

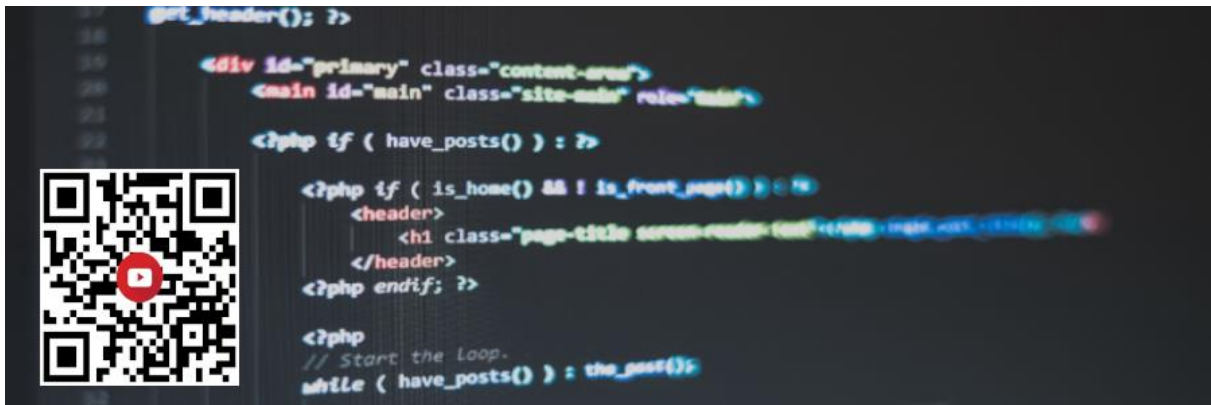


SISTEMAS WEB I

Sumário

| | |
|---|----|
| Introdução | 3 |
| 1 - Conhecendo a linguagem PHP | 5 |
| 2 - Variáveis e Tipos de dados | 12 |
| 3 - Operadores | 20 |
| 4 - Estrutura de controle..... | 24 |
| 5 - Conceito sobre Array | 29 |
| 6 - Método de envio com GET e POST..... | 32 |
| 7 - Tipos de Funções | 37 |
| 8 – Tipos de Cookies e Sessões..... | 41 |
| 9 – Conexão com o banco de dados..... | 46 |
| 10 – Introdução ao conceito POO e classes com PHP | 58 |

Introdução



A linguagem de programação PHP (Hypertext Preprocessor) foi criada em 1994, pelo Ramus Lerdorf, com a filosofia open source, código aberto voltada para web, sendo que para escrever ela precisa ter conhecimento front end da linguagem de marcação HTML, fazendo ilusão a sua lógica de conhecimento o código escrito em PHP, que vem derivado do C e do Perl, vai ser interpretado pelo um servidor web, pre-processado e gerando um HTML para o usuário, nesse caso o usuário só vai receber dados e HTML.

Com essa linguagem podemos desenvolver inúmeras plataformas para WEB, com mais de 20 anos essa linguagem hoje se encontra em diversas empresas de grande e médio porte, podemos até falar que hoje quase 65% dos sistemas Web tem PHP, sendo hoje uma linguagem fortemente moderna e segura com a sintaxe POO. Vamos abordar tudo sobre a linguagem de programação PHP, procedural uma arquitetura muito usado segura e vamos aplicar boas práticas no código que vai ser visto nessa apostila.

Toda linguagem procedural vem de um conceito fácil de entender, linha a linha como se fosse uma receita de bolo, como vimos muito nas aulas de lógica a onde um código depende de outro para rodar uma aplicação.

No tema dez vai ser visto uma pequena introdução sobre o PHP POO, que vamos colocar os pontos principais e fundamentais para você programador também começar a desenvolver seu sistema Web em PHP POO.

1 - Conhecendo a linguagem PHP



1.1 Introdução

PHP (PHP, Hypertext pré – processor) é uma linguagem de programação voltada para desenvolvimento de sistema web, foi criada pelo engenheiro Rasmus Lerdof, com o objetivo de guardar dados via formulários, dentro do escopo Open Source (código aberto), lançado na versão 1.0, em 1994, tendo várias versões e atualização, sendo hoje uma linguagem segura e implantada com a arquitetura orientada ao objeto. Baseado e escrito em linguagem de programação C.

Em Junho de 1995, Rasmus liberou o seu código fonte do PHP ferramentas para o público, o que permitiu que desenvolvedores usassem da forma como desejassem. Isso permitiu - e encorajou - usuários a fornecerem correções para bugs no código, e em geral, aperfeiçoá-lo.

Versões do PHP procedural até a POO

- 1.0 PHP/FI – Primeira versão feita ainda em C, tinha uma básica GCI mais já estava sendo embutido no HTML, por isso ganhou esse nome de PHP/FI, (Pesonal Home Page / Forms Interpreter)
- 2.0 Já a segunda versão ganhou dois programadores que viu o PHP como uma linguagem do futuro eles se chamavam de Zeev e Andi ambos Israelense nessa versão ganhou o nome de PHP.
- 3.0 Oficialmente agora com o nome PHP, já com script próprios, entrou no mercado de desenvolvimento de sistema WEB.
- 4.0 Nessa versão que foi lançado no ano de 2000, venho com duas grandes novidade um suporte melhor para conexão com o banco de dados e as primeiras expressões POO. Lembrando que até aqui toda a linguagem era escrito em processo de procedural

5.0 Esta versão venho cheio de novas, aplicação POO, o facebook adotou na época essa linguagem para o seu projeto.

6.0 Está versão ficou conhecida como a versão de segurança, algo que todos os programadores reclamavam muito, PHP não é seguro, agora sim uma linguagem limpa, segura e organizada

7.0 Versão para mobile e mais segurança com API.

8.0 API para servidores Web, tornando mais rápido nas suas aplicação e atual.

A Linguagem PHP necessita da linguagem de marcação de texto HTML, embutidos no seu front end.

A linguagem que vamos estudar é procedural ou também conhecida como estruturada o que significa isso, é a maneira como vamos escrever nosso código essa maneira de codificar, é escrever bastante código, linha a linha podemos até comparar com a linguagem de marcação de texto, HTML um texto que não podemos ser reutilizado, para quem está iniciando na programação é a maneira mais fácil de iniciar essa trajetória como Back – End.

1.2 Ambientação e instalação de IDE



A linguagem de PHP, pode ser escrita em qualquer sistema operacional, sendo mais indicada e rápida no Linux por ser nativo, com a linguagem PHP roda do lado do servidor, precisamos ter um servidor WEB, APACHE ou ISS entre outros. Vamos usar nesse livro o servidor web APACHE.

Para escrever e rodar o PHP temos que ter as seguintes ferramentas;

- IDE para escrever;

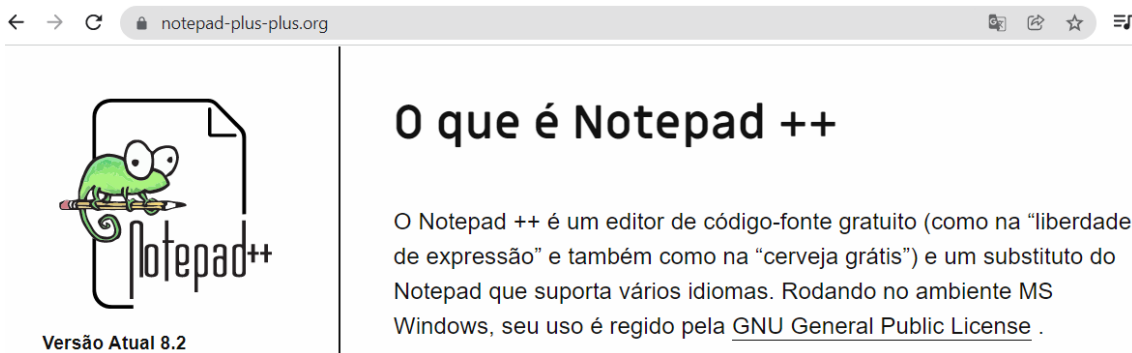
- Linguagem de programação PHP;
- Servidor Web (no caso vamos rodar o APACHE);
- SGBD(Sistema de gerenciamento de banco de dados);

IDE (Integrated Development Environment) ou ambiente de desenvolvimento integrado é um software usado para escrever as linguagem de programação e linguagem de marcação de texto, podemos usar qualquer IDE para essa finalidade, por ser um código aberto.

Vamos usar o IDE NotePad++, por ser tudo livre para escrever o HTML, CSS e o PHP.

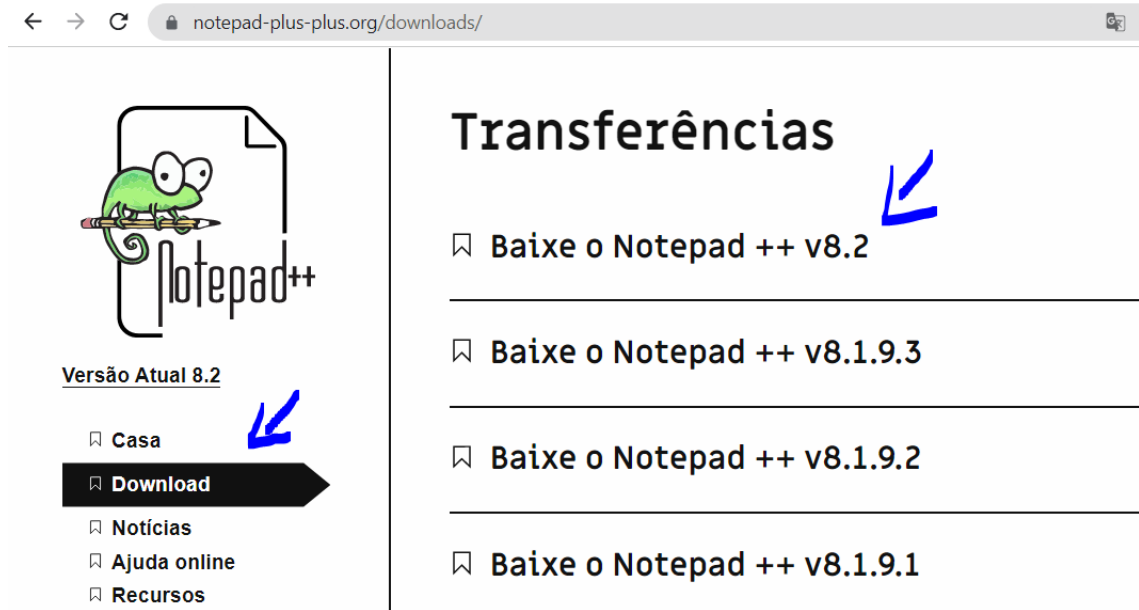
Segue os passos para a instalação:

