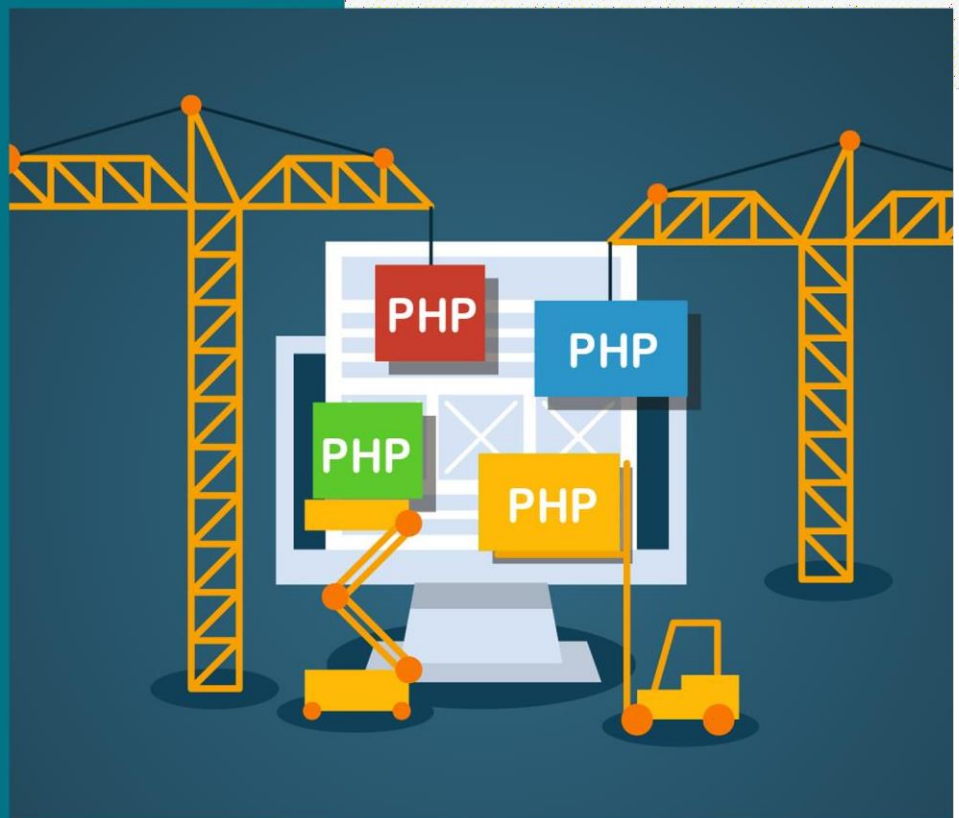


---

# AGENDA 1

---

## PHP: INTRODUÇÃO A SERVER-SIDE



GEEaD - Grupo de Estudos de Educação a Distância  
Centro de Educação Tecnológica Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO  
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO  
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS  
PROGRAMAÇÃO MOBILE I

**Expediente**

Autor:

*GUILHERME HENRIQUE GIROLI*

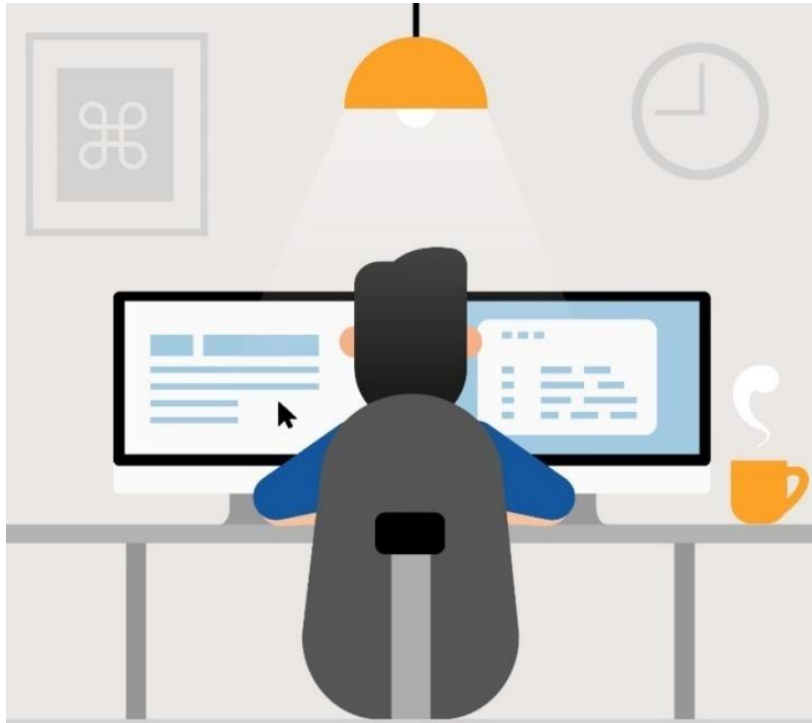
Revisão Técnica:

