PHP - Introdução

PHP - Hypertext Preprocessor, originalmente Personal Home Page é uma linguagem de programação de criação de scripts que foi projetada especificamente para a Web para permitir que desenvolvedores web criem site web gerados dinamicamente.

Dentro do seu HTML você coloca o seu código PHP, esse código será executado toda a vez que a página for visitada.

O código PHP é então interpretado no servidor WEB e gera HTML ou outro resultado que será exibido ao usuário.

É uma linguagem de código aberto e livre, ou seja, utiliza sem pagar nada e tem acesso ao código-fonte, podendo alterá-lo.

Muito da sua sintaxe é emprestada de C, Java e Perl com algumas características específicas do PHP juntas.

Principal característica: suporte a um grande número de Bancos de Dados: Oracle, MySQL, PostGreSQL, Dbase, etc.

Pode ser desenvolvida de forma estruturada ou orientada a objetos.

Histórico - Rasmus Lerdorf, membro da equipe Apache, é o criador do PHP.

A primeira parte do PHP foi desenvolvida em 1994 – um script em PerlCGI para monitorar as pessoas que acessavam seu site pessoal.

Em 1995, devido a demanda, ele criou o Personal Home Page Tools (PHP construction kit).

A versão 2 foi lançada a seguir incluindo uma ferramenta para analisar consultas SQL.

Em 1997, já era utilizado por mais ou menos 50.000 sites, muito grande para uma única pessoa administrar.

Formou-se uma pequena equipe para manter o projeto: Zeev Suraski e Andi Gutmans, PHP3 e PHP 4.

Com o tempo foi evoluindo e parou numa versão 5.6 , melhor dizendo PHP 5. Houve um processo que ficou por muito tempo com uma única versão, não esteve atualizações. A versão 5 é até hoje ainda uma das mais utilizadas na internet. Os desenvolvedores já estão atualizando para as versões mais novas, mas uns 50% ainda dá para dizer utiliza-se da versão 5.6, devido passar um longo período de tempo sem atualização foi acumulando usuários.

Esse processo de ficar um longo período de tempo sem atualizações representou um ponto negativo, o mercado evoluiu, e o PHP não evoluiu na mesma velocidade por assim dizer. Tanto que não existe a versão PHP 6, houve um pulo direto do PHP 5 para o PHP 7 diretamente. A quantidade de atualizações foi muito grande que não validava a atualização 6.

Com a atualização para a versão PHP 7, mudou completamente em se tratando de performance, ficou muito mais rápido, vários recursos atualizados que no mercado já estavam utilizando, que já veio muitas dessas novas arquiteturas, linguagens, métodos e padrões nessa nova versão.

E a cada versão do PHP 7 para o PHP 7.1 houve mudanças significativas, assim como do 7.1 para a versão 7.2, 7.3 e 7.4 agora também com várias modificações.

Hoje mais de 80% dos sites no mundo ainda são feitos em PHP.

Grandes sites como Wikepedia, Yahoo, facebook e entre outros, são desenvolvidos na linguagem PHP.

Nesse curso vamos focar nessa versão mais nova que é PHP 7.4. Essa é a última versão antes do PHP 8, que será lançado mais adiante.

A linguagem PHP é uma linguagem moderna, com excelente performance, recursos novos utilizados por tecnologia de ponta, que fez com que novamente torna-se uma das linguagens mais utilizadas hoje em dia no mundo todo.

A partir dessa introdução sobre a linguagem PHP agora então vamos começar do básico até a instalação e configurações para desenvolverem nessa linguagem fácil de aprender.

Ferramentas Necessárias

Em geral o que precisa:

- O PHP instalado na máquina pessoal.

Por que quando se faz um site, faz antes no próprio computador e depois coloca-se online. Isso significa que transferimos nos arquivos para um servidor que faz com que outras pessoas consigam acessar.

Existe forma de transformar o próprio computador em um servidor, que simula um servidor no nosso próprio computador. Isso inclusive possibilita com que pessoas que estão usando uma mesma rede por exemplo, possam acessar pelo id da máquina.

Hoje nas versões mais novas do PHP, conseguimos com o próprio PHP gerar um servidor, mas é um processo um pouco limitado. O ideal é trabalhar com um ambiente mais próximo possível de um servidor real.

- Um servidor instalado para interpretar os códigos que vamos fazer.

Esse servidor pode ser uns dos mais usados que é o Apache ou Nginx. Sendo o Apache o mais utilizado atualmente nos servidores que vamos utilizar para colocar o site no ar.

- Um banco de dados

Usamos para armazenar os dados para uso posterior. Como lista de usuários, produtos, etc. Um site armazenas informações e para isso utilizamos banco de dados. Existe infinitos bancos de dados e o PHP suporta basicamente todo tipo de banco de dados. O banco de dados mais utilizado é o MySQL.

