Minicurso PHP

Michel Arruda

Apresentação

- Michel Arruda
- Graduando em Ciência da Computação na UFRRJ
- Desenvolvedor web na Fundação Coppetec

O que é ?

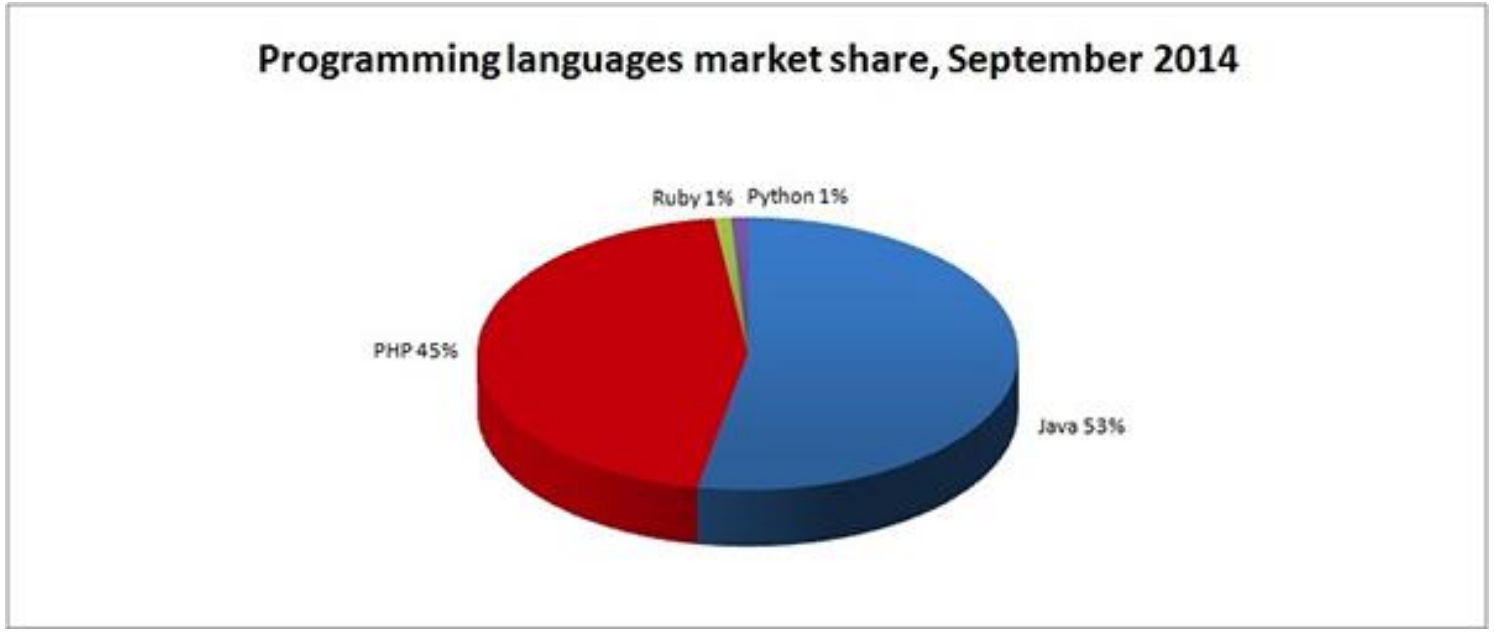
- Acrônimo para "PHP: Hypertext Preprocessor"
- Linguagem de script altamente utilizada
- Open Source
- Scripts PHP são executados no servidor
- É gratis para download e uso