*Eliana Cristina Nogueira Barion*

Revisão Gramatical:

*Juçara Maria Montenegro Simonsen Santos*

Editoração e Diagramação: *Flávio Biazim*



Você já pensou em quantos sites você utiliza diariamente? Ou no quanto a internet facilita seu dia a dia?

Podemos destacar como principais exemplos: transações bancárias, estudos e pesquisas, utilização das redes sociais, compras em lojas virtuais dos mais diversos segmentos. Com tantos acessos e utilização de sites, você já pensou como eles funcionam? Como você consegue fazer uma compra e como a mercadoria chega até a sua casa?



Para exercer essas funções, esses tipos de sites precisam basicamente de dois componentes: um **cliente** e um **servidor web**. Imagine que você esteja consultando o seu extrato bancário! Muitos dos processos necessários para que as informações sejam apresentadas corretamente estão sendo executados no **servidor web** do banco. **O site do banco precisa de dados**, por exemplo: os dados da conta do cliente, os dados cadastrais do cliente, das transações financeiras realizadas, além de outras informações para garantir que ao final do processo, você obtenha as informações, conforme suas solicitações.

Vamos entender ou compreender como fazer aplicações web que permitem que os sites tenham a capacidade de realizar diversas **funcionalidades e serviços**? Nesta agenda, veremos um pouco de SERVER-SIDE (Lado do Servidor) e a introdução para **linguagem de programação PHP** (Hypertext Preprocessor – Pré processador de hipertexto) que nos fornecerá ferramentas e recursos para realizar as mais diversas funcionalidades de uma página WEB!



O desenvolvimento de um site pode ser muito complexo, do ponto de vista da programação. O mercado oferece diversas linguagens que seguem diferentes paradigmas. Vamos aprender a linguagem PHP (Hypertext Preprocessor – Pré processador de hipertexto), uma das mais conhecidas e utilizadas devido a sua tipagem dinâmica, utilizando o conceito Server-side como alicerce.



Sempre que você acessa um site, seja pelo computador, celular ou tablet, utiliza um navegador como o Internet Explorer, Chrome, Firefox, Safari etc. Esses navegadores conhecem basicamente três tecnologias: HTML, CSS e Javascript. O **HTML** tem a responsabilidade de **estruturar o conteúdo das páginas**, o **CSS** define a **formatação e a aparência do conteúdo** e o **Javascript** adiciona **interatividade a uma página web**.

Mas isso não é o suficiente para desenvolver sites mais complexos, como por exemplo: o site do banco que você utiliza para fazer suas transações ou um site de compras. Vamos pensar juntos: Onde ficarão armazenadas as informações da sua conta, como o seu extrato e os históricos de transferências ou últimas compras realizadas? Será que apenas o navegador consegue armazenar e exibir todas essas informações utilizando apenas essas três tecnologias apresentadas até agora?

Lembre-se de que o código que é executado no navegador está rodando na sua máquina, desta forma, todas essas informações teriam que estar no seu computador! Será que isso daria certo? Como fica, por exemplo, a questão da segurança dos seus dados?

O que acontece, nesse caso, é que quando você digita o endereço do site do banco no navegador do seu computador e pressiona a tecla “Enter”, ele faz uma solicitação, que chamamos de **requisição para o servidor do banco** (outro computador). Esse servidor, por sua vez, processa essa requisição, consulta o seu banco de dados e devolve uma resposta para o seu navegador. Esse código que é executado do lado do servidor utiliza linguagens, como: **Perl, Ruby, Python, PHP, Java e C#**.



