

# Introdução

O desenvolvimento back-end é uma parte essencial do mundo da programação e do desenvolvimento de software. Essa disciplina concentra-se nas entradas invisíveis de aplicativos e sites, garantindo que tudo funcione perfeitamente, mesmo que os usuários finais não vejam diretamente o trabalho realizado nos bastidores. O desenvolvedor **back-end** é o arquiteto por trás da cortina, projetando e construindo sistemas que alimentam a lógica de negócios, o armazenamento de dados e a comunicação com o front-end, que é a parte do software que os usuários interagem.

Para desempenhar seu papel, os desenvolvedores back-end usam uma variedade de linguagens de programação, como **Python, Java, Ruby, PHP e muitas outras**, além de **frameworks e tecnologias** específicas para garantir que os aplicativos sejam robustos e eficientes. Eles também lidam com bancos de dados, modelagem de dados e servidores, garantindo que os dados sejam armazenados com segurança e acessados rapidamente.

Além disso, o **desenvolvimento back-end desempenha um papel crucial na escalabilidade** dos aplicativos, permitindo que eles cresçam para acomodar um número crescente de usuários e funcionalidades. **Através de APIs (interfaces de programação de aplicativos), o back-end permite que diferentes sistemas sejam comunicados**, abrindo portas para integrações com terceiros. No final, o desenvolvimento back-end é a espinha dorsal da experiência do usuário, garantindo que tudo funcione sem problemas e de forma eficiente. É um campo desafiador e em constante evolução que desempenha um papel vital no desenvolvimento de aplicativos e sites modernos.

## Conhecendo a Linguagem PHP

### Habilidades

- Compreender o uso de linguagens de alto nível e sua integração com serviços essenciais para sistemas web;

### 1. Introdução ao PHP

A linguagem de programação **PHP (Hypertext Preprocessor)** foi criada em 1994, por **Rasmus Lerdorf**, com a filosofia **open source**, código aberto voltada para web, sendo que para escrever a mesma **precisa ter conhecimento front end da linguagem de marcação HTML**, fazendo alusão a sua lógica de conhecimento o código escrito em PHP, que vem **derivado do C e do Perl**, vai ser **interpretado por um servidor web**, pre-processado e gerando um HTML para o usuário, **nesse caso o usuário só vai receber dados e HTML**. Com essa linguagem podemos desenvolver inúmeras plataformas para WEB, com mais de 20 anos essa linguagem hoje se encontra **em diversas empresas de grande e médio porte**, podemos até falar que hoje **quase 65% dos sistemas Web em PHP**, sendo hoje uma linguagem fortemente tipada moderna e **segura com a sintaxe POO**. Vamos abordar tudo sobre a linguagem de programação PHP, procedural uma arquitetura muito usada e segura e vamos aplicar boas práticas no código que vai ser visto nessa apostila.

Toda **linguagem procedural vem de um conceito fácil de entender, linha a linha como se fosse uma receita de bolo**, como vimos muito nas aulas de lógica onde um código depende de outro para rodar uma aplicação.

**PHP (PHP, Hypertext pré – processor)** é uma linguagem de programação voltada para desenvolvimento de sistema web, foi criada pelo engenheiro Rasmus Lerdorf, com o objetivo de **guardar dados via formulários**, dentro do escopo Open Source (código aberto), lançado na versão 1.0, em 1994, tendo várias versões e atualização, sendo hoje uma linguagem segura e implantada com a arquitetura orientada ao objeto. **Baseado e escrito em linguagem de programação C.**

Em junho de 1995, Rasmus liberou o seu código fonte do PHP ferramentas para o público, o que permitiu que desenvolvedores usassem da forma como desejassem. Isso permitiu - e encorajou - usuários a fornecerem correções para bugs no código, e em geral, aperfeiçoá-lo.

**1.0 PHP/FI** – Primeira versão feita ainda em C, tinha uma básica GCI mais já estava sendo embutido no HTML, por isso ganhou esse nome de PHP/FI, (Personal Home Page / Forms Interpreter).

