



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Charqueadas

Desenvolvimento Back-end I

Linguagem PHP

Prof. Sérgio Yoshimitsu Fujii
sergiofujii@ifsul.edu.br

Linguagens *Server-side*

Tipos de linguagens

Linguagens de desenvolvimento web

- **Client-side**
(Também conhecida como *front-end*);
- **Server-side**
(Também conhecida como *back-end*).

Linguagens *Server-side*

Client-side

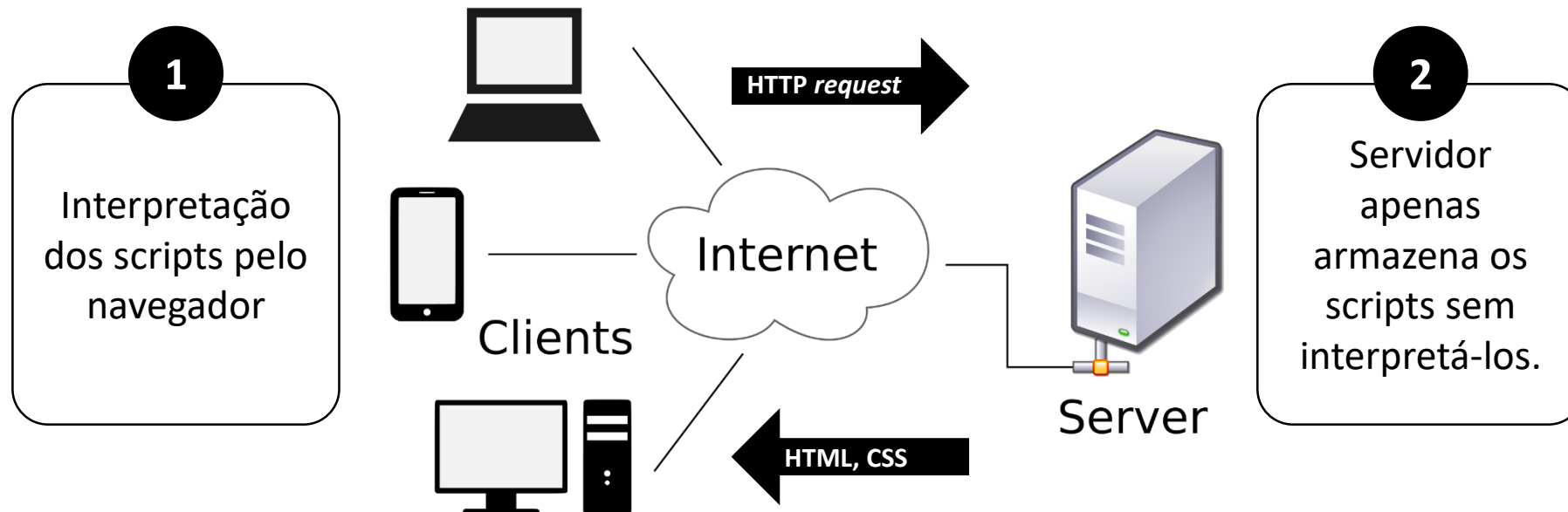
Principais características

- Quem vai processar essa linguagem é o **navegador** (Chrome, IE, Firefox, etc...). Significa que o cálculo é executado no lado do cliente;
- A principal vantagem é a economia de **bandwidth** (largura de banda), fornecendo uma resposta mais rápida de sua aplicação por não haver processamento externo.
- Exemplos de linguagens:
 - HTML, CSS, JavaScript.