Veja o caso do Zeca! Ele já desenvolveu alguns sites como *freelancer* e, como ele começou a ter cada mais clientes com maiores necessidades, os sites ficaram mais complexos, seus novos projetos começaram a precisar de novas tecnologias.

E foi assim que o Zeca começou a estudar novas tecnologias. Ele aprendeu a **a utilizar uma linguagem de programação server-side!**



***Você sabe como funciona essas linguagens? Mergulhe no tema desta agenda e descubra!***



Todo site que você acessa pelo navegador de Internet (Chrome, Internet Explorer, Mozilla, Firefox etc) está hospedado em um computador servidor e então, ao acessar qualquer site na internet, por meio do seu navegador de internet, está fazendo solicitações, ou seja, **requisições a esse servidor**, e, não apenas você, mas outras pessoas também estão fazendo requisições pela internet a todo o momento. Caso muitas pessoas estejam acessando o site ao mesmo tempo, cabe ao **servidor gerenciar esses acessos** de maneira correta. Portanto, quando você acessa um site está de certa forma, acessando uma série de estruturas na internet (**protocolos** de comunicação e **estruturas físicas**) e por trás de tudo isso existe um computador do tipo **servidor** controlando todos esses processos.

No servidor, há linguagens que podem ser instaladas para fazerem alguns procedimentos, como se **conectarem** a um banco de dados ou fazer **rotinas** de programação. Nas primeiras agendas, aprendemos algumas linguagens que não são processadas do lado do servidor, mas sim, do **lado do usuário (cliente)**, como o **HTML**, **Javascript** e **CSS**. Apesar de que estes arquivos (.html , .css ou .js) estejam hospedados (alocados) nesse servidor, eles são baixados pelos navegadores de Internet e quem os processa são os computadores do usuário.



Assim, concluímos que essas linguagens não teriam **acesso ao banco de dados**, sendo processadas pelos computadores, pois seria um risco à **segurança** e **integridade dos dados**, então, a melhor escolha é que a linguagem seja processada do **lado do servidor**, concorda?



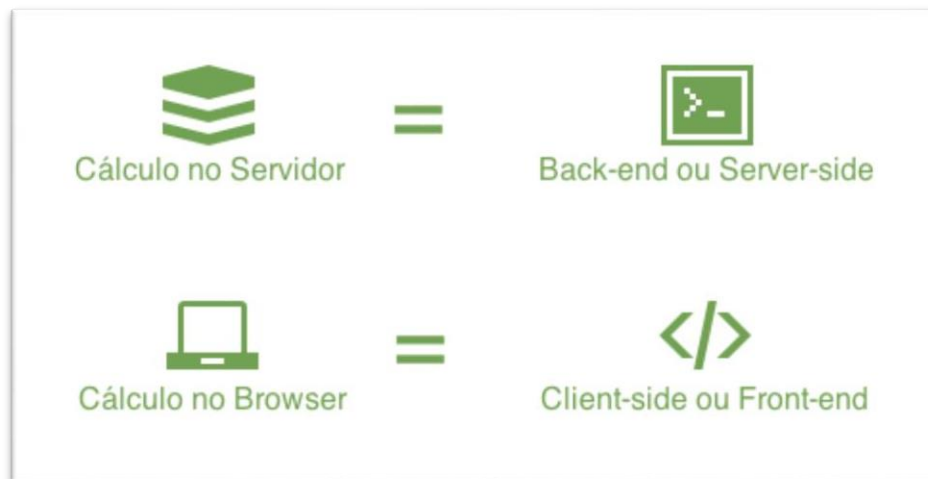


Imagem 03 - Adaptado de <http://tableless.github.io/iniciantes/assets/img/client-side-back-end.png>

A imagem 03 representa dois tipos de processamento: **lado do cliente** e **lado do servidor**. Com esses conceitos conseguiremos iniciar a programação em linguagem **server-side**, porém precisaremos de um **servidor web**.

Não há como trabalhar do lado do servidor sem que você tenha um servidor instalado. Você poderia acessar um servidor web externo, mas isso poderia gerar alguns custos de manutenção.

