

Programação Web

Introdução ao PHP

Prof. George Oliveira - Programação Web - IFRR

Quem sou eu?

George Almeida de Oliveira

Professor de Informática – IFRR

Mestre em Computação Aplicada – UNISINOS

Especialista em Engenharia de Sistemas – ESAB

Analista de Sistemas – IFRR

Email: professorgeorgeoliveira@gmail.com

Facebook: <https://www.facebook.com/professorgeorge.oliveira>

Prof. George Oliveira - Programação Web - IFRR

O que é PHP?

- PHP é uma sigla para **"PHP: Hypertext Preprocessor"**
- PHP é um amplamente utilizado, linguagem de script open source (código aberto).
- scripts PHP são **executados no servidor**
- PHP é **gratuito** para baixar e usar

Prof. George Oliveira - Programação Web - IFRR

O que o PHP pode fazer?

- PHP pode gerar páginas com **conteúdo dinâmico**
- PHP pode criar, abrir, ler, escrever, apagar e fechar **arquivos** no servidor
- PHP pode coletar dados de **formulários**
- PHP pode enviar e receber **cookies**
- PHP pode adicionar, excluir, modificar dados em seu **banco de dados**
- PHP pode ser usado para **controlar o acesso** de usuário
- PHP pode **criptografar os dados**

Prof. George Oliveira - Programação Web - IFRR

O que é um arquivo PHP?

- código PHP são executados no servidor, e o resultado **é retornado** para o navegador como **HTML** simples.
- arquivos PHP tem extensão **".php"**
- arquivos PHP pode conter texto, HTML, CSS, JavaScript e código PHP.

Prof. George Oliveira - Programação Web - IFRR

Por que PHP?

- PHP é executado em **várias plataformas** (Windows, Linux, Unix, Mac OS X, etc.)
- PHP é **compatível** com quase todos os **servidores** usados hoje (Apache, IIS, etc.)
- PHP **suporta** uma ampla gama de **bases de dados**
- PHP é **gratuito**. Baixa-lo a partir do recurso oficial PHP: www.php.net
- PHP é **fácil de aprender** e executado de forma eficiente no lado do servidor

Prof. George Oliveira - Programação Web - IFRR

Operadores String

Operador	Nome	Exemplo	Resultado
.	Concatenação	\$txt1 . \$txt2	Concatena o valor de \$txt1 e \$txt2
.=	Concatenação atribuição	\$txt1 .= \$txt2	Adiciona \$txt2 em \$txt1

Operadores Lógicos

Operador	Nome	Função	Exemplo	Resultado
&&	E	Conjunção	\$x && \$y	True if both \$x and \$y are true
	Ou	Disjunção	\$x \$y	True if either \$x or \$y is true
!	Não	Negação	!\$x	True if \$x is not true
And	E	Conjunção	\$x and \$y	True if both \$x and \$y are true
Or	Ou	Disjunção	\$x or \$y	True if either \$x or \$y is true
Xor	Xou	Disjunção Exclusiva	\$x xor \$y	True if either \$x or \$y is true, but not both

Menor prioridade

Prof. George Oliveira - Programação Web - IFRR

13

Tabela-Verdade

Mostra os resultados das aplicações dos operadores lógicos conforme os valores dos operadores envolvidos.

Negação (!)		Conjunção (&&)				Disjunção ()				Ou Exclusivo(Xor)			
\$A	!(SA)	\$A	\$B	\$A && \$B		\$A	\$B	\$A \$B		\$A	\$B	\$A xor \$B	
V	F	V	V	V		V	V	V		V	V	F	
F	V	V	F	F		V	F	V		V	F	V	
		F	V	F		F	V	V		F	V	V	
		F	F	F		F	F	F		F	F	F	

Prof. George Oliveira - Programação Web - IFRR

14

Operadores de Incremento e Decremento

Operador	Nome	Descrição
++\$x	Pré-incremento	Incrementa 1 em \$x, então retorna \$x
\$x++	Pos-incremento	Retorna \$x, então incrementa 1 em \$x
--\$x	Pré-decremento	Decrementa 1 em \$x, então retorna \$x
\$x--	Pos-decremento	Retorna \$x, então decrementa 1 em \$x

OBS:

