



PHP01 – Introdução a programação em PHP

Sobre esta apostila

Esta e demais apostilas da Íparos são desenvolvidas como material de apoio as aulas ministradas, seja online ou offline.

Como acreditamos na democratização do conhecimento todas as nossas apostilas estão sobre a licença da Creative Commons permitindo seu compartilhamento desde que não seja modificada ou utilizada para fins comerciais.

Por fim lembramos que esta apostila é constantemente atualizada. Sempre consulte o site para novas versões.

Sumário

Sobre esta apostila.....	2
Sobre o PHP.....	5
Performace vs Tempo.....	5
Segurança.....	5
História do PHP.....	6
PHP 6.....	6
Frameworks em PHP.....	7
Resumindo.....	7
CMS em PHP.....	9
Programação estruturada x orientada a objetos.....	10
Editores de PHP.....	11
Como funciona uma linguagem de servidor.....	12
Instalação do PHP.....	13
Testando a instalação do PHP.....	14
Como inserir o PHP na página.....	15
Configurações do PHP.....	16
Configuração do Apache.....	18

Sobre o PHP

Embora exista o PHP-GTK que permite o desenvolvimento de aplicativos desktop com o PHP ele foi construído para ser uma linguagem de programação de servidor e hoje é a mais expressiva no mercado.

A grande vantagem do PHP é que ele permite uma curva de aprendizado muito boa para os iniciantes com uma programação estruturada ou soluções avançadas com orientação a objeto convivendo juntos sem muitos problemas.

Outra característica de sucesso é que sua sintaxe é similar ao C/C++ e sua orientação a objetos similar ao Java facilitando a migração de programadores desktop para o PHP.

Performance vs Tempo

O PHP é uma linguagem de script, ou seja, ela não é compilada para um binário e depois executada. Em resultado a sua performance não é muito boa comparada a outras linguagens como Java porém o tempo necessário para desenvolver o PHP é muito inferior que a grande maioria das linguagens e por isso que ela é hoje um padrão entre sites de pequeno, médio e grande portes.

O desempenho do PHP está muito vinculado ao do Apache (onde o PHP executa) e sua boa configuração é fundamental para um servidor sadio. Porém o problema de desempenho em qualquer site dinâmico é sempre no banco de dados. O banco de dados que vamos utilizar é o MySQL que é muito rápido porém isso fica para depois.

Segurança

O PHP é muito seguro porém a grande maioria dos sites em PHP tem falhas de segurança graves. Isto existe por dois motivos:

1- servidores mal administrados. Quando você assina um plano de hospedagem você deve ver qual é a estrutura do provedor e principalmente o seu conhecimento da tecnologia utilizada. Uma maneira fácil de ver se o provedor possui uma boa política de segurança é saber qual é a versão do PHP utilizada e não aceite nada inferior as versões atuais disponíveis no site do PHP (<http://www.php.net/downloads.php>)

2- programador inexperiente. É muito fácil programar em PHP e nem sempre o material disponível na internet é de boa qualidade. Junto essa combinação faz com que a grande maioria dos programadores de PHP aprenderam a programar de forma insegura e não tem ideia do que estão fazendo de errado.

História do PHP

O PHP nasceu como PHP/FI que significava 'Personal Home Pages Tools / Forms Interpreter' foi criado por Rasmus Lerdorf em 1995 como um conjunto de scripts simples escritos em Perl para sua página pessoal. Mais tarde ele escreveu uma implementação em C maior que entre outras novidades se comunicava com o banco de dados e disponibilizou o código fonte a comunidade.

Em 1997 a **versão 2** já obteve cerca de 50.000 sites (cerca de 1% da internet) porém teve uma vida curta sendo rapidamente substituído pela versão 3 ainda em 1997.

A **versão 3** foi totalmente reescrita por Andi Gutmans e Zeev Suraski e é a primeira do PHP como conhecemos hoje e o mesmo foi renomeado para somente PHP significando 'Hypertext Preprocessor'. A sua principal novidade é a capacidade de receber extensões facilmente o que aumentou em muito a contribuição da comunidade. Em 1998 o PHP 3 já contava com cerca de 10% dos servidores da internet.

O **PHP 4** viu o nascimento da Zend implementando o Zend Engine que aumentou muito a performance do PHP além de melhorar o suporte às extensões e solidificar muitas práticas de programação.

A versão atual, **PHP 5**, foi lançado em 2004 e tem uma melhora sensível da performance com o Zend Engine 2.0 e um sólido suporte à programação orientada a objetos.

