

Universidade de Itaúna

Isis Estevan Mendonça

ELIXIR E PHP

Itaúna

2022

PHP	2
1.História:	2
2.Conceito	4
3.Por Que Usar PHP?	4
Vantagens:	4
Desvantagens:	5
4. Onde Usar?	5
5.Paradigma:	5
6. Aplicações:	6

Elixir	6
História	6
Conceito	7
Por que Elixir	7
Vantagens e Desvantagens	8
Vantagens	8
Desvantagens	8
Quem está usando o Elixir?	8
Paradigma	1

PHP

1.História:

Criado em 1994 por Rasmus Lerdorf, a primeira encarnação do PHP foi um simples conjunto de binários Common Gateway Interface (CGI) escrito em linguagem de programação C.

Em Junho de 1995, Rasmus liberou o código fonte do PHP para o público, o que permitiu que desenvolvedores o utilizassem da forma como desejassem. Isso permitiu e encorajou usuários a fornecer correções para bugs no código e, em geral, aperfeiçoá-lo. Em 1997 e 1998, PHP/FI teve o apoio de milhares de usuários ao redor do mundo. Uma pesquisa Netcraft de Maio de 1998 apontou que cerca de 60.000 domínios relataram ter cabeçalhos contendo “PHP”, indicando que o servidor de hospedagem de fato tinha o PHP instalado. Este número pode ser equiparado com aproximadamente 1% de todos os domínios da Internet na época.

PHP 3.0

Dentro da história do PHP, o PHP 3.0 foi a primeira versão que se assemelha com o PHP como existe hoje. O PHP/FI se encontrava ainda ineficiente e não tinha os recursos necessários para prover a aplicação e-commerce que Andi Gutmans e Zeev Suraski, de Tel Aviv, Israel, estavam desenvolvendo para um projeto da universidade. Andi, Rasmus e Zeev decidiram colaborar no desenvolvimento de uma nova e independente linguagem de programação, que foi lançada com novo e removeu a impressão do limitado uso pessoal que o nome PHP/FI 2.0 tinha mantido.

PHP 4.0

Em 1998, logo após o PHP 3.0 ter sido oficialmente lançado, Andi Gutmans e Zeev Suraski começaram a trabalhar em uma reescrita do core do PHP. Os objetivos do projeto eram melhorar a performance das aplicações complexas e melhorar a modularização do código base do PHP. oficialmente lançados em maio de 2000.

PHP 5.0

O PHP 5 foi lançado em julho de 2004. O time de desenvolvimento do PHP incluiu dezenas de desenvolvedores trabalhando em algo relacionado ao PHP e apoiando projetos como PEAR, PECL, documentação e infraestrutura de rede subjacente de bem mais de uma centena de servidores web em seis dos sete continentes do mundo.

HHVM – Hack

Entre a versão 5 e a versão 7 do PHP que temos hoje se passaram 11 anos. Neste meio tempo muitas empresas adotaram o PHP como linguagem principal para desenvolvimento de seus sistemas, como por exemplo, o Facebook. A rede de Mark Zuckerberg, por sua vez, ao ver que se passaram anos sem uma evolução significativa e sem o lançamento de novas versões do PHP, começou a desenvolver a linguagem Hack baseada em PHP e executada em cima do HHVM (HipHop Virtual Machine).

PHP 7

Em dezembro de 2015.....Ei!! E o PHP 6??

Houve um PHP 6. Ou pelo menos uma tentativa dele. Porém, o resultado não foi satisfatório e a versão não foi publicada oficialmente, mas existiam informações sobre essa versão. Logo poderia haver confusão se esta nova versão se chamasse PHP 6. Então, para evitar esses problemas, a equipe do PHP resolveu chamar a nova versão de PHP 7. Em dezembro de 2015, após anos de desenvolvimento, é lançado oficialmente o PHP 7. Entre as novidades mais significativas da versão 7 do PHP está a sua performance, em alguns casos até 9 vezes mais rápida,

2. Conceito

PHP é um Pré-Processador de Hipertexto, é também um subconjunto de linguagens de scripts, porém o PHP costuma ser mais usado para comunicação do lado do servidor (back-end).

Uma linguagem de script serve para automatizar a execução de tarefas num ambiente de tempo de execução especial. Isso inclui dizer para uma página estática (construída com HTML e CSS) para executar ações específicas com regras que você definiu anteriormente.

A ideia inicial era acompanhar o tráfego do site pessoal do criador. Os anos passaram e o criador desenvolvia scripts, o que aumentava os recursos que o site dele tinha.

Resumo: Como mencionamos acima, trata-se de uma linguagem de script criada para comunicações do lado do servidor. Consequentemente, ela é capaz de lidar com várias funções de backend como coletar formulários de dados, gerenciar arquivos do servidor, modificar bases de dados e muito mais.