**2.0** Já a segunda versão ganhou dois programadores que viram o PHP como uma linguagem do futuro eles se chamavam de Zeev e Andi, ambos Israelense nesta versão ganhou o nome de PHP.

**3.0** Oficialmente agora com o nome PHP, já com script próprios, entrou no mercado de desenvolvimento de sistema WEB.

**4.0** Nessa versão que foi lançada no ano de 2000, há duas grandes novidades, um suporte melhor para conexão com o banco de dados e as primeiras expressões POO. Lembrando que até aqui toda a linguagem era escrita em processo de procedural.

**5.0** Esta versão venho cheio de novas, aplicações POO, o facebook adotou na época essa linguagem para o seu projeto.

**6.0** Esta versão ficou conhecida como a versão de segurança, algo que todos os programadores reclamavam muito, PHP não é seguro, agora sim uma linguagem limpa, segura e organizada.

**7.0** Versão para mobile e mais segurança com API.

**8.0** API para servidores Web, tornando mais rápido nas suas aplicações e atual.

A Linguagem PHP **necessita da linguagem de marcação de texto HTML**, embutida no seu front-end. A linguagem que vamos estudar é procedural ou também conhecida como estruturada o que significa isso, é a maneira como vamos escrever nosso código essa maneira de codificar, é escrever bastante código, linha a linha podemos até comparar com a linguagem de marcação de texto, HTML um texto que não pode ser reutilizado, para quem está iniciando na programação é a maneira mais fácil de iniciar essa trajetória como Back End.

## 1.1. Ambientação e instalação de IDE

A linguagem de PHP, pode ser escrita em qualquer sistema operacional, sendo mais indicada e rápida no Linux por ser nativo, com a linguagem PHP rodando do lado do servidor, precisamos ter um servidor WEB, APACHE ou ISS entre outros. Vamos usar neste livro o servidor web APACHE.

Para escrever e rodar o PHP temos que ter as seguintes ferramentas:

- IDE para escrever;
- Linguagem de programação PHP;
- Servidor Web (no caso vamos rodar o APACHE);
- SGBD (Sistema de gerenciamento de banco de dados);

**IDE** (Integrated Development Environment) ou ambiente de desenvolvimento integrado é um software usado para escrever as linguagens de programação e linguagem de marcação de texto, podemos usar qualquer IDE para essa finalidade, por ser um código aberto.

Vamos usar o IDE NotePad ++, por ser tudo livre para escrever o HTML, CSS e o PHP.

*Siga os passos para a instalação:*

1. **Acesse o site** - <https://notepad-plus-plus.org/>

## 2. Fazer o Download

Clique na guia download e baixe a última versão e escolha a versão do seu sistema operacional sendo 32 ou 64 bits.

## 3. Instalação

Clique no link instalador depois de ter escolhido o tipo de bits que o seu sistema operacional se encontra, faça o download e clique em português e avançar;

**Servidor WEB** é um tipo de serviço que vai receber o código que vamos escrever e enviar o que o cliente solicitou, o código fonte não fica do lado do cliente somente os dados que foi pedido pelo usuário, diferente do HTML, que fica do lado do cliente. Vamos usar o APACHE como servidor de web o mais usado no mundo inteiro.

**SGBD** (Sistema de gerenciamento de dados) é onde vamos guardar os dados do nosso futuro sistema, que vamos desenvolver o SGBD que vamos usar vai ser o MYSQL, mas poderia ser qualquer outro para WEB.

**Linguagem de programação** é o comando da própria linguagem que o IDE, no caso o Servidor Web, que vai interpretar.

Para rodar tudo isso na nossa máquina local, vamos precisar instalar um simulador de servidor Web, que contém APACHE, SGBD e a própria Linguagem, vamos instalar o XAMPP. O XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte às linguagens PHP e Perl.

## Instalando o XAMPP

### 1.Download do XAMPP

<https://www.apachefriends.org/index.html>. Clique na guia download e escolha a última versão do Xampp.

### 2.Instalando o XAMPP

Depois do download, clica no setup que se encontra no canto inferior na esquerda e somente clica em avançar.

Pronto, agora que instalamos o IDE, NOTEPAD ++ e o XAMPP, vamos colocar a mão na massa, agora vamos programar.

