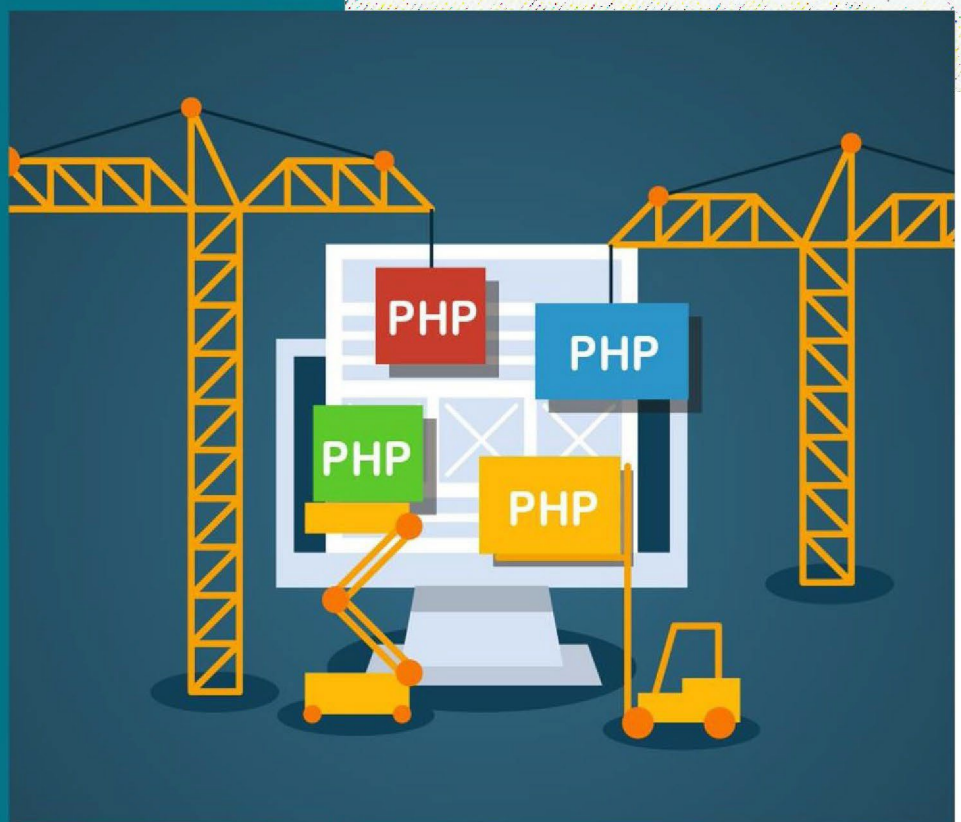

AGENDA 1

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
DESENVOLVIMENTO WEB

Expediente

Autor:

GUILHERME HENRIQUE GIROLI

Revisão Técnica:

