



**UNIVERSIDADE FEDERAL DA BAHIA**  
**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**  
**MATA76 – LINGUAGENS PARA APLICAÇÃO COMERCIAL**

**TRABALHO SEMESTRAL**  
**ETAPA TEÓRICA**

**PHP**

Aluno: Gabriel Almeida Rocha

Salvador - 2008

## **Introdução**

O PHP é uma linguagem interpretada de propósito geral mas que é especialmente adequada para desenvolvimento Web. É bastante utilizada na criação de páginas Web dinâmicas, mas também oferece suporte ao desenvolvimento de aplicações desktop com o uso da biblioteca GTK. Possui também uma interface de linha de comando. O PHP atua como um “server-side scripting”, ou seja são scripts que rodam do lado do servidor fornecendo conteúdo dinâmico. O PHP age recebendo uma entrada de texto contendo tags PHP’s (geralmente dentro de um arquivo html) e gera uma saída, geralmente conteúdo HTML. Este comportamento parece-se muito com o funcionamento de uma aplicação do tipo CGI, que na verdade foi de onde se originou o PHP (Ver seção 1.0). A partir do PHP 4.0 a entrada é compilada para o bytecode de uma máquina virtual, a Zend Engine. De acordo com [5], “*dentre as características podemos destacar:*

- *Velocidade e robustez*
- *Estruturado e orientação a objetos*
- *Portabilidade – independência da plataforma - escreva uma vez, rode em qualquer lugar;*
- *Tipagem fraca.*
- *Sintaxe similar a linguagem C/C++ e PERL. “*

### **1. Em que contexto a linguagem surgiu?**

O PHP surgiu inicialmente como um simples conjunto de ferramentas úteis para utilização em projetos Web. Em 1993, Rasmus Lerdof, trabalhava numa companhia brasileira como desenvolvedor de aplicações CGI escritas em C. Após algum tempo escrevendo os mesmos aplicativos, Lerdof tinha uma biblioteca com cerca de 30 pequenos programas CGI, e.g um interpretador de formulários, acesso a banco de dados(mSQL) dentre outros. Ele resolveu então juntar todos numa única biblioteca C e criou um “*parser*” (analisador léxico) que pegasse tags de arquivos HTML e então substituisse a saída pelas chamadas às funções correspondentes na biblioteca C. Lerdof então, resolveu expandir este parser para reconhecer estruturas condicionais, de controle (loop’s), funções, etc, dando o nome a este conjunto de scripts de “Personal Home Page Tools”.

Rasmus, então em 1995 resolveu publicar o código-fonte desta ferramenta, agora com o nome de PHP/FI (Personal Home Page/Forms Interpretor) para ela pudesse ser melhorada pelos usuários e usá-la da maneira que bem entendessem.

### **2. Quais os principais nomes (pessoas ou empresas) envolvidos na criação da linguagem ?**

- Rasmus Lerdof - Programador canadense. Criador da primeira versão do PHP. Trabalha

atualmente na Yahoo

- Zeev Suraski – Programador israelita. Criador da versão 3.0 do PHP. Criou junto com Andi Gutmans a Zend Engine que veio com a versão 4.0 e também a empresa Zend Technologies.
- Andi Gutmans - Programador israelita. Criador da versão 3.0 do PHP. Criou junto com Zeev Suraski a Zend Engine que veio com a versão 4.0 e também a empresa Zend Technologies.

### **3. Quantas versões anteriores possuiu a linguagem? Quais as principais evoluções que ocorreram?**

O PHP atual sucede o que chamou-se PHP/FI. Citando [3] “*Ele (PHP/FI) usava variáveis no estilo Perl, interpretação automática de variáveis vindas de formulário e sintaxe embutida no HTML. A sua própria sintaxe era similar a do Perl, porém muito mais limitada, simples, e um pouco inconsistente.*” O FI era mais um CGI que analisava consultas SQL e tornava fácil a criação de formulários e tabelas baseadas nestas consultas.

Algumas alterações ao original PHP/FI 1.0 foram realizadas, principalmente reescrita de algumas funcionalidades, como o acesso a banco de dados atualizando-o para versão 2.0. Neste ponto, o conjunto de ferramentas torna-se uma linguagem simples embutida no código HTML. Em [4], Konstantin Morshnev enuncia: “*...isto eliminou a necessidade de numerosos scripts Perls permitindo ao usuário colocar scripts diretamente no código HTML Isto aumentou a performance geral das páginas pois eliminou o overhead dos fork's dos scripts Perl. Tornou mais fácil também a manutenção das páginas colocando todos os componentes num simples arquivo HTML.*”

#### **3.1. PHP 3.0**

Apesar das inovações, o PHP/FI foi logo descontinuado como o intuito dos esforços serem direcionados para uma nova versão, mais robusta, sendo lançado em Junho de 1998. O significado do acrônimo agora também foi modificado: tornou-se um acrônimo recursivo significando PHP: Hypertext Processor. O PHP 3.0 foi criado por dois programadores israelitas Zeev Suraski e Andi Gutmans, que enxergaram no php uma excelente solução para aplicações e-commerce e decidiram reescrever todo o núcleo do php. Foi oferecida uma maior infra-estrutura para acesso a banco de dados e uma grande extensibilidade, permitindo aos usuários adicionar novos módulos a linguagem.

