# The PHP

Antigamente, PHP significava **Personal Home Page**, mas hoje adicionaram mais um nome no acrônimo, se tornando: “**PHP: Hypertext Processor**”.

Tela de celular

Descrição gerada automaticamente com confiança médiaComo o próprio nome já fala, é um Processador de Hipertexto (HTML), ou seja, a linguagem foi projetada pra pegar o HTML e processar ele de uma forma para que possa ser misturado no HTML lógica de programação e acesso a banco de dados.

Assim como muita outras linguagens, o PHP foi desenvolvido em C.

Uma imagem contendo Esquemático

Descrição gerada automaticamente

O antecessor direto do PHP que realizava uma função similar de criar páginas dinâmicas na web era o **Perl**.

Antes do PHP, **Perl** era amplamente utilizado para processar formulários HTML, gerar conteúdo dinâmico e interagir com servidores web, especialmente através de scripts **CGI** (*Common Gateway Interface*). Perl era uma escolha popular para desenvolvedores que precisavam de uma maneira flexível e rápida de lidar com dados de formulários e gerar páginas dinâmicas, o que são funções centrais do PHP.

Além do Perl, outros métodos e linguagens eram usados na época para manipular conteúdo dinâmico em sites, como **CGI** com C, Python ou mesmo Shell Scripts. No entanto, o **Perl** se destacou por ser mais acessível e flexível para desenvolvedores web, assim como o PHP veio a ser posteriormente.

Agora que já sabemos o que é o PHP, temos também que entender como ele foi criado, e um pouco da história dele.

Por isso alguns comandos do PHP parece uma mistura de Perl com C, por exemplo:

“echo” – vem do perl

“print” – vem do C

# PHP, no passado

Mas tem um porém, ele não criou o PHP para ser lançado como linguagem, e sim criou apenas um modelo de linguagem com scripts para ele mesmo, e jogou seu projeto aberto ao público, o público gostou, pegou os scripts e popularizou o PHP.

A história do PHP realmente começou de uma forma muito informal e pessoal. Quando **Rasmus Lerdorf** criou o que ele inicialmente chamou de **"Personal Home Page Tools"** em 1994, seu objetivo era bem simples: ele queria rastrear visitas ao seu próprio site. Esses scripts eram escritos em **C** e foram feitos para resolver um problema particular que ele tinha.

Com o passar do tempo, ele disponibilizou esses scripts publicamente, permitindo que outras pessoas os utilizassem e modificassem. A ideia era ajudar outras pessoas que quisessem algo semelhante. O projeto era muito mais uma coleção de ferramentas do que uma linguagem formal, e a comunidade teve um papel significativo em seu desenvolvimento inicial.

Ao perceber a popularidade crescente e a adoção por parte de outros desenvolvedores, Rasmus começou a investir mais no projeto. Ele reescreveu o PHP, transformando-o em uma ferramenta mais robusta, que poderia ser usada por mais pessoas para criar páginas web dinâmicas. Esse reescrita foi fundamental para que o PHP passasse de um simples conjunto de scripts para uma linguagem real, com versões controladas e recursos novos, permitindo que mais pessoas colaborassem e contribuíssem.

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamenteFoi nessa fase de reescrita que **PHP/FI** (Personal Home Page / Form Interpreter) nasceu, por volta de 1995. Essa versão foi uma grande melhoria e possibilitou a criação de páginas dinâmicas de uma forma mais ampla e acessível. A partir daí, a comunidade começou a se expandir, contribuindo cada vez mais para o desenvolvimento do PHP.

**Adoção pela Comunidade e Crescimento:**

* Com o crescimento da comunidade e o uso cada vez mais amplo, ficou evidente que a linguagem precisava de uma estrutura mais formal e de recursos adicionais para atender às necessidades de desenvolvimento web que surgiam.
* Por isso, **Zeev Suraski** e **Andi Gutmans** se juntaram ao projeto em 1997 e reescreveram o núcleo do PHP. Esse esforço levou ao lançamento do **PHP 3**, que foi a primeira versão realmente estável e popular da linguagem, e foi nessa versão que o nome "PHP: Hypertext Preprocessor" foi adotado.
* Eles também desenvolveram o **Zend Engine**, que se tornou o coração do PHP, e possibilitou o lançamento do **PHP 4** e versões subsequentes, transformando PHP em uma das linguagens mais populares do mundo para desenvolvimento web.

Portanto, a evolução do PHP é uma história interessante de como uma ferramenta pessoal criada para resolver uma necessidade específica acabou se transformando, com a colaboração da comunidade e reescritas formais.

# Evolução e “Má fama”

O PHP ainda hoje, tem uma “má fama” da sua versão 4 e 5 por falta de ferramentas na linguagem voltadas para paradigmas da programação, que no caso POO.