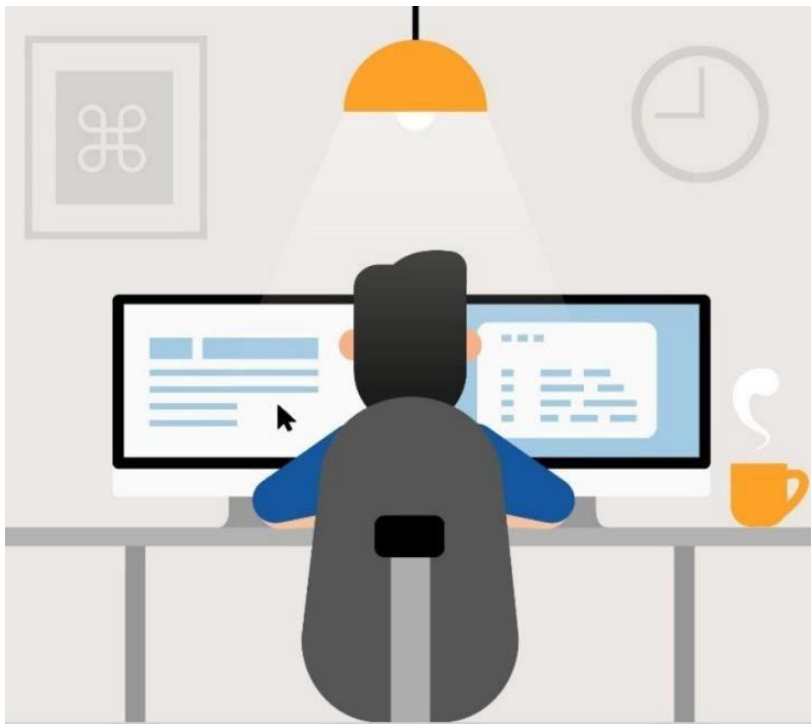
Eliana Cristina Nogueira Barion

Kelly Cristiane de Oliveira Dal Pozzo

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 **utilizar uma linguagem de programação server-side!**



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



MERGULHANDO
NO TEMA...



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?

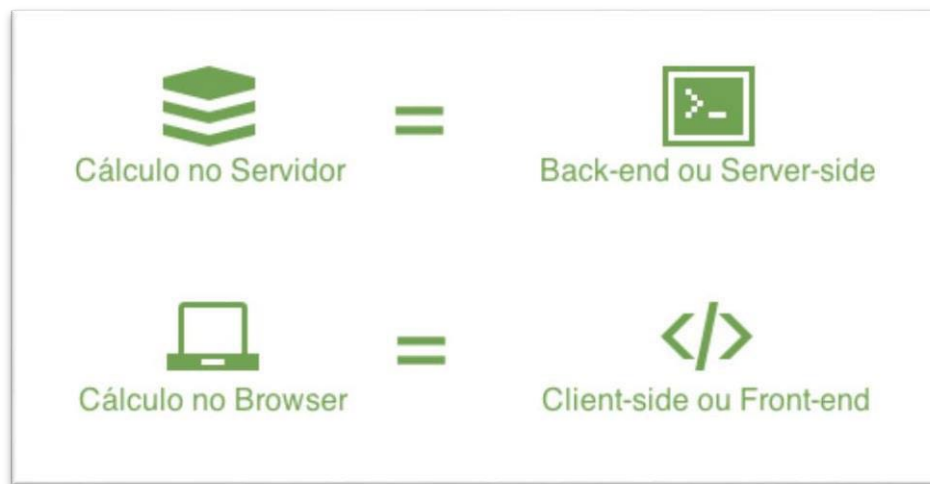


Imagem 03 - Adaptado de <http://tableless.github.io/iniciantes/assets/img/client-side-back-end.png>

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
XAMP	https://www.apachefriends.org/index.html
WAMP	https://wampserver.aviatechno.net/
EasyPHP	https://www.easyphp.org/
AppServ	https://www.appserv.org/
Zwamp	http://zwamp.sourceforge.net/
UwAmp	https://www.uwamp.com/en/?page=download

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.



O mesmo processo de instalação foi apresentado no componente Tecnologia da Informação II, caso já tenha instalado o Xampp por lá, não é necessário realizar a instalação novamente.

Configuração do Ambiente

Para acompanhar as agendas deste módulo, será necessário configurar um ambiente de desenvolvimento que permita trabalhar tanto com a linguagem de programação PHP quanto com o Banco de Dados MySQL, conteúdo que você também está acompanhando na agenda do componente Tecnologia da Informação II. Para atender a essa necessidade, oferecemos duas opções de ferramentas: XAMPP e USBWebserver.

Ambas as ferramentas foram selecionadas por sua praticidade e capacidade de atender às demandas dos conteúdos abordados nos dois componentes técnicos apresentados no módulo. No entanto, é importante ressaltar que **você deve escolher apenas uma das opções** e seguir o passo a passo indicado para configurá-la corretamente.