- Editor de código

Precisamos de um editor de código para por nossos códigos em arquivos. Esses arquivos podemos editar no próprio bloco de notas do computador, mas é mais profissional, interessante e produtivo usar um editor de código que utiliza recursos com colorir o código, possibilitando marcar onde inicia ou termina um recurso. Fica visivelmente mais interessante para o programador, de uma forma que possibilita uma melhor elaboração, manutenção na edição dos códigos, com recursos com o autocompletar, ou seja, dá sugestões entre outros.

Existe muitos editores de código no mercado, mas o mais utilizado atualmente é o editor da Microsoft VScode ou Visual Studio Code. Tem outro muito utilizado também que é o Sublime Text. São editores muitos similares, funcionamento muito parecido.

Inicialmente é isso que vamos precisar para começar a programar em PHP. Todos esses conseguem de forma gratuita e são software muitos leves que funciona em qualquer plataforma como macOS, windows, Linux.

Editor de Código (VScode)

Para instalar o editor de código VCcode basta acessar o site oficial que é:

https://code.visualstudio.com/

Entrando na página, geralmente já sugere a instalação de acordo com o sistema operacional que tem na sua própria máquina.

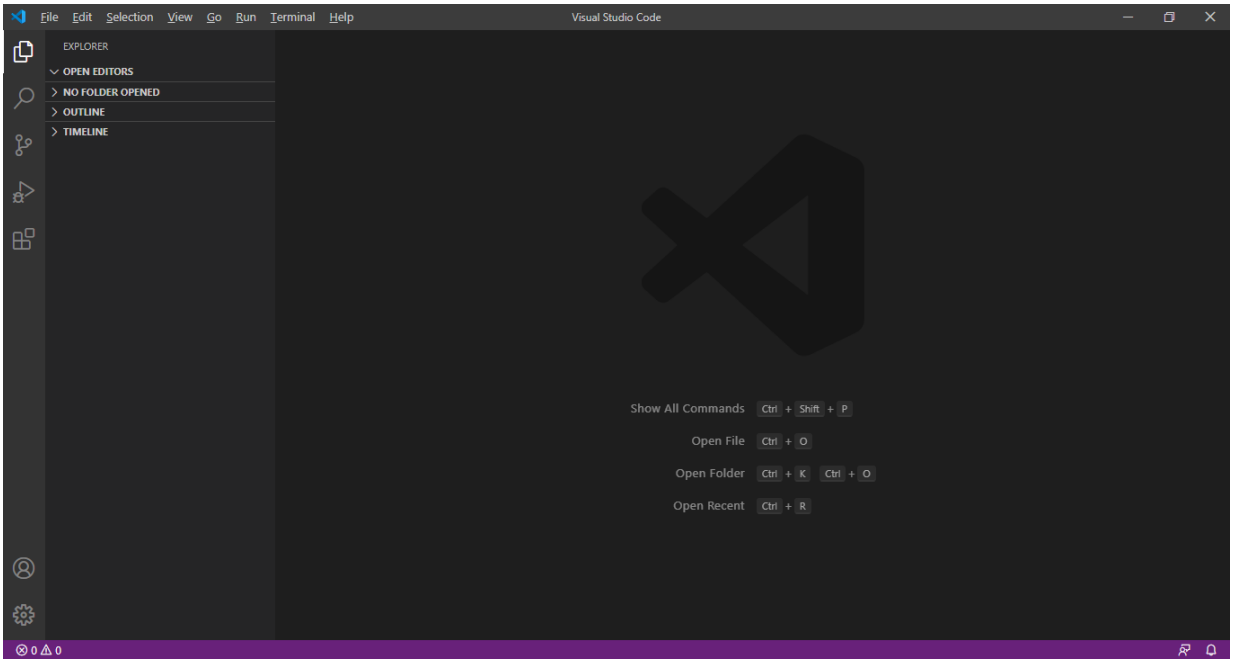

Figura1 - imagem é da área de trabalho do VScode

Geralmente nessa coluna da direita após ser instalada vai abrir com mensagens de boas-vindas e clicando no X que tem logo acima nas abas fechar essa tela de boas-vindas e aparece uma igual exatamente essa que está na figura 1.

A direita tem os menus principais que são:

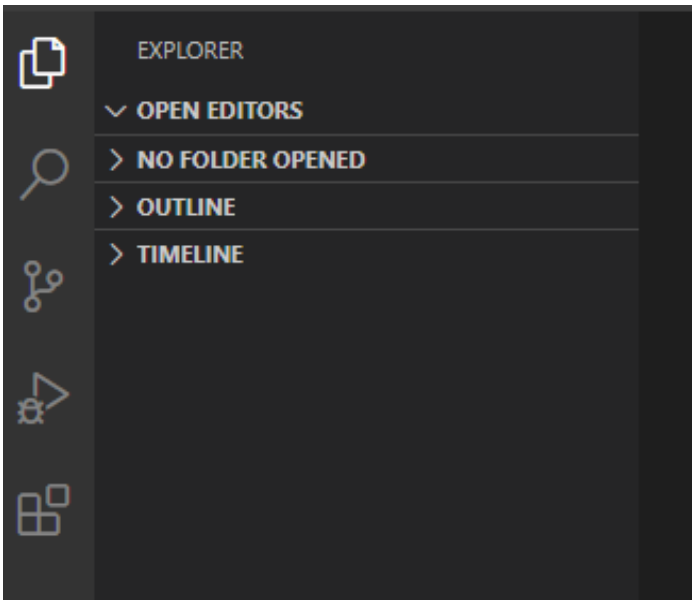

Figura 2 - menus principais

O primeiro de cima para baixo é o explorar, que são os arquivos do seu projeto. Como podem ver nessa imagem está mostrando que por enquanto não tem