Motivação para aprender

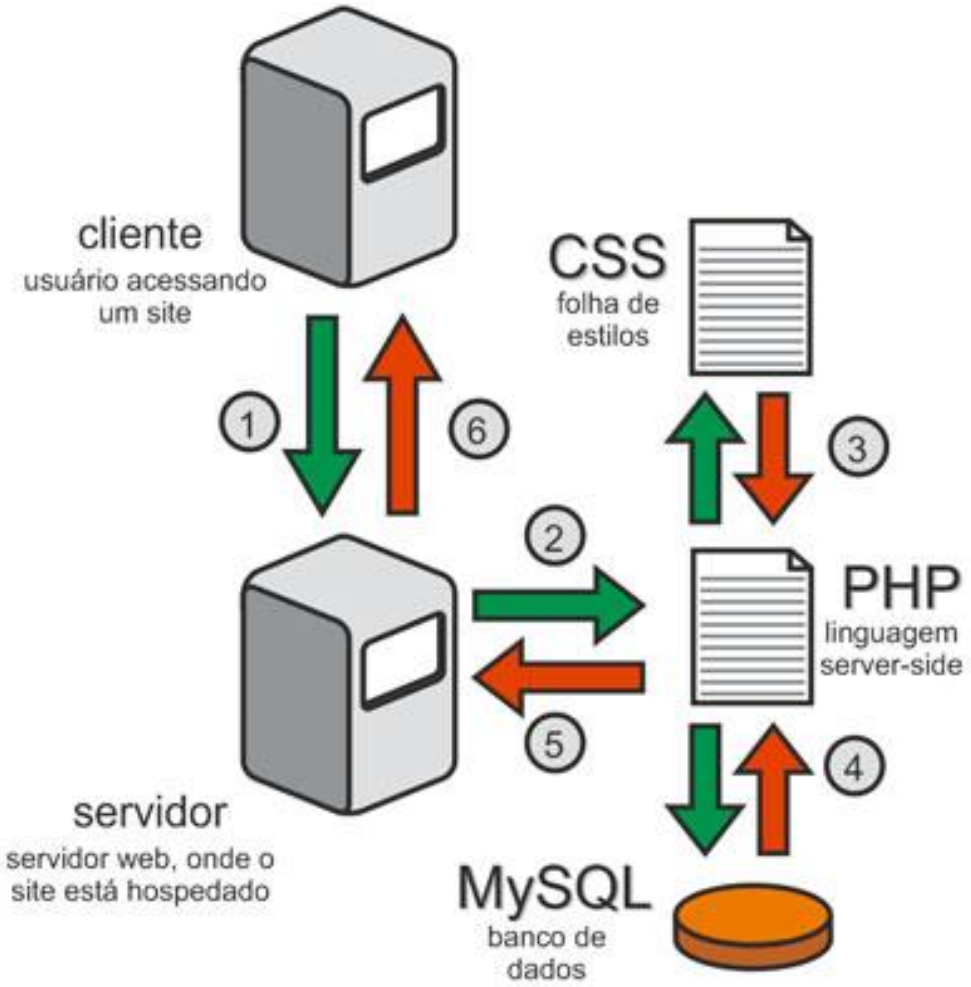
- Executa em diversas plataformas (Windows, Linux, Mac OS X)
- Compatível com a maioria dos servidores utilizados (Apache, IIS)
- Suporta a utilização da maioria dos bancos de dados existentes
- É gratis
- Fácil de aprender e eficiente no Server Side

Motivação para aprender

- Grande fatia do mercado WEB



Arquitetura cliente servidor



Como instalar

- Instalar servidor (Apache, IIS, Nginx)


- Instalar PHP
- Instalar Banco de Dados (MySQL, PostgreSQL)

Hello World

```php
<?php
    echo "Hello World !!";
?>
```

Comentários

```php
<?php
        /* Primeiro Script PHP */

        //Exibição do Hello World
        echo "Hello World !!";
?>
```

Case-Sensitive

- Todas as palavras-chave (por exemplo, if, else, enquanto, eco, etc.), classes, funções e funções definidas pelo usuário não são Case-Sensitive.

```php
<?php
        ECHO "Hello World!<br>";
        echo "Hello World!<br>";
        EcHo "Hello World!<br>";
?>
```

Case-Sensitive

- Todas variáveis são case-sensitive

```php
<?php
        $color = "red";

        echo "My car is " . $color . "<br>";
        echo "My house is " . $COLOR . "<br>";
        echo "My boat is " . $coLOR . "<br>";
?>
```

Variáveis

- Declarando

```php
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

Variáveis

- Começa com o sinal $, seguido do nome da variável
- Começar com uma letra ou o caractere sublinhado
- Não pode começar com um número
- Contém apenas caracteres e sublinhados alfanuméricos (AZ, 0-9 e _)
- São case-sensitive

Escopo de variável

- O escopo de uma variável **é a parte do script em que a variável pode ser referenciado/usado**.
- PHP tem três escopos diferentes variáveis:
  - Local;
  - Global;
  - Estático

Escopo de variável