PHP 6

Há muito tempo esperado a versão 6 do PHP estava para ser lançada mas depois de uma discussão dentro da comunidade ficou decidido que as modificações existentes não eram significativas para mudar a versão e o que seria o PHP 6 se transformou em PHP 5.3

Embora a grande maioria dos provedores de hospedagem continuem evitando utilizar a versão 5.3 (e por consequência utilizando uma versão depreciada do PHP) pouca coisa mudou de fato se o desenvolvedor já escrevia seus códigos de forma segura. Uma lista mais detalhada se encontra em:

<http://php.net/manual/en/migration53.php>

Frameworks em PHP

Segundo a wikipedia frameworks são uma abstração que une códigos comuns entre vários projetos provendo uma funcionalidade genérica.

Ou seja, basicamente um framework é utilizado para você reutilizar códigos e seu uso começa a ficar muito interessante quando você vai criar sites complexos e com muitas funcionalidades.

Ao adotar um framework você vai estar diminuindo o tempo gasto ao desenvolver e manter seu código, vai ser mais fácil trabalhar em equipe e mais importante: vai impor ao seu código práticas boas e seguras.

O lado ruim de adotar um framework é que primeiro você deve aprender ele e depois seu desempenho. Nenhum framework vai ser mais rápido que o PHP Core.

Temos hoje 3 grandes frameworks no mercado dentro de um universo muito grande cheio de bons frameworks. Vamos a eles:

CakePHP: tem uma boa comunidade brasileira, produção de código muito rápido com o bake, baixa curva de aprendizado, parecido com o rails e funciona em qualquer provedor de hospedagem. Porém é muito restritivo, difícil de manter o código e difícil de testar.

Codelgniter: tem uma boa performance, fácil de aprender e permite a produção rápida de códigos. Embora possua uma boa comunidade a empresa que mantém o framework não tem comprometimento com o PHP e já ocorreu no passado problemas dela com a comunidade.

Zend Framework: possui um código organizado, muitos componentes extras, só funciona com PHP 5.2 ou superior e é muito focado em padrões facilitando a manutenção do código (caso ele seja bem escrito). Porém a documentação confusa, aprendizado lento e tem muito uso errado do framework.

Resumindo

Você quer desenvolver as coisas rápido, baixa curva de aprendizado e muito resultado: Codelgniter com boa performance e CakePHP com uma produção muito rápida.

Você quer alto controle sobre tudo, código bem feito e documentado: Zend Framework.

Além dos frameworks existem também as bibliotecas de códigos. As duas mais conhecidas são a PEAR (<http://pear.php.net/>) e o site PHP Classes (<http://www.phpclasses.org/>)

CMS em PHP

CMS significa Content management system que é inglês para sistema de gerenciamento de conteúdo e é um tipo de framework onde a necessidade de programação para a criação de sites dinâmicos é mínima ou nula. Entre os diversos CMS se destacam:

Sites (portais): Joomla e Drupal

Blogs: WordPress (também pode ser utilizado para sites)

Ensino a distância: Moodle

Programação estruturada x orientada a objetos

Existem dois tipos básicos de programação: estruturada e orientada a objetos.

A programação estruturada é feita para reduzir o nível de complexidade e portanto indicada para soluções pequenas ou para iniciantes e isto ocorre pelos seguintes fatores:

- 1- desenvolvimento contínuo das fases mais fáceis as mais difíceis gradualmente
- 2- o sistema é dividido em módulos cada um com sua função é praticamente independentes
- 3- uso limitado de estruturas de controles para cada módulo

A programação orientada a objetos como o próprio nome diz é orientada a objetos, ou seja, toda a base do código do sistema está voltado a elementos e não a tarefas. Embora a programação orientada a objetos tenha grandes vantagens a programação estruturada em reutilização de código, adaptações futuras, grandes sistemas e em trabalho em equipe ela é bem mais difícil de entender pois o programador só vai ver o código funcionando de fato quando todo ou praticamente todo o sistema estiver pronto.