Opção 1: Instalação do ambiente XAMPP:

Acesse o site oficial: <https://www.apachefriends.org/index.html>

Faça o download da versão recomendada de acordo com o seu Sistema Operacional:

The screenshot shows the XAMPP website with the following content:

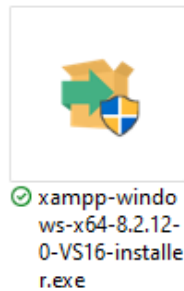
- What is XAMPP?**
XAMPP is the most popular PHP development environment
XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.
- Download** (Click here for other versions)
- XAMPP for Windows** 8.2.12 (PHP 8.2.12)
- XAMPP for Linux** 8.2.12 (PHP 8.2.12)
- XAMPP for OS X** 8.2.4 (PHP 8.2.4)

New XAMPP release 8.2.12, 8.1.25 and 8.0.30

Hi Apache Friends!
We just released a new version of XAMPP for Windows for PHP versions 8.2.12, 8.1.25 and 8.0.30. New versions for Linux and OS X will come soon! You can download these new installers...

[Read more »](#)

Após o download, clique no instalador para iniciar a execução:



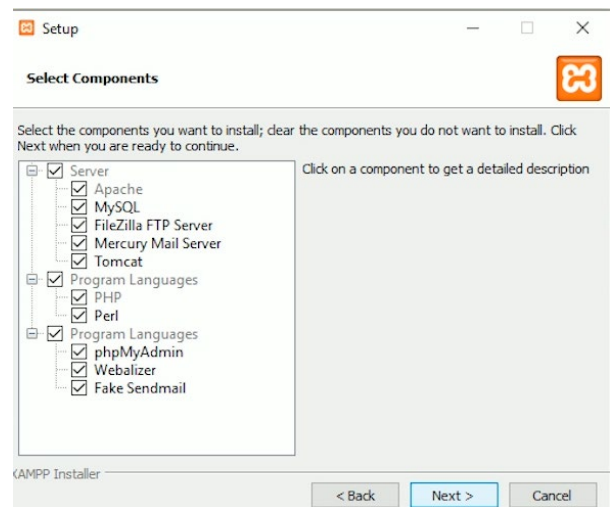
Durante a instalação, selecione os componentes:

Apache (servidor web).

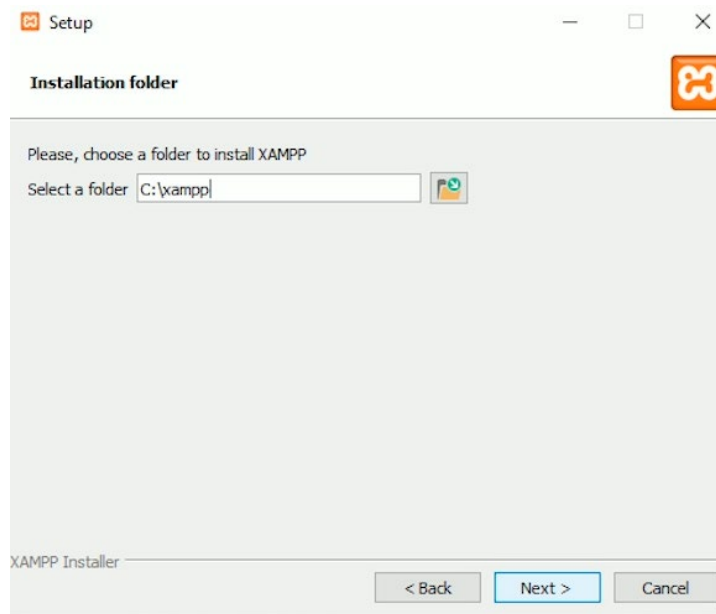
MySQL (banco de dados).

PHP (linguagem de programação).