Linguagens *Server-side*

Client-side

Funcionamento



Linguagens *Server-side*

Server-side

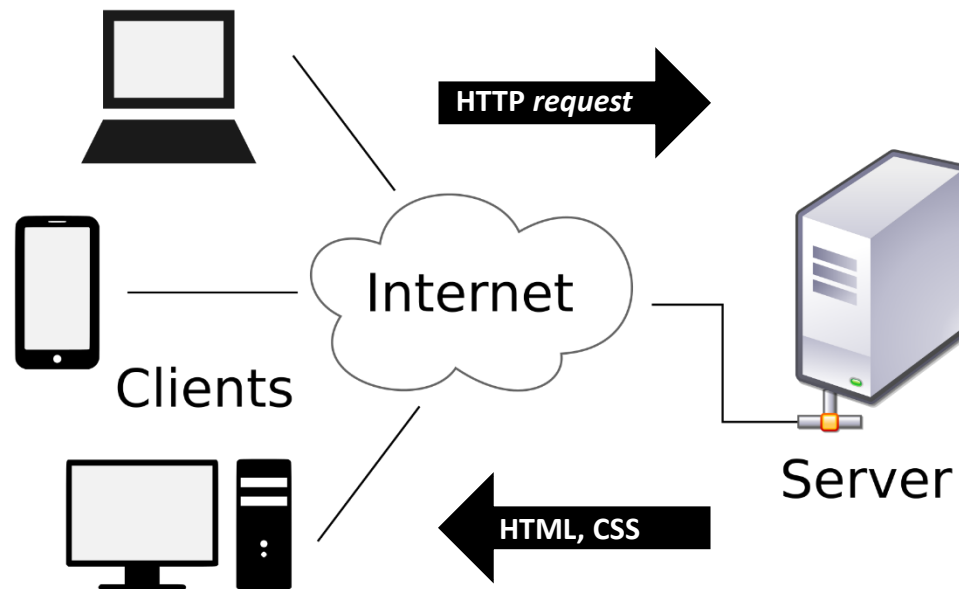
Principais características

- Aplicações que rodam no **servidor**. Isso quer dizer que você vai escrever um código onde o servidor vai processá-lo e então vai mandar para o seu navegador a resposta;
- Exemplos de linguagens:
 - **PHP**, ASP.

Linguagens *Server-side*

Server-side

Funcionamento



1

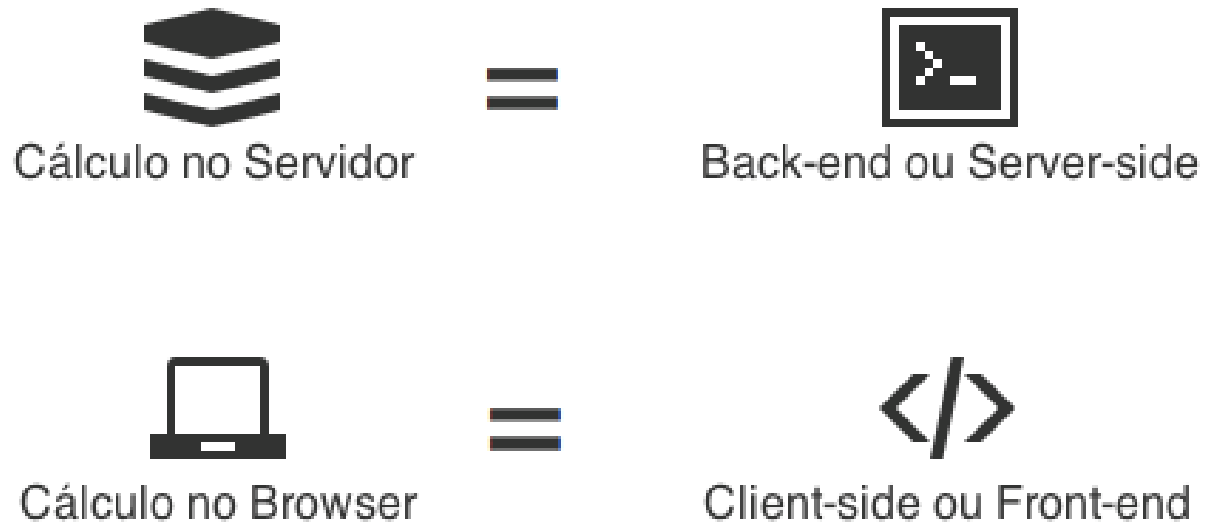
Processamento da requisição no servidor.