### 1.3 Script do lado do cliente

O Script do lado do cliente nada mais é do que o código fonte fica do lado do navegador, assim o servidor WEB, joga todos os códigos juntos com o conteúdo na máquina do cliente, exemplo o HTML, CSS e Javascript.

### 1.4 GUI e suas finalidade do lado do servidor

O Script do lado do servidor e o código gerado pelo o programador no caso o PHP, fica **no servidor WEB**, e esse mesmo servidor manda somente os dados que o usuário solicitou, junto com o HTML, nesse caso nunca os usuários vão ver o seus algoritmos pensados e elaborados pelo programador.

**GUI - Graphical User Interface** (Interface Gráfica do Usuário, em português), é um modelo de interface entre o **humano e a máquina que permite a interação com os dispositivos digitais através de elementos gráficos.**

### 1.5 Sintaxe estrutura e POO

Para que possamos conhecer e escrever de acordo com as boas práticas temos que

compreender a sintaxe de cada linguagem de programação, tipos regras e declarações, e no PHP não fica de fora dessas regras. A sintaxe nada mais é que a organização da escrita que temos que ter na hora de codificar. Hoje pode-se escrever o PHP na arquitetura estrutural e POO, na estrutura escrevemos mais, já na POO utilizamos o que já foi codificado. Vamos entender agora o PHP estrutura, sendo que ambos usam a mesma sintaxe.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Olá Mundo em PHP</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      text-align: center;
    }
    h1 {
      color: #333;
      font-size: 3rem;
      margin-bottom: 20px;
    }
    p {
      color: #666;
      font-size: 1.2rem;
    }
  </style>
</head>
<body>
  <div class="container">
    <?php
      $mensagem = "Olá Mundo!";
      echo "<h1>$mensagem</h1>";
    ?>
    <p>Este é um exemplo de página PHP estilizada com HTML e CSS.</p>
  </div>
</body>
</html>
```

Vale frisar que a linguagem PHP, fica embutido no HTML. Depois de escrever esse código fonte salvamos com a extensão “.php”, no servidor Web e logo em

seguida iniciamos os serviços de servidor, que no nosso caso é o XAMPP.

## RESUMO

No capítulo introdutório, exploramos o PHP, uma linguagem de programação de código aberto criada em 1994 por Rasmus Lerdorf. Inicialmente usada para armazenar dados de formulários web, o PHP evoluiu para uma linguagem poderosa e segura para sistemas web. Destacamos que o PHP é uma linguagem de servidor, que gera páginas HTML dinâmicas no servidor antes de enviá-las aos usuários. Discutimos a evolução do PHP, a importância da integração com HTML, ferramentas essenciais, como o Notepad ++, servidores web como o Apache, SGBD como o MySQL e a sintaxe PHP. Este capítulo preparou o terreno para uma exploração mais profunda dos conceitos e práticas do PHP.

## ATIVIDADES

1. Descreva o propósito e as características de um ambiente de programação orientada a objetos (POO) em PHP. Quais são as vantagens de adotar a POO em comparação com a programação procedural?
2. Dê exemplos de scripts do lado do cliente e do lado do servidor em PHP. Como esses dois tipos de scripts se diferenciam em termos de funcionalidade e localização da execução?
3. Explique o conceito de GUI (Graphical User Interface) e sua relevância no desenvolvimento web. Como a GUI é aplicada em sistemas web baseados em PHP?
4. Destaque a importância da sintaxe na programação PHP. Quais são os principais elementos da sintaxe PHP que os programadores devem dominar para escrever código eficiente e legível?
5. Como podemos definir a linguagem PHP?
6. O que precisamos fazer para codificar a linguagem PHP?
7. Qual é a finalidade do XAMPP?
8. Qual é a extensão do PHP?
9. Qual é a diferença entre comentários de uma linha e comentários de várias linhas em PHP? Como você adicionaria esses comentários a um código PHP para documentação e depuração?
10. Em PHP, o que é uma estrutura condicional e por que ela é importante? Dê um exemplo de uso de uma estrutura "if" em um script PHP para tomar decisões com base em condições.

---