```php
<?php
        //global
        $x = 5;

        function myTest() {
                //local
            $x = 10;
            echo "<p>Variavel x dentro da funcao: $x</p>";
        }

        myTest();

        echo "<p>Variavel x fora da funcao: $x</p>";
?>
```

Escopo de variável

```php
<?php
        $txt = "Hello world!";
        //global
        $x = 5;
        $y = 10.5;

        function myTest() {
            //local
            $x = 10;
            echo "<p>Variavel x dentro da funcao: $x</p>";
            //global
            global $x;
            echo "<p>Variavel x de fora da funcao: $x</p>";
        }

        myTest();
?>
```

Escopo de variável

- Variáveis globais ficam armazenadas em um array chamado $GLOBALS[índice].
- O índice = nome da variável.
- Acessível a partir de dentro de funções
- Pode ser utilizado para atualizar as variáveis globais diretamente.

```php
<?php
    //global
    $x = 5;
    $y = 10.5;

    function myTest() {
        //global atualizada
        $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
        echo "<p>Variavel global Y: ".$GLOBALS['y']."</p>";
    }

    myTest();
?>
```

Escopo de variável

- Static
- Normalmente, quando uma função é executada, todas as variáveis são excluídas. No entanto, às vezes queremos uma variável local não deve ser excluída. Precisamos dela para um outro trabalho. Para fazer isso, use a palavra-chave estática quando você declarar a variável:

```php
<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}
?>
```

Tipos de Dados

- String
  - $x = "Hello world!";
  - $y = 'Hello world!';
- Integer
  - $x = 5985;
- Float
  - $x = 10.365;
- Boolean
  - $x = true;
  - $y = false;
- Array
  - $cars = array("Volvo","BMW","Toyota");
- Object
  - class Car {
    function Car() {
    $this->model = "VW";
    }
    }
  - $herbie = new Car();
- NULL
  - $car = null;
- Resource
  - É o armazenamento de uma referência às funções e recursos externos
  - Ex: Tipo de dados de recursos é uma chamada de banco de dados.

Exibição de Dados

- echo e print
  - echo "Hello world!
";
  - print "Hello world!
";

- print_r
  - print_r(array);

- var_dump($variavel);

Strings

- strlen
  - echo strlen("Hello world!"); // Saída: 12
- str_word_count
  - echo str_word_count("Hello world!"); // Saída: 2
- strrev
  - echo strrev("Hello world!"); // Saída: !dlrow olleH
- strpos
  - echo strpos("Hello world!", "world"); // Saída: 6
- str_replace
  - echo str_replace("world", "Dolly", "Hello world!"); // Saída: Hello Dolly!

Constantante

- define(*name*, *value*, *case-insensitive*)
  - define("GREETING", "Welcome to W3Schools.com!");
    - echo GREETING;
  - define("GREETING", "Welcome to W3Schools.com!", true);
    - echo greeting;

Operadores

- Operadores aritméticos

Operador	Nome	Exemplo
+	Adição	$x + $y
-	Subtração	$x - $y
*	Multiplicação	$x * $y
/	Divisão	$x / $y
%	Módulo	$x % $y
**	Potenciação	$x ** $y

- Operadores de atribuição

Atribuição	Mesmo que..
x = y	x = y
x += y	x = x + y
x -= y	x = x - y
x *= y	x = x * y
x /= y	x = x / y
x %= y	x = x % y

Operadores

- Operadores de comparação

Operador	Nome	Exemplo
==	Igual	$x == $y
===	Identico (Mesmo tipo)	$x === $y
!=	Diferente	$x != $y
<>	Diferente	$x <> $y
!==	Não idêntico	$x !== $y
>	Maior que	$x > $y
<	Menor que	$x < $y
>=	Maior igual que	$x >= $y
<=	Menor igual que	$x <= $y

- Operadores de Incremento / Decremento

Operador	Nome
++$x	Pré-incremento
$x++	Pós-incremento
--$x	Pré-decremento
$x--	Pós-decremento

Operadores

- Operadores lógicos

Operador	Nome	Exemplo
and	And	$x and $y
or	Or	$x or $y
xor	Xor	$x xor $y
&&	And	$x && $y
||	Or	$x || $y
!	Not	!$x

- Operadores de string

Operador	Nome	Exemplo
.	Concatenação	$txt1 . $txt2
.=	Concatenação com atribuição	$txt1 .= $txt2

Operadores

- Operadores de Array

Operador	Nome	Exemplo
+	União	$x + $y
==	Igual	$x == $y
===	Identico	$x === $y
!=	Diferente	$x != $y
<>	Diferente	$x <> $y
!==	Não identico	$x !== $y

Estruturas condicionais

- **If**

  if (*condicao*) { bloco }
- **if...else**

  if (*condicao*) { bloco; }
  else { bloco; }
- **if...elseif....else**

  if (*condicao*) {
  bloco;
  } elseif (*condicao*) {
  bloco;
  } else {
  bloco;
  }

- **Switch**

  switch (*n*) {
  case *label1:*
  bloco;
  break;
  case *label2:*
  bloco;
  break;
  case *label3:*
  bloco;
  break;
  default:
  bloco;
  }

Estruturas de repetição

- While

  while(condicao){ bloco };

- Do... While

  do { bloco } while (condicao);

- For

  for(inicia contador; condição; incremento){ bloco }

- Foreach

  foreach ($array as $valor) { bloco; }

Vetores e matrizes

- Arrays indexados – chaves numéricos

