



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE COMPUTAÇÃO

SEMINÁRIO PHP

Alunos: Lara Galvani Moura

Fúlvio Taroni Monteforte

Professor: Andrei Rimsa Álvares

Belo Horizonte – MG

Novembro de 2020



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE COMPUTAÇÃO

SEMINÁRIO PHP

Relatório apresentado como parte das exigências da disciplina Linguagens de Programação, do Curso Superior em Engenharia de Computação do CEFET-MG.

Alunos: Lara Galvani Moura

Fúlvio Taroni Monteforte

Professor: Andrei Rimsa Álvares

Belo Horizonte – MG

Novembro de 2020

1. Introdução

O presente relatório tem por objetivo introduzir a linguagem de programação PHP, ressaltando seu contexto histórico, aplicações, exemplos, características, futuro da linguagem e outros aspectos. Além disso, um vídeo com os mesmos assuntos abordados no relatório foi gravado e pode ser acessado através do seguinte link: <https://www.youtube.com/watch?v=fbQLylgxlak&feature=youtu.be>.

A linguagem de programação PHP (Hypertext Preprocessor) é caracterizada por rodar do lado do servidor, além do fato de permitir a criação de sites WEB dinâmicos, possibilitando uma interação com o usuário através de formulários, botões, links, parâmetros de URLs. A partir do uso da linguagem em questão, é importante ressaltar que também é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não deixar o código fonte exposto para o cliente. Dentre os fatores responsáveis pela popularização do PHP é possível citar, principalmente, o fato de ser open source (ou seja, qualquer pessoa pode sugerir e, posteriormente, realizar mudanças na estrutura da linguagem). Além disso, o fato da linguagem ser em código aberto faz com que seja possível qualquer indivíduo programar em PHP e, posteriormente, comercialize o projeto (pois não há restrições de uso ligadas a direitos autorais).

A linguagem foi criada em 1994, por Rasmus Lerdorf, e inicialmente era um simples conjunto de binários Common Gateway Interface (CGI) escrito em C. Em 1995, Rasmus liberou o código fonte para o público, fazendo com que fosse possível que cada desenvolvedor usasse a linguagem de acordo com sua necessidade e, a partir disso, muitos usuários passaram a contribuir com a melhoria do código, reportando falhas e fornecendo correções.

Apesar de ser classificada como uma linguagem de uso geral, o PHP é majoritariamente utilizado no desenvolvimento de aplicações Web, de modo a favorecer a conexão entre a interface do usuário e o servidor. É importante ressaltar que uma aplicação Web normalmente é composta pelo Front-End (parte visual do site, ou seja, interface responsável pela interação do usuário com a aplicação), pelo Back-End (concentração das tarefas a nível de servidor, ou seja, onde a lógica do programa é concentrada, onde são definidas as regras de negócio, onde as consultas ao banco de dados são realizadas) e o Banco de Dados (responsáveis

pelo armazenamento de informações relevantes para a aplicação). É possível realizar muitas tarefas programando em PHP como a implementação de sistemas de fórum, lojas virtuais, sistemas de gerenciamento de conteúdo, dentre outras possibilidades. Mesmo não sendo a linguagem de programação mais moderna do mercado, uma pesquisa do W3Techs realizada em 2020 revelou que o PHP é utilizado por 78,8% dos sites que possuem uma linguagem de programação do lado do servidor reconhecida. Ademais, grandes empresas utilizam a linguagem em questão e, dentre estas, podemos citar o Facebook, Yahoo!, Wikipedia, WordPress, Slack, Google, Baidu, Spotify e Uber.