Outra grande característica adicionada foi o suporte a orientação a objeto, o que ajudou a atrair a atenção de muitos desenvolvedores C++ e Java. Ao final de 1998, milhares de usuários utilizavam php para criação de páginas dinâmicas e cerca de 10% dos servidores Web tinha um módulo php instalado, mostrando sua enorme aceitação.

#### **3.2. PHP 4**

Acompanhando a grande aceitação da versão 3.0 Zeev e Andi continuaram a reescrever o núcleo do PHP culminando na Zend Engine 1.0, uma máquina virtual que processa/interpreta os

scripts php. A Zend Engine é o coração da versão 4.0 destacou-se da versão 3.0 em termos de velocidade e confiabilidade. Dentre outras características podemos destacar a adição de referencias, tipo Booleano, saída bufferizada, suporte para muitos servidores Web, programação orientada a objeto estendida, inclusão da biblioteca para expressões regulares PCRE (Perl Compatible Regular Expressions) entre outras.

Neste ponto, o PHP está sendo usado por centenas de milhares de desenvolvedores e uma estimativa de 20% de domínios da Internet.

### **3.3. PHP 5**

Em Julho de 2004, o PHP 5.0 é anunciado, com a nova Zend Engine 2.0. Dentre as características podemos destacar a programação orientada a objeto melhorada, SQLite embutida, manipulação de exceções usando uma estrutura try..catch, melhores ferramentas XML, iterators entre outras.

## **4. Quais aplicações comerciais já foram desenvolvidas com esta linguagem?**

A lista é bastante extensa, mas cito aqui alguns portais/sites que utilizam php como tecnologia subjacente:

- 1 . Wikipedia: <http://en.wikipedia.org/wiki/Wikipedia>
- 2 . FaceBook: <http://en.wikipedia.org/wiki/Facebook>
- 3 . Yahoo: <http://www.yahoo.com>

## **5. A linguagem foi desenvolvida para ser interpretada ou compilada? “Independente de máquina”? Quais os requisitos mínimos e desejáveis para se executar um programa nesta linguagem?**

O “core”(núcleo) do PHP é a Zend Engine, um máquina virtual que interpreta o código php, que roda em conjunto(como um módulo) do servidor de página, tornando a linguagem portanto independente da plataforma – escreva uma vez, rode em qualquer lugar – sendo bom ressaltar que esta máquina virtual é transparente ao usuário. O PHP suporta a maioria dos SO’s incluindo Linux, Windows, OpenBSD, Mac OS X dentre outros. Tem suporte também a grande maioria dos Web Servers tais como: Apache, Microsoft IIS, Xitami, OmniHTTPd e outros. Os requisitos mínimos para a execução de um aplicativo php é limitado pelos requisitos mínimos de execução de um servidor de página.

## **6. Sintaxe**

(As informações desta seção foram retiradas do Manual do PHP, que vem na distribuição da

linguagem.)

## 6.1. Instruções condicionais

**if-else:** if (expr)  
    comando  
else  
    comando

Ex: if (\$n >= \$maior)  
    \$maior = \$n  
else  
    echo "n eh menor";

**switch:** switch(expressão) {  
    case valor: comando  
    .  
    .  
    .  
    [default]: comando

Ex: switch (estado) {  
    case 0: echo "Aguardando locação..."; break;  
    case 1: echo "Locando..."; break;  
}

Existe uma alternativa ao Switch que é o

**if-elseif:** if (expressão)  
    comando  
elseif  
    comando  
else  
    comando

## 6.2 Instruções de repetição

**for:** for(expre1; expre2;expr3)  
    comando;

Ex: for (\$i = 1; \$i <= 10; &i++)  
    echo \$i;

**while:** while (expre)  
    comando;

Ex: \$i=0;  
    while(\$i <= 10) {  
        echo \$i;  
        \$i++;  
    }

**do-while:** Muito similar ao while, exceto que a condição é testada no final do loop:

do  
    comando;  
while (expr)

Ex: \$i=0;  
do{  
    \$i++;  
    echo (\$i);  
} while(\$i <= 9)