```
$cars = array("Volvo", "BMW", "Toyota");
```

- Arrays associativos - chaves nomeadas

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

- Array Multidimensional – Array contendo array

```
$cars = array ( array("Volvo",22,18),
                array("BMW",15,13),
                array("Saab",5,2),
                array("Land Rover",17,15)
              );
```

Vetores e matrizes

```php
<?php
$cars = array("Volvo", "BMW", "Toyota");
$arrlength = count($cars);

for($x = 0; $x < $arrlength; $x++) {
    echo $cars[$x] . "<br>";
}

$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value . "<br>";
}

?>
```

Funções e procedimentos

```php
<?php
function nomeFamilaNascimento($fnome, $ano) {
    echo "$fnome. Nascido em $ano <br>";
}

nomeFamilaNascimento("Silva", "1957");

function setAltura($minhaAltura = 50) {
    echo "A altura é : $minhaAltura <br>";
}

setAltura(350);
setAltura(); // Setará o padrão
?>
```

Formulários HTML + PHP

```
1 <html>
2 <body>
3         <form action="welcome.php" method="get">
4                 Name: <input type="text" name="nome"><br>
5                 E-mail: <input type="text" name="email"><br>
6                 <input type="submit">
7         </form>
8 </body>
9 </html>
```

Get e Post

- Ambos GET e POST criam uma matriz (por exemplo, array (chave => valor, chave2 => valor2, chave3 => valor3, ...)).
- Esta matriz contém pares de chave/valor, onde as chaves são os nomes dos controles de formulário e os valores são os dados de entrada do usuário.
- Ambos GET e POST são tratados como $ _GET e $ _POST.
- Estes são superglobal
- $ _GET É um array de variáveis passadas para o script atual através dos parâmetros de URL.
- $ _ POST é um conjunto de variáveis passadas para o script atual através do método HTTP POST.

Criando o Welcome.php

```php
<?php
    echo "Welcome ".$_GET["nome"]."<br>";
    echo "Your email address is: ".$_GET["email"];
?>
```

Validação de dados de formulário

```php
<?php
        $nome = $_GET["nome"];
        $email = $_GET["email"];

        if (empty($nome)) {
                echo "Por favor, preencha o seu nome.";
        } else if(empty($email)) {
                echo "Por favor, preencha o seu email.";

        } else {
                echo "Welcome ".$nome."<br>";
                echo "Your email address is: ".$email;
        }
?>
```

Sessões

- Armazenam informações de usuário para ser usado em várias páginas
- Por padrão, as variáveis de sessão duram até que o usuário feche o navegador.
- Armazenam informações sobre um único usuário, e estão disponíveis para todas as páginas em um único aplicativo.

Sessões

```php
<?php
// Iniciando sessao
session_start();