1. Passo acessar o site - <https://notepad-plus-plus.org/>



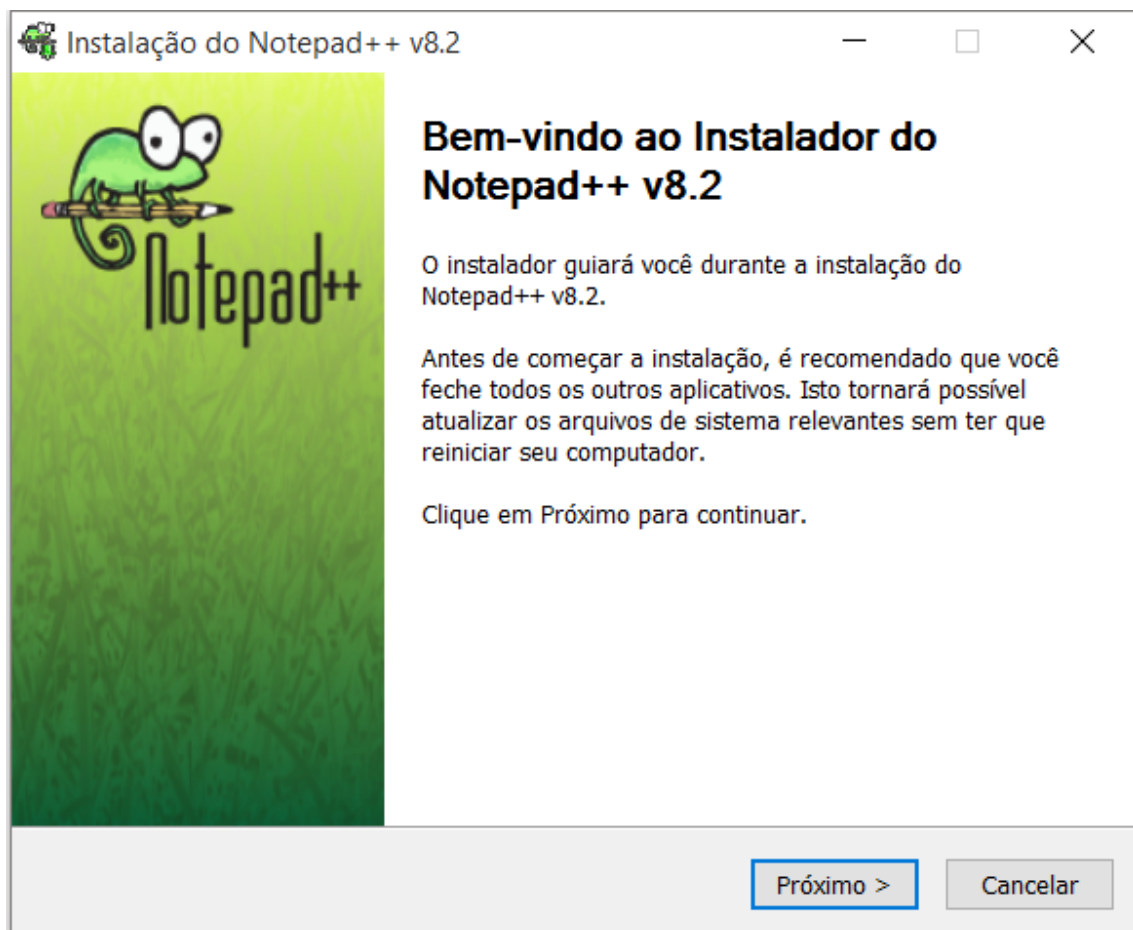
2. Fazer o Download

Clicar na guia download e baixar a última versão e escolher a versão do seu sistema operacional sendo 32 ou 64 bits.



3. Instalação

Clicar no link instalador depois de ter escolhido o tipo de bits que o seu sistema operacional se encontra, faz o download clicar em português e avança;



Servidor WEB é um tipo de serviço que vai receber o código que vamos escrever e enviar o que o cliente solicitou, o código fonte não fica do lado do cliente somente os dados que foi pedido pelo usuário, diferente do HTML, que fica do lado do cliente. Vamos usar o APACHE como servidor de web o mais usado no mundo inteiro.

SGBD (Sistema de gerenciamento de dados) é onde vamos guardar os dados do nosso futuro sistema, que vamos desenvolver o SGBD que vamos usar vai ser o MYSQL, mais poderia ser qualquer outro para WEB.

Linguagem de programação é os comando da própria linguagem que o IDE no caso o Servidor Web, que vai interpretar.

Para rodar tudo isso na nossa máquina local, vamos precisar instalar um simulador de servidor Web, que contem APACHE, SGBD e a própria Linguagem, vamos instalar o XAMPP. O XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Perl.

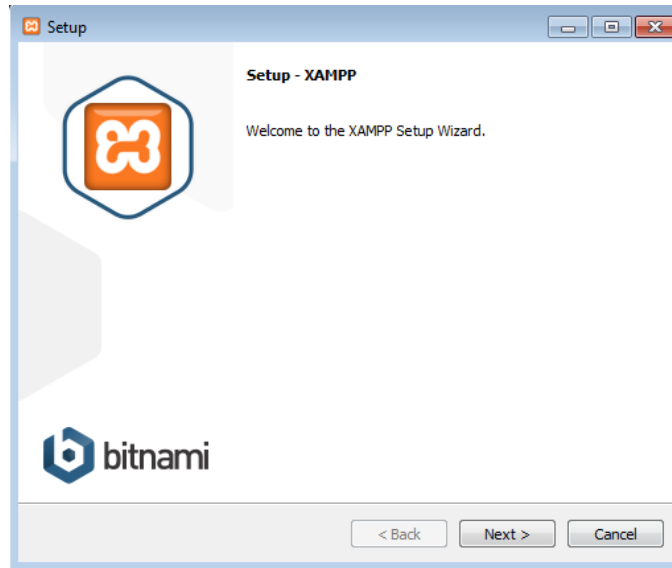
Instalando o XAMPP;

1. Download do XAMPP - <https://www.apachefriends.org/index.html>. Clicar na guia download e escolher a última versão do Xampp.



2. Instalando o XAMPP;

Depois do download, clica no setup que se encontra no canto inferior na esquerda e somente clica em avançar.



Pronto agora que instalamos o IDE, NOTEPAD++ e o XAMPP, vamos colocar a mão na massa, agora vamos programar.

1.3 Script do lado do cliente

O Script do lado do cliente nada mais é do que o código fonte fica do lado do navegador, assim o servidor WEB, joga todos os códigos juntos com o conteúdo na máquina do cliente, exemplo o HTML, CSS e Javascript.

1.4 GUI e suas finalidade do lado do servidor

O Script do lado do servidor o código gerando pelo o programador no caso o PHP, fica no servidor WEB, e esse mesmo servidor manda somente os dados que o usuário solicitou, junto com o HTML, nesse caso nunca os usuários vão ver o seus algoritmos pensando e elaborados pelo programador.

GUI - **Graphical User Interface** (Interface Gráfica do Usuário, em português), é um modelo de interface entre o humano e a máquina que permite a interação com os dispositivos digitais através de elementos gráficos.

1.5 Sintaxe estrutura e POO

Para que possamos conhecer e escrever de acordo com as boas práticas temos que compreender a sintaxe de cada linguagem de programação, tipos regras e declarações, e no PHP não fica de fora dessas regras. A sintaxe nada mais é que a organização da escrita que temos que ter na hora de codificar. Hoje pode escrever o PHP na arquitetura estrutura e POO, na estrutura escrevemos mais, já na POO reutilizamos o que já foi codificado. Vamos entender agora o PHP estrutura, sendo que ambos usa a mesma sintaxe.

```
1 <html>
2 <body>
3 <?php
4     echo "Olá mundo";
5 ?>
6 </body>
7 </html>
```

Vale frisar que a linguagem PHP, fica embutido no HTML. Depois de escrever esse o código fonte salvamos com a extensão “.php”, no servidor Web e logo em seguida iniciamos os serviços de servidor, que no nosso caso é o XAMPP.

Atividade

- 1) Como podemos definir a linguagem PHP?
- 2) O que precisamos fazer para codificar a linguagem PHP?
- 3) Qual é a extensão do PHP?
- 4) Qual é a finalidade do XAMPP?
- 5) Para que serve o APACHE?
- 6) Explique sobre a sintaxe do PHP;
- 7) Qual é a função do comando echo “ ” ?

Desafio

Escreva um programa que imprimi na tela o seu nome completo e da escola;

2 - Variáveis e Tipos de dados



O PHP que vamos estudar como já foi dito do tema 01, vai ser a versão 6.0, na sintaxe procedural, toda linguagem de programação tem a sua documentação diante disso o vamos falar sobre a documentação do PHP, que é muito importante para nos programadores. Para acessar a documentação do PHP vamos entrar nesse link <https://www.php.net/>.

Nele vamos encontrar as informações atuais, dicas de novos comandos expressões, funções e até API, a dica é todo vez que surgir uma dúvida acesse esse site e procure sobre o que você precisa.

Na parte superior da tela na direita tem a caixa de procura basta você digitar o comando que você tem dúvida que o site vai lhe direcionar para uma ajuda objetiva e rápida.



2.1 Sintaxe do PHP

Como vimos no conteúdo passado entendemos que a sintaxe é a organização de qualquer linguagem, agora vamos nos aprofundar mais, sobre a sintaxe do PHP;

Para comentar no PHP usamos // para uma única linha e /* comentário em bloco */

Exemplo de comentário:

```
<?php

// for

for ($i=1; $i<=5; $i++) {
    I
}
```

Agora de uma maneira mais clara, estamos usando php com as tags de html, junto do seu escopo;

```
<html>
  <head>
    <title>PHP Teste</title>
  </head>
  <body>
    <?php echo "<p>Olá Mundo</p>"; ?>
  </body>
</html>
```

2.2 Tipos de variáveis

Variável como vimos nas aulas de lógica é um local temporário, para guardar dados de vários tipos, exemplo inteiro, decimal, texto e lógico. Agora vamos falar desses tipos só que usando o PHP. Os tipos de dados do usado no PHP, são básicos e compostos. Para declara uma variável no PHP, temos que usar o \$ do dinheiro sempre na frente do nome da variável, lembrando que seguimos os exemplos básicos para declarar qualquer tipo de variável.