\$x++ é o mesmo que \$x=\$x+1

\$x-- é o mesmo que \$x=\$x-1

Prof. George Oliveira - Programação Web - IFRR

15

Operadores Relacionais / de Comparação

Operador	Nome	Exemplo	Resultado
==	Igual	\$x == \$y	Returns true if \$x is equal to \$y
===	Idêntico	\$x === \$y	Returns true if \$x is equal to \$y, and they are of the same type
!=	Diferente	\$x != \$y	Returns true if \$x is not equal to \$y
<>	Diferente	\$x <> \$y	Returns true if \$x is not equal to \$y, or they are not of the same type
!==	Não idêntico	\$x !== \$y	Returns true if \$x is not equal to \$y, or they are not of the same type
>	Maior que	\$x > \$y	Returns true if \$x is greater than \$y
<	Menor que	\$x < \$y	Returns true if \$x is less than \$y
>=	Maior ou igual a	\$x >= \$y	Returns true if \$x is greater than or equal to \$y
<=	Menor ou igual a	\$x <= \$y	Returns true if \$x is less than or equal to \$y

Prof. George Oliveira - Programação Web - IFRR

16

Prioridade de Execução (Precedência de Operadores)

Operador	Descrição
()	Parênteses
**	Potenciação
- ! ++ --	Negativo, negação, incremento e decremento
* / %	Multiplicação, divisão e resto da divisão
+ - .	Adição, subtração e concatenação
> < >= <=	Maior que, menor que, maior ou igual e menor ou igual
== != <> === !==	Igual e diferente
&&	E
= += -= *= /= %=	Operadores de atribuição
AND	E com menor prioridade
XOR	Ou exclusivo
OR	Ou com menor prioridade

Prof. George Oliveira - Programação Web - IFRR

17

Operador Ternário

É um operador de expressão condicional.

(avaliação condicional) ? (caso afirmativo) : (caso negativo);

O interpretador PHP avalia a primeira expressão. Se ela for verdadeira, a expressão retorna o valor do caso afirmativo. Senão, retorna o valor do caso negativo.

```
<?php //exemplo com operador ternário
$resultado = 7.5;
$msg = $resultado >= 7 ? 'Aprovado' : 'Reprovado';
echo $msg;
?>
```

Prof. George Oliveira - Programação Web - IFRR

18

Cuidado com os Operadores Lógicos

```
<?php //cuidado com a precedência(ordem dos operadores)
$verdadeiro = true;
$falso = false;

$comparacao1 = $verdadeiro && $falso; // Como se fosse:
$comparacao2 = $verdadeiro AND $falso; // ($comparacao2 = $verdadeiro ) AND $falso;

var_dump($comparacao1); //o resultado será false
var_dump($comparacao2); //o resultado será true
?>
```

O nível de precedência do operador `=` é mais elevado que o do operador `AND`. Porém, o nível de precedência do operador `&&` é mais elevado que o de `=`.

Prof. George Oliveira - Programação Web - IFRR

19

Estruturas de Controle no PHP

SELEÇÃO – Estrutura de Decisão ou Condicional

- Simple – **IF**
- Compostas – **IF .. ELSE**
- Compostas – **IF .. ELSEIF .. ELSE**
- Escolha – **SWITCH CASE**

REPETIÇÃO – Estruturas de Repetição (laço de repetição)

- Pré-Teste – **WHILE**
- Pós-Teste – **DO .. WHILE**
- Variável de Controle – **FOR**

Prof. George Oliveira - Programação Web - IFRR

20

Prática: arquivo *estruturadecisao1.php*

IF	ELSE	ELSEIF
<pre><?php \$x = 3; if (\$x % 2 != 0) { echo "Número ímpar"; } ?></pre>	<pre><?php \$idade = 20; if (\$idade < 18) { echo "Menor de idade"; } else { echo "Maior de idade"; } ?></pre>	<pre><?php \$hora = date("H"); echo "hora: \$hora
"; if (\$hora < 12) { echo "Bom dia!"; } elseif (\$hora < 18) { echo "Boa Tarde"; } else { echo "Boa Noite!"; } ?></pre>
Resultado Esperado Número é ímpar	Resultado Esperado Maior de idade	Resultado Esperado irá depender da hora em que o código for executado