2. Motivação

Do ponto de vista histórico da linguagem, é possível dizer que o PHP é na verdade o sucessor de um produto chamado PHP/FI. Criado em 1994 e previamente conhecido como “Personal Home Page Tools”, pelo groenlandês Rasmus Lerdorf, o PHP originalmente foi usado para acompanhamento de visitas para seu currículo online. Com o passar do tempo, novas funcionalidades foram implementadas e, diante disso, o código foi reescrito e disponibilizado para o público em 1995, dessa vez fornecendo uma estrutura para que fosse possível desenvolver aplicações Web simples e dinâmicas. O PHP originalmente foi escrito em C e seus scripts eram basicamente um conjunto de binários CGI (Common Gateway Interface).

A linguagem em questão foi desenvolvida com o intuito de ser similar ao C, para que fosse mais fácil de ser utilizada por programadores habituados com C, Perl e outras linguagens similares. Quando inventada, a linguagem tinha variáveis parecidas com as do Perl e também contava com sintaxe HTML embutida.

O PHP passou por diversas versões e atualmente está na 7, que foi lançada em 2015, e trouxe consigo uma grande evolução no quesito velocidade (pois consegue ser até duas vezes mais rápido que o PHP 5, além de ter trazido uma melhora relevante em termos de uso de memória).

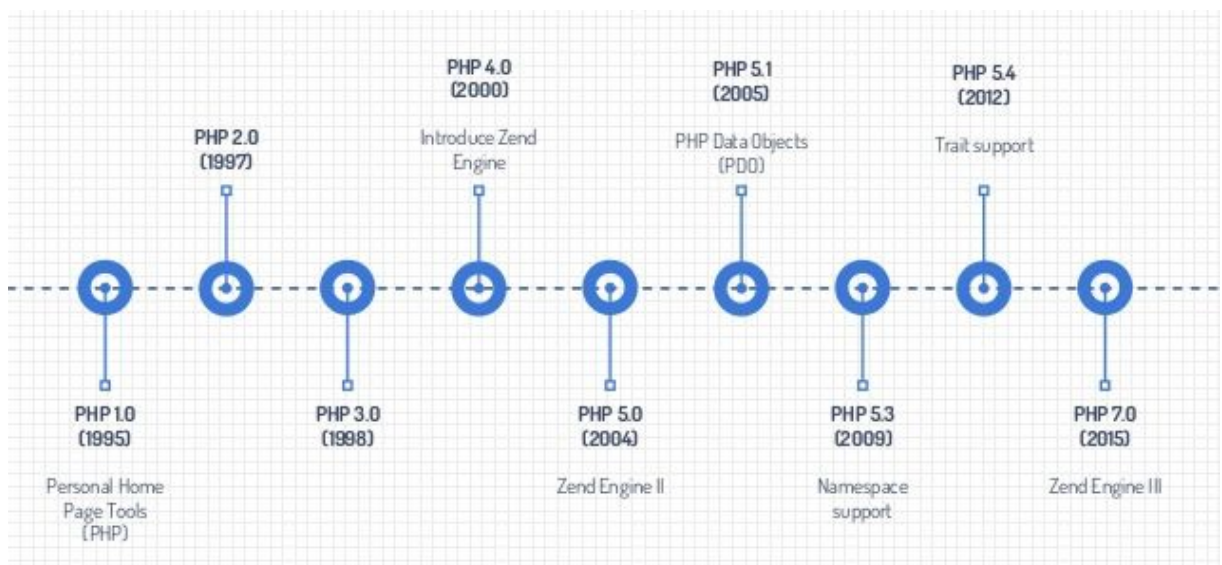


Figura 1: Evolução do PHP ao longo dos anos

3. Características da linguagem

O PHP é uma linguagem caracterizada por ter tipagem fraca e dinâmica. Pode-se dizer que a tipagem dinâmica está ligado ao fato de que a própria linguagem escolhe o tipo do dado automaticamente de acordo com o valor atribuído à variável em tempo de execução dinamicamente. Além disso, tipagem fraca está ligada ao fato de que a linguagem é capaz de realizar conversões automaticamente entre tipos diferentes de dados.