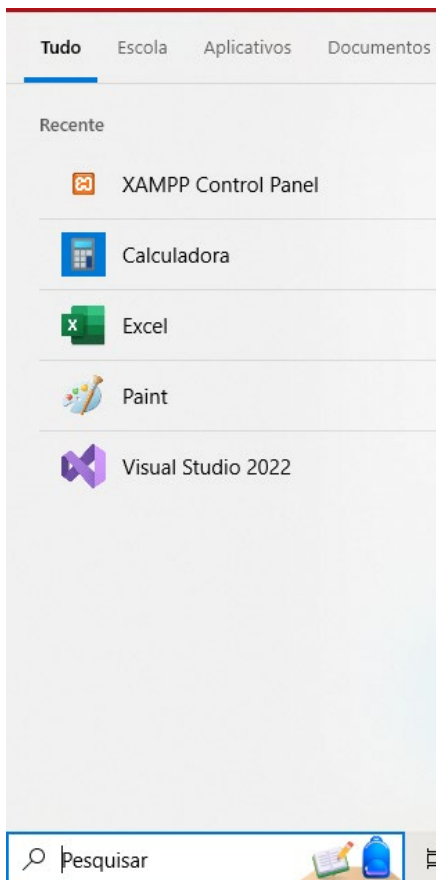
phpMyAdmin (interface gráfica para gerenciar bancos de dados).



Normalmente, instala-se programas na pasta Arquivos de Programas, mas o ideal é manter o Xampp com o caminho mais curto possível – na maioria dos casos, C:\xampp.



Após concluir instalação, abra o Painel de Controle do XAMPP.

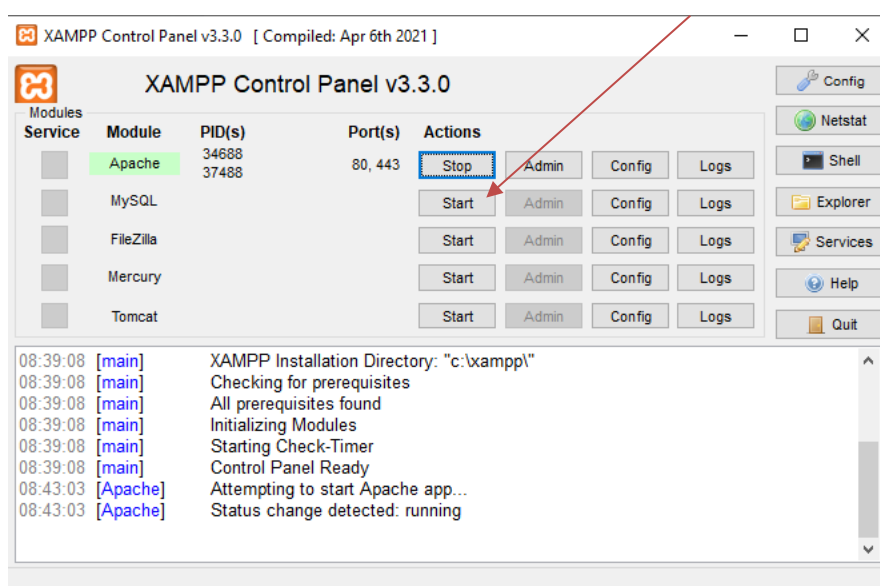


Dica:

Caso o Painel de Controle não apareça automaticamente, o ícone fica disponível ao lado do relógio.



Inicie os serviços Apache e MySQL:



A pasta htdocs é um diretório especial dentro do ambiente de desenvolvimento configurado pelo servidor Apache, sendo a pasta raiz padrão do servidor. Essa pasta é utilizada para armazenar os arquivos que serão processados e disponibilizados pelo servidor web, como páginas PHP, HTML, JavaScript e CSS. Estando localizada dentro do diretório onde o XAMPP foi instalado. Por exemplo:

No Windows: C:\xampp\htdocs.