nenhum arquivo aberto. Essa área, quando o usuário abrir um projeto no editor e clicar numa pasta, ela vai ler os arquivos e mostrar nessa área os arquivos da pasta. Então quando abrir os arquivos e o usuário clicar 2 vezes num arquivo desses, vai aparecer o conteúdo desse na área da coluna a direita. Podem-se abrir vários arquivos ao mesmo tempo, ele vai abrindo abas para esses arquivos.

A lupa serve para buscar alguma coisa no arquivo do seu projeto.

O Source Control é um controle de versão, quem sabe mexer com gif, já sabe lidar com esse controle de versão, que serve para controle de versões de códigos.

O quarto de cima para baixo é o Run and debug, que não vamos utilizar a princípio.

Os últimos desse é o Extensions, que servem baixar extensões ou plugins, que vai ajudar o programador na elaboração do seu código. Um exemplo de plugin para instalar é o terminal, veja na figura a seguir.

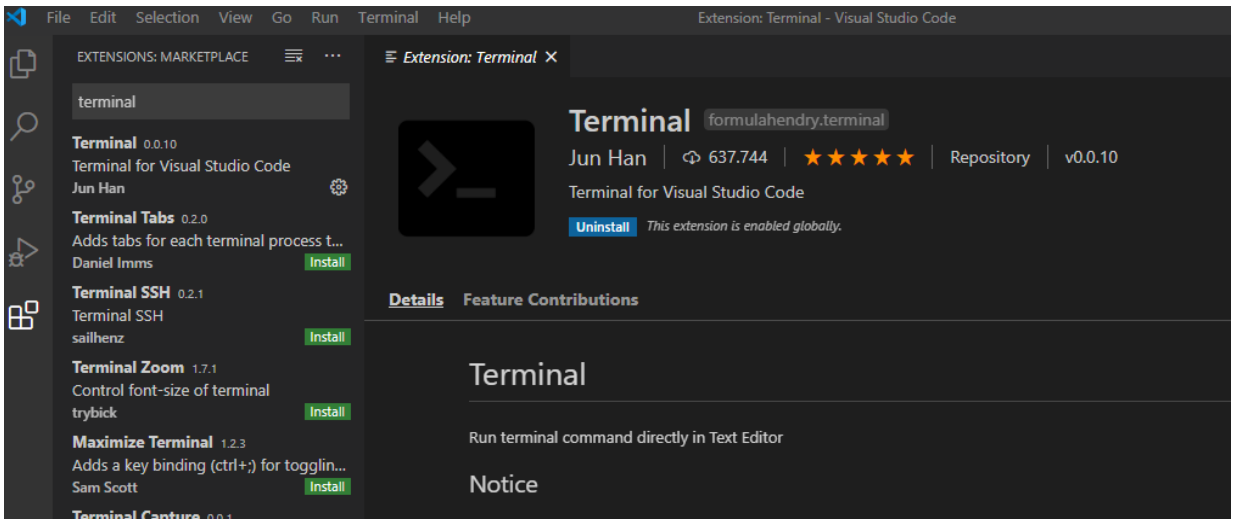

Figura 3 - instalação do plugin terminal 0.0.10

Observe a imagem da Figura 4, mostra na parte inferior direita da página do VScode uma seta apontando a direita, quando o usuário clica nele abre um terminal.

Figura 4 - seta de abertura do prompt do terminal

Como podem ver na Figura 5 mostra essa tela do terminal aberta. Abre um prompt de comando dentro do próprio editor de texto.

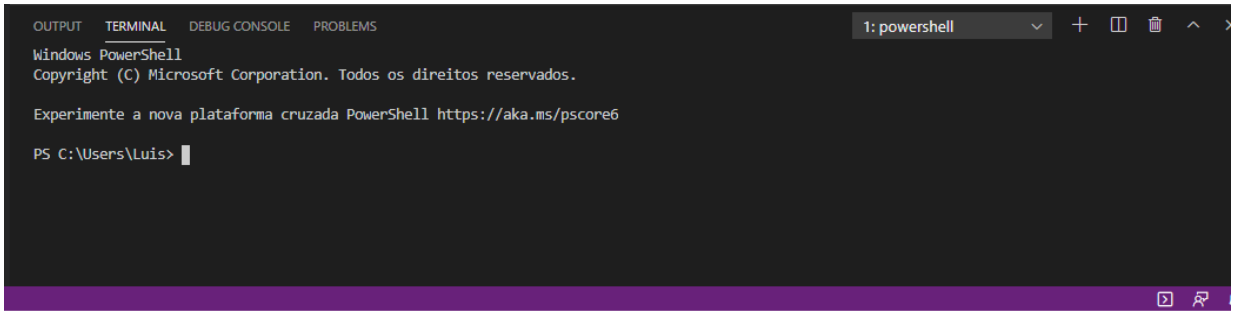

Figura 5 - tela do terminal

Como abrir um projeto dentro do próprio editor? Basta clicar no Open Folder ou simplesmente arrasta a pasta de onde ela está para essa área.