A linguagem em questão possui similaridades com as linguagens C e Perl (em termos de sintaxe), possui operadores aritméticos e lógicos, estruturas de controle e repetição, oferece interfaces para a maioria dos sistemas gerenciadores de Bancos de Dados comerciais, possui suporte para a manipulação de arquivos (possibilidade de realização de qualquer tipo de operação como criação, edição, deleção, entre outros), tem a função de correio eletrônico (ou seja, é capaz de enviar e receber mensagens através de seu próprio servidor Web) e é uma linguagem portátil.

O PHP não foi originalmente criado para ser uma linguagem orientada a objetos (modelo de análise baseado na composição e interação entre diversas unidades chamadas de objetos), entretanto a partir da versão 3 este recurso passou a ser suportado. É importante ressaltar que antes da linguagem oferecer suporte a orientação a objetos, os desenvolvedores PHP utilizavam programação estruturada (técnica que tem como objetivo construir programas claros, legíveis, eficientes e de fácil manutenção) ou orientação a funções alternativamente.

Ademais, o PHP é uma linguagem interpretada, ou seja, o código não é inteiramente convertido para linguagem de máquina de uma só vez (o trabalho de conversão é realizado gradativamente, sempre que uma declaração ou função é executada, por exemplo). O PHP também é uma linguagem de programação em forma de script, e isso significa que as instruções são interpretadas em tempo de execução. No caso específico do PHP, os scripts do lado do servidor são interpretados no servidor, e os scripts do lado do cliente são interpretados pelo aplicativo do cliente.

É válido ressaltar que o código PHP pode ser incorporado ao código HTML (mas também pode ser usado em conjunto com outros sistemas da Web) e um arquivo da linguagem em questão contém tags PHP e termina com a extensão “.php”.

Do ponto de vista fluxo de uma aplicação PHP, pode-se dizer que a linguagem é responsável por toda a lógica necessária para que o navegador consiga exibir alguma informação para o usuário (desde a requisição, passando pelo processo de consulta ao banco de dados se necessário, até o momento em que o HTML pronto deverá ser retornado para que o navegador possa renderizar e exibir para o usuário).

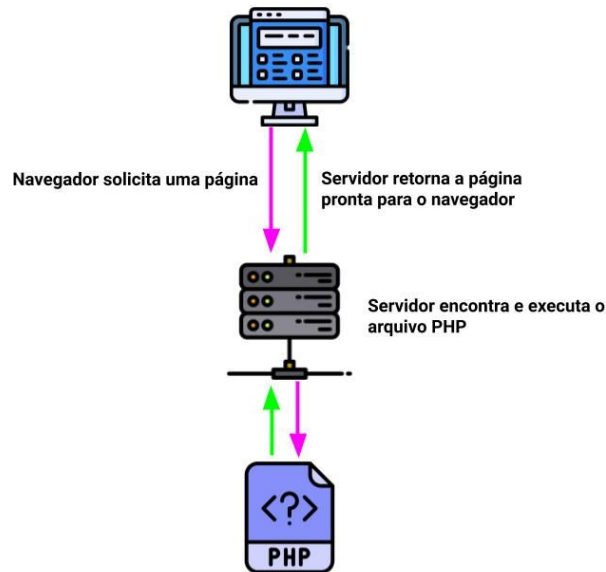


Figura 2: Fluxo de uma aplicação PHP

4. Exemplos

É importante ressaltar que todos os trechos de código estão disponíveis no Github através do seguinte link: <https://github.com/laaragm/PHP-Examples>.

O primeiro exemplo, mostrado abaixo, mostra um *“hello world”* em PHP.



```
<!DOCTYPE html>
<html>
<body>

<?php
    echo "Hello World!";
?>

</body>
</html>
```

Hello World!

Figura 3: Hello World em PHP

Como é mostrada na imagem abaixo, um ponto interessante é que é possível utilizar tags HTML no meio do código em PHP.



```
<!DOCTYPE html>
<html>
<body>

<?php
    echo "<h1>Hello World!</h1>";
?>

</body>
</html>
```

Hello World!