Prof. George Oliveira - Programação Web - IFRR

21

Prática: arquivo *estruturadecisao2.php*

<pre><?php \$versao = 1; switch (\$versao) { case 1: echo "Atualizando versão 1 ...
"; case 2: echo "Atualizando versão 2 ...
"; case 3: echo "Atualizando versão 3 ...
"; default: echo "Versão atualizada!"; } ?></pre>	<pre><?php \$sexo = "m"; switch (\$sexo) { case "m": case "M": echo "Sexo masculino"; break; case "f": case "F": echo "Sexo feminino"; break; default: echo "Sexo não informado"; } ?></pre>
Resultado Esperado Atualizando versão 1 ... Atualizando versão 2 ... Atualizando versão 3 ... Versão Atualizada!	Resultado Esperado Sexo masculino

Prof. George Oliveira - Programação Web - IFRR

22

Prática: Arquivo *estruturaderepeticao1.php*

While	Do ... While
<pre><?php \$x = 1; while (\$x <= 5) { echo "O número é: \$x
"; \$x++; } ?></pre>	<pre><?php \$y = 1; do { echo "O número é: \$y
"; \$y++; } while (\$y <= 5); ?></pre>
Resultado Esperado O número é: 1 O número é: 2 O número é: 3 O número é: 4 O número é: 5	Resultado Esperado O número é: 1 O número é: 2 O número é: 3 O número é: 4 O número é: 5

Prof. George Oliveira - Programação Web - IFRR

23

Prática: Arquivo *estruturaderepeticao2.php*

<pre><?php for (\$x = 0; \$x <= 10; \$x++) { echo "O número é: \$x "; } ?></pre>	<p>Resultado Esperado</p> <p>O número é: 0 O número é: 1 O número é: 2 O número é: 3 O número é: 4 O número é: 5 O número é: 6 O número é: 7 O número é: 8 O número é: 9 O número é: 10</p>
--	---

OBS: Este tipo de repetição é ideal quando sabemos com antecedência quantas vezes o bloco de instruções deveria ser executado.

OBS: Não utilizar quando não se sabe a condição de encerramento/parada da estrutura de repetição.

Prof. George Oliveira - Programação Web - IFRR

24

Exemplos: Procedimento vs Funções

Procedimentos

```
function adicao ($a, $b){
    $x = $a + $b;
    echo "o resultado da soma é: $x";
}
adicao(2, 3);
adicao(4, 6);
```

o resultado da soma é: 5
o resultado da soma é: 10

OBS: Um procedimento não retorna valor.

OBS: Não é possível criar duas funções com o mesmo nome.

Prof. George Oliveira - Programação Web - IFRR

25

Variável Global praticando: *escopoglobal.php*

Uma variável declarada **fora** de uma função tem um escopo global e só pode ser acessada fora de uma função:

```
<?php
$y = 7; //Escopo global
function TestaEscopoGlobal() {
    //Usando y dentro dessa função irá gerar um erro
    echo "variável y dentro função é: $y ";
}
TestaEscopoGlobal();
echo "variável y fora função é: $y ";
?>
```

Prof. George Oliveira - Programação Web - IFRR

26

Variável Local praticando: *escopolocal.php*

Uma variável declarada **dentro** de uma função tem um escopo local e só pode ser acessada dentro dessa função:

```
<?php
function TestaEscopoLocal() {
    $x = 5; //Escopo local
    echo "variável x dentro função é: $x ";
}
TestaEscopoLocal();
// Usando x fora da função irá gerar um erro
echo "variável x fora função é: $x ";
?>
```

Prof. George Oliveira - Programação Web - IFRR

27

A palavra-chave: *global* praticando: *palavraglobal.php*

Uma variável criada com o comando **global** pode ser acessada em qualquer parte do programa, dentro e fora das funções.

Para criar uma variável global usamos a palavra reservada **global** antes do nome da variável durante sua declaração.

```
<?php
$a = 2;
$b = 4;
function soma() {
    global $a, $b; // reconhece as variáveis do escopo global
    global $c; // torna $c uma variável global
    $c = $a + $b;
}
soma();
echo "c: $c"; // $c não é mais uma variável local
?>
```

Prof. George Oliveira - Programação Web - IFRR

28

Variáveis PHP Global - Superglobais

No PHP as variáveis superglobais são:

\$GLOBALS	Tem referências de todas as variáveis disponíveis no escopo global
\$_SERVER	Informação do servidor e do ambiente de execução
\$_GET	Variáveis HTTP GET
\$_POST	Variáveis HTTP POST
\$_REQUEST	Variáveis de Requisição HTTP
\$_COOKIE	Variáveis de cookies
\$_SESSION	Variáveis de sessão
\$_FILES	Variáveis de Upload HTTP
\$_ENV	Variáveis de ambiente

Prof. George Oliveira - Programação Web - IFRR

29

Funções: isset x empty x is_null

A tabela abaixo descreve as função de cada um dos comandos:

FUNÇÃO	DESCRIÇÃO
isset()	verifica se uma variável está setada* e não é nula.
empty()	verifica se uma variável está vazia .
is_null()	verifica se uma variável é NULL (nula)

*Setada (is set) = se uma variável existe na memória.