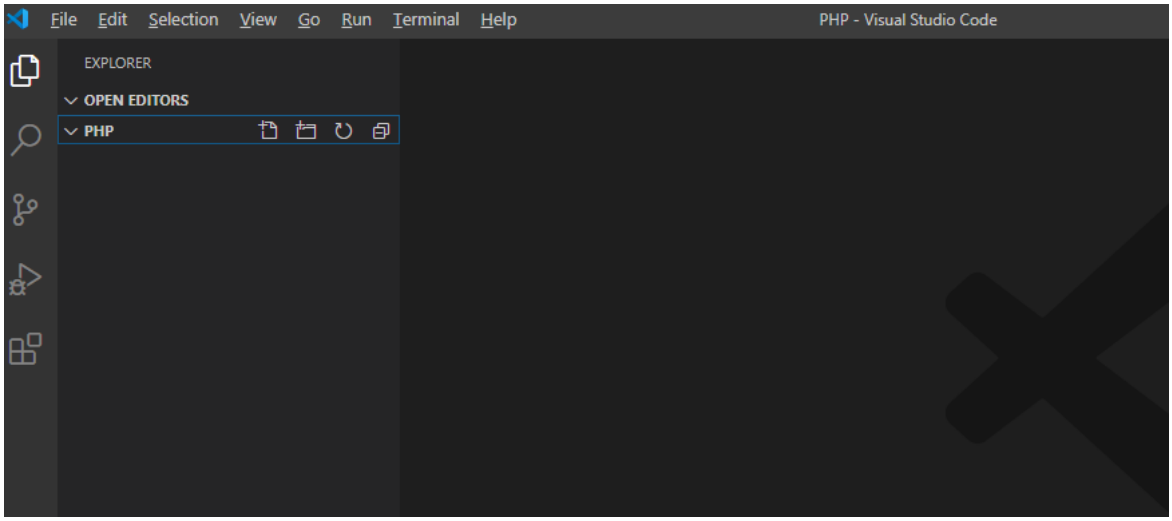

Figura 6 - pasta criada PHP

Na Figura 6 ao lado da pasta PHP observem a direita dessa pasta está os ícones para criar novo arquivo, nova pasta, atualizar recolher.

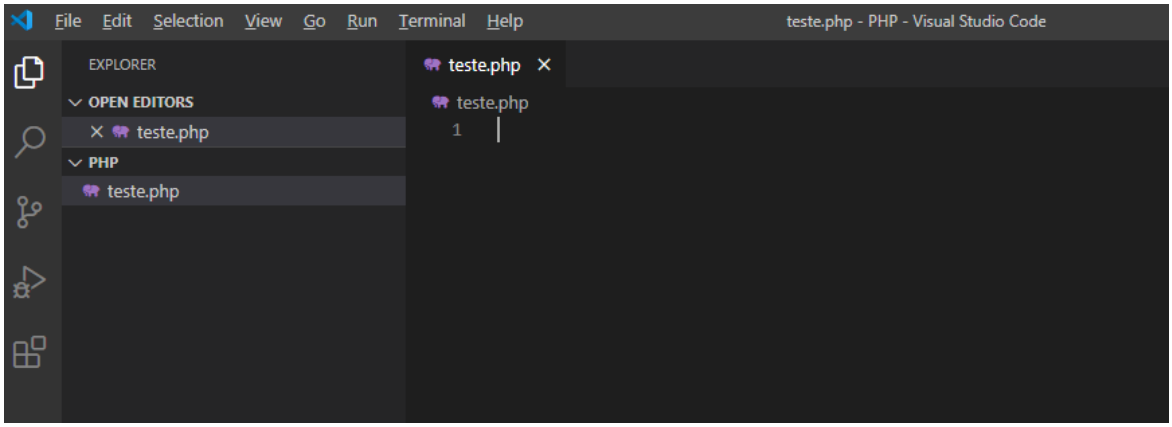

Figura 7 - imagem arquivo teste.php

Na figura 7, mostra o arquivo teste.php que foi criado e automaticamente abre na área de edição, a direita, como pode ser visto o nome na aba teste.php.

Instalando o PHP 7.4 (XAMPP)

Com a instalação do PHP, vamos instalar o Apache, MySQL, phpMyAdmin e todas as outras ferramentas. Vamos instalar um pacote com todas essas ferramentas que é o XAMPP um dos servidores mais utilizados e tem versões para todas as plataformas, macOS, windows, Linux.

Se quiser um específico para windows tem o WAMPSERVER, específico para macOS tem o MAMP.

WAMP - termo que significa Windows, Apache, MySQL, PHP.

