diferencia-entre-javascript-y-php-en-desarrollo-web/>

<https://www.php.net/manual/es/resource.php>

<https://www.samuelaguilera.com/post/20-recursos-para-programadores-de-php>