Vamos falar sobre os tipos de variável de dados básicos;

String

String é um tipo de variável que só suporta texto, podendo ser uma letra uma palavra ou até um texto, conhecida também como caracter. Só que aqui no PHP é chamada string.

```
1 <?
2 //tipo de variável do tipo texto
3 $nome = "Anselmo";
4 $idade = "38";
5 ?>
```

Inteiro

Inteiro ou também chamado de int é um tipo de variável que só recebe números inteiros, não aceita números quebrados exemplo {... -4 -3 -2 -1 0 1 2 3 4 ...}.

```
1 <?
2 //tipo de variável do tipo integer
3 $valor1 = 250;
4 $idade = 38;
5 ?>
```

Float

Float é um tipo de variável que recebe números inteiros e quebrados, conhecido como decimal. Também falado ponto flutuante

```
1 <?
2 //tipo de variável flutuante
3 $salario = 3.000;
4 $altura = 1.75;
5 ?>
```

Boolean

Booleano ou lógico é um tipo de variável que recebe somente dois valores verdadeiro (true) e Falso (false), também 1 para verdadeiro e 0 para falso.

```

1 <?
2 //tipo de variável do tipo booleano
3 $filhos = false;
4 $curso = true;
5 ?>

```

Tipos de variáveis compostos

Array

Um array é um tipo de variável que pode armazenar mais de um valor ao mesmo tempo, possui um identificador que chamamos de índice.

```

1 <?
2 //tipo de variável do tipo array
3 $carros = array('gol', 'uno', 'palio');
4 ?>

```

Objeto

Objeto é um tipo de variável que recebe valores de classes, usado no paradigma POO, que vamos estudar mais no próximo modulo, segue um exemplo de como instanciar uma variável.

```

1 <?
2 //tipo de variável do tipo objeto
3 $clientes = new cadastro();
4 ?>

```

2.2 Comando de exibição print e echo

Print

Comando para exibir uma string na tela, por não ser uma função real não precisa usar parênteses com ele.

```

1  <?
2  //Elemento print
3  print "Seja bem vindo!";
4  ?>

```

Echo

Comando echo tem a mesma função do print, sendo o mais usado quando usamos o PHP, sendo assim ele também não é uma função real não precisa usar parênteses.

```

1  <?
2  //Elemento echo
3  $clientes = "Davi lopes"
4  echo $clientes;
5  echo "Seja bem vindo!";
6  ?>

```

2.3 Tipos de dados com funções ambiente

Funções são comando feito pelo desenvolvedor, criando uma espécie de rotinas, ao chamar essas rotinas será executada o conjunto todo. Podemos também ter uma função da própria linguagem que chamamos de função de ambiente.

Exemplo de função ambiente;

| Variável Ambiente (função) | Descrição |
|----------------------------|-----------|
|----------------------------|-----------|

| | |
|-----------------|--|
| Phpinfo() | Usado para trazer as informações da versão do PHP, APACHE e segurança do servidor local. |
| REMOTE_USER | Usada para obtermos o nome do usuário, no servidor. |
| SERVER_SOFTWARE | Para obter a versão do software do servidor |

Exemplo de função definida pelo programador

Funções definidas pelo usuário

Uma função pode ser definida usando a seguinte sintaxe:

```
<?php
function imprime ($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Exemplo de função.\n";
    return $valor_retornado;
}
?>
```

2.4 Concatenação

Este elemento é muito usado em qualquer linguagem de programação, sua finalidade é unir um conjunto de string, para exibir na tela. Usamos o ponto (.) para concatenar no PHP.

Segue um exemplo

```
1 <?
2 //Elemento concatenar
3 $aluno = "Carol almeida";
4 $curso = "Técnico em informática";
5 echo "Aluno".$aluno."Curso".$curso;
6 ?>
```

2.5 Conversão de dados

As variáveis declaradas no PHP, não tem logo na sua criação um tipo de dados específico ela vai se adaptar ou virar um tipo de dados conforme os dados que se atribui a ela, isso vale também para variáveis que vem derivadas de outras, Chamamos de coerção.

Veja alguns exemplos

```
8 $f = 5/3;
9 var_dump($f);
```

Linha 8 – Variável \$f está recebendo um cálculo de divisão, logo a variável vai ser do tipo float, valor decimal

Linha 9 – var_dump é uma função que identifica o tipo de dados de qualquer variável.

Função settype

Essa função permiti converte uma variável já declarada em qualquer outro tipo de dados.

Veja o exemplo

```
8 $g = 5/3;
9 settype($g, 'integer');
10 echo $g;
```

Linha 8 – Variável \$g está recebendo um cálculo de divisão, logo o tipo de dados dela é decimal

Linha 9 – A Função settype está convertendo a variável \$g em inteiro.

Linha 10 – Echo vai exibir o total do cálculo só que em valor inteiro.

2.6 Tipos de erros

Erros de lógica é o mais comum entre os desenvolvedores, quando isso acontece é quando você e a máquina não consegue ter o mesmo pensamento, nesse caso é bom descansar a mente e depois voltar a programar.

Outro erro comum é o de sintaxe, quando o programador digita errado na tela nesse caso vai aparecer essa mensagem.

Parse error: syntax error, unexpected '\$idade' (T_VARIABLE), expecting ';' or ','
in **C:\xampp\htdocs\aula02\index.php** on line **16**

Veja que “Parse error: Syntax”, vai ajudar você entender que foi digitado um comando errado que não existe, no final dessa mensagem está aparecendo “line 16” significa que o problema está na linha 16.

Outros erros que vão aparecer na sua aplicação.

| Constante | Descrição |
|---------------------|--|
| E_ALL | Todos erros e avisos são exibidos. |
| E_ERROR | Erros fatais de tempo de execução. Eles indicam erros que não podem ser recuperados, como um problema de alocação de memória. |
| E_RECOVERABLE_ERROR | Erro fatal captável. Indica que ocorreu um erro provavelmente perigoso, mas não deixou a Engine em um estado instável. |
| E_WARNING | Avisos em tempo de execução (erros não fatais). A execução do script não é interrompida. |
| E_PARSE | Erros de parse em tempo de compilação. Os erros de parse devem ser gerados apenas pelo parser. |
| E_NOTICE | Notificação de tempo de execução. Indica que o script encontrou algo que poderia indicar um erro, mas também poderia ocorrer no curso normal da execução de um script. |

Atividade

- 1) Qual é a função de uma variável?
- 2) Quais os tipos básicos de variáveis que temos no PHP?
- 3) Qual é a diferença de variável do tipo booleano e float?
- 4) O que é concatenar?
- 5) Quais são os tipos de variáveis compostas?
- 6) Qual é a função do comando print?

Desafio

- 01) Faça um programa que mostre o nome, idade, escola de um aluno do nosso curso, usando variáveis, echo e o concatenar;
- 02) Faça uma pesquisa desses comando na documentação do PHP explicando cada um deles;

`session_unset();`

`mysql_query();`

`isset();`

`include();`

md5());

3 - Operadores



Dentro de qualquer linguagem de programação temos os operadores, que tem a função que indica o tipo de operação matemática se é de texto, logico entre outros aqui será abordado os principais, esses operadores será executada gerando novos valores a partir do seu chamado. E a linguagem de programa PHP, usa vários tipos de operadores que vamos ver neste capítulo.

3.1 Tipos de operadores

O PHP oferece vários tipos de operadores, vejamos alguns exemplos como unário, binário, ternário, matemáticos e lógicos dentre outros.

3.2 Operadores aritméticos

Os operadores aritméticos são responsáveis por fazer os cálculos tradicionais usado na nossa matemática.

Segue os exemplos

| Sintaxe | Operação |
|-------------|------------------|
| \$v1 + \$v2 | Adição |
| \$v1 - \$v2 | Subtração |
| \$v1 * \$v2 | Multiplicação |
| \$v1 / \$v2 | Divisão |
| \$v1 % \$v2 | Resto da divisão |
| -\$v2 | Negação |

Exemplo de uma operação aritmética, usando o operador de soma:

```

7  <?php
8      $v1 = 15;
9      $v2 = 12;
10     $total = $v1+$v2;
11     echo "Total ".$total;
12  ?>

```

3.3 Operadores de comparação

Operadores de comparação ou também chamados de condicionais, tem a finalidade dentro da sintaxe do PHP de comparar valores e dados na programação.

| Sintaxe | Operação |
|-------------|---------------|
| \$b == \$c | Igualdade |
| \$b === \$c | Idêntico |
| \$b != \$c | Não igual |
| \$b <> \$c | Diferente |
| \$b !== \$c | Não idêntico |
| \$b < \$c | Menor |
| \$b > \$c | Maior |
| \$b <= \$c | Menor e igual |
| \$b >= \$c | Maior e igual |

Exemplo usando um operador de comparação:

```

1  <?
2      //fluxo de dados simples usando o ternário
3      //com operador de comparação
4      $b = 10;
5      $c = 100;
6      echo ($b>$c?"b é maior q c":"C é maior B");
7  ?>

```

3.4 Operadores String

Na linguagem de programa do PHP, temos dois operadores de STRING, no caso junção de dois ou mais texto.

| Sintaxe | Operação |
|-------------------|----------------------------|
| \$nome . \$sobre | Concatenação "União" |
| \$nome .= \$sobre | Atribuição de concatenação |

Exemplo de operador de String

```

14 <?php
15     $nome = "Carlos";
16     $sobre = "Fives";
17
18     $todo = $nome . $sobre;
19     echo $todo. "<br/>";
20     $nome .= $sobre;
21     echo $nome;
22 ?>

```

3.5 Operadores incremento e decremento

No PHP temos dois operadores de aumentar e diminuir respectivos valores, incremental e decrementar valores dentro de variáveis. Eles podem ser pré -incremental e pós – decremental.