MAMP - termo que significa MAC, Apache, MySQL, PHP.

O XAMPP usa esse mesmo conceito.

Todos são servidores de softwares livres.

Para instalar é só acessar o site oficial:

https://www.apachefriends.org/pt_br/index.html

Figura 8 - Site para download xampp

Como podem ver na Figura 8 no cabeçalho depois do Apache tem MariaDB que é o banco de dados MySQL basicamente, mais melhorado que vem com o pacote de instalação do XAMPP. Observem na parte inferior da imagem tem as instalações para cada plataforma e a versão do PHP 7.4.

Quando clicar numa dessas instalações irá fazer o download e automaticamente ele já abre após com essa tela a seguir:

Figura 9 - software de abertura na instalação do xampp

Como podem ver na Figura 9, essa janela tem três abas na parte superior.

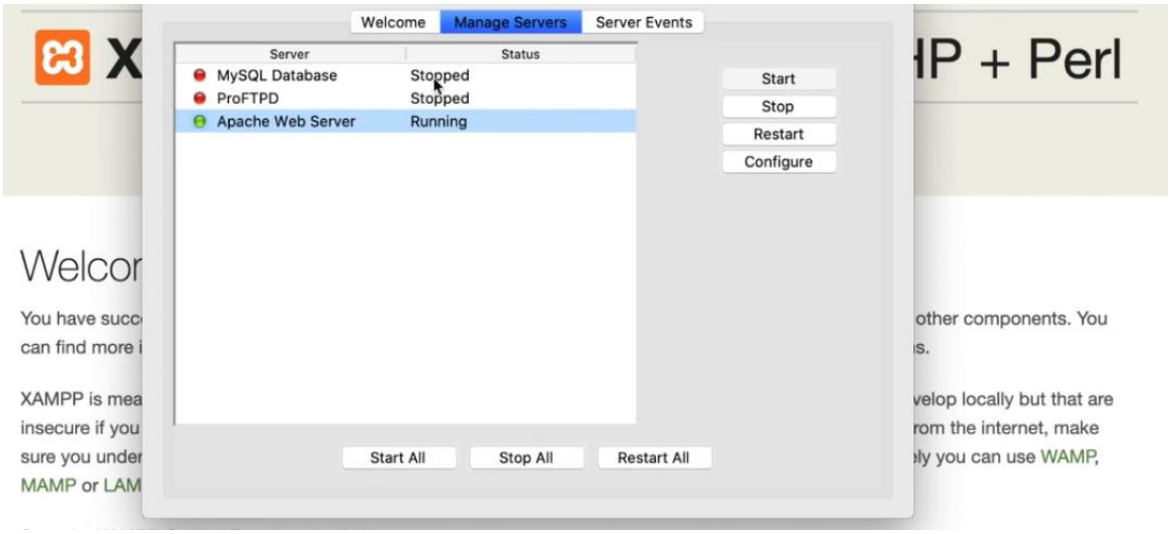

Figura 10 - Aba Manage Servers

Nessa Figura 10 mostra a aba do Manage Server que mostra os servidores e como podem ver o Apache está rodando. O MySQL não está rodando, quer dizer que o banco de dado não está executando. Se quiser parar o Apache basta clicar no botão a direita Stop ou então Stop All, que para todos.

A partir de agora vamos ver como funciona os servidores, como começar a trabalhar, onde estão os arquivos etc.

Como funciona o PHP?