### 1. **\*\*Propósito e Características da Programação Orientada a Objetos (POO) em PHP\*\*:**

A programação orientada a objetos em PHP (POO) permite aos desenvolvedores organizar o código em objetos, que combinam dados e comportamentos relacionados. As características incluem encapsulamento, herança, polimorfismo e abstração. As vantagens de adotar a POO em comparação com a programação procedural incluem modularidade, reutilização de código, facilitação da manutenção e extensibilidade do software.

### 2. **\*\*Exemplos de Scripts do Lado do Cliente e do Lado do Servidor em PHP\*\*:**

Um exemplo de script do lado do cliente em PHP pode ser um script embutido em uma página HTML para validar um formulário antes de ser enviado para o servidor. Um exemplo de script do lado do servidor em PHP pode ser um script que processa os dados enviados por um formulário

HTML e os armazena em um banco de dados. A diferença entre eles é que o script do lado do cliente é executado no navegador do usuário, enquanto o script do lado do servidor é executado no servidor web.

3. **\*\*Conceito de GUI (Graphical User Interface) e sua Relevância no Desenvolvimento Web\*\*:**

A GUI refere-se à interface gráfica que permite a interação entre o usuário e o sistema. No desenvolvimento web, a GUI é aplicada na criação de páginas web com elementos visuais, como botões, formulários e menus, para facilitar a interação do usuário com o site ou aplicativo web. Em sistemas web baseados em PHP, a GUI é geralmente implementada usando HTML, CSS e JavaScript para criar uma experiência de usuário intuitiva e agradável.

4. **\*\*Importância da Sintaxe na Programação PHP\*\*:**

A sintaxe na programação PHP é crucial para escrever código eficiente e legível. Os principais elementos da sintaxe PHP que os programadores devem dominar incluem declaração de variáveis, estruturas condicionais (como if, else), loops (como for, while), funções e classes. Uma sintaxe adequada facilita a compreensão do código por outros desenvolvedores e ajuda na detecção de erros.

5. **\*\*Definição da Linguagem PHP\*\*:**

PHP é uma linguagem de programação de código aberto amplamente utilizada para desenvolvimento de aplicativos web. Ela permite a criação de páginas dinâmicas e interativas, processando dados do lado do servidor e gerando conteúdo HTML para ser exibido no navegador do usuário.

6. **\*\*Codificação da Linguagem PHP\*\*:**

Para codificar em PHP, é necessário um editor de texto ou uma IDE (Integrated Development Environment), como Notepad++, Visual Studio Code ou PHPStorm. Além disso, é preciso um servidor web configurado para interpretar e executar os scripts PHP, como Apache, Nginx ou o servidor embutido do PHP. O código PHP é geralmente salvo em arquivos com a extensão ".php".

7. **\*\*Finalidade do XAMPP\*\*:**

O XAMPP é um pacote de software que inclui um servidor web Apache, um servidor de banco de dados MySQL e as linguagens de script PHP e Perl. Ele é usado para criar um ambiente de desenvolvimento local em um computador, permitindo aos desenvolvedores testar e desenvolver aplicativos web antes de implantá-los em um servidor de produção.

8. **\*\*Extensão do PHP\*\*:**

A extensão de arquivo padrão para scripts PHP é ".php".

9. **\*\*Diferença entre Comentários de Uma Linha e de Várias Linhas em PHP\*\*:**

Em PHP, um comentário de uma linha é precedido por "//", enquanto um comentário de várias linhas é delimitado por "/\*" no início e "\*/" no final. Os comentários são usados para documentar o código e fazer anotações para outros desenvolvedores. Eles são úteis para explicar o propósito de partes do código e para depurar problemas.

10. **\*\*Estrutura Condicional em PHP e sua Importância\*\*:**

Uma estrutura condicional em PHP, como a instrução "if", permite que o código tome decisões com base em condições específicas. Por exemplo, um script PHP pode usar uma estrutura "if" para verificar se um usuário está logado antes de permitir o acesso a determinadas páginas do site. As estruturas condicionais são importantes para controlar o fluxo de execução do programa e realizar diferentes ações com base em diferentes situações.