Prof. George Oliveira - Programação Web - IFRR

30

Tabela comparativa

No PHP, os possíveis valores para cada uma das funções são apresentados na tabela abaixo:

Valor da variável (\$var)	isset(\$var)	empty(\$var)	is_null(\$var)
"" (uma string vazia)	bool(true)	bool(true)	bool(false)
" " (espaço)	bool(true)	bool(false)	bool(false)
FALSE	bool(true)	bool(true)	bool(false)
TRUE	bool(true)	bool(false)	bool(false)
array() (um array vazio)	bool(true)	bool(true)	bool(false)
NULL	bool(false)	bool(true)	bool(true)
"0" (0 como uma string)	bool(true)	bool(true)	bool(false)
0 (0 como um integer)	bool(true)	bool(true)	bool(false)
0.0 (0 como um float)	bool(true)	bool(true)	bool(false)
var \$var; (uma variável declarada, mas sem um valor)	bool(false)	bool(true)	bool(true)

Prof. George Oliveira - Programação Web - IFRR

31

INCLUDE x REQUIRE INCLUDE_ONCE x REQUIRE_ONCE

INCLUDE()

- Incluir código do arquivo quantas vezes for solicitado.
- Não trava se o arquivo não existir.

REQUIRE()

- Incluir código do arquivo quantas vezes for solicitado.
- Se não encontrar o arquivo, trava!

INCLUDE_ONCE()

- Incluir um arquivo apenas uma vez.
- Não trava se o arquivo não existir.

REQUIRE_ONCE()

- Incluir um arquivo apenas uma vez.
- Se não encontrar o arquivo, trava!

Prof. George Oliveira - Programação Web - IFRR

32

Arrays indexados

Existem duas maneiras de criar arrays indexados:

Automaticamente: O índice é atribuído através do comando `array()`

```
<?php
$nomes = array("Alex", "Leonardo", "Vanessa");
?>
```

Manualmente: O índice é atribuído pelo programador.

```
<?php
$nomes[0] = "Alex";
$nomes[1] = "Leonardo";
$nomes[] = "Vanessa"; //será adicionado no fim do array:(índice 2)
?>
```

Também é possível combinar os dois.

```
<?php
$frutas = array("Laranja", "Banana", "Limão");
$frutas[] = "Melancia"; //será adicionado no índice 3
$frutas[8] = "Tomate"; //será adicionado no índice 8
?>
```

Prof. George Oliveira - Programação Web - IFRR

33

Arrays associativos

Usam o conceito de "chave-valor". Podem ser criadas de 2 maneiras:

Através do comando `array()`:

```
<?php
$idades = array("Artur"=>15, "Kelvin"=>26, "Rodrigo"=>19);
?>
```

Individualmente:

```
<?php
$idades["Artur"] = 15;
$idades["Kelvin"] = 26;
$idades["Rodrigo"] = 19;
?>
```

Prof. George Oliveira - Programação Web - IFRR

34

Arrays associativos: acessando elementos

- `$array['chave']`: Acesso direto (e específico) a um único elemento.
- `foreach($array as $valor)`: percorrer todos elementos e acessando seus valores.
- `foreach($array as $chave => $valor)`: percorrer todos os elementos, acessando suas chaves e valores.

```
<?php
$idades = array("Artur"=>15, "Kelvin"=>26, "Rodrigo"=>19);

echo "Chave-Valor: <br>"; //Acesso um único elemento
echo "Kelvin tem: ". $idades["Kelvin"]. " anos <br>";

echo "Chave-Valor: <br>"; //Acesso um único elemento
echo "Kelvin tem: ". $idades["Kelvin"]. " anos <br>";

foreach($idades as $valor) {
    echo "Idade: $valor <br>";
}

foreach($idades as $chave => $valor) {
    echo "Chave: $chave e valor: $valor <br>";
}

foreach($idades as $chave => $valor) {
    echo "Nome: $chave e idade: $valor <br>";
}
```