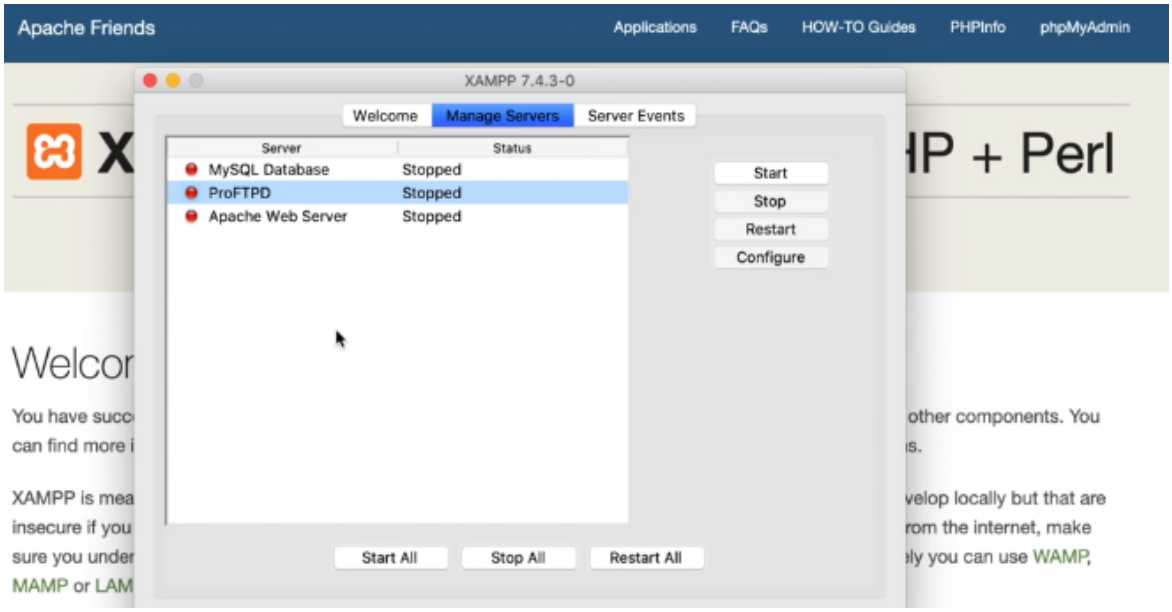

Figura 11 - Manage Severs

Primeiro iniciamos o Apache, MySQL ou seja, esses três que aparecem na Figura 11, como podem perceber o botão está em vermelho significando que não estão rodando.

Para isso basta clicar no botão Start All pode ser que demore um pouco para eles ficarem rodando, isso é, quando ficarem na cor verde. Podem acompanhar esse carregamento do servidor na aba Server Events na parte superior.

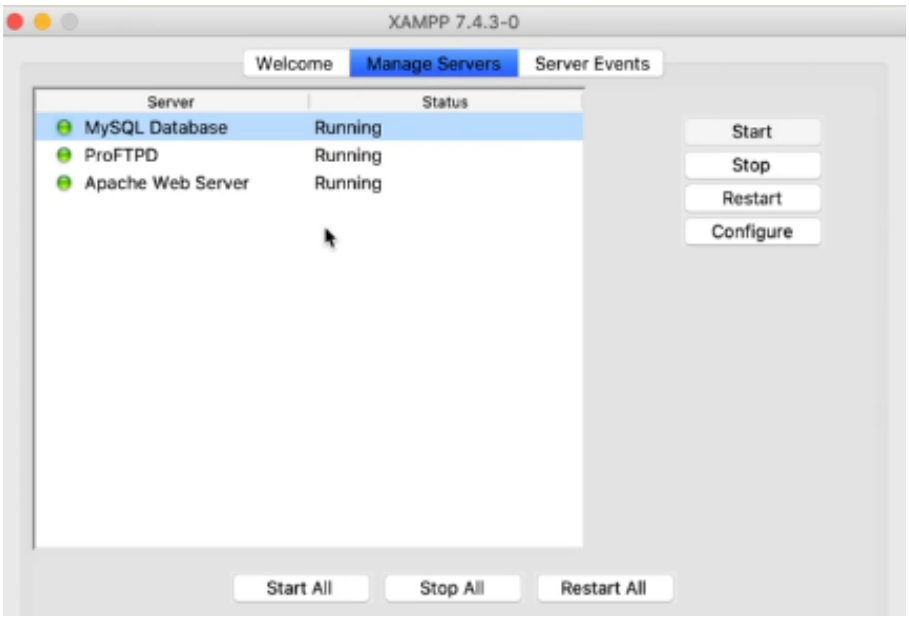

Figura 12 - Servidor funcionando

Na Figura 12, mostra que todos estão carregados, funcionando por estarem na cor verde. Isso significa, se for à pasta de instalação do xampp, pasta essa

htdocs, como podem perceber vai estar uns arquivos nela, que podem selecionar todas e deletar. Essa pasta htdocs é a pasta que é acessada pelo servidor, servidor Apache como podem ver na Figura 12, que roda na porta 80. Para acessar esse servidor, existe várias formas, primeiro existe um IP que toda máquina tem, e ela direciona para a própria máquina, um IP que funciona dentro de seu computador que é o 127.0.0.1. Se forem na barra de endereço do navegador e digitar esse IP e dar ENTER, vai aparece a tela conforme Figura 13, se vocês tiverem deletado todos os arquivos do htdocs. Caso contrário vai abrir coisas do XAMPP.

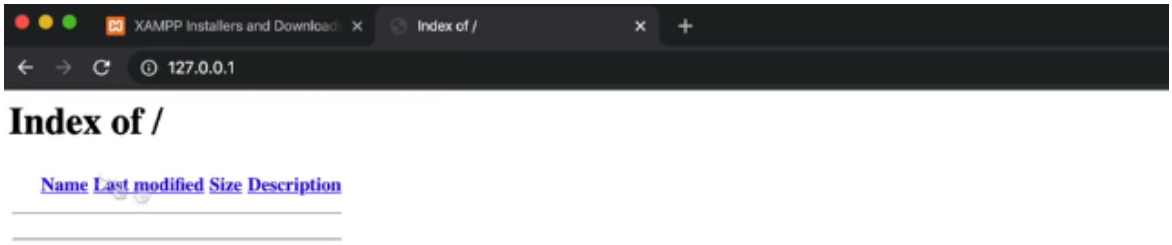

Figura 13 - Imagem pasta htdoc

Existe um atalho para esse IP 127.0.0.1 que é o localhost. Se for ao navegador e digitar no campo de endereço localhost vai acessar essa mesma janela conforme Figura 13. Pode ser que digitando só localhost ele vai para o google como pesquisa por esse termo, será necessário digitar junto http://localhost. para acessar htdocs.