Editores de PHP

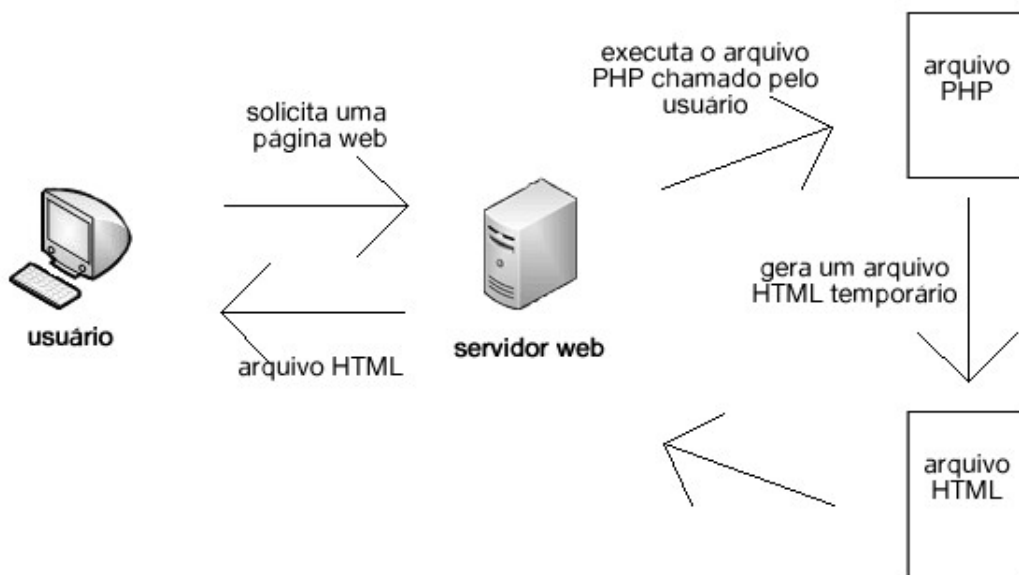
Existem diversos editores para o PHP cada um com suas características, vantagens e defeitos. Não é muito estranho ver programadores PHP utilizando mais de um editor para necessidades diferentes. Entre os diversos editores são mais populares:

Gratuitos e multiplataforma: Aptana, Eclipse (junto com o PHP Development Tools – pdt), Netbeans

Comerciais: Dreamweaver, Zend Studio, PHPEdit, Delphi for PHP

Como funciona uma linguagem de servidor

O maior problema de desenvolvimento de sites é a grande variedade de navegadores e computadores acessando os sites o que limita em muito as possibilidades de programação. A vantagem da linguagem de servidor é que ela exatamente não depende da configuração do seu usuário pois todo o script é processado direto no servidor. Ou seja, o usuário solicita uma página contendo uma programação, o servidor processa esta programação e depois envia ao usuário um arquivo HTML temporário.



São linguagens de servidor: PHP, ASP, ASP.NET, JSP e ColdFusion. Entre essas linguagens a melhor solução é o PHP pois é uma linguagem fácil de aprender, tem o melhor desempenho e é totalmente gratuito. Outra boa solução é o ColdFusion da Adobe que é uma poderosa linguagem de programação e muito fácil de aprender porém tem um custo de desenvolvimento muito elevado.

Instalação do PHP

Para ter o PHP funcionando em um computador para testes você vai precisar:

- 1- Servidor Web
- 2- Instalar o Plugin do PHP no servidor Web
- 3- MySQL
- 4- Administrador do MySQL

No vídeo é ensinado como instalar:

- 1- Apache
- 2- O PHP habilitando pelo menos o plugin do MySQL
- 3- MySQL
- 4- phpMyAdmin

Também existem instaladores automáticos em que os mais famosos são o EasyPHP e o Xampp.

Testando a instalação do PHP

Após instalar o PHP e o MySQL basta ver se eles estão funcionando corretamente. Para tanto basta salvar o seguinte código como teste.php e executar ele através do servidor (ex.: <http://localhost/teste.php>):

```
<?php phpinfo(); ?>
```

A função “phpinfo()” faz com que o PHP faça uma saída de texto com todas as opções de configurações do servidor. Caso isto não apareça o PHP não está funcionando corretamente. Se o servidor “localhost” não for localizado é porque o seu servidor web (IIS ou Apache) não está funcionando corretamente.