## Servidores Web

Você pode instalar e configurar manualmente o servidor **Apache**, a linguagem **PHP** e o **banco de dados**, porém há soluções muito mais simples para começar o desenvolvimento de seus projetos. Trata-se de **pacotes de softwares** que já instalam e configuram o servidor web, a linguagem **PHP** e **banco de dados**, e há diversas opções no mercado, como:

| Ferramenta     | Site Oficial / Download   |
|----------------|---|
| <b>XAMP</b>    | <a href="https://www.apachefriends.org/index.html">https://www.apachefriends.org/index.html</a> |
| <b>WAMP</b>    | <a href="http://www.wampserver.com">http://www.wampserver.com</a>                               |
| <b>EasyPHP</b> | <a href="https://www.easyphp.org/">https://www.easyphp.org/</a>                                 |
| <b>AppServ</b> | <a href="https://www.appserv.org/">https://www.appserv.org/</a>                                 |
| <b>Zwamp</b>   | <a href="http://zwamp.sourceforge.net/">http://zwamp.sourceforge.net/</a>                       |



Todas essas ferramentas citadas anteriormente necessitam de instalação, porém, no mercado existem versões portáteis disponíveis e arquivos executáveis que não necessitam de instalação, sendo possível desenvolver e demonstrar seus sites **PHP** em qualquer lugar e a qualquer momento, utilizando apenas um pendrive ou similar.

Neste curso utilizaremos a ferramenta **USBWebServer**, ela é gratuita e é possível realizar seu download por meio do link: <https://usbwebserver.yura.mk.ua/>.

Perceba que o site oferece duas versões: a 8.6.1 (PHP 5) e a 8.6.2 (PHP7), conforme mostra a imagem 04.



Imagem 04—Links para download da ferramenta USBWebserver.

Neste curso vamos utilizar a versão **PHP 7**, então clicaremos no segundo link de download. (Imagem 05)

[usbwebserver\\_v8.6.2.zip](#) - **PHP 7.1**

Imagem 05 – Link para download do PHP 7.1.

Depois de realizar o download do arquivo, basta descompactá-lo em uma pasta como demonstrado pela imagem ao lado.

Execute o aplicativo **usbwebserver** (Imagem 07), clicando duas vezes sobre o mesmo.

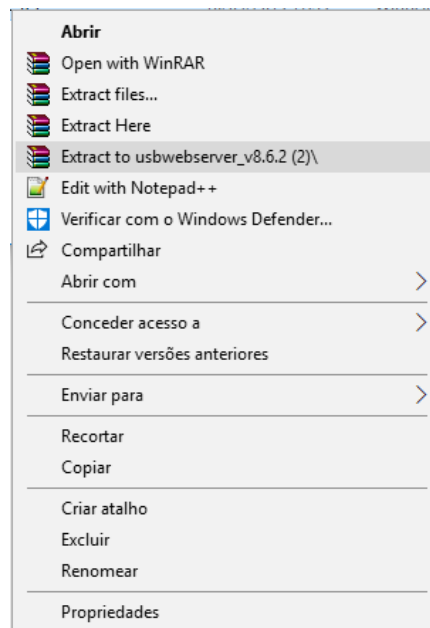


Imagem 06 – Descompactar arquivo baixado da internet

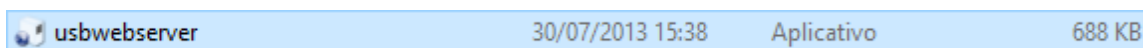


Imagem 07 – Aplicativo a ser executado conhecido como usbwebserver.

Se seu **firewall** estiver ativo, irão aparecer duas mensagens como na Imagem 08, pois trata-se de **permissões** para acesso externo ao servidor apache e ao banco de dados dessa ferramenta. Caso tenha outros computadores disponíveis em sua rede e você deseja que eles acessem o **apache** e **mysql**, clique em **permitir**, caso contrário clique em cancelar.