Exemplos:

Incremental (++)

Pré e pós incremental

Decremental (--)

Pré e pós decremental

Significa que se o sinal for colocado antes da variável, primeiro será executado subtraído ou acrescentado sendo pré, logo o sinal sendo depois fica sendo pós decrementa ou incremental, veja um exemplo na sintaxe do PHP.

```

23 <?php
24 //Incremento pré e pós
25     //++
26     ++$a
27     $a ++
28 //decremento pré e pós
29     //--
30     --$a
31     $a--
32 ?>

```

3.6 Operadores lógicos

Operadores lógicos são usados para a parte lógica de entrada de fluxo de dados, das operações de comparações. Veja os exemplos

| Sintaxe | Operação |
|-------------|---|
| \$a && \$b | E o valor será verdadeiro quando as duas variáveis são verdadeiras |
| \$a \$b | OU – O valor será verdadeira quando umas das variáveis é verdadeiras |
| \$a xor \$b | Exclusivo – quando ambas forem verdadeiras |
| ! \$a | Não – A variável será verdadeiro quando o valor de \$a não for verdadeiro |

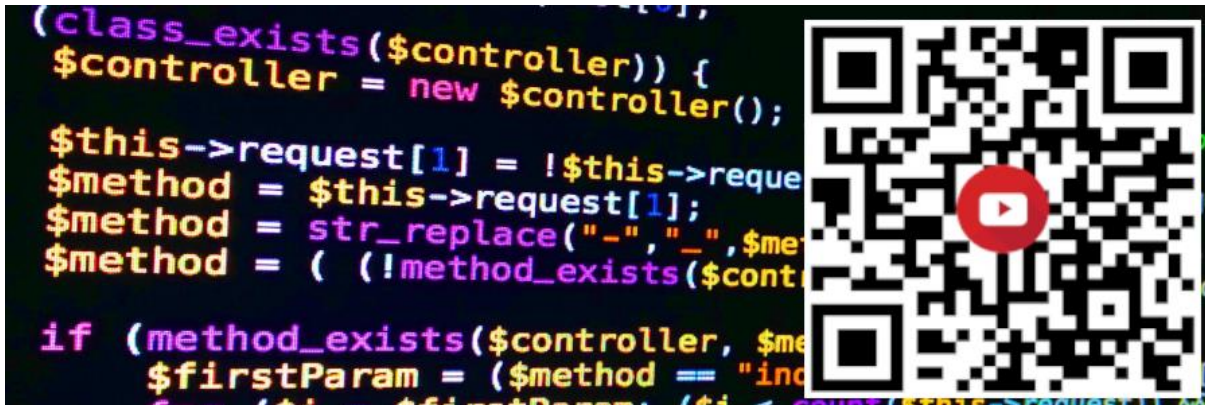
Atividade

- 1) Qual é a função dos operadores?
- 2) Qual é o caractere usado para definir uma variável no PHP?
- 3) Qual é o nome do operador que usamos para cálculos matemáticos?
- 4) Qual é a diferença do operador && e ||?
- 5) Qual é a diferença no PHP, usando um sinal de igual e dois sinal de igual?

Desafio

- 01)Faça um algoritmo no PHP, que solicita três valores e soma os mesmo, usando os elementos echo;
- 02)Faça um algoritmo no PHP, que solicita três notas de um aluno e mostre na tela a média dele e o nome, usando os elementos echo e ternário;

4 - Estrutura de controle



4.1 Introdução e conceito

Estrutura de controle em qualquer linguagem de programação é o fluxo de controle dos dados de entrada e saída, vejamos alguns exemplos práticos na sintaxe do PHP.

4.2 If e else simples e composto

Dentro da programação vamos ter muitas situações de decisão, aí que entra o fluxo de controle conhecido como condicional. Caso essa decisão ou expressão seja avaliada como verdadeiro ou falso a mesma vai partir para uma sequência de comando que será logo executada. Vamos falar agora da estrutura if e depois do else, uma das estruturas mais usadas dentro de qualquer linguagem de programação.

If

Caso a expressão seja verdadeira (true), a programação vai seguir um caminho.

```
34 <?php
35 if(10==10) {
36     echo "10 é igual a 10";
37 };
38 ?>
```

If e else

Já a expressão if com else, temos agora o outro lado da questão caso seja falso, determinamos uma sequência lógica.


```

34 <?php
35 if(10==10) {
36     echo "10 é igual a 10";
37 }
38 else
39 {
40     echo "10 não é igual a 10";
41 };
42 ?>

```

Elseif

Agora vamos tratar-se de uma combinação entre else e if, colocado anteriormente nesse capítulo. Agora vamos colocar if e else um dentro do outro chamado de composto. Estrutura de fluxo uma dentro da outra.

```

44 <?php
45 $hora =date('H');
46 if($hora >= 7 && $hora <=12)
47 {
48     echo "Acorda, Bom dia!";
49 }
50 elseif($hora >= 13 && $hora<=18)
51 {
52     echo "Hora do café, Boa tarde!";
53 }
54 elseif($hora>19 && $hora <= 23)
55 {
56     echo "Tomar banho,Boa noite";
57 }
58 else
59 {
60     echo "";
61 }
62 ?>

```

4.3 Switch

O comando condicional Switch no PHP ou em qualquer linguagem de programação segue a mesma sintaxe, e com a mesma similaridade do If e else, a diferença em ambos é que o switch usa somente a clausula case no caso somente se for verdadeiro.

```

63 <?php
64 $dia = 1;
65 switch($dia) {
66     case 1:
67         echo "Segunda";
68         break;
69     case 2:
70         echo "Terça";
71         break;
72     case 3:
73         echo "Quarta";
74         break;
75     default:
76         echo "Opção inválida";
77 }
78 ?>

```

4.4 Comando de repetição

Comando de repetição tem a função de determinar o número de vezes que um bloco de uma sequência de instruções pode ocorrer, uma repetição colocada pelo o programador, ajuda a diminuir uma codificação dentro da sintaxe. Vamos falar nesse capítulo dos principais comandos de repetição usados no PHP, sendo while, do .. while, for e foreach.

While

O mais simples de ser colocado em ação, porém nesse caso usando o While sabemos o início mas não o fim, da repetição. Veja o exemplo.

```

1 <?php
2
3 $num=1;
4
5 while ($num<=5)
6 {
7     echo $num."<br/>";
8 }
9 ?>

```

Do ... while

Muito parecido com o while, porém mesmo sendo falso ele repete a primeira vez, não precisando fazer a rotina de verificação.

```
<?php
    $fin = 1000;
    $serie = 1;
    $numeroAnterior = 1;
    $numeroposterior = 1;

    do
    {
        echo($serie . ", ");
        $serie = $numeroAnterior + $numeroposterior;
        $numeroAnterior = $numeroposterior;
        $numeroposterior = $serie;
    }
    while($serie < $fin)
?>
```

For

Este comando é o loop que repete o início e fim específico determinado pelo programador, diferente do while que só sabemos o início.

```
<?php
    for($i=0;$i<10;$i++){
        echo $i;
    }
?>
```

Foreach

O comando foreach é um laço de repetição usado para Array, sendo mais fácil a interação dentro de um conjunto usando matriz e vetor, também usado para interagir uma lista de dados vindo de uma base de dados. Na prática vamos ver ele quando entrarmos no capítulo de banco de dados.

```
1 <?php
2 $salaries = array();
3 $salaries["Anna"] = 5000;
4 $salaries["John"] = 6000;
5 $salaries["Aamir"] = 10000;
6
7 foreach($salaries as $key_arr => $val_arr){
8     echo $key_arr . " = " . $val_arr . "<br />";
9 }
10 ?>
```

4.5 Break

O comando ou também conhecido como um elemento dentro da sintaxe do PHP, será usado ou chamada para parar uma instrução, podendo ser de loop laço de repetição ou até de uma condição

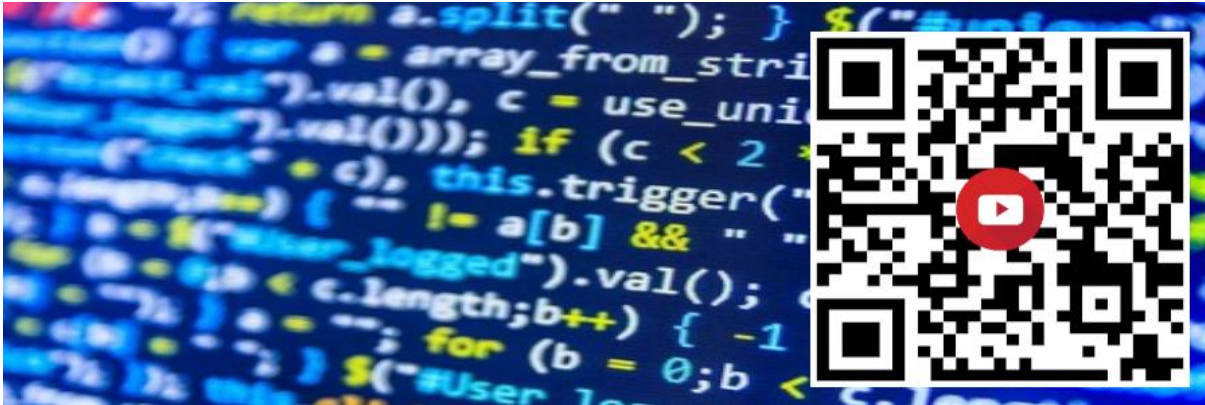
4.6 Continue

O comando ou também chamado de elemento dentro da sintaxe do PHP, vai ser usado quando paramos uma instrução no Break, sendo então para a lógica voltar a correr usamos o continue.

Atividade

- 1) Qual é função de uma condição dentro da programação?
- 2) Explique sobre o bloco if e else;
- 3) Qual é a diferença entre o if e Swicth?
- 4) A onde aplicamos os comandos de fluxo de controle for e while?
- 5) Qual é a função do laço de repetição na programação?
- 6) Faça um algoritmo de uma tabuada 1 ao 10, usando o laço de repetição for.
- 7) Faça um programa em PHP que defina uma variável com o peso e a altura de uma pessoa. Calcule e mostre o IMC.
- 8) Faça uma página em HTML5 que leia o placar de um jogo de futebol (os gols de cada time, um valor em cada variável do tipo int). Informe se houve empate ou se a vitória foi do 1º ou do 2º time. A entrada pode ter o nome dos dois times.
- 9) Faça um programa em PHP que mostre todos os números inteiros de 100 a 200 com incremento de 2 em 2.
- 10) Faça um programa em PHP que apresente todos os valores ímpares no intervalo de 500 a 1000

5 - Conceito sobre Array



Vamos agora falar um pouco sobre Array, sendo uma rotina de tarefas muito usado em qualquer linguagem de programação, dentro do PHP vamos ver a importância desse comando ou função.

A finalidade de um Array é guardar vários dados economizando linhas de códigos, porque ao declarar uma variável do tipo Array ela pode guardar muitos dados e organizando por índice.