**foreach:** Utilizado para percorrer arrays:

foreach (array\_expression as \$value)  
    comando ou  
foreach (array\_expression as \$key => \$value)  
    comando

Ex: \$Nomes = array('Pedro', 'Joao', 'Antonio', 'Marcos');

```

foreach($Nomes as $nome)
    echo $nome . "<br>";
ou
foreach($Nomes as $pos => $nome)
    echo $pos . " - " . $nome . "<br>";

```

## 6.3. Definição de funções e objetos

### 6.3.1 Funções

A definição de função segue o estilo:

```

function qualquer_funcao($arg_1,$arg_2,/*...,*/$arg_n) {
    // Qualquer código PHP válido
    /* incluso chamadas/definições de funções e definições de
       classes */
    return $retval;
}

```

O PHP possui um tipo especial de função que é chamado de função condicional, onde a declaração da função pode ocorrer dentro de um bloco if, tornando sua “existência” condicional:

```

$a = false;
if ($a) {
    function my_funcao() {
        echo "Minha função";
    }
}
foo();

```

Este código retornaria um erro de chamada a função não definida, o qual não ocorreria caso \$a=true.

O PHP também possui funções recursivas mas a um número limitado ha 100-200 níveis de recursão.

### 6.3.2. Classes e Objetos

O PHP a partir da versão 3.0 passa a oferecer suporte a orientação a objetos. Praticamente todos os recursos de que uma linguagem orientada a objetos oferece foram implementados no PHP, mas ainda não está completa. No anexo 1 segue um código que demonstra algumas das principais características OO implementadas na linguagem.

As classes são definidas através da palavra reservada *class*, seguido do nome da classe e um par de chaves contendo as declarações de membros e métodos da classe. Ex:

```

class Classe
{
    //declaração de membros
    public $var = 'valor de inicialização';

    //declaração de método
    public function mostraVar() {
        echo $this->var;
    }
}

```

Como foi dito, o PHP suporta herança simples (múltipla não é permitida) através da palavra chave *extends* (ver anexo 1). A linguagem também suporta polimorfismo de sobrecarga e

sobrescrita (ver anexo 1).

OBS: Todas as funções e classes em PHP tem escopo global – elas podem ser chamadas fora de uma função mesmo se foi declarada dentro da mesma e vice-versa.

#### 6.4. Comentários

```
// Comentário de uma linha
/* Comentário de multiplas
   linhas */
# Outro comentário de uma linha
```

#### 6.5 Tratamento de exceções

A linguagem possui um controle de exceção similar a muitas outras linguagens utilizando os blocos *try..catch*. Exceções podem ser lançadas utilizando-se o operador *throw*.

*Ex:*

```
function raiz($x) {
    if ($x < 0) {
        throw new Exception("Valor inválido de domínio");
    }
    else return sqrt($x);
}
try{
    echo "raiz de 16 = " . raiz(16) . "<br>";
    echo "raiz de -5 = " . raiz(-5) . "<br>";
} catch (Exception $e) {
    echo "Erro: " . $e->getMessage();
}
```

#### 6.6. Mecanismo de acesso ao banco de dados

O PHP oferece suporte a uma vasta gama de banco de dados como: InterBase, PostgreSQL, Sybase, DB2. Focarei no MySQL por ser talvez o mais utilizado junto com a linguagem - A dupla PHP + MySQL muito provavelmente é a solução mais utilizada para comércio eletrônico. Algumas das funções mais utilizadas para a utilização com o MySQL:

**mysql\_connect(\$db\_host, \$db\_user, \$db\_pass ) :** Abre uma conexão para um servidor MySQL localizado no host “\$db\_host”, com a conta de usuário “\$db\_user” e senha “\$db\_pss” Retorna um handler que referencia a conexão com o banco ou FALSE caso não obtenha sucesso.

**mysql\_select\_db(string \$db\_name):** Escolhe o banco cujo nome é “\$db\_name” dentro do servidor para ser utilizado como banco ativo. Retorna TRUE em caso de sucesso e FALSE c.c.

**mysql\_query (string \$query):** Executa a consulta “\$query”.

**mysql\_fetch\_array (\$result):** Retorna um array contendo os dados da consulta armazenada em \$result.

Ver no anexo 2 um exemplo de utilização do PHP + MySQL.

#### 6.8. Uso de bibliotecas/controles feitos na própria linguagem.

A linguagem oferece uma imensa variedade de bibliotecas para as mais variadas funções como compressão de arquivos: bzip2, zip; manipulação de XML, documentação: PHDoc; bibliotecas de acesso a bancos como o MySQL, cliente FTP e etc, sendo necessário apenas o registro no arquivo de configuração php.ini e o download da biblioteca externa, se necessário.