Verifica se o usuário está logado;
Busca informações no banco de dados;
Coleta e trata dados de um formulário HTML, etc.

Linguagens *Server-side*

Client-side x Server-side

Resumindo ...



Linguagens *Server-side*

HTTP *request*

Como funciona uma requisição HTTP

- Quando o navegador mostra uma página da internet, esse processo é realizado utilizando o HTTP (*Hypertext Transfer Protocol*);
- O HTTP é um protocolo de pedido e resposta, o que significa que o computador envia uma solicitação ao pedir uma determinada página ou arquivo, e o servidor na internet envia de volta essa página ou arquivo.

Linguagens *Server-side*

HTTP *request*

Como funciona uma requisição HTTP

- Além de solicitar um determinado arquivo, o HTTP envia várias informações para o servidor, entre elas:
 - O endereço IP;
 - A versão do navegador que está sendo usado;
 - Qual página fez a HTTP *Request*;
 - O idioma em uso, etc.

A linguagem PHP

O que é PHP

Hypertext Preprocessor

- PHP é o acrônimo de *Hypertext Preprocessor* (inicialmente foi chamado de *Personal Home Page Tools*);
- PHP é uma linguagem de desenvolvimento desenvolvida por desenvolvedores web;
- PHP é uma linguagem de criação de scripts do lado do servidor, que pode ser incorporado ao HTML ou ser utilizado de forma independente;
- PHP é código aberto e livre.



A linguagem PHP

O que é PHP

Hypertext Preprocessor

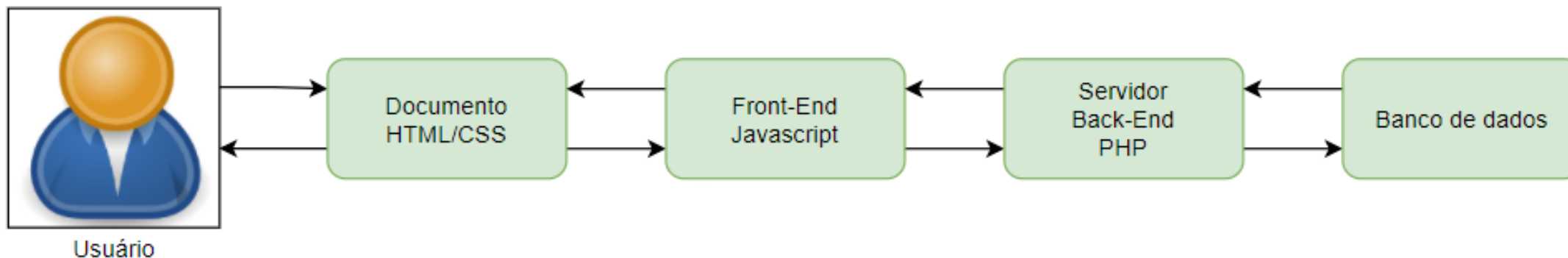
- Foi criado por **Rasmus Lerdorf** em 1994/1995;
- Primeiramente fazia estatística de acessos do seu currículo online;
- Atualmente, está na versão 8.1 (2022).



Linguagem PHP

PHP - (PHP: Hypertext Preprocessor)