No macOs também acessa colocando o nome do usuário do sistema seguido com ponto e local, ou seja, como exemplo no meu sistema:

sandra.local.

Normalmente se usa localhost no lugar do IP 127.0.0.1.

O htdocs é a pasta oficial e acessível no seu navegador. Isso quer dizer se alguém acessar o seu servidor só poderá acessar o que está dentro dessa pasta htdocs.

Normalmente deixamos as nossas pastas dos projetos dentro dessa pasta htdocs.

Para servir de exemplo, vamos criar uma pasta chamada php nessa pasta htdocs.

Agora para acessar essa pasta via navegador é necessário digitar **localhost/php** e dar Enter.

Observem que por enquanto só mostra a pasta por que não temos nada armazenado nela, nenhum arquivo.

O servidor quando ele acessa uma página a primeira coisa que ele faz é procurar um arquivo chamando index.html, se não achar vai procurar

index.php. Tem como configurar no servidor qual a ordem de procura por esse arquivo index, mas geralmente essa é a ordem.

Conforme até agora como exemplo criado anteriormente a pasta php, não existe esse arquivo index no htdocs, lembrando que deletamos todos os arquivos antes e assim ele mostra a lista das pastas, no nosso caso vai mostrar até aqui somente a pasta php.

Mas se criarmos um arquivo chamado index.php dentro dessa pasta php no htdocs, quando digitarmos no navegador localhost/php/ ele vai mostrar esse arquivo, por que conseguiu achar o arquivo index.php. Ele conseguindo achar, lê esse arquivo se não tem nada nesse arquivo não exibirá nada na página do navegador.

Se colocarmos nesse arquivo index.php, a frase Olá Mundo!!! e salvar, ir no navegador e atualizar mostrará essa frase.

Primeiros comandos no PHP

Para executar os primeiros comandos no PHP é necessário saber como funciona internamente o PHP.

Sempre que acessarmos um site como nosso exemplo anterior ou apertamos um F5, o Apache recebe a requisição (exemplo requisição pra pasta php) e manda para o local correto. Carregou o arquivo agora ele vai interpretar eventuais códigos que tem dentro desse arquivo. Exemplo do arquivo index.php, só existe por enquanto só a frase Olá Mundo!!! não precisou interpretar nada. Mas se tiver um código php dentro desse arquivo aí chama o php para interpretar o código php. Ele vai interpretar somente códigos php que tiver nesse arquivo e depois manda para o Apache e o apache junta com o que não é interpretável e manda para o usuário.

Resumindo:

Apache recebe requisição

Manda para local correto

Chama o php para interpretar o código php

0 php manda o resultado para o apache

O apache junta com o que não é interpretável e manda para o usuário.

Por Exemplo:

No arquivo index.php dentro da pasta htdocs, vamos colocar um script php dentro dele agora. Como iniciar um código PHP ?

Digitando <?php ?>

Tudo que colocarmos entre <?php ?> é código php.

Sintaxe Básica - Marcadores dos comandos PHP

O interpretador identifica quando um código é PHP pelas seguintes tags:

```
<?php
    comandos
?>
```

Para exibirmos conteúdo para o usuário ou texto a partir de um código PHP, utilizamos os comandos:

echo

Inicie colocando as tags para indicar que é um programa PHP <?php ?> Adicione o comando para imprimir “Olá Mundo”

```
<?php
    echo " Ola Mundo! ";
?>
```

Lembrando que o arquivo index.php está dentro da pasta php, na pasta htdocs no xampp. Após salvar as alterações no index.php, precisamos ir na URL do navegador e digitar:

localhost/php/index.php

ou

localhost/php/

Alterando o conteúdo desse arquivo index.php:

```
<?php
    echo "Sandra";
    echo "Amorim;
?>
```

Vamos salvar, executar no navegador, como estamos já utilizando-o basta atualizar (F5) e o resultado que mostra para o usuário é:

SandraAmorim

Percebam que o comando para imprimir na tela, não dá a quebra de linha.

Vamos simplificar esses comandos:

```
<?php
        echo "Sandra Amorim";
?>
```

Agora a saída na tela será:

Sandra Amorim

Observação:

Sempre que no seu arquivo tiver só códigos php, não é necessário colocar a tag de encerramento ?> no script php, exemplo:

```
<?php
        echo "Sandra Amorim";
```

A tag de encerramento não é necessária nesse caso. Algo lógico, iniciou um script php e não tem encerramento, abrangendo todo arquivo.

Comentários

Há dois tipos de comentários em código PHP:

• Comentários de uma linha:

Delimitado por duas barras (//)

• Comentários de mais de uma linha:

Tem como delimitadores os caracteres " /* " para o início do bloco e " */ " para o final do comentário.

TESTE comentando seu código!!!

Tipos de Variáveis

Nome das variáveis:

Toda variável em PHP tem seu nome precedido pelo caracter $ + um literal (letra ou “_”);

Ou seja, segui as normas de nomear variáveis.

