10 de octubre de 2024

Elaborado Por: Jair Garcia

Reconocer aspectos generales sobre programación a través de un ejercicio practico

Tabla de contenido

[Introducción 2](#_Toc180001104)

[Conclusiones (al final) 3](#_Toc180001105)

[Desarrollo 4](#_Toc180001106)

[¿ cómo convertir el código php en bytecode? 4](#_Toc180001107)

[R// 1. Usar PHP Opcache 4](#_Toc180001108)

[2. Usar herramientas como PHP-AST 4](#_Toc180001109)

[3. **Compilar PHP usando un código de bytesphc** 5](#_Toc180001110)

[¿ 2. citar ejemplos de php a bytecode. 5](#_Toc180001111)

[¿Cómo funciona el compilador JIT? 7](#_Toc180001112)

[¿Investigar y alistar las diferentes extensiones de PHP con su descripción? 7](#_Toc180001113)

[1. PDO (Objetos de datos PHP) 7](#_Toc180001114)

[2. MySQLi (MySQL mejorado) 7](#_Toc180001115)

[3. rizo 8](#_Toc180001116)

[4. OpenSSL 8](#_Toc180001117)

[5. Dios bendiga 8](#_Toc180001118)

[6. mbstring (cadena multibyte) 8](#_Toc180001119)

[XML 8](#_Toc180001120)

[8. intl (Internacionalización) 8](#_Toc180001121)

[9. bcmath (Calculadora binaria) 8](#_Toc180001122)

[10. exif 8](#_Toc180001123)

[11. Información del archivo 8](#_Toc180001124)

[12. código postal 9](#_Toc180001125)

[13. imagick 9](#_Toc180001126)

[14. redis 9](#_Toc180001127)

[15. pdo\_mysql 9](#_Toc180001128)

[16. mysqli 9](#_Toc180001129)

[17. JABÓN 9](#_Toc180001130)

[18. caché 9](#_Toc180001131)

[19. jabón 9](#_Toc180001132)

[20. APCU 9](#_Toc180001133)

[¿Que es un servidor web. Arquitectura del sevidor Web? 10](#_Toc180001134)

[Bibliografía 11](#_Toc180001135)

Introducción

El lenguaje de programación PHP ha sido un pilar fundamental en el desarrollo web desde su creación, facilitando la construcción de aplicaciones dinámicas y la interacción con bases de datos. Este trabajo se propone explorar varios aspectos claves relacionados con PHP y su funcionamiento.  
Justificación

La elección de investigar sobre la conversión del código PHP a bytecode, el funcionamiento del compilador Just-In-Time (JIT), las extensiones de PHP y la arquitectura del servidor web se fundamenta en la creciente relevancia de PHP en el desarrollo de aplicaciones web. Moderno. A medida que las aplicaciones se vuelven más complejas y demandantes en cuanto a rendir ,la conversión a bytecode y el uso del compilador JIT son fundamentales para mejorar la eficiencia de ejecución de aplicaciones PHP.

Conclusiones (al final)

En conclusión, la investigación sobre la conversión del código PHP a bytecode, el funcionamiento del compilador Just-In-Time (JIT), las extensiones de PHP y la arquitectura del servidor web revelan aspectos cruciales que contribuyen al desarrollo de aplicaciones web eficientes y escalables.

# Desarrollo

## ¿ cómo convertir el código php en bytecode?

### R// 1. Usar PHP Opcache

PHP Opcache es una extensión que compila el código PHP en bytecode para mejorar el rendimiento. Si deseas ver el bytecode generado, puedes habilitar el Opcache y luego usar herramientas para inspeccionarlo.