// Seta variaveis de sessão
$_SESSION["cor"] = "Preto";
$_SESSION["animal"] = "Cachorro";
echo "Variaveis setadas";
?>
```

Sessões

```php
<?php
session_start();

echo "Cor: " . $_SESSION["cor"] . ".<br>";
echo "Animal: " . $_SESSION["animal"] . ".";
?>
```

Retorno de erro no formulário

```
1  <html>
2  <body>
3          <form action="welcome2.php" method="get">
4                  Name: <input type="text" name="nome"><br>
5                  E-mail: <input type="text" name="email"><br>
6                  <input type="submit">
7          </form>
8          <div>
9          <?php
10                 session_start();
11                 if(isset($_SESSION["erro"])) {
12                         echo $_SESSION["erro"];
13                 }
14         ?></div>
15
16 </body>
17 </html>
```

Retorno de erro no formulário

```php
<?php
        $nome = $_GET["nome"];
        $email = $_GET["email"];

        session_start();

        if (empty($nome)) {

                $_SESSION['erro'] = "Por favor, preencha o seu nome.<br>";
                header("location: formularioErro.php");

        } else if(empty($email)) {

                $_SESSION['erro'] = "Por favor, preencha o seu email.<br>";
                header("location: formularioErro.php");

        } else {
                echo "Welcome ".$nome."<br>";
                echo "Your email address is: ".$email;
                unset($_SESSION['erro']);
        }
?>
```

```html
<html>
<body>
	<form action="10.2 - welcomeCompleto.php" method="post">
		Name: <input type="text" name="nome"><br>
		E-mail: <input type="text" name="email"><br>
		Professor: <input type="text" name="professor"><br>
		Nota: <input type="text" name="nota"><br>
		<input type="submit" name="inserir" value="Inserir" />
		<input type="submit" name="excluir" value="Excluir" />
		<input type="submit" name="selecionar" value="Selecionar" />
		<input type="submit" name="alterar" value="Alterar" />
	</form>

	<div>
	<?php
		session_start();
		if(isset($_SESSION["erro"])) {
			echo $_SESSION["erro"];
		} elseif (isset($_SESSION["sucesso"])) {
			echo $_SESSION["sucesso"];
		}
	?>
	</div>
</body>
</html>
```

Include x Require

- Incluem um arquivo dentro do código PHP.
- Require exige a inclusão do arquivo.
- Include não exige a inclusão e se arquivo não encontrado, o script continuará executando.

- include ‘arquivo.php‘
- include_once ‘arquivo.php’
- require ‘arquivo.php‘
- require_once ‘arquivo.php’

Orientação a objetos

```php
<?php
class Pessoa {

        private $nome = "Manoel";

        function showNome(){
                echo $this->nome;
        }
}

$pessoa = new Pessoa();
$pessoa->showNome();
```

Construtores e destrutores

```php
<?php
class Pessoa {

        private $nome;

        function __construct($nome){
                $this->nome = $nome;
        }

        function __destruct(){
                echo "<br>Destruido<br>";
        }

        function showNome(){
                echo $this->nome;
        }
}

$pessoa = new Pessoa("Manoel");
$pessoa->showNome();
```

Herança

```php
<?php
class Pessoa {

        private $nome;

        function __construct($nome){
                $this->nome = $nome;
        }

        public function getNome(){
                return $this->nome;
        }
}
```

```php
class Professor extends Pessoa {

        private $salario;

        function __construct($nome, $salario){
                parent::__construct($nome);
                $this->salario = $salario;
        }

}

$aluno = new Aluno("Manoel", 10);
$aluno->showAtributos();
?>
```

Herança

```
15 class Aluno extends Pessoa {
16
17         private $nota;
18
19         function __construct($nome, $nota){
20                 parent::__construct($nome);
21                 $this->nota = $nota;
22         }
23
24         public function showAtributos(){
25                 echo $this->getNome()." - ".$this->nota;
26         }
27
28 }
```

Polimorfismo