Figura 4: Utilização de tags HTML em PHP

O trecho de código abaixo mostra como declarar e utilizar variáveis na linguagem em questão.


```

<!DOCTYPE html>
<html>
<body>

<?php
    $name = "Lara";
    $name2 = "Fúlvio";
    echo "Relatório sobre PHP feito por:<br><li>"
    . $name . "<br><li>" . $name2 . "<br>";
?>

</body>
</html>

```

Relatório sobre PHP feito por:

- Lara
- Fúlvio

Figura 5: Declaração e utilização de variáveis em PHP

Outro exemplo interessante consiste em uma das características principais do PHP de ser uma linguagem de tipagem fraca e dinâmica, o que faz com que não seja necessário declarar o tipo da variável, além da realização de operações entre tipos diferentes (pois as conversões são realizadas automaticamente).

```

<!DOCTYPE html>
<html>
<body>

<?php
    $var = "5";
    $var2 = 10;
    echo $var + $var2;
?>

</body>
</html>

```

15

Figura 6: Tipagem fraca e dinâmica

Como mencionado anteriormente, o PHP possui similaridades com as linguagens C e Perl, em termos de sintaxe, e isso é mostrado no exemplo abaixo.

```

<!DOCTYPE html>
<html>
<body>

<?php
    $x = 1;
    while ($x <= 10){
        echo "$x <br>";
        $x++;
    }
?>

</body>
</html>

```

12345678910

Figura 7: Similaridade, em termos de sintaxe, com C e Perl

O exemplo a seguir mostra como declarar, inicializar e iterar sobre *arrays* em PHP.

```

<!DOCTYPE html>
<html>
<body>

<?php
    $array = array("lp", "llp", "aeds 2", "laeds
2", "aoc", "laoc");
    echo "Disciplinas: <br>";
    foreach($array as $item) {
        echo "<li> $item";
    }
?>

</body>
</html>

```

Disciplinas:

- lp
- llp
- aeds 2
- laeds 2
- aoc
- laoc

Figura 8: Arrays em PHP

O próximo trecho de código mostra como pode ser realizada a modificação direta de elementos de um *array* dentro de um laço. É importante ressaltar que o “*unset*” é utilizado para quebrar a referência de “*\$value*” para com o último elemento do *array*.

<!DOCTYPE html>	
<html>	
<body>	
<?php	
\$array = array(2, 4, 6, 8, 10);	4
foreach (\$array as &\$value) {	8
\$value = \$value * 2;	12
}	16
unset(\$value);	20
foreach (\$array as \$item) {	
echo "\$item ";	
}	
?>	
</body>	
</html>	

Figura 9: Modificação de valores de um array em um laço

O próximo exemplo mostra um pouco sobre execução de métodos presentes em classes em PHP.

<?php	
class TheTestClass	
{	
function hello()	
{	
echo "Testing class function.";	Testing class function.
}	
}	
\$testClassObject = new TheTestClass;	
\$testClassObject->hello();	
?>	

Figura 10: Execução de métodos presentes em classes

Outro exemplo interessante consiste na construção de strings utilizando interpolação, pois é o mais rápido se comparado com concatenação e heredoc.

```
<?php
    $names = array("Lara", "Fúlvio");
    $sentence = "Dupla de LLP: {$names[0]} e
{$names[1]}";
    echo $sentence;
?>
```

Dupla de LLP: Lara e Fúlvio

Figura 11: Interpolação de strings

Um exemplo interessante é o uso de ternários, que é uma alternativa para substituir o comando *"if"*, em alguns casos, além de apresentar uma escrita bem enxuta.

```
<?php
    $value = 5;
    $secondValue = 5;
    $condition = $value > $secondValue ? 'true' : 'false';
    $result = $condition ?? 'true';
    echo $result;
?>
```

false

Figura 12: Ternários

A forma como comparações entre variáveis são realizadas é um fator onde deve-se prestar muita atenção ao escrever um código, pois dependendo da comparação é possível obter resultados indesejados. O exemplo a seguir retrata justamente os casos mencionados, mostrando principalmente um recurso do PHP que possibilita a comparação entre tipos e valores (e não somente comparação entre valores).

```
<?php
    echo "Comparação de valor: ";
    if (false == 0){
        echo 'true';
    } else {
        echo 'false';
    }

    echo "<br>Comparação de valor e tipo: ";
    if (false === 0){
        echo 'true';
    } else {
        echo 'false';
    }
?>
```