#### **6.9. Forma e disponibilidade para acesso a bibliotecas/controles desenvolvidos em outras linguagens (método de interação entre sistemas)**

Oferece - ainda em forma experimental, através de uma biblioteca externa - a invocação de métodos de objetos Java, a partir do PHP. É necessário o registro da biblioteca no arquivo de configuração php.ini.

7. A linguagem possui utilitários ou formas de automação/auxílio na documentação?

Através do PHPDoc, muito similar ao JavaDoc. Gera uma documentação em HTML, a partir de comentários especiais inseridos no código. Os comentários podem possuir tags como @author, @param (autor e parâmetros, respectivamente) dentre outras, que serão utilizados pelo aplicativo gerador da página de documentação para a formatação da mesma.

**8. Pode ser utilizado para desenvolver aplicações Web? E “desktop”?**

O PHP é utilizada principalmente para aplicações Web, sendo provavelmente a mais utilizada para processamento server-side, gerando conteúdo dinâmico.

Através da biblioteca gráfica GTK, a linguagem pode ser utilizada para criação de aplicações desktop.

#### **10. Quais os ambientes de desenvolvimento integrados (IDEs) disponíveis?**

1. Zend Studio – A mais conhecida e talvez a mais completa  
Fabricante: Zend Licença: comercial  
Características: Suporte a vários DB's, integração com um servidor cvs, auto-completar.  
Preço: 99 a 1500 dólares. Site: <http://www zend com/en/products/studio>
  2. Dev-PHP – Muito similar ao já conhecido Dev-Cpp para C.  
Fabricante: ? Licença: open-source  
Características: Preview do script, navegação do código, integração com o GTK.  
Site: <http://sourceforge net/projects/devphp/>
  3. NetBeans 6.5 – IDE popular para Java.  
Fabricante: Sun Licença: open-source  
Características: Auto-completar, navegação do código, etc.  
Site: <http://www netbeans org/kb/trails/php html>

**11. A linguagem é padronizada (ANSI/ISO)? Qual a referência (documento, data, ano)?**

Não.

## ANEXO 1

```
<?php
    class Quadrilatero{
        private $base;
        private $altura;

        public function __constructor($b, $a) {
            $this->base = $b;
            $this->altura = $a;
        }
        public function setBase($base) {
            $this->base = $base;
        }
        public function getBase() {
            return $this->base;
        }
        public function setAltura($altura) {
            $this->altura = $altura;
        }
        public function getAltura() {
            $this->altura = $altura;
        }
        public function getArea(){
            $area = $this->base * $this->altura;
            return $area;
        }
    }
    class Quadrado extends Quadrilatero{

        public function Quadrado($lado){
            $this->setBase($lado);
            $this->setAltura($lado);
        }
        public function getArea(){
            $area = $this->getBase() * $this->getBase();
            return $area;
        }
    }
    class Retangulo extends Quadrilatero{

        public function Retangulo($b, $a){
            $this->setBase($b);
            $this->setAltura($a);
        }
    }
    $q = new Quadrado(5);
    $r = new Retangulo(4,7);
    echo "Área do quadrado: " . $q->getArea() . "<br>";
    echo "Área do retângulo: " . $r->getArea();
?>
```

## ANEXO 2

```
<?
// Fazendo a conexão com o servidor MySQL
$conexao = mysql_connect("localhost","root","");
mysql_select_db("Cliente",$conexao);
?>
<table border="1"><tr>
    <td><b>Nome</b></td>
    <td><b>Idade</b></td>
</tr>
<?
$query = "SELECT nome,idade FROM Cliente ORDER BY nome";
$clientes = mysql_query($query,$conexao);
while ($row = mysql_fetch_array($clientes)) {
    ?>
<tr>
    <td><? echo $row['nome']; ?></td>
    <td><? echo $row['idade']; ?></td>
</tr>
<?
}
?>
</table>
```

---

### Referências:

- [1] - “**PHP on Hormones**”, In: <http://itc.conversationsnetwork.org/shows/detail3298.html>
- [2] - “**Interview - PHP's Creator, Rasmus Lerdorf**”, In: <http://www.sitepoint.com/article/phps-creator-rasmus-lerdorf/>
- [3] - “**PHP: Hypertext Preprocessor**”, In: <http://br.php.net/>
- [4] - “**PHP/FI Versio 2.0**”, In: <http://www.moko.ru/doc/php/doc.html>
- [5] - “**PHP History**”, In: [http://www.nusphere.com/php/php\\_history.htm?/](http://www.nusphere.com/php/php_history.htm?/)
- [6] - “**PHP Manual**”, In: <http://www.php.net/manual/en/>