3. Por Que Usar PHP?

PHP não é a única linguagem de script server-side disponível. Entretanto, ela tem uma vantagem sobre seus competidores se você está rodando um site WordPress. Com ele, você consegue criar aplicações para efetuarem alguma tarefa que o usuário determinar. Essas aplicações são compiladas dentro de um servidor, chamado de server-side, ou, script do lado do servidor. Ainda falando em scripts, com o PHP você pode criar um script e utilizá-lo sem a necessidade de ter um navegador. Você precisa somente do interpretador. Essa função é muito útil para programadores que desejam executar o script para criação de APIs. O recurso é chamado de execução de script via linha de comando.

Vantagens:

- **Fácil Aprendizado** : PHP é baseada em outras linguagens de Devs muito difundidas na área, como C, Java, Pearl e C++. Por isso, a curva de aprendizado é curta e rapidamente é possível começar a programar com ela. Além disso, tem ótima documentação sobre suas funções e exemplos práticos. E mais! Por ter uma ampla comunidade que a utiliza, se você tiver qualquer problema será fácil encontrar soluções na internet.
- **Código aberto**: Por ser uma linguagem de código aberto, você pode usá-la gratuitamente. Na prática, isso também significa que a comunidade está sempre trabalhando para corrigir possíveis bugs e criar novas funcionalidades a cada dia que passa.
- **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.
- **Multiplataforma**: PHP roda em diversos sistemas operacionais, o que facilita em muito sua programação e usabilidade. Ela funciona tanto em Linux e Windows, como em outros sistemas como por exemplo Mac OS, Free BSB, Risc OS entre outros.
- **Compatível com diversos bancos de dados**: PHP pode ser integrado a praticamente todos os bancos de dados do mercado, como MySQL, Oracle, SQLite, DB2, entre outros.

Desvantagens:

- Falta de continuidade: Desde quando foi criado, ele apresentou diversas versões. Isso poderia ser um ponto positivo, todavia, tornou-se uma desvantagem, pois muitas vezes não possuía compatibilidade entre elas, e fez com que vários erros ocorressem.
- Suporte a Datas: Com o PHP é possível fazer diversos tipos de cálculos, mas ainda assim a função não é tão completa e não apresenta tão benéfica. Principalmente se comparar a outros tipos de linguagens de programação, por exemplo, ASP.
- Documentação incompleta: Nas vantagens mostramos que o PHP é uma linguagem de código aberto, e por esse motivo, novas funções primeiro são incluídas na linguagem para depois serem divulgadas. Contudo, isso gera certos “atrasos” para utilizar dessas novas funções, porque os documentos demoram para serem incluídos de forma oficial no site do PHP.

4. Onde Usar?

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (como HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS e provavelmente outros. O PHP também é suportado pela maioria dos servidores web atualmente. Com o PHP, portanto, você tem liberdade de escolha de sistema operacional e de servidor web. Além disso, você pode escolher entre utilizar programação estruturada ou programação orientada a objeto (OOP), ou ainda uma mistura das duas.

5.Paradigma:

PHP é uma linguagem multi paradigma . Ela é considerada uma linguagem imperativa, Linguagens imperativas são caracterizadas por três conceitos: variáveis, atribuições e sequência. Os paradigmas de programação imperativo são focados em instruções exatas que devem ser passadas ao computador na sequência em que serão executadas.

Ela também é considerada orientada a objetos, onde o paradigma orientado a objetos traz boas vantagens, como a reutilização de código, a legibilidade e manutenibilidade do código, a natural modularização e a produção de código mais acessível, já que as estruturas criadas geralmente representam aspectos também existentes no mundo real.

Paradigma funcional: PHP não é uma linguagem funcional, mas algumas técnicas da programação funcional podem ser utilizadas para melhorar o nosso código: melhor legibilidade e tornar mais fácil a manutenção do mesmo são vantagens que adquirimos ao usar o paradigma funcional de programação. PHP, não é 100% funcional e mudamos a maneira de programar para o modo funcional e não é uma tarefa fácil. Podemos começar aplicando alguns desses simples princípios e tornando o nosso código mais simplificado, mais legível, testável e com poucos efeitos colaterais

O interessante do PHP é que ele possui um suporte razoável para todos esses paradigmas, embora alguns sejam mais bem explorados do que os demais. A linguagem é muito conhecida por ser de fácil aprendizado. Vejo um movimento da comunidade PHP em direção ao uso de outros paradigmas, mas menos para programação funcional, o que é uma pena. Enfim, seria bom ver os três paradigmas funcionando em conjunto permitindo soluções melhores em todas as áreas.

6. Aplicações:

E-commerce: As lojas virtuais têm uma necessidade muito marcante, que é a comunicação frequente com bancos de dados robustos e repletos de elementos importantes. Afinal, há muitas fotos, vídeos e outras mídias relacionadas aos produtos que são vendidos. Assim, toda vez que precisa carregar páginas, é necessário se conectar com esses bancos de dados

Plugins WordPress: O WordPress é mais uma grande empresa que baseia toda a programação de suas aplicações na linguagem PHP. Ela é a base dessa plataforma, o que, por si só, já ajuda a entender por que o WordPress é uma escolha cada vez mais recorrente no desenvolvimento de sites. Uma das grandes bases do WordPress, os plugins ajudam diferentes projetos a manter seus sites mais completos e com funcionalidades específicas que suportam a gestão. Assim como a linguagem dos sites, plugins também precisam ter uma comunicação dinâmica e leve com o servidor, e é justamente por isso que PHP é a escolha ideal nesse caso.