Chave-Valor:
Kelvin tem : 26 anos

Idade: 15
Idade: 26
Idade: 19

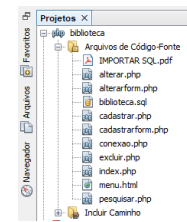
Chave: chave e valor
Artur : 15
Kelvin : 26
Rodrigo : 19

Prof. George Oliveira - Programação Web - IFRR

35

PHP e Banco de Dados

Projeto Prático



Prof. George Oliveira - Programação Web - IFRR

36

Praticando: *conexao.php*

O código anterior pode ser reescrito de uma maneira mais simples e continuar com a mesma função:

```
<?php
//endereço, usuário, senha, base de dados (já deve existir);
$conexao = @mysqli_connect("localhost", "root", "", "loja")
or die("Erro: " . mysqli_connect_error());

echo "Conexão Realizada com Sucesso";
?>

(MySQLi Procedural)
```

Prof. George Oliveira - Programação Web - IFRR

37

Prática: Arquivo *cadaststrarform.php*

Formulário para adicionar um produto

```
<html>
<body>
  <form method="post" action="cadaststrar.php">
    Nome: <input type="text" name="nome"/> <br>
    Descrição: <input type="text" name="descricao"> <br>
    Valor: <input type="text" name="valor"/> <br>
    <input type="submit" value="Cadastrar"/>
    <input type="reset" value="Limpar"/>
  </form>
</body>
</html>
```

Prof. George Oliveira - Programação Web - IFRR

38

Prática: Arquivo *cadaststrar.php*

```
<?php
$nome = $_POST['nome']; //recuperar dados do formulário
$descricao = $_POST['descricao'];
$valor = $_POST['valor'];

require 'conexao.php';

$sql = "INSERT INTO produtos (nome, descricao, valor)
VALUES ('$nome', '$descricao', $valor)";
if(mysqli_query($conexao, $sql)){
  echo "Registro inserido";
}else{
  echo "ERRO: " . mysqli_error($conexao);
}

OBS: não esqueça das aspas simples ''
(MySQLi Procedural)
?>
```

Prof. George Oliveira - Programação Web - IFRR

39

Prática: Arquivo *pesquisar.php*

```
<?php
require 'conexao.php';
$sql = "SELECT * FROM produtos";
$resultado = mysqli_query($conexao, $sql);
if (mysqli_num_rows($resultado) > 0) {
  while($registro = mysqli_fetch_assoc($resultado)) {
    echo "id: " . $registro['id']; //nome da coluna do Banco de dados
    echo "<br> nome: " . $registro['nome'];
    echo "<br> descrição: " . $registro['descricao'];
    echo "<br> valor: " . $registro['valor'];
    echo "<a href=excluir.php?id=".$registro['id']. "> Excluir</a> - ";
    echo "<a href=alterarform.php?id=".$registro['id']. "> Alterar</a>";
    echo "<br> ";
  }
}else{
  echo "<br> Nenhum resultado encontrado. <br>";
}
?>
```

Prof. George Oliveira - Programação Web - IFRR

40

Prática: arquivo *excluir.php*

```
<?PHP
require 'conexao.php';
if (isset($_GET["id"])) { //consultar e recuperar id do registro
  $id = $_GET["id"];
  $sql = "DELETE FROM produtos WHERE id=$id";
  if(mysqli_query($conexao, $sql)){
    echo "Registro deletado";
  }else{
    echo "ERRO: " . mysqli_error($conexao);
  }
} else {
  echo "Registro não informado";
}
?>

(MySQLi Procedural)
```

Prof. George Oliveira - Programação Web - IFRR

41

Prática: arquivo *alterarform.php* ...

```
<?PHP
if (isset($_GET["id"])) { //consultar e recuperar id do registro
  $id = $_GET["id"];
  require 'conexao.php';
  $sql = "SELECT * FROM produtos WHERE id=$id";
  $resultado = mysqli_query($conexao, $sql);
  $registro = mysqli_fetch_assoc($resultado);
  $id = $registro['id'];
  $nome = $registro['nome'];
  $descricao = $registro['descricao'];
  $valor = $registro['valor'];
} else {
  die("Registro não informado");
}
?> .... <HTML> <body>...
```

