

# Criptografia PHP:

## Criptografando código fonte com PHP

Este artigo demonstra como criptografar o código fonte utilizando base64 nativa do PHP.

O envio e recebimento de informações sigilosas é uma necessidade desde tempos atrás. Com o avanço da tecnologia, a chegada da internet e a facilidade de transmitir dados de maneira extremamente rápida, a criptografia tornou-se uma ferramenta fundamental nos dias de hoje para o envio e recebimento desse tipo de informação, onde apenas o emissor e o receptor tenham acesso a essas informações.

A transmissão de dados e o armazenamento de informações têm que possuir segurança e para isso ocorrer são utilizadas criptografias que ajudam a embaralhar as informações, evitando que pessoas não autorizadas vejam.

O objetivo deste artigo é demonstrar um algoritmo simples que utiliza base 64, que **criptografa** o código fonte em **PHP**, focando a segurança do código fonte.

A criação do aplicativo será dividida em duas partes: a primeira será o desenvolvimento da página denominada **index.php**, que será a interface do usuário final podendo ele carregar seu arquivos PHP e fazer a criptografia das páginas em tempo real; a segunda parte é denominada **funcao.php** e ela não possui interface com o usuário final, pois é uma classe desenvolvida para pegar o arquivo **PHP** enviado pelo usuário e fazer o processo de criptografia e mudança de nome do arquivo.

## Como é o funcionamento do sistema

Veja na **Figura 1** como será o desenvolvimento do nosso programa.



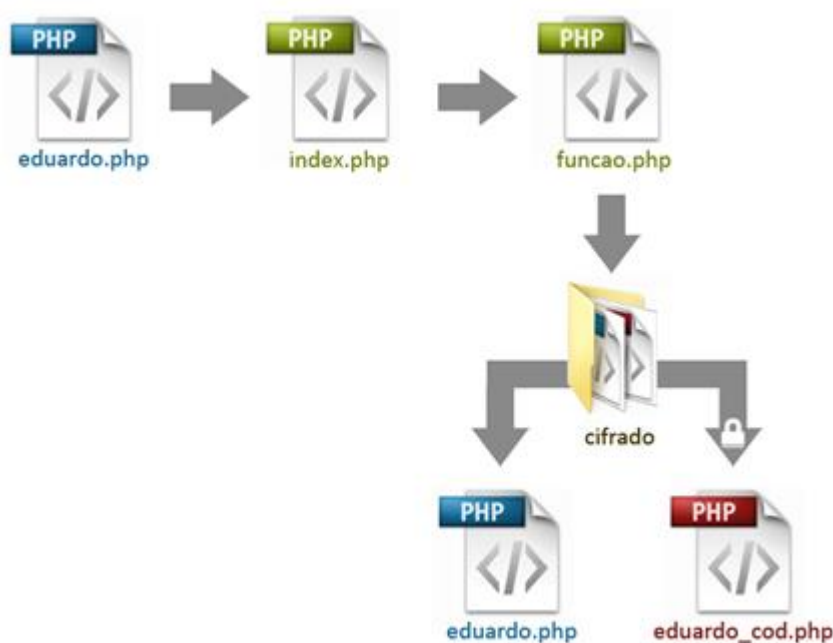
**Figura 1.** Desenvolvimento do programa.

Veja como serão os passos:

1. Usuário faz upload do arquivo em PHP;
2. O sistema faz a criptografia do código usando sistema de Base 64 nativo do PHP;
3. É gerado dois arquivos: o Original e o Cifrado.

## Processo de funcionamento na Prática

Conforme podemos ver na **Figura 2**, o programa gera dois arquivos o original e o cifrado. Na **Listagem 1** temos a classe index.php e na **Listagem 2** temos a classe geradora. Ambas as listagens estão comentadas para que você possa entender o código.