# 2. Usar herramientas como PHP-AST

**1. Instalar php-ast:** Puedes instalar la extensión usando PECL:

pecl install ast

2. **Usar la extensión:** Aquí hay

<?php

$code = '<?php echo "Hello, World!";';

$ast = ast\parse\_code($code, 70); // 70 es la versión de PHP

print\_r($ast);

?>

## 3. **Compilar PHP usando un código de bytesphc**

1. **Instalar phc:** Debes descargar e instalar phcdesde su repositorio

2 .**Compilar el código:** Ejecuta el siguiente

intento

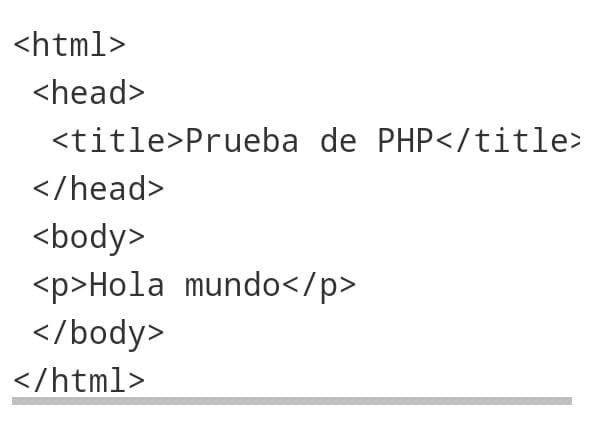
Copiar código

phc -o output\_file.phpc input\_file.php

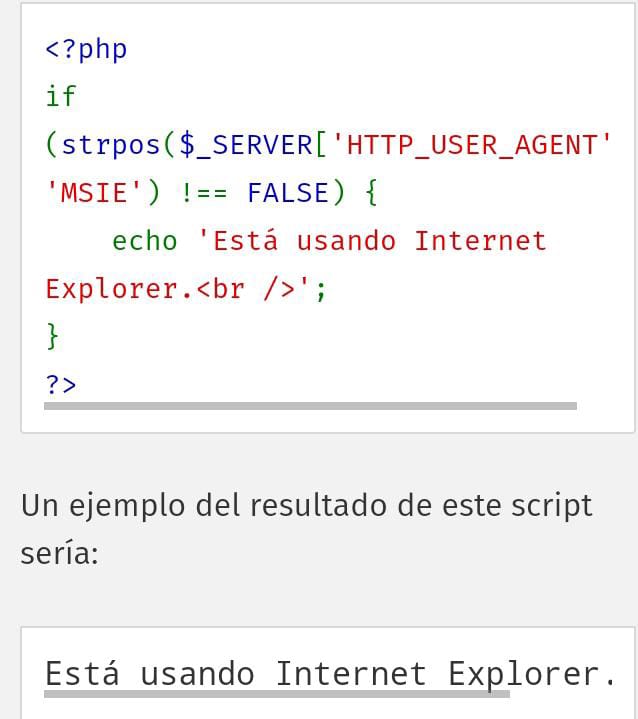
## ¿ 2. citar ejemplos de php a bytecode.

****

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final <?php y ?> que permiten entrar y salir del "modo PHP".



Este programa es extremadamente simple y realmente no es necesario utilizar PHP para crear una página como esta. Lo único que muestra es: Hola mundo empleando la sentencia echo de PHP.



Aquí hemos introducido un par de conceptos nuevos. Tenemos una sentencia if. Si está familiarizado con la sintaxis básica del lenguaje C, debería parecerle lógico.

## ¿Cómo funciona el compilador JIT?

El compilador JIT (Just-In-Time) es una técnica de optimización que compila código en tiempo de ejecución en lugar de hacerlo completamente antes de la ejecución. El funcionamiento del JIT comienza con la interpretación inicial del código, permitiendo que el programa se ejecute rápidamente. A medida que se ejecuta, el JIT identifica las partes del código que se están utilizando con más frecuencia, conocidas como "código caliente". Una vez que se identifica este código, el JIT lo compila en código máquina nativa para la plataforma específica, lo que permite una ejecución más eficiente.

## ¿Investigar y alistar las diferentes extensiones de PHP con su descripción?

## 1. **PDO (Objetos de datos PHP)**

Proporciona una interfaz para acceder a bases de datos de forma segura y eficiente. Permite trabajar con

## 2. **MySQLi (MySQL mejorado)**

Es una extensión que proporciona una interfaz mejorada para interactuar con bases de datos MySQL. Características del soporte a

## 3. **rizo**

Permite realizar solicitudes HTTP y comunicarse con otros servidores a través de una variedad de protocolos (HTTP, FTP, etc.). Es útil para consumir

## 4. **OpenSSL**

Proporción

## 5. **Dios bendiga**

Es una biblio

## 6. **mbstring (cadena multibyte)**

Propuesta

## **XML**

Incluye varias funciones y bibliotecas para manipular y procesar archivos XML. Permite leer, escribir y validar documentos XML utilizando diferentes

## 8. **intl (Internacionalización)**

Ofrece soporte para la internacionalización y localización en PHP. Funciones proporcionadas para formatear fechas, números y mensajes de acuerdo con las convenciones locales.