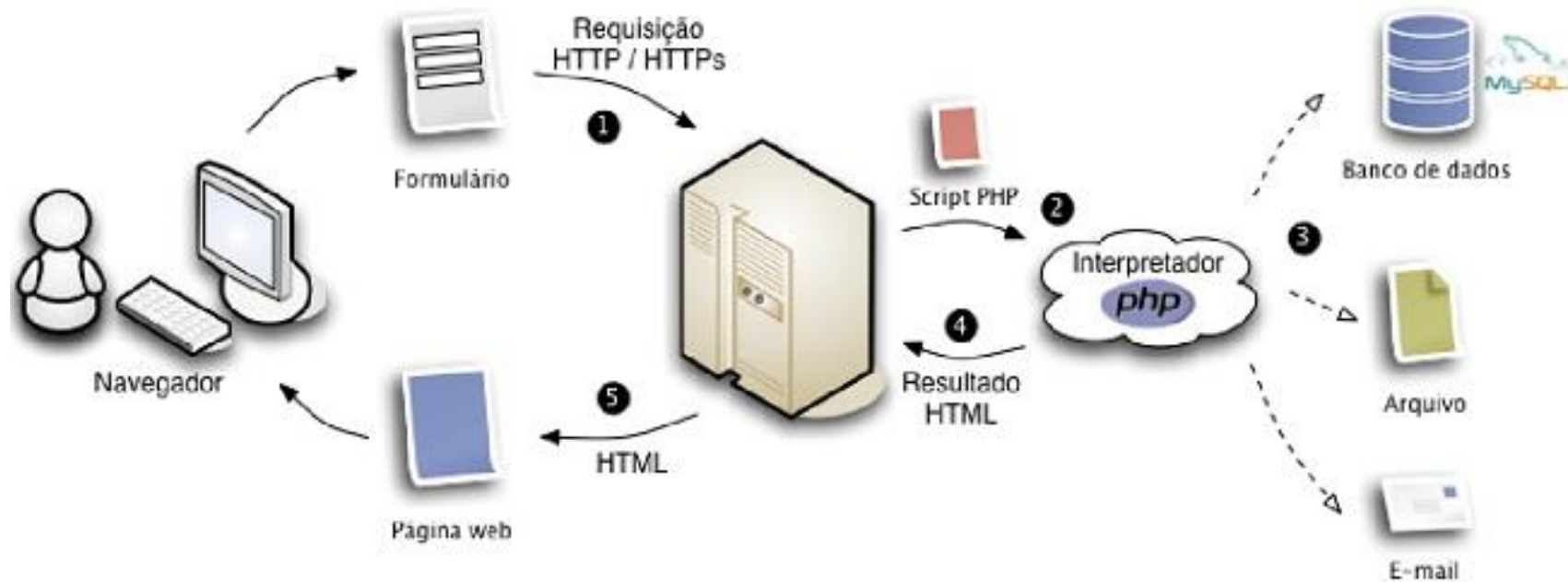
- Linguagem server-side
 - Scripts executados no servidor
- Interação com banco de dados
- Manipulação de informações sensíveis
- Sincronização de informações
- Validação de dados do usuário
- Garantia de código não manipulado pelo usuário



A linguagem PHP

O servidor PHP

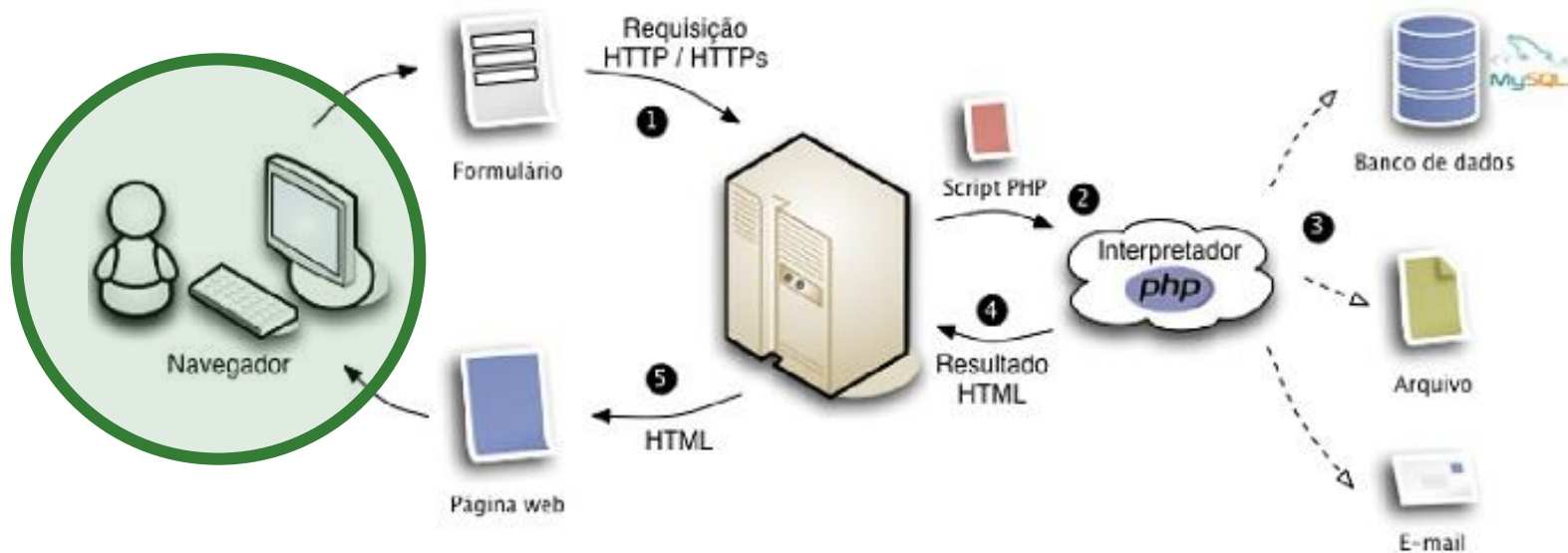
Como funciona?



