



# Introdução ao PHP

**Prof. Fábio Procópio**  
**Prof. João Nascimento**



2

# Relembrando...

- Na aula passada, finalizamos nossos estudos introdutórios sobre a linguagem SQL;
- Nesta aula, começaremos a estudar uma das linguagens de programação mais utilizados no desenvolvimento de aplicações para Web: PHP;
- E aí? Vamos nessa?!?!?





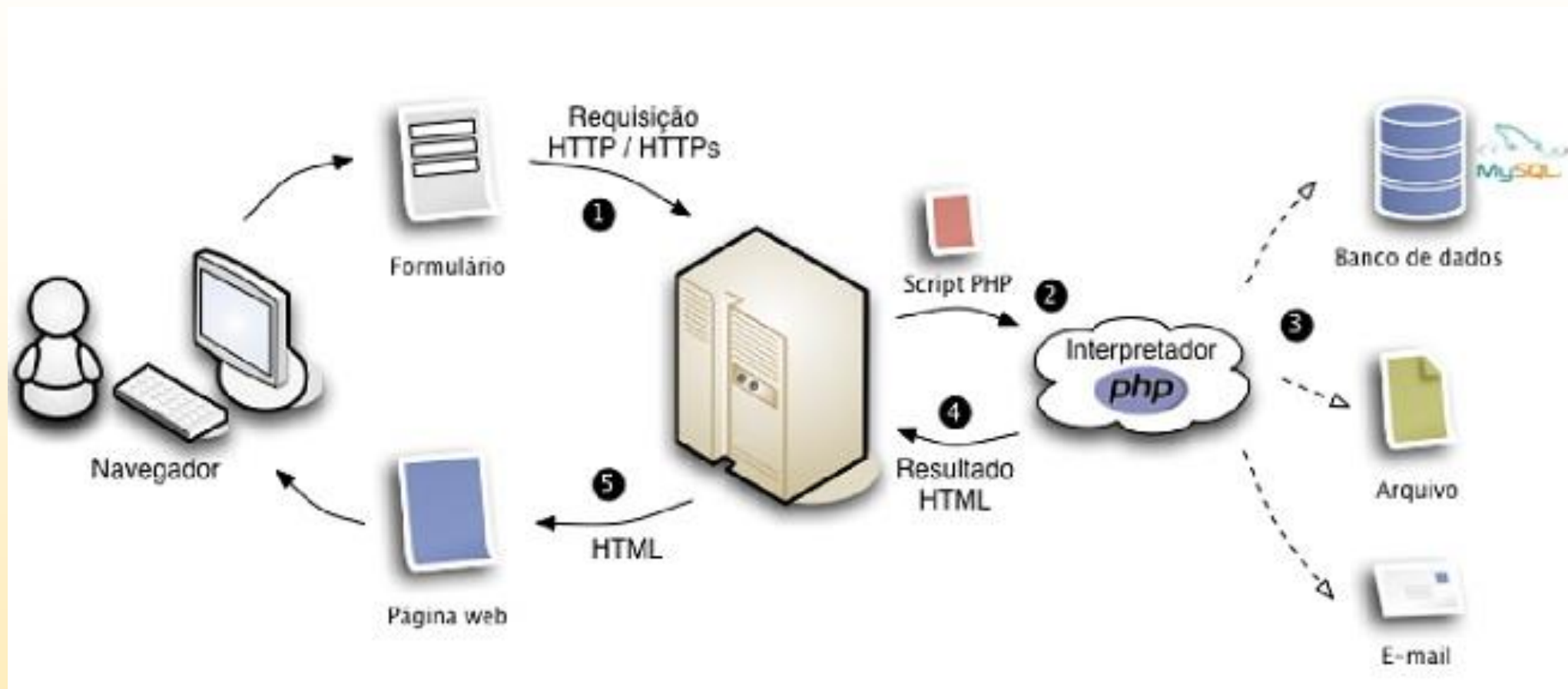
3

## Um pouco de história...

- A primeira versão do PHP surgiu em 1995, escrita por Rasmus Lerdorf
  - Ele escreveu alguns *scripts* na linguagem de programação Perl a fim de criar uma *interface* para analisar as estatísticas de acesso ao seu website
  - Ele batizou esta versão de **Personal Home Page Tools** (PHP Tools)
- A segunda versão surgiu após aprimoramentos nos códigos da versão 1.0
  - Esta versão foi batizada de **Personal Home Page/Forms Interpreter** (PHP/FI)
  - Nesta versão era possível a interpretação de dados enviados a partir de formulários HTML
- Em 1996, Rasmus liberou o código fonte do PHP para a comunidade desenvolvedora
  - A partir daí, o PHP caminhou para se tornar uma linguagem de programação
- Em 1998, os israelenses Zeev Suraski e Andi Gutmans reescreveram PHP/FI e criaram a versão 3.0 batizando-a, finalmente, de **PHP Hipertext Preprocessor**.