```php
<?php
class Pessoa {

        private $nome;

        function __construct($nome){
                $this->nome = $nome;
        }

        public function showAtributo(){ echo $this->nome; }
}

class Aluno extends Pessoa {

        private $nota;

        function __construct($nome, $nota){
                parent::__construct($nome);
                $this->nota = $nota;
        }

        public function showAtributo(){ echo $this->nota; }
}

$aluno = new Aluno("Manoel", 10);
$aluno->showAtributo();
```

Abstração

```php
<?php
abstract class ClasseAbstrata
{
    // Força a classe que estende ClasseAbstrata a definir esse método
    abstract protected function pegarValor();
    abstract protected function valorComPrefixo( $prefixo );

    // Método comum
    public function imprimir() {
        print $this->pegarValor();
    }
}

class ClasseConcreta1 extends ClasseAbstrata
{
    protected function pegarValor() {
        return "ClasseConcreta1";
    }

    public function valorComPrefixo( $prefixo ) {
        return "{$prefixo}ClasseConcreta1";
    }
}
```

Encapsulamento

```php
<?php
class Pessoa {

        public $nome;
        protected $disciplina;
        private $cpf;

        public function showNome(){ echo $this->nome; }
        protected function showDisciplina(){ echo $this->disciplina; }
        private function showCpf(){ echo $this->cpf; }
}
?>
```

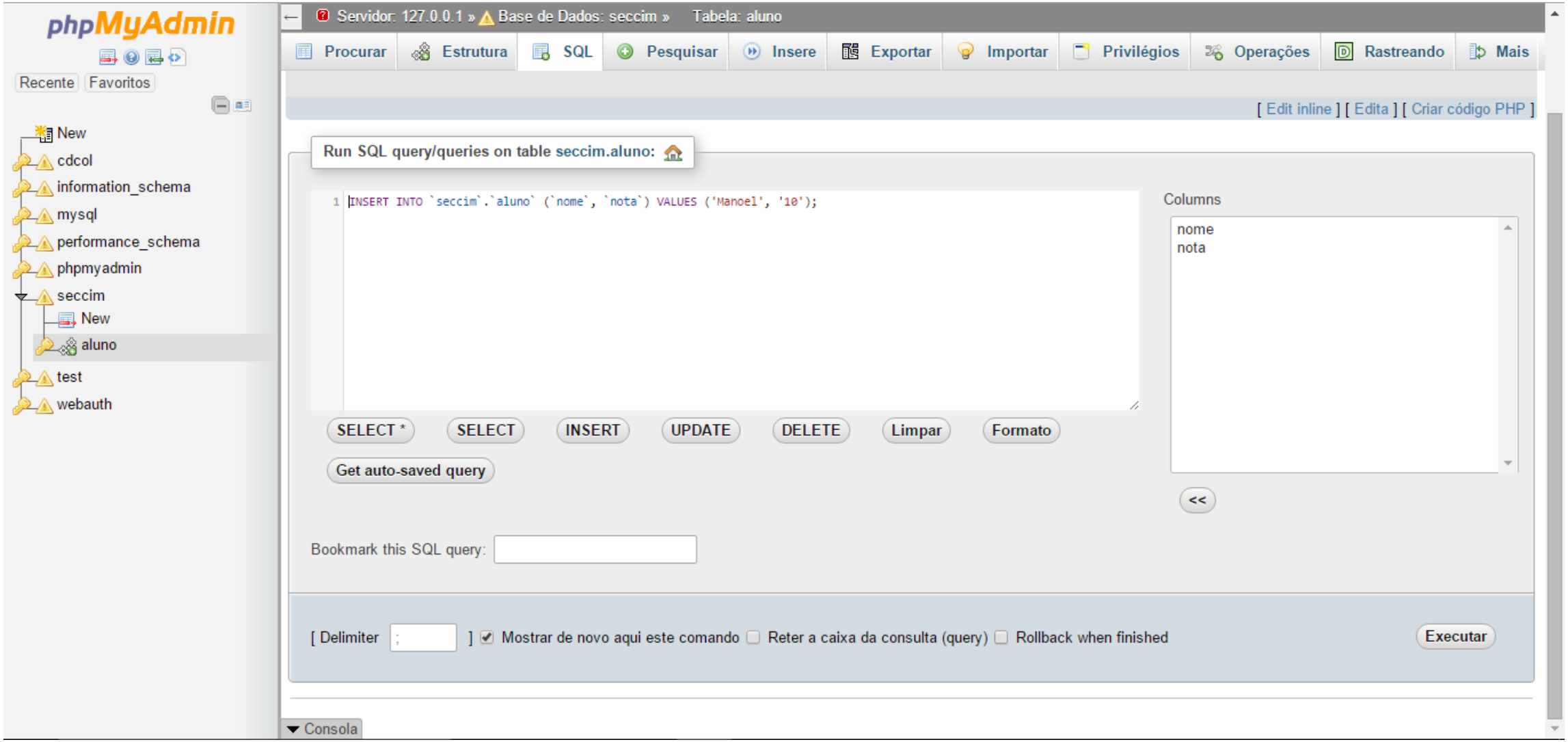
Interfaces

