# Fundamentos

## O que é o PHP

O PHP é uma linguagem de programação que visa auxiliar o HTML de forma a trazer funcionalidades ao site. É uma linguagem de código aberto, ou seja, qualquer um tem acesso ao código fonte. Os programas são executados ao lado servidor (*backend*).

PHP = *Personal Home Page Hypertext Preprocessor*.

## História

Essa linguagem foi criada por Rasmus Lerdorf em 1994 como um “framework” para criação de páginas em HTML. De lá pra cá ela evoluiu bastante e atualmente está na versão 8, sendo que esta ainda não é muito difundida, utiliza-se a 7.4 (*major release*). Obs. Recomenda-se a instalação estável. A ideia inicial era deixar as páginas dinâmicas, pois havia problemas na mudança de estado da página naquela época. Atualmente existem frameworks modernos como o Wordpress que são bastante utilizados, em 2020 estimava-se que 80% dos sites da internet atualmente são feitos em PHP, além de frameworks específicos para nichos como blogs, e-comerces, etc.

## Instalação

### VSCode no Windows

O VSCode é uma das melhores IDEs gratuitas disponíveis no mercado, ele apresenta uma série de vantagens:

* Terminal integrado
* Suporte a extensões
* Versatilidade

Para instalar no Windows

1. acesse [www.code.visualstudio.com](http://www.code.visualstudio.com)
2. Baixe o executável .exe e execute como administrador
3. Aceite os termos de uso
4. Marque:
   1. Adicionar varável de ambiente ao PATH
   2. Abrir arquivos e pastas com botão direito do mouse
5. Clique em install e aguarde a instalação ser concluída
6. Ao abrir o vscode, faça login com a conta Microsoft

### PHP no Windows

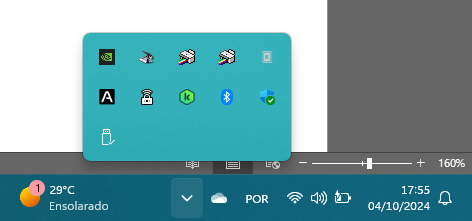
O PHP pode ser instalado no Windows sem a instação de ferramentas, mas recomenda-se o XAMPP (documentação). Ele inclui os principais recursos para desenvolvimento Web como:

* Apache
* MySQL
* Pearl
* PHP

O XAMPP simula um servidor Web com poucos cliques e de forma simples.

Curiosidade: No Linux o servidor Web é executado a todo momento, além de fácil configuração, por isso normalmente as empresas optam por servidores Linux, normalmente Debian ou similar. Recomenda-se uma VM.

Primeiramente instalar o XAMPP

1. Acesse [www.apachefriends.org](http://www.apachefriends.org)
2. Baixar o executável .exe para a versão correspondente do Windows
3. Executar o instalador como administrador
4. Escolher os componentes
5. Escolher a pasta de instalação, é sugerido *“C:\xampp”*
6. Escolher a linguagem
7. Desmarcar o checkbox
8. Prosseguir com a instalação

O xampp pode ser controlado pela bandeja do Windows (imagem ao lado), onde pode ser aberto o painel de controle da aplicação, onde será feita as ativações e desativações de seus componentes. Para testar se o Xampp está funcionando, abra o navegador e digite “localhost”. Se tudo der certo aparecerá a dashboard do Xampp.

### PHP no Linux

No Linux também é possível instalar o php de forma independente, porém é interessante instalar outros componentes necessários, por isso é recomendável a instalação de uma pilha LAMP (Linux, Apache, MySQL e PHP). Dessa forma é possível executar não só o php como também projetos web completos.

1. Atualizar o sistema com sudo apt udate
2. Use o comando sudo apt install apache2 para instalar o apache. A partir dessa etapa, o Apache ficará rodando em segundo plano.
3. O próximo passo é configurar o firewall para permitir aplicações web.
   1. Primeiramente use o comando: sudo ufw app list
   2. Para saber qual a porta principal do Apache, use o comando sudo ufw app info “Apache Full”
   3. Para habilitar o apache: sudo ufw allow in “Apache Full”
   4. Faça o teste para ver se o Apache está funcionando digitando no navegador: “localhost”
4. O próximo é o MySQL:
   1. Instalação usando o comando: sudo apt install mysql-server
   2. Escolha um password para o usuário e root
   3. O comando sudo mysql para abrir e o comando exit para finalizar
5. Finalmente instale o php e pacotes de comunicação com os anteriores:
   1. Use o comando: sudo apt install php libapache2-mod-php php-mysql
6. Pra finalizar digite o comando sudo systemctl restart apache2 para reiniciar o apache com os componentes adicionados.
7. Para verificar se deu certo, digite o comando sudo systemctl status apache2. A bolinha verde indica que deu certo.

Após a instalação, todos os arquivos devem ser armazenados na pasta */var/www/html* para que dessa forma o servidor apache consiga reconhece-los e consequentemente o PHP possa executar.