A linguagem PHP

Funcionamento do servidor PHP

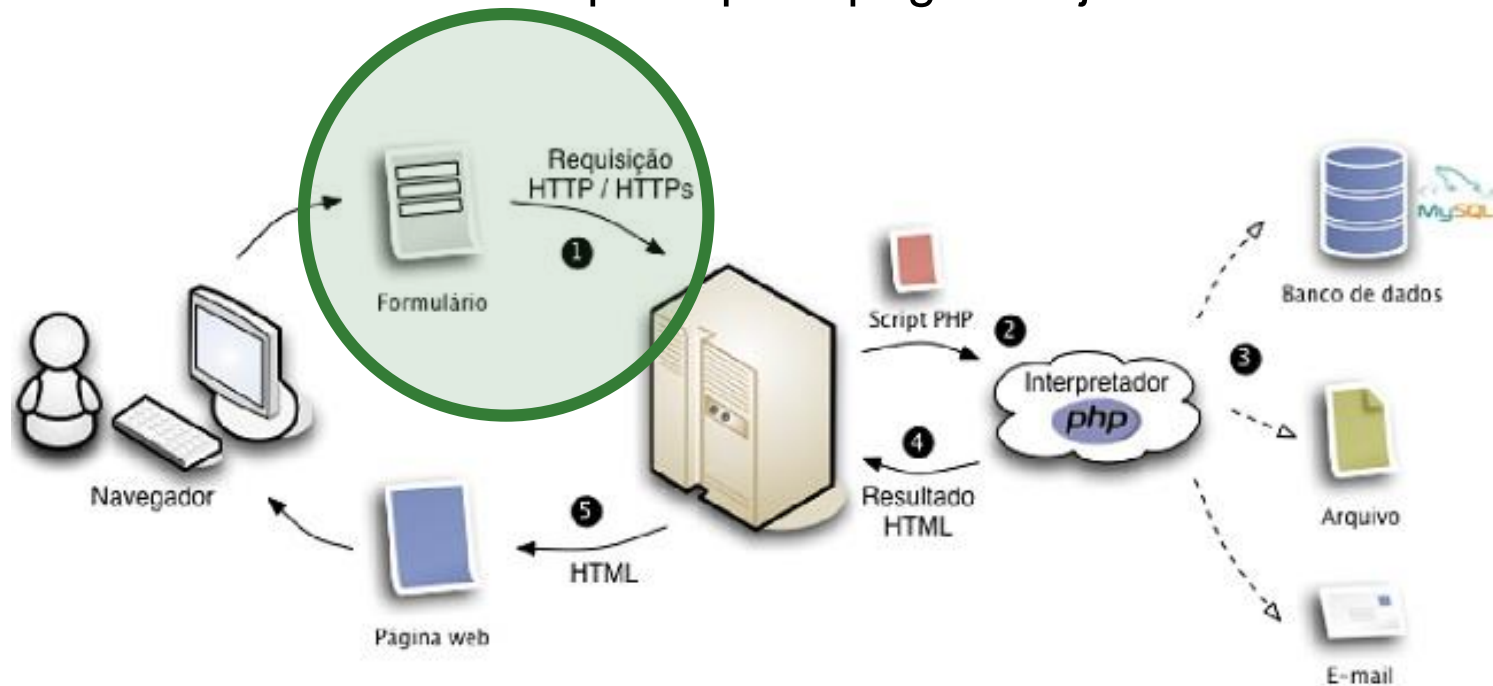
- O usuário acessa um site ou aplicativo web qualquer desenvolvido em PHP através do browser.