```php
<?php
interface AutoMotor {
    public function ligar();
    public function acelerar();
    public function freiar();
    public function desligar();
}

class Moto implements AutoMotor {

    private $motorLigado = false;
    private $velocidade = 0;

    public function ligar() { $this->motorLigado = true; }
    public function desligar() { $this->motorLigado = false; }
    public function acelerar() { $this->velocidade++; }
    public function freiar() { $this->velocidade--; }
}

$moto = new Moto();
$moto->ligar();
?>
```

Objeto dinâmico

```php
<?php
$luiz                = new stdClass();
$luiz->nome          = 'Luiz Otávio';
$luiz->sobrenome     = 'Miranda';
$luiz->idade         = 27;
$luiz->profissao     = 'Desenvolvedor';

var_dump($luiz);
?>
```

Banco de Dados MySQL



Conexão com MySQL

```php
<?php
        $link = mysql_connect('localhost', 'root', 'seccim');

        if (!$link) {
            die('Não foi possível conectar: ' . mysql_error());
        }

        echo 'Conexão bem sucedida';

        mysql_select_db("seccim", $link) or print(mysql_error());

        mysql_close($link);
?>
```

Consulta MySQL

```php
<?php
        //Conecta
        $conecta = mysql_connect("localhost", "root", "") or print (mysql_error());
        mysql_select_db("seccim", $conecta) or print(mysql_error());

        //Consulta
        $sql = "SELECT nome, nota FROM aluno";
        $result = mysql_query($sql, $conecta);

        while($consulta = mysql_fetch_array($result)) {
           print "Nome: $consulta[nome] - Nota: $consulta[nota]<br>";
        }

        mysql_free_result($result);
        mysql_close($conecta);
?>
```

Insere MySQL

```php
<?php
        //Conecta
        $conecta = mysql_connect("localhost", "root", "") or print (mysql_error());
        mysql_select_db("seccim", $conecta) or print(mysql_error());

        //Consulta
        $sql = "INSERT INTO aluno (nome, nota)
                        VALUES ('Renato', 10)";

        if (mysql_query($sql) === TRUE) {
            echo "Inserido com sucesso";
        } else {
            echo "Error: " . $sql . "<br>" . mysql_error();
        }
        mysql_close($conecta);
?>
```

Exclui MySQL

```php
<?php
        //Conecta
        $conecta = mysql_connect("localhost", "root", "") or print (mysql_error());
        mysql_select_db("seccim", $conecta) or print(mysql_error());

        //Consulta
        $sql = "DELETE FROM aluno
                        WHERE nome = 'Renato'";

        if (mysql_query($sql) === TRUE) {
            echo "Ecluido com sucesso";
        } else {
            echo "Error: " . $sql . "<br>" . mysql_error();
        }
        mysql_close($conecta);
?>
```

Atualiza MySQL

```php
<?php
        //Conecta
        $conecta = mysql_connect("localhost", "root", "") or print (mysql_error());
        mysql_select_db("seccim", $conecta) or print(mysql_error());

        //Consulta
        $sql = "UPDATE aluno
                        SET nome = 'Roberto'
                        WHERE nome = 'Renato'";

        if (mysql_query($sql) === TRUE) {
            echo "Atualizado com sucesso";
        } else {
            echo "Error: " . $sql . "<br>" . mysql_error();
        }
        mysql_close($conecta);
?>
```

Referências

- http://blog.websolute.com.br/market-share-uso-software-brasil-mundo-setembro-2014/
- http://php.net/manual/pt_BR/
- http://www.w3schools.com/php/