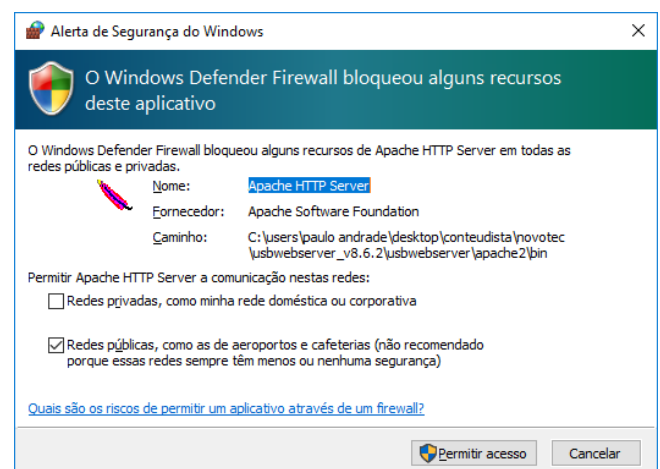
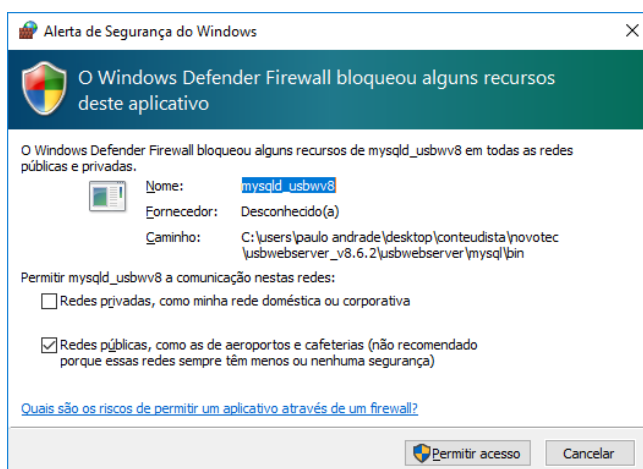


Imagem 08 – Permissão de Acesso externo com verificação do firewall.

Logo após, você deverá selecionar a linguagem que deseja utilizar (Imagem 09).

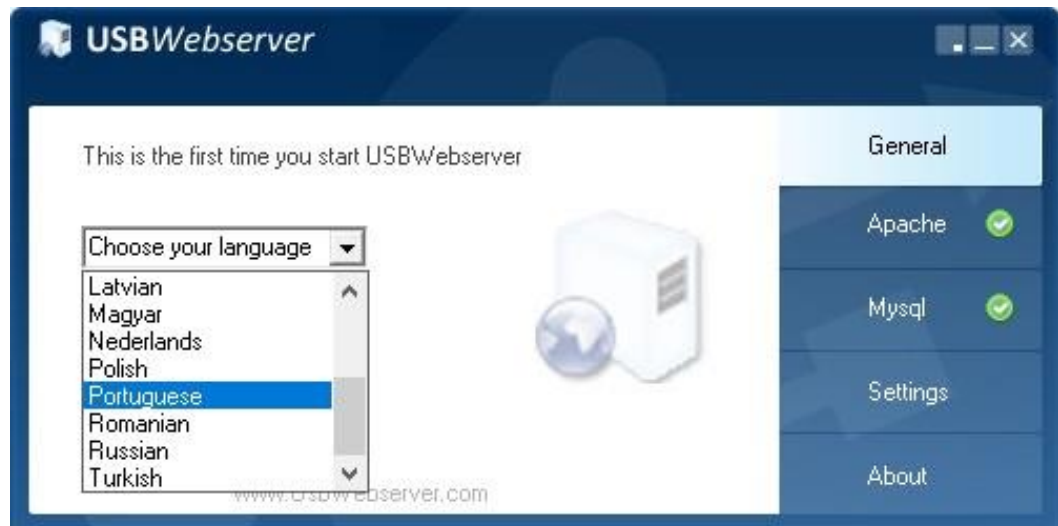


Imagem 09 – Escolha da linguagem, como a opção Portuguese.

Nas guias **apache** e **mysql** deverão aparecer duas bolinhas verdes, indicando que os serviços estão funcionando de forma adequada. (Imagem10)



Imagem 10 – Botões e serviços iniciados da forma adequada.