Prof. George Oliveira - Programação Web - IFRR

42

... ainda no arquivo *alterarform.php*

OBS: Precisamos trazer o formulário preenchido, para isso, colocamos os valores das variáveis dentro dos campos do formulário usando o atalho `<?= $variavel ?>`

```
... <HTML>
<body>
<form method="post" action="alterar.php">
  <input type="hidden" name="id" value="<?= $id ?>" />
  Nome: <input type="text" name="nome" value="<?= $nome ?>" />
  Descrição: <input type="text" name="descricao" value="<?= $descricao ?>" />
  Valor: <input type="text" name="valor" value="<?= $valor ?>" />
  <input type="submit" value="Salvar" />
  <input type="reset" value="Limpar" />
</form>
</body>
</html>
```

Prof. George Oliveira - Programação Web - IFRR

43

Prática: arquivo *alterar.php*

```
<?php
$id = $_POST['id']; //recuperar id do registro
$nome = $_POST['nome']; //recuperar dados do formulário
$descricao = $_POST['descricao'];
$valor = $_POST['valor'];

require 'conexao.php';

$sql = "UPDATE produtos SET nome='$nome',
        descricao='$descricao', valor=$valor WHERE id=$id";
if(mysql_query($conexao, $sql)){
    echo "Registro atualizado";
}else{
    echo "ERRO: ".mysql_error($conexao);
}
?>
```

(MySQLi Procedural)

Prof. George Oliveira - Programação Web - IFRR

44

Cookie x Sessão

CARACTERÍSTICA	SESSÃO	COOKIE
Local da Informação	No Servidor	No Cliente
Segurança	Mais Seguro	Menos Seguro
Disponibilidade	Sempre	As vezes
Tempo de Duração	Menor	Grande

Prof. George Oliveira - Programação Web - IFRR

45

Características de um cookie

- Cookies são pequenas informações que são armazenadas no disco rígido do cliente.
- Trafegam pelo cabeçalho http.
- Informações que ficam **armazenadas no cliente**.
- Tem os parâmetros **nome e valor**.
- O programa manda armazenar no cliente para no futuro reavê-la.
- **Não recomendado** para informações **sigilosas**

Prof. George Oliveira - Programação Web - IFRR

46

Trabalhando com cookie

```
Criar/alterar/excluir:
setcookie(nome,valor,expiração,pacote,domínio);
Recuperar um valor:
echo $_COOKIE[cookie_name];

<?php
setcookie("usuario", "Bruce Wayne", time()+3600);
echo $_COOKIE['usuario'];
setcookie("usuario", "Diana", time()+3600);
setcookie("usuario", "", time()-3600);
?>
```

OBS: Para excluir um cookie, use a função setcookie () com uma data de expiração no passado, **valor negativo**:

Prof. George Oliveira - Programação Web - IFRR

47

Características de uma sessão

- As variáveis de sessão **armazenam informações** sobre um único usuário.
- Estão disponíveis para todas as páginas de uma aplicação.
- É destruído no momento que o **navegador for fechado**.
- É utilizado principalmente na **segurança do sistema**.
- As informações **ficam no servidor**.

Prof. George Oliveira - Programação Web - IFRR

48

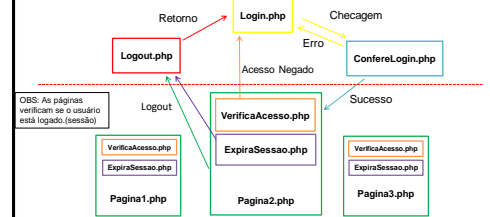
Trabalhando com sessão

Obrigatório -> `session_start();`
 Criar -> `$_SESSION['usuario'] = "Barry Allen";`
 Recuperar -> `echo $_SESSION['usuario'];`
 Alterar -> `$_SESSION['usuario'] = "Diana";`
 Excluir -> `unset($_SESSION['usuario']);`
 Excluir -> `session_unset();`
 ?>

Prof. George Oliveira - Programação Web - IFRR

49

Controle de Acesso



Prof. George Oliveira - Programação Web - IFRR

50

Referências

VÍDEO AULAS:

<http://www.cursoemvideo.com/course/curso-php-iniciante/>
https://youtu.be/F7KzJ7eEAc?list=PLHz_AreHm4dm4beCCmW4xwpmLf6EHY9k (Youtube)

APOSTILA:

http://www.cin.ufpe.br/~ags/2464_php_com_mysql.pdf

W3SCHOOLS: Documentação:

<http://www.w3schools.com/php/>

Prof. George Oliveira - Programação Web - IFRR

51