Caso você queira testar a comunicação do PHP com o MySQL a melhor maneira é abrir o phpMyAdmin. Se você seguiu corretamente o procedimento de instalação basta abrir a página <http://localhost/mysql>

Como inserir o PHP na página

Você pode inserir o código PHP a qualquer momento em um arquivo HTML. Para que o servidor reconheça que o código naquela parte é em PHP e não HTML ele deve ser marcado da seguinte maneira:

```
<?php código em php ?>
```

O código não só pode ser inserido em qualquer lugar de um arquivo HTML mas como ele pode se alternar com o HTML tal como o exemplo a seguir:

```
<?php if(stripos($_SERVER['HTTP_USER_AGENT'],'microsoft')){ ?>
    <strong>Você está usando o Internet Explorer</strong>
<?php } else { ?>
    <strong>Você não está usando o Internet Explorer</strong>
<?php } ?>
```

Por fim o arquivo HTML que utilizar código em PHP deverá ser salvo com a extensão .php para que o servidor reconheça e processe o código. O PHP faz diferença entre letras minúsculas e maiúsculas e o espaço e quebra de linhas são ignorados na hora de execução sendo necessário o uso do ponto e vírgula para identificar o fim de uma função.

ATENÇÃO

Existem também a tag curta, tag ASP e a tag script como formas de abrir o código PHP sendo que a mais popular é a tag curta:

```
<? código em php ?>
```

Todas essas versões estão marcadas para serem descontinuadas em versões posteriores do PHP por problema de compatibilidade.

Configurações do PHP

O arquivo de configuração do PHP é o `php.ini` e ele pode estar localizado em qualquer local do computador. A página resultado da função `phpinfo()` reflete todas as configurações do `php.ini` e a sua localização em: "Loaded Configuration File"

Abaixo temos as configurações mais importantes deste arquivo:

```
short_open_tag = Off  
asp_tags = Off
```

Quando na opção `on` permitem inserir o PHP com tags curtas ou tags `asp` respectivamente.

```
precision = 14
```

Quantidade de casas decimais possíveis no PHP.

```
max_execution_time = 30
```

Quantidade em segundos que o PHP pode rodar até um timeout.

```
max_input_time = 60
```

Quantidade de tempo que o servidor vai aguardar o recebimento de dados do cliente

```
memory_limit = 128M
```

Limite de memória para o PHP

```
error_reporting = E_ALL | E_STRICT
```

Que tipo de erros vai ser relatado. Quando tiver um `~` na frente significa uma ordem para omitir este erro, exemplo:

`E_ALL`: mostre todos os erros

`E_ALL & ~E_DEPRECATED`: mostre tudo menos os avisos de código depreciado

`E_ALL | E_STRICT`: mostre todos os erros e avisos de padrão de códigos

```
display_errors = On
```

Determina se os erros vão ser mostrados na tela ou não.

```
display_startup_errors = On
```

Erros de inicialização do PHP.


```
log_errors = On
```

Determina se o PHP vai manter os erros em um log ou não.

```
register_globals = Off
```

Determina se o PHP vai trabalhar com variáveis globais. Variáveis globais eram comuns no PHP 3 e a partir do PHP 4 foram consideradas uma grande falha de segurança.

```
post_max_size = 8M
```

Tamanho máximo do formulário (incluindo arquivos anexos) a ser recebido do usuário

```
upload_max_filesize = 2M
```

Tamanho máximo de um arquivo recebido do usuário.

```
date.timezone = America/Sao_Paulo
```

Zona horária padrão. Esta configuração está comentada por padrão (com o ; no início)

```
session.use_only_cookies = 1
```

Força a utilização de cookies para autenticar sessões (valor 1) e diminui o roubo de sessão.

Configuração do Apache

O arquivo de configuração do Apache é por padrão o httpd.conf localizado na pasta conf dentro da raiz do Apache. Dependendo do sistema ou da versão do Apache esse arquivo pode estar dividido em vários outros.

O Apache possui duas configurações interessantes. Uma é permitir que arquivos index.php sejam utilizados como indexadores de pasta:

```
<IfModule dir_module>
    DirectoryIndex index.html
    DirectoryIndex index.php
</IfModule>
```

Outra configuração interessante é o suporte a ler os arquivos .htaccess permitindo assim o uso de URL amigáveis. Caso esteja colocando o site em um provedor de hospedagem é bem possível que o apache já esteja configurado para tanto porém em um servidor local de testes precisamos mudar a sua configuração.

A primeira coisa a ser feita é habilitar o mod_rewrite localizando e descomentando (retirando o #) a seguinte linha:

```
LoadModule rewrite_module libexec/apache2/mod_rewrite.so
```

Agora precisamos habilitar a leitura de fato do arquivo htaccess. No apache tem grupos de comandos para diretórios diferentes, você deve localizar a configuração de diretório onde quer habilitar a URL amigável e aplicar o AllowOverride all

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

<Directory "C:/Program Files/Apache Software
Foundation/Apache2.2/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
```