5.1 Introdução sobre tipos e Array

No PHP, os array podem ser criados por meio do construtor de linguagem `array()`, sendo composto por parênteses e chaves, na sintaxe.

5.2 Criando um Array e seus tipos

Array do tipo vetor:

Os dados guardado em um Array do tipo vetor são organizado por colunas e chamado por índice, todos os tipos de Array tem uma sequência logica iniciado por zero, então para chamar um valor dentro de um Array, chamamos por esse número que é rotulado de índice.

Variável \$cor está recebendo três valores veja como fica a alocação desse tipo de variável.

| | | |
|-----------|--------|--------|
| 0 Amarelo | 1 Azul | 2 Rosa |
|-----------|--------|--------|

Cada valor alocado nessa variável, vai ser organizado em índice que inicia com zero, para chamar cada uns desses valores vamos chamar eles por números que chamamos de índice ou chave.

Veja o exemplo

```
81 <?php
82     $cor = array("Amarelo","Azul","Rosa");
83     //exibir todos os dados
84     echo $cor;
85     //exibir somente o índice zero que é amarelo
86     echo $cor[0];
87 ?>
```

Array do tipo matriz:

Os dados guardado em um Array do tipo matriz, são guardados e organizados também pelo índice, só que nesse caso guardamos em uma tabela fictícia baseado em linhas e múltiplas colunas. Veja o exemplo

```
90 <?php
91 $cadastro = array
92 (
93     array("Anselmo","Renata","Lucas"),
94     array("Ana","Ticiane","Roberta"),
95     array("Daniela","Adriana","Maria"),
96 );
97 //linha 0 e coluna 1, vai exibir Roberta
98 echo cadastro[0][1]
99 //linha 2 e coluna 0, vai exibir Daniela
100 echo [2][0]
101 ?>
```

5.3 Modificando e mesclando

Para modificarmos um array, é pela sua configuração explícita de valores neles. Para isso, basta inserirmos valores no array, com a chave específica para cada dados entre colchetes. Veja o exemplo

```

104 <?php
105 $cadastro = array
106 (
107     array("n1"=>"Anselmo", "n2"=>"Renata"),
108     array("n3"=>"Ana", "n4"=>"Ticiane"),
109     array("n5"=>"Daniela", "n6"=>"Adriana"),
110 );
111 //agora vamos chamar pelo nome do indice vai
    exibir Anselmo e Daniela
112 echo cadastro[n1][n5];
113
114 ?>
    
```

Para mesclar valores dentro de um array temos que usar uma função específica que é `array_merge`, mesclar é unir duas ou mais array em uma só, uma junção de dados.

```

116 <?php
117 $alunas = array("Andrea", "Julia", "Mariana");
118 $alunos = array("Carlos", "Sergio", "Luiz");
119 //usando a função array_merge vai exibir todos
    //os alunos
120 print_r(array_merge($alunas, $alunos));
121
122 ?>
    
```

Atividade prática

- 1) O que é array?
- 2) Qual é a diferença entre vetor e matriz?
- 3) O que é índice?
- 4) O vetor e a matriz inicia sua organização com o número 1 ou 0?
- 5) Qual é a finalidade do comando `array_merge()`?

Desafio

Faça uma matriz que suporte todos os seu filmes e gêneros que você mais assistiu no ano de 2021, e exiba dois filmes com o seu gênero;

6 - Método de envio com GET e POST



Dentro das linguagem de programação precisamos ter um maneira para enviar os dados entre a própria linguagem e para a base de dados, para enviar os dados que se encontra no formulário do html, vamos utilizar o método post e get, que colocamos dentro da tag form no parâmetro method.

6.1 Introdução sobre o conceito de envio GET e POST com HTML 5

Os métodos que vamos ver agora são usados na sintaxe de estrutura do PHP, sendo o get meio de enviou pela url e o post por maneira de aquisição de dados, ambos seguros mais com maneira de envios diferentes. Os dois métodos que vamos estudar agora precisa do formulário do html.

6.2 Método POST e GET

Método GET

Nesse método os dados são enviados juntamente com o nome da página, no caso pela url, método muito usado por ser tratar de uma maneira rápida de envio ele não suporta dados com um poder muito grande. Esse método é considerado o padrão do HTML, caso não especificado no method, ele que sempre vai está disponível.

Formulário do HTML

```

1 <html>
2 <head>
3     <title>Aula de método</title>
4 </head>
5 <body>
6     <form action="pag.php" method="get">
7         Nome: <input type="text" name="cxnome">
8         <input type="submit" value="Enviar">
9     </form>
10 </body>
11 </html>

```

Form é a tag usado no html para criar um formulário

Action é para onde os dados vão quando clicar no botão enviar no caso vai para a pagina "pag.php".

Name é o parâmetro usado para rotular a caixa do input.

Method é o método que vamos usar para enviar os dados no caso agora vai ser o GET.

Pagina que vai receber os dados

Pag.php

```

124 <?php
125 $nome = $_GET["cxnome"];
126 echo "Nome: " . $nome;
127 ?>

```

\$nome é a variável que está recebendo o nome pela a caixa input dado o **name de cxnome**.

\$_GET[" "] é o parâmetro que sempre vamos usar para receber os dados do formulário, dentro dos colchetes colocamos o nome do input.

Método POST

No método POST, não é possível passar dados no caso parâmetros pela URL, somente pelo formulário, por aquisição de informação do usuário. Ele também vai seguir a mesma sintaxe usado no GET, considerado um método mais seguro por não mostrar na url para onde

os dados estão sendo enviados. Método muito usado para guardar dados no servidor de dados, com maior poder de gerenciar dados, para criar um CRUD, esse método é o mais recomendado.

Formulário do HTML

```
1 <html>
2 <head>
3     <title>Aula de método</title>
4 </head>
5 <body>
6     <form action="pag.php" method="post">
7         Nome:<input type="text" name="cxnome">
8         <input type="submit" value="Enviar">
9     </form>
10 </body>
11 </html>
```

Seguindo o mesmo exemplo do método get, só mudamos para POST no parâmetro method.

Pagina que vai receber os dados

```
124 <?php
125 $nome = $_POST["cxnome"];
126 echo "Nome: ".$nome;
127 ?>
```

Seguindo a mesma ideia do método get, agora a variável muda para `$_POST[" "]`, para receber do formulário que está usando o método post.

6.3 Include e Require suas diferenças e particularidades

Vamos agora incluir arquivos em páginas PHP, usando os comandos `require()` e `include()`, que são responsáveis por realizar a leitura das tag PHP e executar dentro delas, por via de código.

A grande diferença entre esses dois elementos que vamos falar agora é que o `include()`, trabalha em tempo real, exemplo podemos usar ele nos laços de repetição, já o `require()` não tem esse time real.



Include_once e require_once

Estes dois comando também chamado de expressões trabalha de maneira idêntica ao `include` e `require` padrão, a única diferença entre eles quando usamos o `_once`, o arquivo será incluído uma única vez ou seja executado só um de cada vez, por isso que vamos usar esses dois comando nas conexão de banco de dados.

Veja o exemplo:

Arquivo: pagina.php

```
130 <?php
131     $escola = "Proz - Educação Profissional";
132 ?>
```

Arquivo: pagina2.php

```
134 <?php
135     include "pagina.php";
136
137     echo "Escola ".$escola;
138 ?>
```

As páginas que vão receber os arquivos, no cabeçalho da página precisamos digitar o comando `include` ou `require`, ai tudo que temos nessa página podemos reutilizar.

Atividade prática

- 1) Qual é o método usado para enviar dados via URL?
- 2) Qual é a função dessas duas expressões de variáveis, `$_GET[" "]` e `$_POST[" "]`?
- 3) Explique sobre o `include` e `include_once`;
- 4) Explique sobre essa linha de comando `<form action="pag.php" method="POST">`
- 5) Qual é a principal diferença entre `include` e `require`?

Desafio

Faça um quiz usando PHP que:

- Na primeira tela receba o nome do participante;
- Faça 6 perguntas em sequência usando páginas diferentes;
- Na 7ª página exiba o resultado com a quantidade de acertos do participante juntamente com um botão para zerar o resultado e voltar ao início.

7 - Tipos de Funções



As funções são rotinas criadas para efetuar diversos serviços, estes serviços são tarefas que tem uma finalidade dentro do código fonte que estamos escrevendo, basta utilizar a função adequada para que não aja a necessidade de repetir um mesmo código já escrito por diversas vezes, chamando simplesmente esta ação em qualquer lugar do seu código.

7.1 Introdução sobre funções

Temos dois tipos de funções dentro do PHP, as nativas e as internas, as nativas são funções desenvolvidas pelos os engenheiro de software que organiza a linguagem do PHP, já as funções consideradas internas são as que vamos criar no decorrer no nossos projetos, software e sistema web.

Vamos agora criar nossas próprias funções, dentro do PHP para isso temos que respeitar algumas regras e boas práticas da linguagem seguindo sua sintaxe.

7.2 Funções Ambientes nativas e criação

Sintaxe de uma função

```

141 function nomedafuncao(Arg ..... ){
142     codigo;
143     return
144 }
145 ?>
    
```

Function – Comando para criar uma função no php, sempre minúscula.

Nome da função sempre temos que seguir as boas práticas da programação igual quando declaramos uma variável.

Código – Onde colocamos o trecho da tarefa do código.

Return – Caso a função tem um retorno no argumento é usado de forma opcional.

Segue um exemplo de uma função simples sem retorno

```
146 <?php
147     function soma () {
148         $v1 = 10;
149         $v2 = 20;
150         $total = $v1+$v2;
151         echo "Total ".$total;
152     }
153     //para executar a função
154     soma ();
155 ?>
```

Função simples com o objetivo de fazer um cálculo de adição, para chamar a função em qualquer local do código basta colocar o nome dela com os parênteses no final do nome.

Veja um outro exemplo de função com argumento;

```
146 <?php
147 $cx1 = $_POST["cxvalor1"];
148 $cx2 = $_POST["cxvalor2"];
149
150 function soma($valor1,$valor2){
151     $v1 = $valor1;
152     $v2 = $valor2;
153     $total = $v1+$v2;
154     return $total;
155 }
156 //para executar a função
157 echo soma ($cx1,$cx2);
158 ?>
```

Uma função com argumento, usando o retorno de duas variável.