Comparação de valor: true
Comparação de valor e tipo: false

Figura 13: Operadores para comparar valores e tipos

Apesar de PHP não contar com estruturas de dados conhecidas como dicionários nativamente, mesmo assim é possível utilizar estruturas parecidas que contenham chave e valor, e o exemplo seguinte ilustra justamente esse fato.

```
<?php
$array = [
    "lp" => "91",
    "lp" => "93",
];
echo var_dump($array);
echo "<br> {$array['lp']}";
?>
```

```
array(2) { ["lp"]=> string(2) "91" ["lp"]=> string(2) "93" }
91
```

Figura 14: Estruturas de dados com chave e valor

Falando de Orientação a Objetos, é importante ressaltar que as interfaces permitem a especificação de métodos que classes concretas deverão implementar (funcionam como um contrato). Em PHP, para implementar uma interface, deve-se utilizar a palavra reservada *“implements”* e, além disso, para indicar que uma estrutura é uma interface, deve-se utilizar a palavra reservada *“interface”*. O exemplo a seguir mostra um pouco sobre interfaces e como fazer com que classes concretas as implementem, lembrando que todos os métodos da interface devem ser implementados pela classe concreta.

```
<?php
interface ProductInterface {
    public function buy($name, $value);
    public function sell($name, $value);
}

class Product implements ProductInterface {

    public $Name;
    public $Value;

    public function buy($name, $value) {
        echo 'We will be buying the {$name} for {$value}';
    }

    public function sell($name, $value) {
        echo 'We will be selling the {$name} for {$value}';
    }
}
?>
```

Figura 15: Interfaces e implementação

JSON é um formato compacto responsável pela troca de dados de maneira simples e rápida entre sistemas, de modo que os dados possam ser utilizados por qualquer linguagem de programação. Com essa informação, é importante ressaltar que em PHP é possível utilizar o JSON para fazer o “*encoding*” e “*decoding*” de determinado tipo de dados da seguinte forma:

```
<?php
$info = array(
    "Lara"=>20,
    "Fulvio"=>22
);
$encoder = json_encode($info);
echo "{$encoder} <br>";

$decoder =
var_dump(json_decode($encoder));
echo $decoder;
?>
```

```
{"Lara":20,"Fulvio":22}
object(stdClass)#1 (2) { ["Lara"]=> int(20) ["Fulvio"]=> int(22) }
```

Figura 16: Encoding e decoding

Para realizar a leitura e escrita de arquivos, o PHP conta com os métodos “*fopen*” e “*fread*”, com sintaxe semelhante ao C, e o próximo exemplo mostra suas respectivas sintaxes.

```
<?php
$fileName = "test.txt";
$theFile = fopen($fileName, "r") or die("The file
couldn't be opened");
echo fread($theFile, filesize($fileName));
fclose($myfile);
?>
```

```
The file couldn't be opened
```

Figura 17: Leitura e escrita de arquivos

5. Conclusão

Mesmo sendo uma linguagem consideravelmente antiga em relação a outras, PHP continua sendo muito popular na WEB. É uma linguagem relativamente simples e flexível principalmente pelo fato de suportar os paradigmas Funcional e Orientação a Objetos, além do fato de que a tipagem não obrigatória e a comunidade ativa são fatores muito positivos para um crescimento contínuo, em termos de uso, da linguagem. Atualmente a linguagem está em sua versão 7 e há uma previsão para que uma nova versão seja lançada no final de 2020. Algumas das novas funcionalidades que virão no PHP 8.0 incluem um compilador JIT (Just In Time) e a extensão do OPcache (para que não haja necessidade de carregar e analisar scripts em cada requisição), o que ilustra muito bem o crescimento contínuo da linguagem através de novos recursos, correção de “bugs” e extensão da documentação.