# Esquema de funcionamento



Fonte: MARINHO (2014) - Cliente-Servidor-PHP



# Características

- Algumas características do PHP são:
  - É **interpretada pelo servidor** e, por isso, é chamada de *server-side* (lado do servidor)
  - Diferente de páginas construídas usando apenas HTML, PHP pode gerar **conteúdo dinâmico**
  - Em geral, é embutida dentro do código HTML
  - Herdou características de C, de C++, de Java e de Perl
  - Hoje é utilizado em mais de 75% dos servidores que oferecem serviços web



6

## Projeto *Apache Friends* → XAMPP

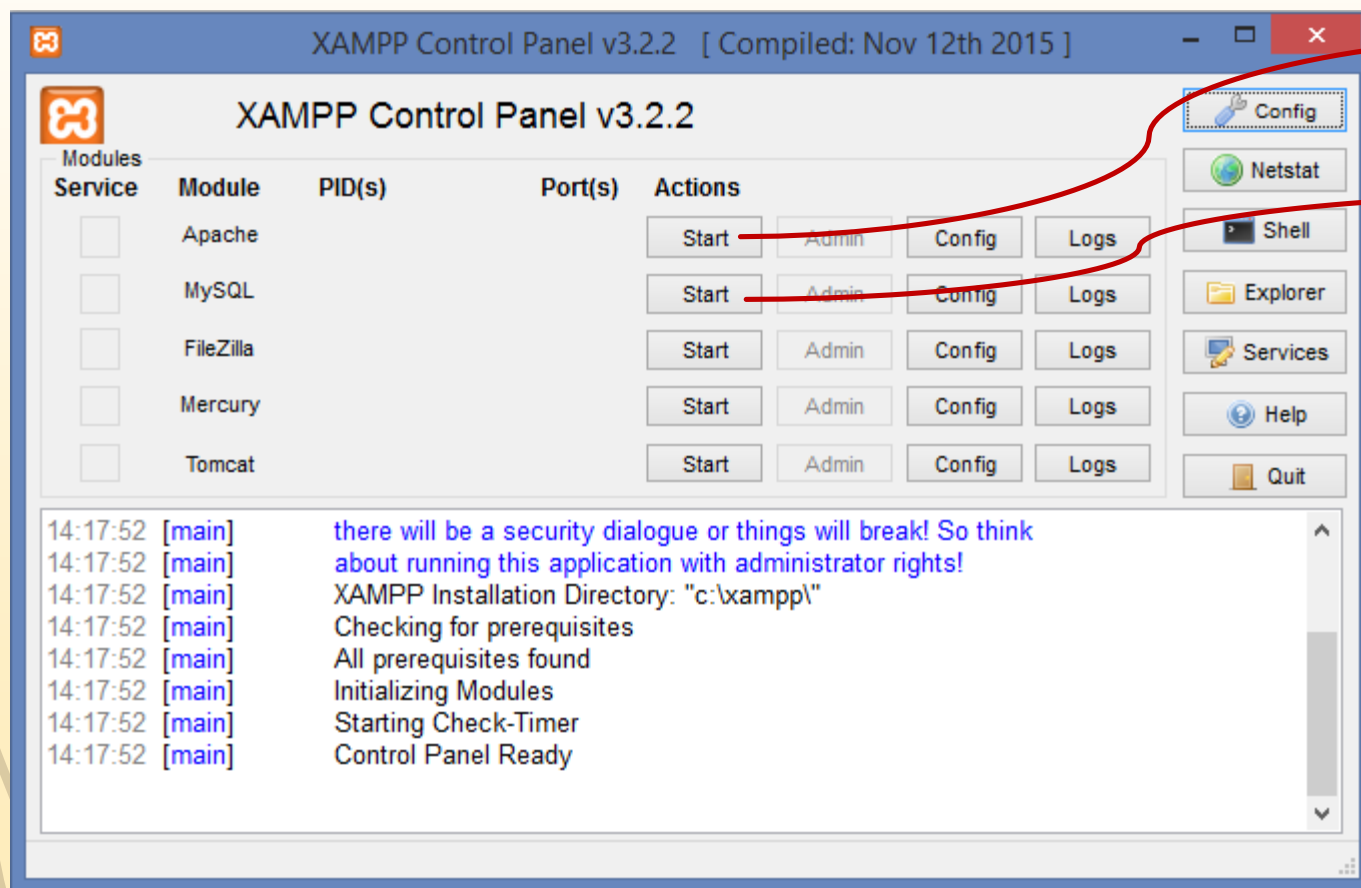
- *Apache Friends* é um projeto sem fins lucrativos desenvolvido para disponibilizar o servidor Apache;
- Dentro do *Apache Friends*, foi construído o XAMPP
  - XAMPP é fácil de instalar e contém PHP, MariaDB e Perl
  - XAMPP é uma boa alternativa para desenvolvimento local
  - Se ainda não instalou o XAMPP, [baixe-o aqui](#)
  - Caso tenha dúvidas de como instalá-lo, [clique aqui](#)



7

# XAMPP – Painel de controle

- Acesse o painel de controle do XAMPP e inicialize os serviços Apache e MySQL:



Clique neste botão para iniciar o servidor web Apache.

