

Artigo de Paradigmas da Programação sobre PHP

Cibely Araujo e Silva

Instituto Federal de Brasília - Campus Taguatinga

Estrutura de Dados e Algoritmos

Professor: Roberto Duarte Fontes

Abstract. *The purpose of this article is to discuss the PHP programming language, in order to show who it was developed by, how it came about, its main characteristics, development platforms, which paradigm PHP is associated with. Show whether it is readable, reliable and what its cost is, in addition to showing what it is used for, its job market and whether it is important to see it in Bachelor and/or Degree courses.*

Resumo. *O objetivo deste artigo é dissertar sobre a linguagem de programação PHP, a fim de mostrar por quem ela foi desenvolvida, como surgiu, suas principais características, plataformas de desenvolvimento, qual paradigma o PHP está associado. Mostrar se é legível, confiável e qual é seu custo, além disso mostrar para quem é utilizado, seu mercado de trabalho e se é importante vê-lo nos cursos de Bacharelado e/ou Licenciatura.*

1. Introdução

O PHP criado por Rasmus Lerdorf em 1994 era um conjunto de binários Common Gateway Interface (CGI) escrito na linguagem de programação C, Rasmus mandava o link da sua página contendo seu currículo para várias empresas, mas ele gostaria de monitorar quem estava acessando. Por isso, desenvolveu um script que colhia as informações de quem estava acessando sua página. Assim surgindo o Personal Home Page Tools ou PHP Tools e mais futuramente chamado de PHP.

PHP é uma linguagem de script open source para desenvolver aplicações para web e também para criar sites. Por ser uma linguagem de servidor o processamento dela é feito dentro do servidor a cada requisição do usuário, além de ser em código aberto, em que qualquer pessoa pode realizar mudanças na estrutura dessa linguagem.

Ela é capaz de lidar com várias funções backend como, por exemplo, coletar dados de um formulário, modificar base de dados, gerenciar arquivos do servidor, etc.

Para que as pessoas não vejam o código-fonte, basta escrever o código em um arquivo PHP e integrá-lo em um arquivo HTML.

2. Características com exemplos de código

2.1. Velocidade e robustez

As versões mais novas do PHP tornam o processamento no servidor extremamente veloz e consegue executar diversas funções sem alterar em quase nada a velocidade do servidor.

2.2. Estruturado e orientado a objetos

Podemos observar no código abaixo que a linguagem PHP é bem estruturada e orientada a objetos, em que ela possui a construção de uma classe, define seus atributos, define o método e exibe os atributos na classe.

```
<?PHP

#Criação da classe
class Produto{

    #Definição dos atributos da classe
    public $Codigo;
    public $Descricao;
    public $Preco;
    public $Quantidade;

    #Definição do método da classe
    function ImprimeEtiqueta(){

        #Exibe os valores associados aos atributos da classe
        echo 'Codigo: ' . $this->Codigo . "<br /> \n";
        echo 'Descrição: ' . $this->Descricao . "<br /> \n";
        echo 'Preço: R$ ' . $this->Preco . "<br /> \n";
        echo 'Quantidade: ' . $this->Quantidade . "<br /> \n";

    }

}

?>
```

Figura 1. Exemplo de código em php com a classe de produto

2.3. Portabilidade

Não importa a arquitetura utilizada, o mesmo código pode ser utilizado em qualquer servidor, seja no Windows, no Linux ou no MacOS.

2.4. Tipagem dinâmica

Não é necessário especificar o tipo de dados na definição de variáveis, isso é identificado internamente. Ou seja, o compilador irá manipular uma variável de acordo com o valor que está sendo definido.

```

$valor = "Lorem ipsum";
if (is_string($valor)) {
    echo "Oi! Sou uma string!"; // Irá entrar aqui
}

$valor = 123;
if (is_numeric($valor)) {
    echo "Agora sou um número!"; // Irá entrar aqui
}

```

Figura 2. Exemplo de variáveis em PHP

2.5. Sintaxe similar a C/C++ e o Perl

A sintaxe do PHP se parece muito com outras linguagens de programação como a linguagem C.

```

#include <stdio.h>
#include <conio.h>
int main(void)
{
    int contador; //variável de controle do loop

    for(contador = 1; contador <= 10; contador++)
    {
        printf("%d ", contador);
    }

    getch();
    return(0);
}

```

Figura 3. Exemplo de estrutura de repetição em C

```

<?php
    for ($i=0; $i < 5 ; $i++) {
        echo "Laço de número: " . $i;
    }
?>

?>

```

Figura 4. Exemplo de estrutura de repetição em PHP

2.6. Open-source

É uma linguagem de código aberto, ou seja, todos podem usar o PHP gratuitamente e ainda ajudar a criar novas funcionalidades e corrigir bugs.

```
<!DOCTYPE html>
<html>
<head>
<title>Example</title>
</head>
<body>

<?php
    echo "Hi, I'm a PHP script!";
?>

</body>
</html>
```

Figura 5. Exemplo de código em PHP com HTML

3. Plataforma de desenvolvimento e execução

Frameworks são bibliotecas de arquivos que armazenam funções básicas, fornecendo uma fundação para desenvolver projetos de forma eficiente. Existem diversos frameworks que realizam o desenvolvimento e execução de códigos em PHP, a seguir serão apresentados os mais utilizados e suas respectivas características.