Obs.: Caso aconteça de uma ou ambas bolinhas ficarem vermelhas, com o símbolo de exclamação é porque algo impediu a inicialização do Apache e/ou MySql. (Imagem 11)

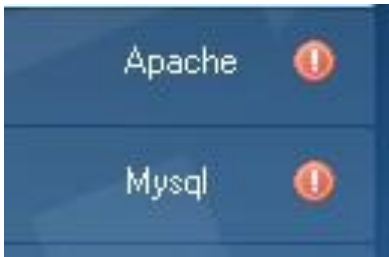


Imagem 11 – Erro na inicialização dos serviços, Apache e Mysql com símbolo de exclamação

#### Alguns dos possíveis problemas e respectivas soluções.

**1 – O local do arquivo:** quando você descompactou o USB Webserver, este procedimento foi realizado dentro de alguma pasta, caso alguma das pastas da hierarquia tenha algum caracter especial como “o”, os serviços podem deixar de iniciar. **Como resolver:** troque **Usb WebServer** de pasta.

**2 – Outro Servidor Web executando na máquina:** pode acontecer de existir na máquina outro servidor web. **Como resolver:** clique na guia configurações e **troque a porta apache de 8080 para 8081**, ou outra próxima (**8082**). **Obs:** “Caso você realize esta troca na hora de acessar suas páginas deverá utilizar: ”porta”. Exemplo: se trocou para a porta **8081**, deverá digitar na URL: **localhost:8081**.

**3- Outro servidor de Banco de dados executando na máquina:** pode acontecer de existir na máquina outro servidor de BD. **Como resolver:** clique na guia configurações e troque a **porta mysql de 3306 para 3307**, ou outra próxima (**3308**). **Obs.:** Caso você realize esta troca na hora de acessar, lembre-se de utilizar a porta escolhida.

Agora precisamos testar e verificar se está tudo correto! Para isso teremos dois caminhos:

1- Clicar sobre o botão **localhost** ou

2- Abri o navegador e digitar na URL: “**localhost**” e pressionar ENTER.

O resultado será a exibição do **PHP Info** e **webserver** em funcionamento:



Imagem 12 – Phpinfo, webserver funcionando.

Como discutido anteriormente, no **lado do servidor** estão **todos os arquivos do site**, portanto, ao utilizar um servidor web no computador, todos os arquivos do site ficarão na pasta/diretório do seu servidor, geralmente essa pasta tem o nome de “**www**”, “**public\_html**”, “**htdocs**”, no caso do **USB Wserver** é denominada “**root**”. Para acessá-la, basta abrir a pasta na qual o **USB Webserver** foi descompactado e abrir a pasta **root** ou clicar no botão “**root dir**”, no guia geral do aplicativo **USB WebServer**.

Quando requisitado, tudo o que for **PHP** será processado pelo servidor e enviados apenas os resultados para o cliente.



Depois de tudo isso, vamos criar a primeira página em **PHP**?

Então vamos lá:

1. Dentro da pasta **root**, crie uma pasta denominada “**Agenda 1**”, nesta pasta vamos concentrar todas as atividades.
2. Abra o **Visual Studio Code**, crie um arquivo e salve dentro da pasta recém-criada com o nome “**ola**”, não esquecendo de escolher o tipo **PHP**, clique em salvar. (Imagem 13)

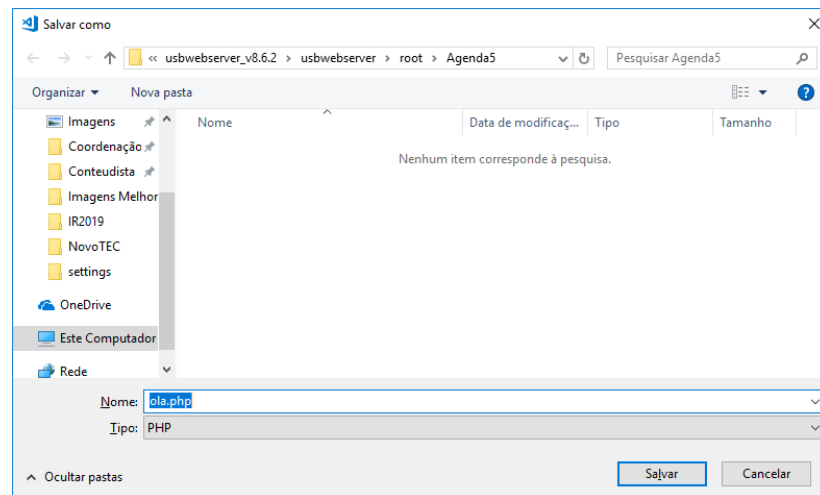


Imagem 13 – Salvando Arquivo PHP.