O PHP realmente enfrentou muitos desafios durante as versões 4 e 5, especialmente no que diz respeito à programação orientada a objetos (POO). Vamos dar uma olhada mais profunda nesse contexto:

## Problemas e Críticas ao PHP 4 e 5

* **PHP 4**: Foi uma versão que tinha suporte muito limitado para POO. Na verdade, a POO no PHP 4 era apenas "superficial". Não havia um verdadeiro modelo de orientação a objetos, e muitos dos conceitos importantes de POO, como herança complexa, encapsulamento adequado e interfaces, eram rudimentares ou não estavam presentes. O PHP 4 se concentrou principalmente em ser uma linguagem de script procedural para a web.
* **PHP 5**: Introduziu um suporte significativo para POO, incluindo **classes, objetos, visibilidade (public, private, protected)**, bem como **interfaces e herança**. No entanto, mesmo com essas melhorias, o PHP 5 ainda tinha algumas limitações. O motor de execução do PHP e o desempenho da linguagem em geral eram áreas problemáticas. Além disso, muitos desenvolvedores ainda tinham uma visão negativa da linguagem, devido à falta de rigor e de ferramentas comuns a linguagens orientadas a objetos mais maduras, como Java e C#.
* **Versão 6**: Nunca foi lançada oficialmente, principalmente devido à complexidade dos recursos planejados. Houve muitas complicações e falhas, especialmente em torno da implementação de suporte nativo para **Unicode**. Esses problemas acabaram levando ao abandono da versão 6.

## PHP 7 - O Retorno com Melhorias Significativas

* **PHP 7** foi um divisor de águas. Ele trouxe melhorias gigantescas em desempenho, tornando o PHP muito mais rápido e eficiente para aplicativos de grande escala.
* **Programação Orientada a Objetos**: PHP 7 introduziu muitos dos recursos necessários para transformar PHP em uma linguagem moderna de desenvolvimento, como **tipos escalares e de retorno**, **exceções melhoradas**, **declarações de tipo**, **operador de coalescência nula** (??) e **operador spaceship** (<=>).
* Essas mudanças ajudaram a transformar o PHP em uma linguagem muito mais robusta, flexível e adequada para desenvolvimento de grandes projetos e sistemas com código limpo e bem estruturado.

## PHP 8 - Uma Linguagem Moderna e Completa

* **PHP 8** trouxe uma série de novos recursos que elevaram o PHP a outro patamar. Alguns dos destaques são:
  + **JIT Compiler (Just-in-Time)**, que melhora o desempenho da execução do código em certas situações.
  + **Aprimoramento de Tipagem**: PHP 8 fortaleceu ainda mais o suporte a tipos, tornando o código mais seguro e previsível.
  + **Atributos**: Uma maneira de adicionar metadados às classes e métodos, semelhante às **annotations** em Java.
  + **Expressões Match**, **Propriedades tipadas**, **Operador de atribuição nula (??=)**, entre outros, que tornam a linguagem mais expressiva e moderna.
* A **Programação Orientada a Objetos** agora é completamente suportada e bastante madura no PHP 8. Recursos como **Herança**, **Polimorfismo**, **Interfaces**, **Traits**, e **Encapsulamento** são implementados de forma robusta. Isso significa que hoje o PHP é uma linguagem poderosa e moderna para desenvolvimento, compatível com paradigmas como POO e até mesmo **Programação Funcional**.

## A Superação da "Má Fama"

Devido à limitação das versões 4 e 5, PHP ganhou uma "má fama" por ser inadequado para grandes projetos e não ser uma linguagem séria de programação orientada a objetos. No entanto, isso mudou radicalmente com as versões 7 e, especialmente, com a versão **8**.

Hoje, o PHP é uma linguagem **madura**, **desempenhável** e capaz de ser utilizada em projetos de grande escala, com POO de forma plena e organizada. Frameworks modernos como **Laravel**, **Symfony**, e **CodeIgniter** também ajudaram a melhorar a percepção da linguagem, trazendo uma estrutura mais limpa e eficiente para o desenvolvimento de software.

Portanto, essa “má fama” do PHP, em grande parte, é um reflexo do passado, e não da realidade atual. O PHP é uma ferramenta poderosa que, com suas últimas versões, está à altura de muitas das linguagens modernas, permitindo o desenvolvimento de aplicações robustas, escaláveis e de alta qualidade.

# Fontes:

**A evolução do PHP :**

https://www.youtube.com/watch?v=TwNmvk-F7E8&list=PLHz\_AreHm4dlFPrCXCmd5g92860x\_Pbr\_&index=4

**“Má fama” do PHP :**

<https://www.tabnews.com.br/pedrorvidal/php-o-bode-expiatorio-favorito-da-internet>

**Por que o mascóte é um elefante ?**

<https://www.youtube.com/watch?v=4kSJOJEi0aQ&list=PLHz_AreHm4dlFPrCXCmd5g92860x_Pbr_&index=5>