PHP é case sensitive, ou seja, as variáveis $nome e $NOME são diferentes, cuidado ao definir os nomes das variáveis.

Tipo de Dados

Inteiro: é qualquer número do conjunto Z = {..., -2, -1, 0, 1, 2, ...}. positivo ou negativo, que pode ser representado na base decimal, hexadecimal (preceder o número com um 0X) ou octal (preceder o número com um 0):

$totalDaSoma = 8;

Float (ou Double): consiste no tipo ponto flutuante, de precisão dupla ou número real: Esse valor fracionado tem que ser no padrão americano, com um (ponto) e não vírgula.

$totalDaSoma=6.7;

String: é qualquer texto, palavra, um caractere, ou grupo de caracteres, simbolos que podem ser definidos com aspas simples (') ou duplas (").

$nome = "Sandra";

Boolean: para armazenar True/False.

$presente = true;

Vazia: consiste em uma variável vazia, não tem nenhum valor atribuído a ela.

$vazia = null;

Observem que é diferente se colocar assim:

$vazia = '';

No exemplo da linha anterior significa que é uma variável string, mesmo que vazia.

Declarando Constante

É utilizada a função DEFINE:

Define ('nome variavel', valor)

Para utilizar essa variável, não se usa o $.

Exemplo:

Faça um programa que declare como variável constante o valor de PI = 3,14 A seguir no programa declara uma variável chamada raio e atribui o valor 2 metros a essa variável. Ao final o programa mostra o valor da área desta circunferência de 2 metros de raio.

```
<?php

define ('PI',3.14);

$raio=2;

$area=PI*$raio*$raio;

echo "A área da circunferência de raio = $raio metros é de $area metros quadrados";

?>
```

Mesclando Variáveis

Consiste em concatenar informações.

```
<?php

        $nome = "Sandra";

        $sobrenome = "Amorim";

        //$nomeCompleto = $nome.$sobrenome;

        //$nomeCompleto = $nome.'  '.$sobrenome;

        $nomeCompleto = "$nome $sobrenome";

        //Agora obs. abaixo tirando do comentário essa linha de código

        $nomeCompleto = '$nome $sobrenome';
```

```
    echo $nomeCompleto;
?>
```

Percebam quando coloca uma string com aspas simples, o php vai entender que tudo que tem dentro daquela string é um valor literal, então significa que ó que está dentro é o testo real. Por isso que na linha de comando onde tem $nomeCompleto = "$nome $sobrenome"; vai mostrar da forma que está digitado na string.

NOTE: . (PONTO) concatena variáveis

Execute no navegador para ver o que foi impresso!!!!

Substitua o ECHO por PRINT e veja se é igual.

Mas existem outras formas de concatenar:

```
<?php
    $nome = "Sandra";
    $sobrenome = "Amorim";
    $nomeCompleto = $nome;
    $nomeCompleto .= $sobrenome;
    echo $nomeCompleto;
?>
```

Observem na linha de comando $nomeCompleto .= $sobrenome; foi colocado um .(ponto) antes do igual (=), com isso vai pegar o que já tinha armazenado na variável $nomeCompleto e a seguir adicionar o que tem na variável $sobrenome. Ou seja, é o mesmo que fazer conforme exemplo a seguir.

```
$nomeCompleto = $nome.$sobrenome;
```

Tipo de Operadores

Operadores Aritméticos

Operador	Nome	Exemplo
+	Adição	$a + $b
-	Subtração	$a - $b
*	Multiplicação	$a * $b
/	Divisão	$a / $b
%	Módulo	$a % $b
.	Concatenação	$a.$b

Operadores Lógicos

Operador	Nome	Exemplo
and ou &&	E	($a==0) && ($b==1)
or ou ||	OU	($a==0) \|\| ($b==1)
!	Negação	($a != 0)

Operadores de Comparação

Operador	Nome	Exemplo
==	Igual a	$a == $b
!=	Não igual a	&a != $b
<>	Não igual a	&a <> $b
<	Menor que	&a < $b
>	Maior que	&a > $b
<=	Menor ou igual a	&a <= $b
>=	Maior ou igual a	&a >= $b

Operadores de Atribuição de Combinação

Operador	Equivale a	Exemplo
+=	$a = $a + $b	$a += $b
-=	$a = $a - $b	$a -= $b
*=	$a = $a * $b	$a *= $b
/=	$a = $a / $b	$a /= $b
%=	$a = $a % $b	$a %= $b
.=	$a = $a.$b	$a .= $b

Operadores de Incremento e Decremento

Podem ser utilizados de duas formas:

Antes: ++$a

Depois: $a++

Operador	Objetivo	Exemplos
++$i	Pré-incremento	$a = 10 $c = ++$a //$c recebe 11 e $a fica com 11
$i++	Pós-incremento	$a = 10 $c = $a++ // $c recebe 10 e $a fica com 11
--$d	Pré-decremento	$a = 10 $c = --$a //$c recebe 9 e $a fica com 9
$d--	Pós-decremento	$a = 10 $c = $a-- //$c recebe 10 e $a fica com 9