Segundo a Northeastern University, em um artigo escrito em 2020, o PHP está entre as 10 linguagens de programação mais populares do mundo, além de também ser uma das que mais existe demanda do mercado de trabalho. Ademais, segundo rankings realizados pelo Stack Overflow, o PHP é uma das 30 linguagens com maior remuneração do mundo, apresentando uma média salarial de U\$39.000,00 por ano (equivalente a R\$218.142,60).

Além de ser utilizado por grandes empresas, estar dentre as linguagens mais requisitadas e mais bem pagas do mundo, possuir constantes melhorias através de novas versões, é importante ressaltar que é uma linguagem gratuita, com alta usabilidade, é compatível com os mais diversos sistemas de bancos de dados e plataformas, é também uma linguagem versátil, simples e prática, além de contar com vários frameworks (como Laravel e Zend, por exemplo) estáveis e seguros responsáveis pelo auxílio no desenvolvimento. Diante disso, é fato que o PHP mesmo sendo uma linguagem antiga está em constante desenvolvimento e com muita demanda para os profissionais da área.

6. Referências

PHP Tutorial. Disponível em: <<https://www.w3schools.com/php/>>. Acesso: 20 de Outubro de 2020.

História do PHP. Disponível em: <https://www.php.net/manual/pt_BR/history.php>. Acesso: 20 de Outubro de 2020.

8 dicas de PHP que você deveria conhecer. Disponível em: <<https://blog.locaweb.com.br/geral/8-dicas-de-php-que-voce-deveria-conhecer/>>. Acesso: 20 de Outubro de 2020.

O que é PHP? Saiba tudo sobre esta linguagem. Disponível em: <<https://www.youtube.com/watch?reload=9&v=va4Wlzo211Y>>. Acesso: 21 de Outubro de 2020.

O que é PHP? Disponível em: <<https://rockcontent.com/br/blog/o-que-e-php/>>. Acesso: 21 de Outubro de 2020.

The most popular Programming Languages to learn in 2020. Disponível em: <<https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/>>. Acesso: 21 de Outubro de 2020.

Usage statistics of PHP for websites. Disponível em: <<https://w3techs.com/technologies/details/pl-php>>. Acesso: 22 de Outubro de 2020.

O PHP morreu? Disponível em: <<https://kinsta.com/pt/blog/o-php-morreu/>>. Acesso: 21 de Outubro de 2020.

O que há de novo no PHP 8. Disponível em: <<https://kinsta.com/pt/blog/php-8/>>. Acesso: 22 de Outubro de 2020.

O PHP e o seu ecossistema nunca estiveram tão fortes. Disponível em: <<https://www.treinaweb.com.br/blog/o-php-e-o-seu-ecossistema-nunca-estiveram-tao-fortes/#:~:text=Algumas%20das%20grandes%20empresas%20que,Wikipedia%2C%20Spotify%2C%20Uber%20etc>>. Acesso: 22 de Outubro de 2020.

Introdução às Linguagens de Programação/PHP. Disponível em: <https://pt.wikiversity.org/wiki/Introdu%C3%A7%C3%A3o_%C3%A0s_Linguagens_de_Programa%C3%A7%C3%A3o/PHP>. Acesso: 22 de Outubro de 2020.

PHP. Disponível em: <<https://www.devmedia.com.br/php-quem-e-voce/1797>>. Acesso: 22 de Outubro de 2020.

O que é PHP e para que serve? Disponível em: <<https://marquesfernandes.com/tecnologia/o-que-e-php-e-para-que-serve/>>. Acesso: 22 de Outubro de 2020.