Dentro da pasta htdocs, é possível criar subpastas para organizar os projetos. Por exemplo:

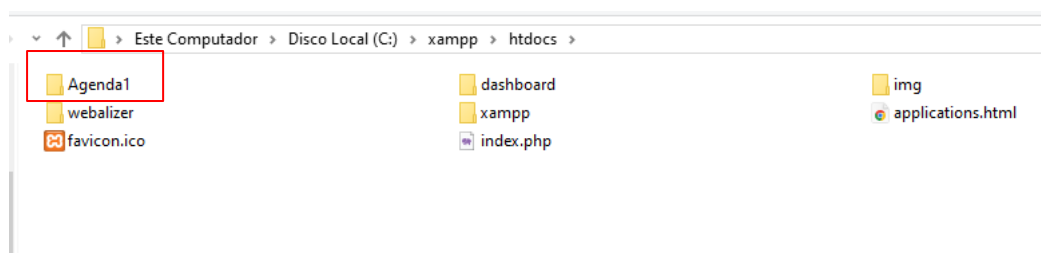
- C:\xampp\htdocs\Agenda1
- C:\xampp\htdocs\projeto_teste

Cada subpasta pode representar um projeto ou uma agenda de estudo, facilitando a navegação e o desenvolvimento.

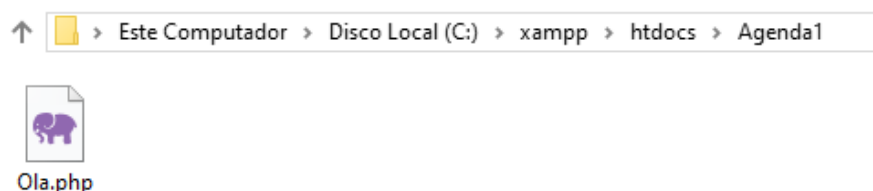
Todos os arquivos colocados na pasta htdocs estarão disponíveis para acesso via navegador, utilizando o endereço <http://localhost/>.

Vamos fazer um teste prático:

Localize a pasta htdocs na pasta raiz do seu computador e crie um novo diretório chamado Agenda1.



Dentro dessa pasta, crie um arquivo chamado ola.php.



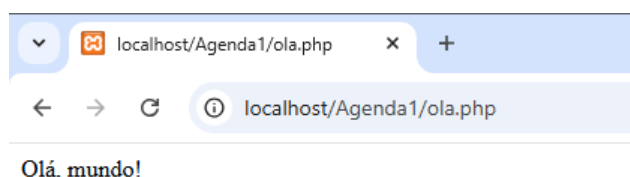
Para criação e manipulação de arquivos PHP você pode utilizar, assim como para arquivos HTML, qualquer editor de textos simples. Porém, para nossas aulas, conforme já feito no módulo anterior, vamos continuar utilizando o editor VSCode.



Adicione o seguinte código no arquivo:

```
<?php  
echo "Olá, mundo!";  
?>
```

Acesse o arquivo pelo navegador, digitando o seguinte endereço: <http://localhost/Agenda1/ola.php>

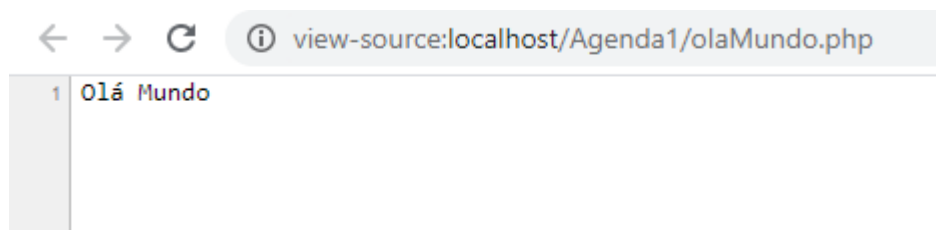


Se tudo estiver correto a frase Olá, Mundo será exibida no navegador.

Caso tenha problemas com a instalação do Xampp você pode utilizar o USBWebserver, seguindo o passo a passo:

https://eadtec.cps.sp.gov.br/midiateca/arquivos/ag1_web_ii_instalacaousbserver.pdf

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.



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.



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!