Scripts de linha de comando: Também é possível disparar scripts PHP diretamente do terminal, usando linha de comando. Dessa forma, não é necessário um servidor ou navegador para executá-lo. Apenas o interpretador do PHP será necessário para que você possa rodar shell scripts com a linguagem.

Elixir

História

José Valim é o brasileiro criador da linguagem de programação Elixir, um projecto de R&D da Plataformatec, uma subsidiária do nubank. Seus objetivos foram permitir uma maior extensibilidade e produtividade no Erlang VM, mantendo a compatibilidade com ferramentas e ecossistema de Erlang (Linguagem da empresa Ericsson) e abstrai sua sintaxe e nos ajuda a resolver problemas de escalabilidade das nossas aplicações ou seja quando você precisar construir sistemas que sejam tolerante a falhas e que processe muitas informações.

Conceito

Elixir é uma linguagem de programação funcional, concorrente, de propósito geral que executa na máquina virtual Erlang (BEAM). Elixir compila em cima de Erlang para fornecer

aplicações distribuídas, em tempo real suave, tolerante a falhas, non-stop, mas também a estende para suportar metaprogramação com macros e polimorfismo via protocolos.

Por que Elixir

Existem alguns pontos e diferenciais que fazem com que o Elixir se destaque frente a outras linguagens de programação. Entre eles:

- **Execução distribuída**
O código Elixir é executado em pequenas threads, chamadas de processos. Esses processos são totalmente isolados e trocam informações através de mensagens
- **Tolerante a falhas**
Na comunidade erlang existe um conceito chamado 'let it crash' ou, em português, 'deixe falhar'. A única certeza que temos de um código que é executado em produção é que as coisas podem dar errado.
- **Ferramentas para desenvolvimento**
Além das ótimas funcionalidades que a linguagem traz por padrão, por se aproveitar da BEAM, vale destacar também a quantidade de ferramentas que fazem parte do dia a dia de desenvolvimento:
 - Mix - Ferramenta de build;
 - Hex - Gerenciador de pacotes;
 - Iex - Execução interativa do elixir;
 - ex_unit - Framework de testes unitários;
 - ex_doc - Geração de documentação;
 - Phoenix - Framework de desenvolvimento web;
 - Ecto - Interação com banco de dados.

Vantagens e Desvantagens

Vantagens

- **Crescimento e escala** - Se a sua aplicação será acessada por milhares ou milhões de usuários você pode se aproveitar dos benefícios de confiabilidade e escalabilidade da BEAM;
- **Gerenciamento de grande quantidade de requisições simultâneas** - Se você não tem milhões de usuários, mas tem um número menor que faz uso intenso do seu produto, você precisa de um sistema que suporte concorrência;
- **Mantenabilidade do código** - Uma das vantagens ao escrever código em Elixir é conseguir manter o código simples. O resultado disso é que muitas empresas escolhem o Elixir pela possibilidade de ter um codebase de fácil manutenção, desde a correção de bugs até o desenvolvimento de novas funcionalidades.

Desvantagens

Assim como em todas as linguagens o Elixir não é bala de prata. Ela não foi feita para cobrir todos os casos de uso possíveis, por mais que possa parecer. Existem, principalmente, algumas situações em que o uso do Elixir não é o ideal:

- Processamento computacional pesado - Não é algo que o core team do Erlang/Elixir está muito preocupado. Além disso, já existem ótimas ferramentas utilizando Python, Matlab ou R;
- Deep Learning - Existem algumas ferramentas rodando em Elixir, mas existem ferramentas de Deep Learning mais robustas e validadas em Python, por exemplo;
- Mobile apps - Hoje o Elixir não é uma linguagem viável para desenvolvimento mobile. Seria muito legal utilizar o Elixir para desenvolver aplicativos!

Quem está usando o Elixir?

Várias empresas ao redor do mundo já adotaram o Elixir para melhorar sua produtividade ao construir aplicações distribuídas. Alguns exemplos são:

- Pinterest - Suporte a mais de 30k de eventos por segundo e mais de 200 milhões de usuários por mês;
- PepsiCo - Desenvolvimento da plataforma de e-commerce;
- Moz - Diminuíram o tempo de resposta de mais de 800 ms para menos de 50 ms.

Paradigma

Elixir é uma linguagem de programação funcional. Em casos como esses, pode-se fazer melhor uso das cores da CPU e escrever códigos mais curtos e explícitos. O conceito de programação funcional contém três pilares: imutabilidade, funções e código declarativo.