**Obs:** Nunca utilize caracteres especiais e espaços em branco para nomear arquivos.

Perceba que após o arquivo ter sido salvo, o **Visual Code** coloca o símbolo do **PHP** demonstrando que se trata de um arquivo **PHP** (Imagem 14).

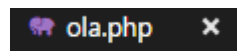


Imagem 14 –Imagem da guia doArquivo no Visual Studio Cod- arquivo salvo com o nome “ola” e extensão “.php”.

3. O próximo passo será a criação dos delimitadores PHP, como apresenta o código a seguir:

```
<?php
?>
```

Precisamos dos delimitadores pois, apesar do arquivo ser php, o mesmo pode possuir códigos HTML, logo os delimitadores identificam para o servidor que estes códigos são PHP.

4. Agora vamos enviar uma mensagem para a tela do navegador. Para isso, vamos utilizar o comando “echo”, este comando é basicamente um dos mais utilizados em sistemas PHP, por ele ser o responsável por enviar dados para o usuário, codifique então:

```
<?php  
echo "Olá Mundo";  
?>
```

Salve o arquivo. Abra seu navegador favorito e digite na url:  
“localhost/Agenda5/ola.php”.

5. O resultado deverá ser como o apresentado na imagem a seguir:

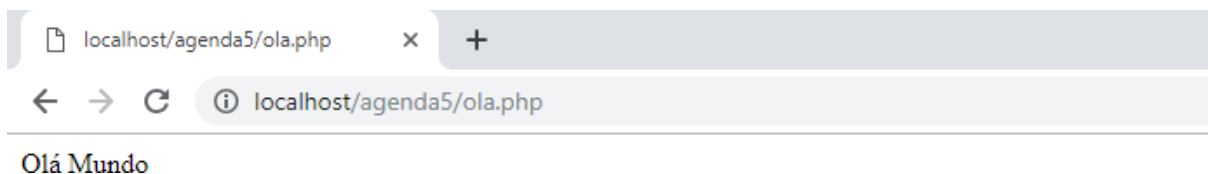


Imagem 15 –Resultado no navegador da página com a frase “Olá Mundo”.

Pronto, criamos a primeira página PHP! Neste momento, é possível verificar uma diferença entre o programado “Olá Mundo” em php e o em html, para verificar basta clicar com o botão direito e clicar em exibir código fonte da página.

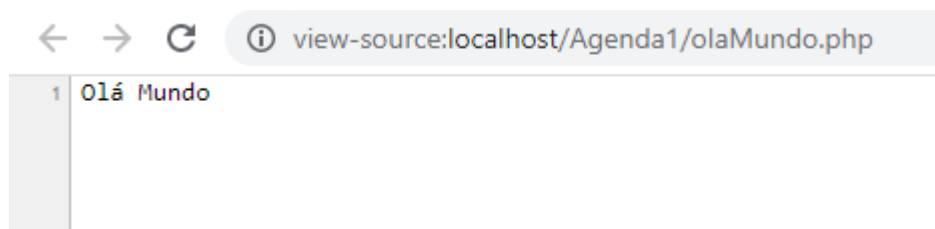


Imagem 16 –Resultado no navegador de exibir o código fonte da página “Olá Mundo”.

Perceba, que os códigos PHP não estão sendo exibidos, qual o motivo para isso? Quando o usuário faz a requisição para o servidor, ele busca a página, interpreta todo o código PHP e envia como resposta apenas o necessário para o navegador interpretar e exibir a página para o usuário que, neste exemplo, é apenas a Frase codificada.



**Obs:** O **USBWebServer** vem com uma configuração para não mostrar no site os erros de codificação. Quando pensamos no produto acabado, esta configuração é muito válida, porém, para o processo de aprendizagem trocar essa configuração será muito útil.