**Figura 2.** Esquema da criptografia

### Listagem 1. INDEX.PHP

```
<?php
/*****
**** Autor: Eduardo Marcos Fiorini ****
**** Data Criação: 07/10/2012 - 02:15 ****
**** Ultima Modificação: 18/12/2012 - 14:42 ****
*****/

?>

<html>
<body>
```

```
<form action="index.php" method="post" enctype="multipart/form-data">
<input type="file" name="Arquivo" id="Arquivo"><br>
<input type="submit" value="Enviar">
<input type="reset" value="Apagar">
</form>
```

```
<?php
if (isset($_FILES["Arquivo"])){
    if (!empty($_FILES["Arquivo"])) {

        //Upload do arquivo.
        $nome_temporario=$_FILES["Arquivo"]["tmp_name"];
        $nome_real=$_FILES["Arquivo"]["name"];
        copy($nome_temporario,"cifrado/$nome_real");

        // Chamar o arquivo com as Funções.
        include_once("funcao.php");
        // Arquivo que irá ser Cifrado.
        $arquivo = "cifrado/$nome_real";
        // Cria a Class Codificar.
        $codificador = new Codificar();
        // Executa a função de Cifragem
        $arquivo_c = $codificador->codificador($arquivo);

        // Se gerar erro apresente o erro.
        if($codificador->erro()){
            $erros=$codificador->pegar_erro();
            echo "<pre>";
            print_r($erros);
        }
        // Se não apresente o resultado.
        else {
            print("<p>O arquivo <u>$nome_real</u> foi cifrado com sucesso, Para testar a
cifragem acesse: <a href='$arquivo_c'><strong>$arquivo_c</strong></a></p>");
        }
    }
}
?>
</body>
</html>
```

## Listagem 2. FUNCAO.PHP

```
<?php
/*****
**** Autor: Eduardo Marcos Fiorini ****
**** Data Criação: 07/10/2012 - 02:15 ****
**** Ultima Modificação: 18/12/2012 - 14:42 ****
*****/
?>
```

```
<?php
// Nivel 0 não Cifra nada.
// Nivel 10 é o Nivel Padrão Recomendado.
// Nivel 30 é o Nivel Maximo acima de 30 pode ocorrer erros e o arquivo ficar muito grande.
define("NIVEL_CIFRAGEM", 10);
```

```
Class Codificar {
    private $arquivo="";
    private $prefixo="cod";
    private $arquivo_c="";
    private $erro=array();
```

```
// Função de Construção.
```

```
public function __construct($prefixo=""){
    if (trim($prefixo)!="") {
        $this->prefixo=$prefixo;
    }
}
```

```
// Função de Codificação do Arquivo.
```

```
public function codificador($arquivo) {
    if (trim($arquivo)=="") {
        $this->erro[]="Nome do arquivo não pode ficar em branco na função ".__FUNCTION__;
        return false;
    }
    if (!is_readable($arquivo)){
        $this->erro[]="Falha ao abrir o arquivo $arquivo na função ".__FUNCTION__;
        return false;
    }
    $this->arquivo=trim($arquivo);

    $ext=end(explode(".", $this->arquivo));
    $pfx=strrpos($this->arquivo, ".");
```

```

$arquivo=substr($this->arquivo,0,$pfx);
$this->arquivo_c=$arquivo_c=$arquivo."_ ".$this->prefixo." ".$Sext;

if(($fp=fopen($arquivo_c,"w+"))===false){
    $this->erro[]="Falha ao abrir o arquivo $arquivo_c ao escrever a função '.__FUNCTION__';
    return false;
}
else {
    fwrite($fp,"<?php \r\n");
    $linha=file_get_contents($this->arquivo);

    $linha=str_replace("<?php","",$linha);
    $linha=str_replace("<?", "", $linha);
    $linha=str_replace("?", "", $linha);
    $linha=trim($linha);
    $linha=$this->codificar_string($linha,NIVEL_CIFRAGEM);
    $linha.="\r\n";
    fwrite($fp,$linha);
    fwrite($fp,"?>");
}
fclose($fp);
return $arquivo_c;
}

```

// Função para codificar o conteúdo do arquivo antes de escrevê-lo.

```

private function codificar_string($string, $levels=""){
    if (trim($levels)=="") {
        $levels=rand(1,9);
    }
    $levels=(int) $levels;
    for ($i=0; $i<$levels;$i++){
        $string=base64_encode($string);
        $string='eval(base64_decode("'.$string.'"))';
    }
    return $string;
}

```

// Função para retornar todos os erros encontrados.

```

public function pegar_erro(){
    return $this->erro;
}

```

// Função para descobrir se havia alguma erro.

```
public function erro(){
    if (count($this->erro)>0) {
        return true;
    }
    else {
        return false;
    }
}
?>
```