### Instalação do MySQL *Workbranch*

Observação

Se a instalação do XAMPP já foi realizada, esse procedimento para instalação do MySQL não é necessário. Isso porque o MySQL já vem incluso no XAMPP, no qual pode ser acessado e configurado no painel de controle do aplicativo.

MySQL é um tipo de linguagem para edição e consulta de banco de dados. Para instalar o MySQL siga os seguintes passos

1. Acesse: [www.mysql.com/downloads/](http://www.mysql.com/downloads/)
2. Escolha a versão Community (grátis)
3. Em seguida escolha MySQL for Windows e escolha a opção de tamanho maior, na qual já vem com todos os pacotes inclusos.
4. Clique em *“No thanks, just stay my download”.*
5. Execute o instalador como administrador
6. Selecione ou a opção Full ou Custom (necessário escolher os pacotes)
7. Escolha a pasta de instalação
8. Ele vai mostrar todos os pacotes que serão instalados
9. A instalação iniciará e ele pedirá para progredir com a instalação
10. Caso tenha instalado o xampp, pode haver um conflito de portas, sendo necessário alterá-la.
11. Ele vai sugerir usar autenticação com senha, defina uma.
12. Configure o serviço que será executado pelo Windows.
13. Ele vai aplicar todas as configurações quando aplicar o execute.
14. Teste a conexão
15. Executar script de execução de dados
16. Finalizar

O próximo passo é adicionar o MySQL às variáveis de ambiente do Windows

1. Vá até a pasta de instalação do MySQL server e copie o caminho no explorador de arquivos
2. Vá em “Editar variáveis de ambiente”
3. Variáveis de usuário
4. Clique em path e em seguida em editar
5. Clique em novo e cole o caminho copiado
6. Confirme tudo

Para verificar se deu certo, use um terminal de sua preferência e digite mysql –version.

## Executando o PHP no Windows

Após a instalação do Xampp, todos os arquivos devem ser armazenados na pasta *C:\xampp\htdocs\* para que dessa forma o servidor apache consiga reconhece-los e consequentemente o PHP possa executar. Dessa forma o primeiro passo é a criação de uma pasta para o projeto. Dentro dela crie um novo arquivo chamado “index.php”

A sintaxe de um código php é:

<?php

echo “Olá Mundo”;

?>

O nome *index* é obrigatório, pois indica que esse será a página inicial do site, de forma similar ao *html*. Outra observação é que os nomes de diretórios e arquivos não podem possuir espaços, letras maiúsculas e caracteres especiais, isso porque isso costuma resultar em erros.

## Localhost

É uma forma de acessar um servidor local em nossa própria máquina, sendo equivalente 127.0.0.1. Desta forma é possível simular como se o arquivo ou site estivesse sendo processado em um servidor.

IP = Internet Protocol

Localhost = O próprio computador

## Sintaxe do PHP

Para que um bloco seja executado é preciso que ele esteja entre **Tags do php**, sendo este representado por:

Abertura: **<?php**

Fechamento: **?>**

Fim de cada instrução: **;**

Primeiramente o código dentro das tags será executado pelo php e depois ele vai imprimir na tela. Lembrando que essa sintaxe é obrigatória!

## Dependências do PHP

As informações e detalhes do php e suas dependências podem ser identificadas por meio da função phpinfo(), isso porque há uma série de componentes que podem estar ou não separados, que servem como pré-requisitos para as aplicações desenvolvidas. Isso é importante pois em uma empresa ou um servidor é imprescindível saber tudo que está instalado nele, pois pode ser que hajam funções já criadas e que podem ser utilizadas. Outro motivo é a segurança, pois executar funções desatualizadas pode colocar em risco.

# Conceitos Básicos

## Case sensivity

Consiste na sensibilidade a casas maiúsculas e minúsculas, isso varia de linguagem para linguagem, no caso do php não há essa diferença, exceto para **variáveis**, conforme veremos mais adiante. A tag <br> assim como o html serve para adicionar uma quebra de linha.

### Instruções de código

As instruções de código são separadas por ponto e vírgula, isso porque é possível adicionar mais de um comando na mesma linha. Quando há uma estrutura maior (ex. condicional), a continuidade é dada por abertura e fechamento de chaves e nesses casos não necessita de ponto e vírgula. Exemplo de condicional abaixo:

If (5 > 2) {

echo “Dentro do if”

}

Else {

echo “Não está dentro do if”

}

## Espaços em branco

Os espaços em branco são ignorados, no entanto é uma má prática e deixa o código desorganizado. Isso porque durante a compilação os espaços são removidos. A única exceção são as strings, ou se houver uma quebra de linha no meio do código.

## Comentários

Linhas de código que são ignoradas e por isso são usados para observações ou cabeçalho. Os comentários podem ser escritos:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| // | Comentário de uma linha | // Este é um comentário |
| # | Comentário de uma linha | # Este é um comentário |
| /\* \*/ | Comentário de múltiplas linhas | /\* Este é um comentário  Resolver isso aqui \*/ |

Obs. Em hipótese alguma escreva informações sensíveis sobre a empresa, APIs ou pessoais.