## 9. **bcmath (Calculadora binaria)**

Permite realizar cálculos aritméticos de alta precisión con números grandes. Es útil para aplicaciones que requieren un alto grado de pre

## 10. **exif**

Permiso

## 11. **Información del archivo**

Funciones proporcionadas para detectar el tipo de contenido de un archivo basándose en su contenido real en lugar de su e

## 12. **código postal**

Permite crear

## 13. **imagick**

Una extensión para la manipulación avanzada de imágenes utilizando la biblioteca ImageMagick. Permite realizar operaciones complejas de procesamiento

## 14. **redis**

Permite la integración con bases de datos Redis, un almacén de datos en memoria. Es útil para almacenamiento en caché, colas de trabajo y almacena.

## 15. **pdo\_mysql**

Extensión de PDO que proporciona soporte específico para bases de datos MySQL, permitiendo el uso de las características avanzadas

## 16. **mysqli**

Extensión que permite la interacción con bases de datos MySQL utilizando características mejoradas, como consultas preparadas y una interfaz orientada a objetos.

## 17. **JABÓN**

Permite l

## 18. **caché**

Optimiza la ejecución del código PHP

## 19. **jabón**

Permite la creación y consumo de servicios web basados ​​en el protocolo SOAP, facilitando la interoperabilidad entre diferentes aplicaciones.

## 20. **APCU**

¿Eres tú?

## ¿Que es un servidor web. Arquitectura del sevidor Web?

Un **servidor web** es un sistema informático que almacena, procesa y entrega contenido web a los navegadores de los usuarios a través de Internet. Cuando un usuario solicita un recurso web (como una página HTML, una imagen, o un archivo de video), el servidor web responde enviando ese recurso al navegador del usuario.

El software que gestiona el hardware del servidor y proporciona La arquitectura de un servidor web puede describirse en varias capas y componentes clave:

1. **Hardware**:

**Servidor físico**: El equipo físico que alberga el software del servidor web.

**Red**: Conexiones de red que permiten la comunicación entre el servidor y los clientes (usuarios) a través de Internet.

1. **Sistema Operativo**:

Servicios básicos. Ejemplos comunes incluyen Linux, Windows Server, y Unix.

1. **Software del Servidor Web**:

Programas que manejan las solicitudes HTTP y responden con contenido. Ejemplos populares son Apache, Nginx, Microsoft IIS y LiteSpeed.

1. **Protocolos de Comunicación**:

**HTTP/HTTPS**: Protocolos utilizados para la comunicación entre el servidor y el cliente. HTTPS incluye una capa de seguridad adicional a través de SSL/TLS.

1. **Base de Datos**:

Muchos servidores web interactúan con bases de datos para almacenar y recuperar información dinámica. Ejemplos de sistemas de gestión de bases de datos son MySQL, PostgreSQL y MongoDB.

1. **Lenguajes de Programación**:

Se utilizan para crear el contenido dinámico que se enviará a los usuarios. Algunos lenguajes comunes son PHP, Python, Ruby y JavaScript.

1. **Aplicaciones Web**:

Programas que se ejecutan en el servidor y generan contenido dinámico, como sistemas de gestión de contenido (CMS) y aplicaciones personalizadas.

1. **Capa de Presentación**:

La interfaz que los usuarios ven en sus navegadores, que se compone de HTML, CSS y JavaScript. Esta capa presenta el contenido y permite la interacción del usuario.

Bibliografía**:** PHP Manual. (n.d.). Introducción a PHP: ¿Qué es PHP? Recuperado el 16 de octubre de 2024, de <https://www.php.net/manual/es/intro-whatis.php#example-1>