Para realizar a alteração, basta entrar na pasta **UsbWebServer**, e depois em uma pasta denominada **settings**. (Imagem 17)

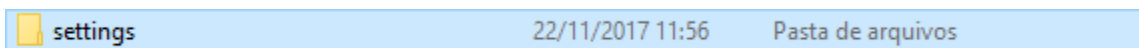


Imagem 17 – Pasta settings.

Dentro desta pasta terá um arquivo denominado **php.ini**, clique com o botão direito sobre ele e escolha a opção “**abrir com**”, escolha o Visual Studio Code ou IDE que você está utilizando para o desenvolvimento e pressione **CTRL + F** (Caso esteja utilizando Visual Studio Code), e coloque para procurar “**display\_errors = Off**”, esta configuração estará provavelmente na linha 477, conforme a imagem a seguir.

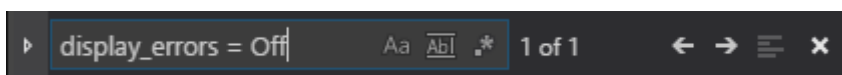


Imagem 18 – `display_errors=Off`.

Substitua o **Off** por **On** e salve, pronto! Quando acontecer algum erro no código php o servidor enviará uma mensagem, para a tela.



Utilizando o que foi visto até agora....

1. Crie um arquivo PHP na pasta Agenda 1.
2. Mostre na tela os dados a seguir:
  - a. Nome Completo;
  - b. Idade;
  - c. Profissão.



Confira a seguir se você conseguiu resolver os desafios propostos!

O código abaixo deve ser criado no **PHP** e salvo na pasta Agenda 1.

```
<?php
echo"Nome: Zeca da Silva. Idade: 19. Profissão: Estudante";
?>
```

### Resultado no Navegador

Ao digitar no navegador **localhost/agenda5/voceNoComando1.php** teremos como resposta:

“Nome: Zeca da Silva, Idade: 19 e Profissão: estudante”, conforme a imagem a seguir.

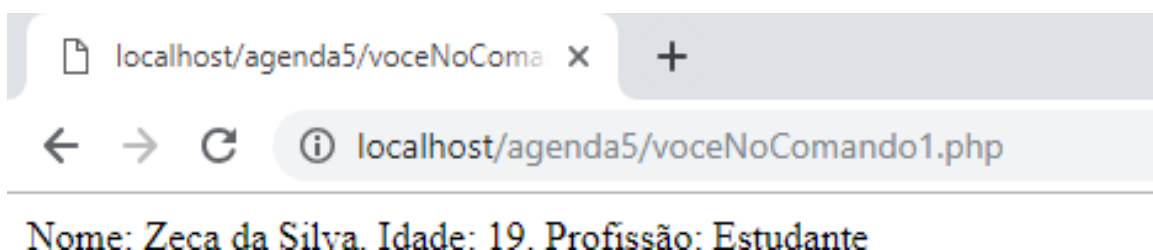


Imagem 19 –Resultado no navegador do Exercício Você no Comando.

### Comentários

**Obs.:** Da mesma forma que **HTML** e **CSS**, o **PHP** também possui comentários. Você pode adicionar comentários em seu arquivo PHP usando “//” comentário em linha

```
<?php
//Comentário.
?>
```

Ou utilizando /\* \*/ para comentário em bloco.

```
<?php
/*
    Comentário em Bloco
*/
?>
```



Não deixe também de assistir o video:



Fonte: Desenvolvedor Back-end: O que é Server-Side e Client-Side - (TekZoom, 2019).

Confira neste vídeo as reais diferenças entre Front-end e Back-end e descubra também o que é Server-Side e Client-Side. Disponível em <https://www.youtube.com/watch?v=bkDJe5UAvfM>. Acessado em 10/02/2020.

### Links

- CAELUM. “Introdução a PHP”. Acessado em 04/08/2023.

<https://www.caelum.com.br/apostila/apostila-html-css-javascript.pdf>

### Livros

- OGLIO, Pablo D. PHP Programando Com Orientacao a Objetos. 4ª Edição. São Paulo. Novatec. 2018. ISBN: 978-8575226919