7.3 Funções de texto, data, codificadas e numéricas

Função de texto – Retorna um carácter específico.

```
160 <?php
161     echo chr(65);
162 ?>
```

Função de data – Permite a formatação da data e da hora de um local.

```
170 <?php
171
172 echo date("d/m/y H:i:s");
173
174 ?>
```

Funções codificadas – retornando uma matriz de string, frações da string original.

```
163 <?php
164
165 $carros = "gol - palio - corsa";
166 $array_carro = explode("-", $carros);
167 var_dump($array_carro);
168 ?>
```

Funções numéricas – Formatar uma string

```
176 <?php
177 echo number_format(2000, 2, ',', '.');
178 ?>
```

Atividade prática

- 1) O que são funções?
- 2) O que uma função retorna quando não usa o comando return?
- 3) Qual é a função chr?
- 4) O que é funções nativas?

5) O que é funções ambiente?

Desafio

- 1) Entrar com nome, sexo e idade de uma pessoa. Se a pessoa for do sexo feminino e tiver menos que 25 anos, imprimir nome e a mensagem: ACEITA. Caso contrário, imprimir nome e a mensagem: NÃO ACEITA.
- 2) Entrar com 3 números e imprimi-los em ordem decrescente (suponha números diferentes).

8 – Tipos de Cookies e Sessões



Vamos aprender sobre os o meio de autenticação que temos na linguagem de programação do PHP, e defini-los e a acessar aqueles que foram enviados. Além dos cookies

8.1 Introdução sobre segurança usando cookies e Sessões

Os cookies são arquivos gerados pelo servidor web responsável por gerenciar e guardar os sites. São arquivos como fotos, vídeo e texto usando uma memória local do seu navegador, cookies nada mais é que pedaços de sites, que acessamos diariamente.

Sessão ou também chamada de session ou variável de sessão é muito parecido com a sessão de login usado nos sistemas operacionais, seria uma espécie de segurança a onde as informações, são armazenados em um local da memória e podendo ser usado em qualquer lugar da tela. As variáveis de sessão armazenam informações sobre um único usuário e estão disponíveis para todas as páginas em um único aplicativo. Exemplo clássico de sessão é o carrinho de compra, em um site de e-commerce, uma espécie de variável que não usa a base de dados e sim a memória do navegador.

8.2 Definição de cookies e aplicando

A definição perfeita de cookies, lembra bem um biscoito restos de arquivos de uma site ou sistema que fica armazenado no seu navegador, quando você acessar novamente aquele site, ele vai ser processado mais rápido. Veja como aplicar ele no nosso sistema usando a linguagem PHP.

Como aplicar o cookies:

Vamos usar a função `setcookies()`, ou da função `strawcookies()` que é nativo do PHP definido antes de qualquer saída porque faz parte do HTTP, do sistema WEB.

```
180 <?php
181
182 if(!isset($_COOKIES['fotodefundo'])) {
183     setcookie("fotodefundo", "url('img/fundo.jpg')",
184         time()-86400);
185     $cookies = "Alterada com sucesso!
186     <br/> F5 para atualizar";
187 }
188 $bk = $_COOKIES["fotodefundo"];
```

Linha 182 – Condicional para verificar se tem a foto de fundo, caso seja verdadeiro vai aparecer essa foto o dia todo, todo vez que ele abrir seu sistema web.

Linha 187 – Variável `$bk` está recebendo a imagem;

```
190 <body style="background-image:<?php echo $bk ?>">
```

Linha 190 – No HTML no corpo do site vamos atribuir a variável `$bk`, para a troca de fundo a cada dia.

```
192 <?php echo $cookies ?>
```

Linha 192 – Mensagem para exibir que foi alterado com sucesso o fundo do site.

8.3 Definição de sessão e aplicando

Sessão é uma variável que guarda um valor logo no início na aplicação e pode ser usada em qualquer lugar da aplicação, muito útil para trabalhar com segurança de dados.

```
195 <?php
196 session_start();
197 $_SESSION["usuario"] = "Anselmo Paulo";
198 echo "Usuário logado ".$_SESSION["usuario"];
```

Linha 196 – Início da sessão temos que sempre usar essa função, para executar a sessão, reforçando em todas as páginas que for usar a variável de sessão.

Linha 197 – Variável atribuída de sessão com o nome usuário que está recebendo o nome de usuário, que pode ser da base de dados ou até um cálculo de compra.

Linha 198 – Exibir o nome do usuário que está na variável.

Em todas as páginas vamos iniciar a sessão e logo vamos chamar a variável, que foi atribuída a ela.

Todo sistema tem o botão logoff ou sair, então essas informações não pode ficar armazenada no navegador, temos que limpar todas as configurações. Veja o comando para finalizar a sessão.

```
1 <?php
2 session_start();
3
4 session_destroy();
5 ?>
6 <script language="javascript" type="text/javascript">
7
8 document.url="index.php";
9 window.location.href="index.php";
10
11 </script>
```

Linha 2 – Início da sessão

Linha 4 – destruindo todas as variáveis de sessão que está na memória do PC, no caso no navegador.

Linha 6 – Iniciando um javascript para retornar a página principal do sistema sem o acesso ao sistema.

Linha 8 e 9 – Tags de javascript para direcionar para a página principal.

8.4 Configurando as sessões

As configurações das sessões são muitas, vou apresentar algumas para configurar vamos no arquivo php.ini, que se encontra na pasta do Xampp.

Session.save_handler

`session_set_save_handler()` define as funções de armazenamento de sessão em nível de usuário que são usadas para armazenar e recuperar dados associados a uma sessão. Isso é mais útil quando um método de armazenamento diferente daqueles fornecidos pelas sessões de PHP é preferido, por exemplo, armazenar os dados da sessão em um banco de dados local.

Session.save_path

`session_save_path ()` retorna o caminho do diretório atual usado para salvar os dados da sessão.

path

Caminho de dados da sessão. Se especificado e não null, o caminho no qual os dados são salvos será alterado. `session_save_path ()` precisa ser chamado antes de `session_start ()` para esse propósito.

Session.name

`session_name ()` retorna o nome da sessão atual. Se nome for dado, `session_name ()` irá atualizar o nome da sessão e retornar o antigo nome da sessão.

Session.auto_start

`session_start ()` cria uma sessão ou retoma a atual com base em um identificador de sessão passado por meio de uma solicitação GET ou POST ou por meio de um cookie.

Session.cookie_lifetime

`session_set_cookie_params` - Definir os parâmetros do cookie da sessão

Atividade Prática

- 1) Que é cookies?
- 2) Qual é a finalidade da função `setcookies()`;
- 3) Explique sobre esse código: `$bk = $_COOKIES["fotodefundo"];`
- 4) O que é sessão?
- 5) Qual é declaração que colocamos no início do código para executar uma sessão?
- 6) Explique sobre esse código: `$_SESSION["usuario"] = "Anselmo Paulo";`
- 7) Todos os navegadores suportar cookies?
- 8) Qual é o código usado para destruir uma sessão?

Desafio

Página 1: Salve com o nome `index.php`

Faça um formulário que receba as seguintes informações:

- Nome:
- Idade:
- Peso:
- Altura:
- Guarde essas informações nas variáveis de sessão

Pagina 2: Salve com o nome pagina2.php

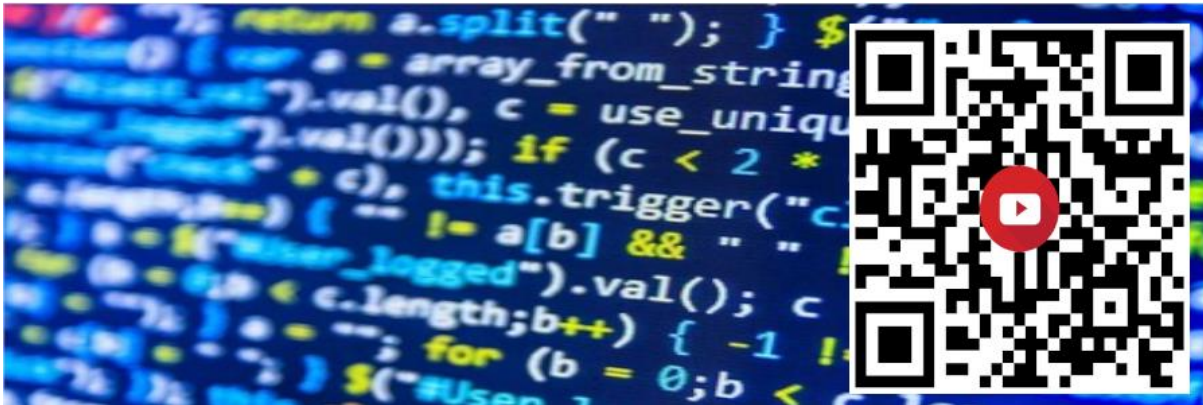
Faça outro formulário com os mesmos campos para digitar os dados de uma outra pessoa e também guarde essas informações em variáveis de sessão.

Pagina 3: Salve com o nome pagina3.php

Nesta tela compare e mostre na tela:

- Quem tem uma altura maior
- Quem é mais velho
- Quem tem o peso maior

9 – Conexão com o banco de dados



Vamos agora aprender uma das funções mais importante de todas as linguagem de programação, podemos até colocar como o ponto central de qual tipo de linguagem, é a conexão com o banco de dados, vamos entender e aprender sobre a linha entre a base de dados com o PHP.

É muito importante, reforçar que o PHP é uma das linguagem mais segura em questão de conexão com o banco de dados, sendo assim é agora que vamos codificar o PHP e com a base de dados.

Após criar o sistema WEB usando a linguagem de programação PHP, devemos selecionar um provedor web para hospedar a nossa base de dados. Este serviço que chamamos de provedor, tem vários tipos de SGBD, e ai que vamos escolher o MySQL.