Clique neste botão para iniciar o servidor de banco de dados MySQL.



8

# O que é necessário para usar PHP?

- 1) Um editor de texto da sua preferência (usaremos o Apache Netbeans IDE)
  - Se você não tem o Apache NetBeans instalado, [baixe-o aqui](#);
  - Para utilizarmos o Apache Netbeans, é necessário instalar Java Development Kit. Se não o tem instalado, [baixe-o aqui](#);
    - Para facilitar a instalação, baixe o arquivo com **jdk-12.0.1\_windows-x86\_bin.exe** ou **jdk-12.0.1\_windows-x64\_bin.exe**, dependendo do seu sistema operacional Windows
  - [Neste vídeo](#), veja como configurar o Apache Netbeans
- 2) Além disso:
  - Um interpretador do PHP
  - Um servidor web como Apache, IIS, Tomcat
  - Um servidor de BD como MySQL, SQL Server, Postgres, Oracle, etc.
- 3) Se você estiver com o XAMPP e com o Apache Netbeans instalados corretamente, já podemos começar!!! 😊





9

# Já posso começar a programar?

► Vamos verificar se está tudo certo com o nosso ambiente de desenvolvimento PHP


- 1) **ATENÇÃO:** certifique-se que o serviço web está sendo executado. Em nosso caso, verifique no XAMPP, o status do Apache
- 2) Em seguida, acesse o Apache Netbeans: **File → New Project**
- 3) Em **Categories**, selecione **PHP**. Em **Projects**, selecione **PHP Application**
- 4) Clique em **Next**. Na tela seguinte, em **Project Name**, digite **OlaMundo**. Clique em **Finish**
- 5) No arquivo **index.php** criado, dentro da tag `<body>`, digite o código a seguir:

```
<?php
    phpinfo();
?>
```

- 6) Clique com o botão direito dentro do *script* digitado e selecione **Run File**;
- 7) Se tudo estiver OK, uma página HTML será exibida com diversas informações sobre o PHP.




# Tudo certo com o PHP!

**PHP Version 7.3.6**

System	Windows NT PROCOPIO_PAIVA 6.3 build 9600 (Windows 8.1 Professional Edition) AMD64
Build Date	May 29 2019 12:05:38
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cmdscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,TS,VC15
PHP Extension Build	API20180731,TS,VC15
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 support	enabled
DTrace support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v3.3.6, Copyright (c) 1998-2018 Zend Technologies





# Tags do PHP

- Todo trecho de um código em PHP deve estar entre as tags **<?php** e **?>**
  - As tags permitem que o servidor web saiba que esse trecho deve ser processado, ou seja, não é um conteúdo HTML
  - A tag **<?php** indica o início de um trecho de código
  - Já a tag **?>** indica o final do trecho de código
  - Exemplo:

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



## Estrutura do *script*

- Um programa PHP pode possuir comandos HTML e códigos PHP;
- Os comandos HTML devem aparecer fora das tags PHP;
- Ao fim de cada comando, deve ser incluído um ponto e vírgula;
- Porém, dentro das tags PHP, só podem aparecer comandos HTML se usarmos os comandos **echo** ou **print** para escrevê-los:

```
<?php
    echo "Nossos cursos:<br>";
    echo "<ul type= 'square' >";
    echo "<li>Informática</li>";
    echo "<li>Sistemas para Internet</li>";
    echo "<li>Mecatrônica</li>";
    echo "</ul>";
?>
```



# Comentários

```
<?php
//Isto é um comentário
#Isto também é um comentário, semelhante ao do Python
/*Por fim, esta também é outra maneira de escrever
comentários. Normalmente, usa-se quando se deseja
comentar mais de uma linha.
*/
?>
```



# Arquivos PHP

```
<?php
    echo ".php - arquivos com essa extensão contêm código PHP. <br>";
    echo ".class.php - arquivos com essa extensão contêm classes PHP. <br>";
    echo ".inc.php - arquivos com essa extensão contêm constantes e funções.";
?>
```



# Principais Referências

- 1) JORNAL DO DEV. **Aula 01 – História do PHP.** Disponível em: <https://jornadadodev.com.br/cursos/curso-de-php-para-incipiantes/aula-01-historia-do-php>. Acessado em: 08 abr 2019.
  
- 2) JORNAL DO DEV. **Aula 02 – Como funciona o PHP.** Disponível em: <https://jornadadodev.com.br/cursos/curso-de-php-para-incipiantes/aula-02-como-funciona-o-php>. Acessado em: 08 abr 2019.