PHP é uma linguagem de script de servidor, e uma ferramenta poderosa para fazer páginas da Web dinâmicas e interativas. Os pontos fortes dessa linguagem é que você pode incluí-la de forma fácil e rápida a um arquivo já existente de HTML, além de ser uma alternativa fácil e simples para utilizar em sites da web.

O php é principalmente usado na comunicação entre computador e servidor, tornando essa comunicação fácil e rápida de ser feita.

Para utilizar o PHP, é necessário salvar o arquivo com a extensão .php. Você pode fazer isso usando o explorador de arquivos ou um editor de código como o Visual Studio Code.

Estando com o seu arquivo php em mãos, ele precisa da estrutura do HTML, ou seja, o head, o body e etc. Como está logo abaixo:

“<!DOCTYPE html>

<html>

<body>

</body>

</html>”

Depois da estrutura base do HTML, para começar a programar em PHP é preciso usar a tag “php”, abaixo está como ela é utilizada:

“<?php

?>”

Dentro dessa tag você irá programar o que quiser em php.

Comandos do PHP

Uns dos primeiros comandos e mais fundamentais para se aprender é como realizar a saída de algo para o web site. O comando de escrita é o “echo”, ele funciona de um jeito muito simples, só é preciso escrever “echo” após, colocar o que você queira que ele escreva entre aspas. Com o echo você pode usar qualquer tag do html. Observe abaixo como que o echo pode ser usado.

“<?php

echo ‘Hello World’

?>”

A concatenação é usada para organizar em várias linhas um echo, além de também pode unir variáveis a frases. Para usar a concatenação é bem simples, quando vc quiser pular uma linha no código sem precisar é somente colocar . (ponto). Veja o exemplo abaixo

Criando/declarando variáveis PHP

No PHP, as variáveis são todas reconhecidas por um $ (cifrão) na frente dela, então quando for escrever uma variável, escreva $nomeDaVariavel.

$variavel

É claro, a declaração de variáveis precisa estar dentro de um script PHP.

Após declaram a variável, você pode deixá-la vazia ou atribuir algum valor inicial, isso é feito somente colocando um sinal de igual e depois escrevendo o valor:

$variavel = “Olá mundo”;

No PHP não precisamos dizer a ele qual tipo de dados é a variável, como em outras linguagens, mas caso seja um texto, precisamos deixar esse texto dentro de aspas.

Para enviar as variáveis para a tela é bem simples, é somente digitar o nome da variável dentro do echo.

Caso só queria escrever a variável na tela, então so colocar:

Echo $variavel

Mas caso queira escrever a variável dentro de um texto ou frase, deixe assim:

Echo “texto $variavel texto”;

Ou

Echo “texto” . $variavel . “texto”;

Inputs para o PHP

Quando você precisa obter o valor de um campo de formulário e aplicar um filtro a ele, utilize a função filter\_input. O filter\_input obtém o valor de um campo de formulário e aplica um filtro a ele, se desejado. O INPUT\_ indica o método do formulário (POST, GET, etc.) de onde o valor será obtido.

<form name="frm" method="post">

    <div>

        <input type="number" name="txt">

    </div>

</form>

<?php

$valor1 = filter\_input(INPUT\_POST, 'txt');

<?php

A diferença do método post, get e etc, já foi explicada no modulo de html, mas resumidamente, o método get expõe todos os inputs na url, já o post os inputs ficam ocultos.

No PHP, as variáveis são acessadas por seus nomes, não por IDs. O id serve para identificar o elemento no HTML, enquanto o name é usado para recuperar o valor do campo.

Ao utilizar o PHP, podemos separar o código em diferentes arquivos e utilizar o código presente no arquivo quando necessário, como se fosse módulos.

Fazer um código dividindo desse jeito irá deixar esse código mais organizado e poderá utilizar quantas vezes que for necessário.

Para incluir o código de um arquivo somente precisa do seguinte script:

<? Php

include\_onde ‘nome do arquivo . extensão’;

?>

Alguns dos arquivos que podemos criar para facilitar é um arquivo para o head, um para a navegação e além de separar os conteúdos das páginas em arquivos.

Agora vamos ver alguns exemplos na pratica desses módulos.

Primeiramente, o index irá ser a base para todas as outras paginas e a primeira pagina que irá aparecer no nosso site vai ser a home.

O código do nosso index irá ser o seguinte:

<!DOCTYPE html>

<html>

    <head>

    <?php include\_once 'cabecalio.php'; ?>

    </head>

    <body>

        <?php include\_once 'navegacao.php'; ?>

        <div class="container mt-4">

            <div class="row">

               <?php

                    $pagina = filter\_input(INPUT\_GET, 'p');

                    if (empty($pagina)) {

                        include\_once 'home.php';

                    } else {

                    if (file\_exists($pagina. '.php')){

                        include\_once $pagina . '.php';

                    }else{

                        echo "<div class='col-sm-12 com-md-12'>"

                    ."<div class='alert alert-danger' role='alert'>"

                    ."ERRO 404, página não encontrada"

                    ."</div>"

                    ."</div>";

                    }

                }

               ?>

            </div>

        </div>

        <?php include\_once 'scripts.php'; ?>

    </body>

</html>

O código possui a estrutura básica do html, o head está em um outro arquivo, a navegação e os scripts do bootstrap tambem.

Podemos notar a variável “Pagina” que irá pegar uma informação da URL do site com o nome como “P” e atribuí-la.

O primeiro if e else, verifica se a variável “pagina” não tem nada, se ela estiver vazia então o conteúdo do arquivo home irá ser atribuído a página, senão entra no segundo if e else.

O segundo if e else verifica se o site tem um arquivo com o nome que esteja na variável “pagina” mais a extensão “. php” e se tiver atribui o conteúdo desse arquivo no site e se não tiver então irá dar o erro 404 de página não encontrada.

A navegação irá ficar do seguinte jeito:

<a class="nav-item nav-link" href="?p=home">Home</a>

<a class="nav-item nav-link" href="?p=categoria ">Categoria</a>

<a class="nav-item nav-link" href="?p=cliente ">Cliente</a>

<a class="nav-item nav-link" href="?p=fornecedor ">Fornecedor</a>

<a class="nav-item nav-link" href="?p=produto ">Produto</a>

Além dessas tags, tambem é preciso da estrutura que você poderá encontrar no Bootstrap ou visualizar a estrutura no modulo do HTML.

A única coisa diferente que o php faz ter é na linkagem, o código “href="?p=X "” faz com que apareça na url essa informação, que no caso seria “?p=X", e então o php poderá interpretá-lo como se fosse um input pelo método GET, pois irá funcionar da mesma forma.

Até agora é somente isso que vimos, daqui a pouco vamos ver como trabalhar com banco de dados, então podemos incluir no próximo mês isso.