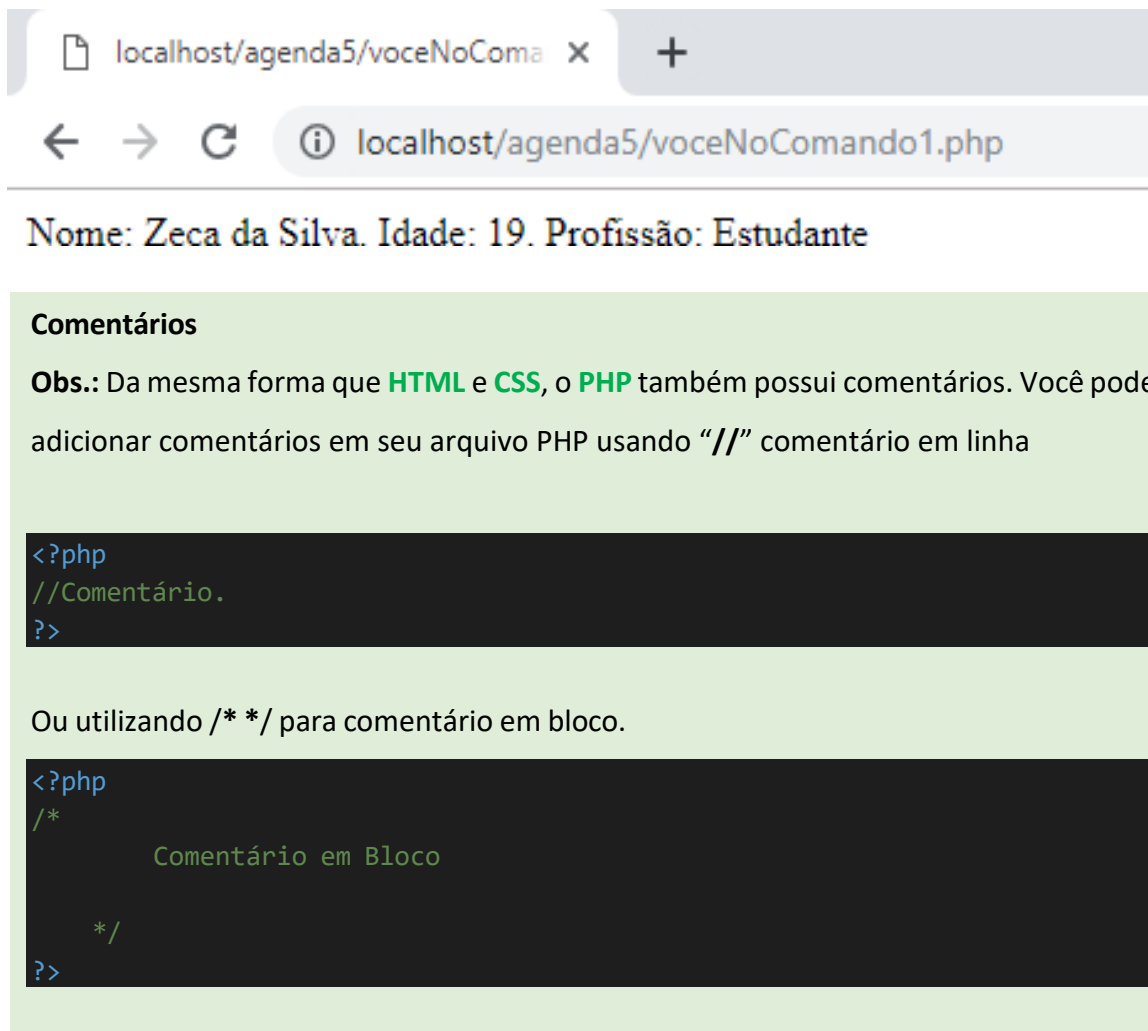
O código abaixo deve ser criado no **PHP** e salvo na pasta Agenda 1.

```
<?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.



Testes Locais

Sempre salve os arquivos dentro de htdocs antes de tentar acessá-los pelo navegador, pois o servidor Apache só reconhece arquivos nesse diretório.

Livros

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