9.1 Introdução de tipos de SGBD

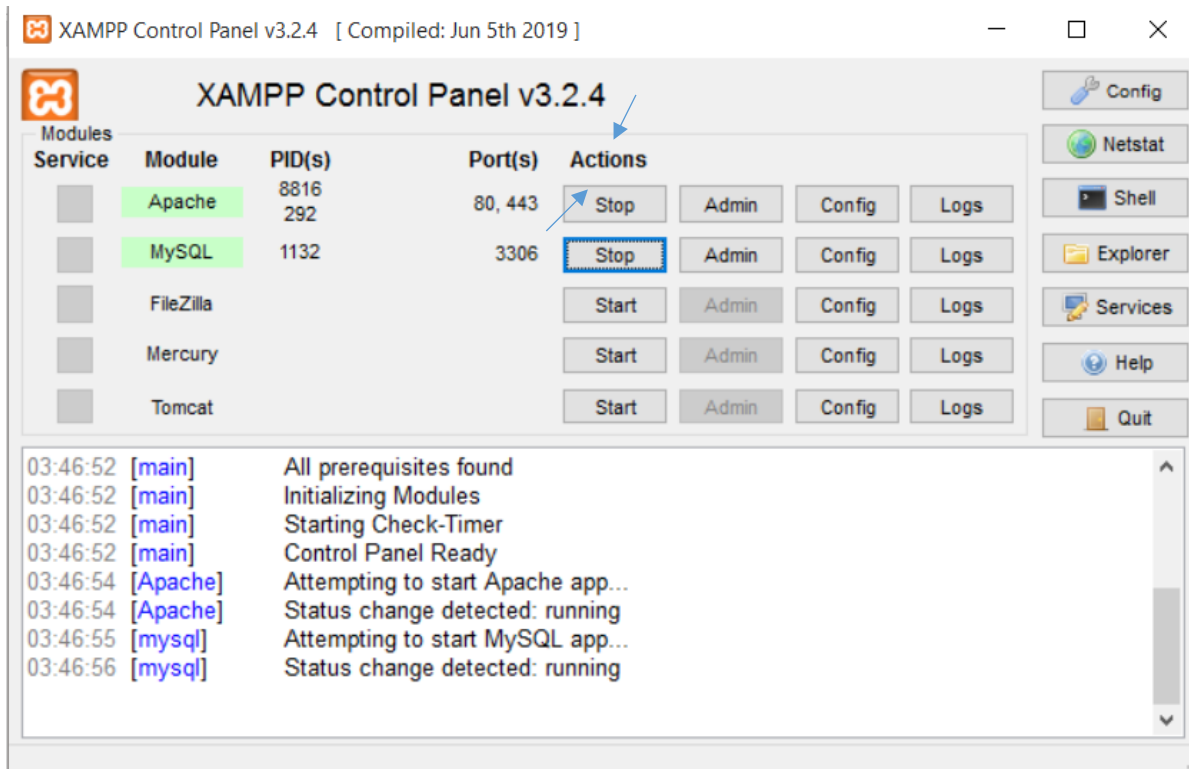
SGBD é uma sigla de sistema de gerenciamento de banco de dados, toda linguagem de programação precisa ter acesso ao banco de dados, então precisamos ter um SGBD que é o gerenciador da base de dados, nesse capítulo vamos falar do MySQL, um SGBD usado parar as grandes redes de computador WEB.

O MySQL é recomentado para trabalhar com o PHP, por ser open source, e conectividade mais leve e segura.

9.2 Criando uma base de dados

Primeiro passo para usar o PHP com conexão com o SGBD, MySql vamos entender como criar o CRUD, segue um passo a passo.

1. Executar o Xampp



Para iniciar o Xampp, vamos no disco local (C:) procura a pasta xampp e depois procura o arquivo xampp-control.

2. Acessar o navegador de sua preferência, e digite na url <http://localhost/phpmyadmin>, vai abrir o gerenciador do banco de dados MySQL.

3. Clicar na guia sql;

4. Para criar uma base de dados vamos digitar esse código

```
28 CREATE DATABASE bdloja2022;
```

5. Para selecionar a base de dados:

```
30 use bdloja2022;
```

6. Código para criar uma tabela;

```
1 create table tbcliente(
2     codigo int auto_increment primary key,
3     nome varchar(90),
4     idade int
5 );
```


7. Inserir dados na tabela

```
7 insert into tbcliente
8 (nome,idade)
9 values
10 ('Mariana',32);
```

8. Código para selecionar todos os dados

```
12 Select *from tbcliente
```

9. Código para selecionar um dado por código ou nome

```
14 /*consulta por codigo*/
15 select *from tbcliente where codigo=1;
16 /*Consulta por nome*/
17 select *from tbcliente where nome="mariana";
```

10. Código para alterar um dado na tabela

```
19 update tbcliente
20 set
21 nome = 'renata',
22 idade = 'california'
23 where
24 codigo=1
```

11. Código para excluir um dado na tabela

```
26 delete from tbclientea where Codigo=1;
```

9.3 Usando PHP com conexão com a Base de dados

Agora que vimos a construção de uma base de dados, em MySql chegou o momento da junção entre a linguagem de programação e o SGBD, vamos criar primeiro um formulário para a entrada de dados e usar essa base que acabamos de criar, para que o aja uma conexão entre o HTML e o PHP vamos criar um arquivo de conexão.



Formulário HTML


```

1 <html>
2 <head>
3   <title>Cadastro de Dados</title>
4 </head>
5
6 <body>
7   <form method="POST" action="inserir.php">
8
9   Nome:<input type="text" name="cnome">
10  idade:<input type="text" name="cidade">
11  <input type="submit" value="Cadastrar">
12
13 </form>
14 </body>
15 </html>

```

Método de envio vai ser o POST, o arquivo que vamos enviar os dados para a base de dados, vai ser o inserir.php.

Conexão

```

1 <?php
2 $server = "localhost";
3 $user = "root";
4 $password = "";
5 $dbname = "bdloja2022";
6
7 $conn = mysqli_connect($server,$user,$password,
8   $dbname);
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Variável \$server – está recebendo o local da base de dados

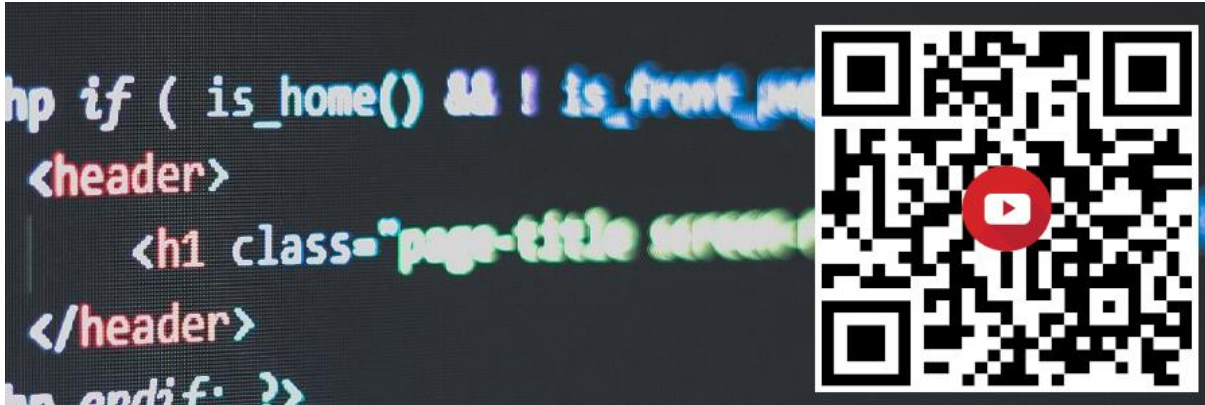
Variável \$user – está recebendo o usuário atual da base de dados

Variável \$password – está recebendo a senha da base de dados

Variável \$dbname – está recebendo o nome do banco de dados

Variável \$conn – está recebendo o comando de conexão do banco de dados mysqli_connect() que vai buscar as informações nas variáveis e fazer uma autenticação, caso tudo esteja certo o PHP vai ler a base de dados.

9.4 Criando um CRUD com PHP



CRUD é uma simplificação das quatro operação básica de qualquer base de dados, sendo que as quatro operações básicas são “C” create ou inserir, “R”, consultar ou select, “U”, alterar ou update, “D” deletar ou excluir. Todo sistema sendo WEB ou desktop, vai usar essas quatro operações, se analisarmos vamos ver que todos os sistema que existe vai ter que ter inserção, consulta, alteração e exclusão. Vamos agora fala de cada uma dessas operações.

CREATE ou Insert

O CREATE é usado para criar a estrutura da tabela, como vimos no subcapitulo acima, já o insert é usado para inserir dados na base de dados.

Insert

```

1  <?php
2      if($_POST['cnome'] != "")
3      {
4          include_once 'conexao.php';
5          $nome = $_POST['cnome'];
6          $idade = $_POST['cidade'];
7
8          $sql = "insert into cliente (nome,idade)
9              values ('$nome','$idade')";
10         $query = mysql_query($sql);
11         echo "Dados cadastrado com sucesso";
12     }
13     else
14     {
15         echo "Dados não cadastrado";
16     }
17 ?>

```

Linha 2 – Condicional para saber se a caixa de input cnome tem dados, caso não tenha dados não vai ser inserido na base de dados.

Linha 4 – Expressão para incluir o arquivo de conexão nesse arquivo.

Linha 8 – Variável \$sql está recebendo a query de inserção de dados na base de dados.

Linha 9 – Variável \$query está recebendo o comando para executar a query de inserir dados na tabela.

Select – Consulta de todos os dados da tabela

```

1 <html>
2 <head><title>Cadastro de Dados</title>
3 </head>
4 <body>
5 <?php
6     include_once 'conexao.php';
7     $consultar = "select *from tbcliente";
8     $executar = mysql_query($consultar);
9     ?>
10
11     <?php
12         while($linha = mysql_fetch_array(
13             $executar))
14         {
15             Nome:<input type="text" value="<?php echo
16                 $linha["nome"];?>">
17             Idade:<input type="text" value="<?php echo
18                 $linha["idade"];?>"><br/>
19         }
20     ?>
21 </body>
22 </html>

```

Linha 6 - Expressão para incluir o arquivo de conexão nesse arquivo.

Linha 7 – Variável \$consultar recebendo a query de consulta do SGBD.

Linha 8 – Variável \$executar recebendo o comando de php para executar a query da linha 7.

Linha 11 – Laço de repetição para buscar todos os dados da tabela cliente, caso seja verdadeiro o que vem do comando mysql_fetch_array, vai ser retornado todos os valores da mesma.

Linha 14 e 15 – Vai exibir os dados do loop do comando while.

Consulta simples por código

```
10 <?php
11     include_once 'caminho.php';
12     $codigo = $_GET['codigo'];
13
14     $consultar = "select *from cliente where
15     codigo $codigo;";
16     $executar = mysql_query($consultar);
17 <?>
18 <div id="listar">
19     <?php
20         while($linha = mysql_fetch_array(
21             $executar))
22         {
23             <?>
```

Linha 11 - Expressão para incluir o arquivo de conexão nesse arquivo.

Linha 12 – Variável \$codigo recebe via método GET o código para fazer a consulta por um código digita pelo usuário, no formulário.

Linha 14 – Variável \$consultar recebe a query de select por where do código que se encontra na tabela.

Linha 17 – Declaração de uma div para exibição no formulário.

Linha 19 – Lista via laço de repetição seguindo o mesmo passo da consulta simples.

Update– Alteração de um dado ou vários dados dentro da tabela.