3.1. Laravel

Quando falamos de frameworks PHP o Laravel é um dos mais famosos por ter uma sintaxe bem apresentada, de fácil aprendizagem e simples de trabalhar. Suas principais características são: extensão de funcionalidades com complementos; funções nativas para trabalhar com gerenciamento de usuários, cache e mais; Integração com plataformas externas, como o AWS (Amazon Web Services).

3.2. CodeIgniter

Utiliza arquitetura Model View Controller (MVC), ou seja, utiliza componentes diferentes para lidar com tarefas específicas. Além de possuir um manual e ser simples de aprender. Suas principais características são: framework leve e focado em desempenho; aprendizagem fácil e manual detalhado; criar aplicativos escaláveis com a arquitetura MVC.

3.3. Symfony

Disputa com o Laravel e o CodeIgniter em quesito popularidade, ele é altamente flexível, bem como possui um pacote que permite a escolha de funções PHP, porém não possui uma boa performance. Suas principais características são: altamente flexível que permite escolher componentes individuais; funcionalidade de testes nativa; excelente manual de aprendizado.

4. Paradigma associado

Cada linguagem de programação tem um paradigma e com o PHP não é diferente, ele pertence a Programação Orientada a Objetos (POO). Ela é um modelo que aproxima o mundo real do mundo virtual, através da criação e interação entre códigos, objetos, métodos, entre outros. O princípio da POO é que tudo que existe são objetos, eles necessitam ser criados para ter um aspecto inicial, os objetos têm atributos que são suas características individuais e tem também suas funcionalidades que são os métodos. As principais características que definem esse paradigma são as classes, atributos, objetos e métodos.

A seguir, será mostrado o porque o PHP pertence ao paradigma citado anteriormente.

4.1. Classes

A Programação Orientada a Objetos começa a partir da criação da classe, as classes são formas de definir os objetos que serão utilizados e instanciados no código, tendo suas propriedades, métodos e funções que serão utilizadas pelo objeto através da classe.

```
<?php
class Usuario{

}
?>
```

Figura 6. Criação de uma classe em PHP

4.2. Atributos e Métodos

Cada classe possui um atributo que são elementos que definem uma classe, são chamados também de variáveis de classe. O método será executado quando a classe for chamada, é nele que colocamos os elementos para iniciar uma

classe. Na função preparaUsuario ela preenche os atributos da classe, quando a classe for instanciada foram criados métodos para os atributos.

Podemos ver o exemplo abaixo para instanciar e acessar os atributos da classe.

```
<?php
class Usuario{
    public $nome;
    public $cpf;
    public $endereco;
    //construtor da classe
    function Usuario(){
        $this->preparaUsuario();
    }

    function preparaUsuario(){
        $this->nome = "Octavio";
        $this->cpf = "99999999999";
        $this->endereco = "Rua Fulano de Tal número 0 apt 999";
    }
} ?>
```

Figura 7. Definindo atributos da classe

5. Legibilidade, confiabilidade e custo

A legibilidade é a facilidade de ler e compreender programas em uma certa linguagem, logo o PHP pode se encaixar nesse quesito, pois é bastante parecido com outras linguagens de programação como a linguagem C.

O PHP pode se encaixar no quesito confiabilidade, pois é capaz de gerar programas que sua execução reflete exatamente o que foi especificado, como uso de apelidos e manipulação de exceções.

Por ser uma linguagem de código aberto é possível instalar nos sistemas operacionais gratuitamente, ou seja, seu custo de manutenção é baixo.

6. Onde se utiliza/indicado para quê

Utilizado na maioria dos sistemas operacionais, incluindo Linux, variantes do Unix, Microsoft Windows, macOS, RISC OS, etc. O PHP tem suporte para servidores web como Apache, o IIS e muitos outros. A linguagem PHP é indicada para criação de aplicativos, sites, softwares e jogos online, mas está mais ligado ao desenvolvimento de e-commerce, pois eles precisam coletar, processar e analisar informações. Indicada também para quem está começando a programar, por ser uma ferramenta simples de trabalho.

7. Mercado de trabalho

Atualmente, o mercado de trabalho em PHP tem sido cada vez mais reconhecido e valorizado. Essa crescente valorização deve-se às melhorias que a linguagem sofreu ao longo dos anos. Pode citar alguns exemplos de grandes sistemas desenvolvidos em PHP, como o Facebook, Joomla, Magento e WordPress, provando que a linguagem pode ser utilizada em pequenos sites e até em grandes plataformas.

8. Associação com nos cursos de Bacharelado e/ou Licenciatura em Computação

Desse modo, após todas as informações citadas neste artigo podemos analisar que o PHP é importante ser visto nos cursos de Bacharelado e/ou Licenciatura em Computação, pois é uma linguagem de fácil acesso, com sua legibilidade fácil de compreender, tem um baixo custo e roda na maioria dos sistemas operacionais.

Consegue atender tanto programadores iniciantes como programadores mais experientes, além de ter diversos frameworks de PHP ela ainda tem suporte para servidores web. Do mesmo modo, que o mercado de trabalho em PHP tem crescido e sido valorizado, em que pequenas e grandes empresas podem usá-lo para desenvolver sites, aplicativos, jogos, entre outros.