```

1  <?php
2      include_once 'conexao.php';
3      $id = $_POST['codigo'];
4      $nome = $_POST['cnome'];
5      $idade = $_POST['cidade'];
6
7      $alterar = "UPDATE tbcliente SET
8      nome = '$nome',
9      idade = '$idade'";
10     $executar = mysqli_query($conn,$alterar);
11     if($executar)
12     {
13         echo "Dados alterado com sucesso";
14         echo "<br>";
15         echo "<a
16         href='listarDadosCliente.php'>Voltar</a>"
17         ;}
18     else
19     {
20         echo "Erro ao alterar os dados";
21     }
22     ?>

```

Linha 2 - Expressão para incluir o arquivo de conexão nesse arquivo.

Linha 3, 4 e 5 – Recebendo os valores do formulário para as alterações.

Linha 7 – Variável \$alterar recebendo a query de alteração.

Linha 10 – Variável \$executar recebendo o comando de alteração a query junto com a conexão.

Linha 11 – Condicional caso a variável \$executar seja verdadeira os dados vão receber as alterações e lista a mudança.

Delete – Usado para excluir dados dentro da tabela, todos ou com clausula.

```

1 <?php
2     include_once 'conexao.php';
3     $id = $_GET['id'];
4     $excluir = "delete from tbcliente where
5     codigo='$id' ";
6     $executar = mysqli_query($conn, $excluir);
7     if($executar)
8     {
9         echo "Cliente excluido com sucesso";
10        echo "<br>";
11        echo "<a
12        href='listarDadosCliente.php'>Voltar</a>";
13    }
14    else
15    {
16        echo "Erro ao excluir os dados";
17    }
18 ?>

```

Linha 2 - Expressão para incluir o arquivo de conexão nesse arquivo.

Linha 3 – Variável \$id recebendo via método GET o id no caso o código para fazer a exclusão.

Linha 4 – Variável \$excluir recebe a query de excluir dados na tabela, usando uma clausula por código.

Linha 5 – Variável \$executar recebe o comando de excluir, vindo da query.

Linha 6 – Condicional para verificar se a variável \$executar tem algo, no caso verdadeiro, para seguir na exclusão.

9.5 Login e senha com PHP

Todo sistema temos que ter um acesso restrito por usuários e função, é ai que vamos criar uma tela de login e senha, essa autenticação é sempre usado em qualquer sistema sendo WEB, Desktop ou mobile.

Formulário em HTML

```

1 <html>
2 <head>
3 <title>Login e Senha</title>
4 </head>
5 <body>
6
7 <center>Usuario e Senha</center>
8 <form action="valida.php" method="post">
9
10
11 USUARIO:<input type="text" name="usuario">
12 SENHA:<input type="password" name="senha">
13 <br/>
14 ADM<input type="radio" name="perfil" value="adm">
15 User<input type="radio" name="perfil" value=
  "user" checked>
16
17 <input type="submit" value="Enviar"/>
18 </form>
19 </body>
20 </html>

```

Linha 7 – Comando de html, para centralizar

Linha 8 – Comando de html para criar um formulário <form>.

Linha 8 – Action comando para indicar a onde vai ser autenticado a conexão com o banco.

Linha 8 – Method de envio POST.

Linha 14 e 15 – Tag do html para definir qual é o tipo de perfil ADM ou Padrão.

Código para validar a conexão.

```

1 <?php
2 SESSION_START();
3 $login=isset($_POST["usuario"])?strtolower($_POST[
  "usuario"]):"";
4 $senha=isset($_POST["senha"])?md5($_POST["senha"]):"";
5 $perfil=isset($_POST["perfil"])?strtolower($_POST[
  "perfil"]):"";
6
7 include("conexao.php");
8
9 $log=mysql_query("select * from tbusuar where user='
  $login' and senha='$senha' and perfil='$perfil') or die(
  mysql_error());

```

Linha 2 – Início da sessão para guardar o login e senha, para colocar as informação nas telas das outras páginas.

Linha 3 – Variável \$login vai receber o usuário via método POST, e converte em letras maiúscula.

Linha 4 – Variável \$senha vai receber a senha do input e converte para md5(), uma criptografia.

Linha 5 – Variável \$perfil vai receber o perfil do input e converter em maiúscula.

Linha 7 – Expressão para incluir o arquivo de conexão nesse arquivo.

Linha 9 – Variável \$log vai receber a execução do comando mysql_query, para consultar o perfil e usuário.

```
11 $linha=mysql_fetch_array($log) ;
12
13 if($login==" " || $senha=="") die("Acesso restrito,
efetue login <br><a
href=\"javascript:history.go(-1)\">Voltar</a>");
14
15 if($login!=$linha["user"] || $senha!=$linha["senha"]) die
16 ("<strong>Nome de usuario ou senha incorreta<br>
17 <a
href=\"javascript:history.go(-1)\">VOLTAR</a></strong>");
18
19
20 if($login==$linha["user"] && $senha==$linha["senha"] &&
$perfil=="adm") {
21 $_SESSION["user"]=$linha["user"];
22
23 echo "
24 <script language='javascript' type='text/javascript'>
25 window.location.href='homepage.php';
26 </script>";
27 }
```

Linha 11 – Variável \$linha está recebendo o login em um array.

Linha 13 – Condicional para verificar se a variável \$login e \$senha estiver vazia retorna para o index.

Linha 20 – Condicional para verificar se \$login, \$senha e \$perfil for verdadeiro acesso permitido e guardar na sessão o nome do usuário.

Atividade prática

- 1) O que é SGBD?
- 2) Qual é a finalidade do Xampp?
- 3) O que é um servidor Web?
- 4) Qual é a função do Apache?
- 5) Qual é o comando para criar um banco de dados
- 6) Explique sobre esse comando, create table tbcliente();
- 7) Qual é o comando para excluir um dado?

- 8) O que é uma query?
- 9) O que é o CRUD?
- 10) Explique sobre este código, `strtolower`;

Desafio

Faça um sistema de cadastro de produto com cinco campos, usando o CRUD na tabela.

10 – Introdução ao conceito POO e classes com PHP



Para muitos desenvolvedores o PHP, orientação a objetos é um conceito que deixa medo, cheio de sintaxes complicadas. Vamos agora aprender todos os conceitos por trás da **programação orientada a objetos** (POO), um estilo de codificação onde ações relacionadas são agrupadas em classes para ajudar na criação de códigos mais compactos e efetivos. A linguagem de programação Orientado ao Objeto além de ser muito organizada trata – se de uma linguagem segura e fortemente tipada, dando mais segurança e deixando o PHP, mais usado nos dias de hoje.

10.1 Introdução sobre POO.

Programação orientado ao objeto é uma maneira de programar de forma organizada, segura podendo reutilizar códigos, muito fácil para trabalhar em equipe podendo usar todas as boas práticas de programação, essa maneira de programar ou também conhecida como arquitetura não é nova no ramo de desenvolvimento, foi desenvolvida praticamente junto com a procedural, como antigamente não tínhamos projetos gigantes como nos dias de hoje então a procedural sempre deu conta do recado, agora com projetos cada vez mais gigantes é necessário programar em equipe, ai que a linguagem de POO vem ganhando mais mercado.

POO é intimidadora para uma grande quantidade de desenvolvedores porque ela introduz uma sintaxe e, á primeira vista, parece ser bem mais complexa que a programação procedural também conhecida de estruturada. Portanto, se analisar bem, POO é bem direta e, fundamentalmente, uma abordagem mais simples para programação.

A linguagem PHP, começou a ser organizada ou introduzida o POO, nos anos de 2005, quando houve uma grande mudança no mercado de grandes projetos para WEB, foi então que o PHP POO, começou a ganhar mercado mundial. Podemos programar somente de forma POO no PHP ou de uma maneira hibrida usando POO e estruturada. Não é recomentado fazer

esse tipo de programação pela fato da segurança. A dica é se estamos programando em procedural termina em procedural, nunca é bom misturar.

10.2 Conceito de classes e construtores.

Definimos classe como sendo uma abstração de um conjunto de objetos, os quais são agrupados por possuírem similaridade em termos de comportamento e características. Sendo assim as propriedades ou os atributos de um objeto são descritos a partir da definição de uma classe.

Classe é uma entidade teórica, ou seja, uma organização usado na POO, ela não existe fisicamente e não ocupa espaço na memória.

Quando criamos um objeto uma determinada classe temos que instanciar o objeto criado para poder existir ele dentro do sue algoritmo.

Para instanciar uma classe vamos usar um comando new. Veja um exemplo abaixo.

```
10  <?php
11  $car1 = new Carro();
12  $car1->fabricante = 'VW';
```

Linha 11 – Variável \$car está sendo instanciada de uma classe, dentro desse arquivo se torna um objeto.

A variável do tipo objeto \$car1 está sendo atribuída o fabricante. O construtor define o valor da densidade logo na criação do objeto. Sendo fundamental para a parte de visualização dos arquivos dentro da POO.

Um dos pilares da orientação a objetos é evitar que classes tenham acesso a um código que não tenha a ver com sua estrutura mãe. Imagine a classe mãe, que tem volume e consequentemente densidade. Se alterarmos a massa de um corpo sem alterarmos a densidade, criamos uma inconsistência dentro da classe. Logo esse é um caso em que não podemos permitir livre acesso a esses campos.

10.3 Visibilidade de classes e subclasses.

A visibilidade de uma classe e das variável é fundamental na programação Orientado ao Objeto, sendo a visão para as outras classes e arquivos dentro dessa organização.

Tipos de visibilidade de variável e classes

Public

Public todos os arquivos, variáveis e classes conseguem visualizar sendo fácil acesso para todos.

Private

Private somente a classe e variável no local a onde foi criado consegue visualizar.

Protect

Protected é pra quando você não quer deixar um atributo public, livre para todos. somente, você quer compartilhar ele com as subclasses.

Atividade teórica

- 1) Qual é a diferença entre POO e procedural?
- 2) Explique sobre o comando new;
- 3) O que é instanciar uma classe?
- 4) O que é uma classe?
- 5) O que é um objeto?
- 6) Explique sobre a visibilidade public;
- 7) Explique sobre a visibilidade private e protect;

Desafio

Faça uma pesquisa sobre a linguagem de programação atual que usa a organização POO, formate nas normas ABNT