A linguagem PHP

Funcionamento do servidor PHP

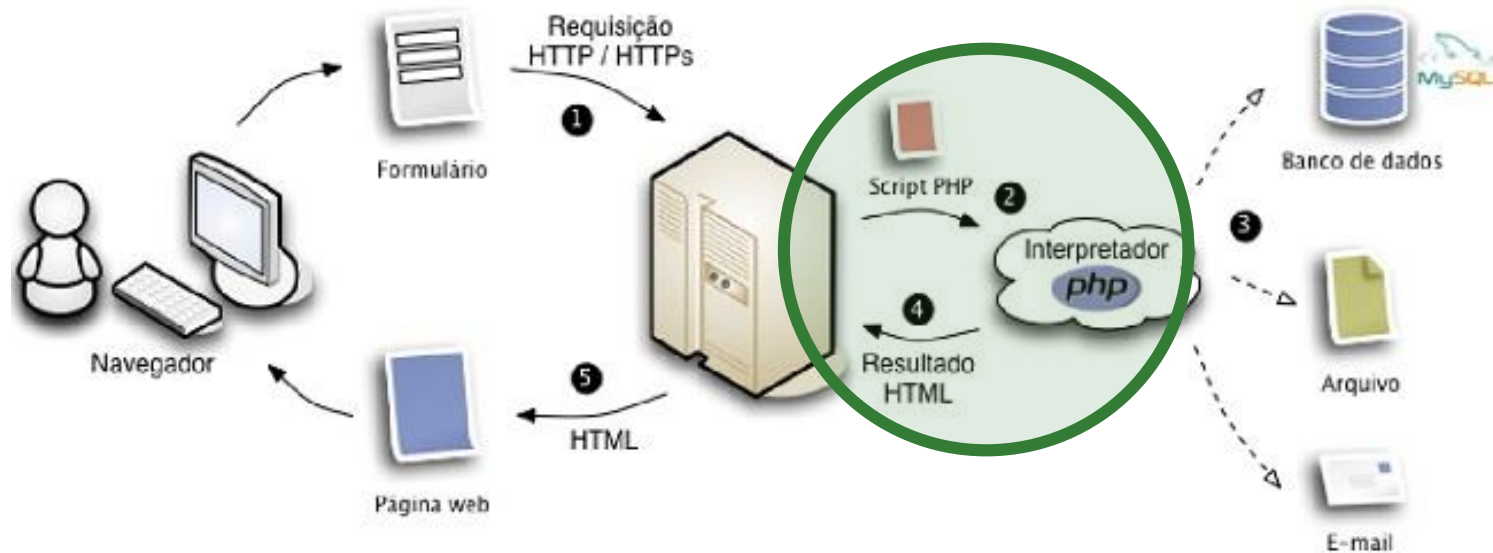
- Ao digitar o endereço, uma requisição HTTP é feita pelo browser ao servidor web solicitando que aquela página seja exibida ao usuário.



A linguagem PHP

Funcionamento do servidor PHP

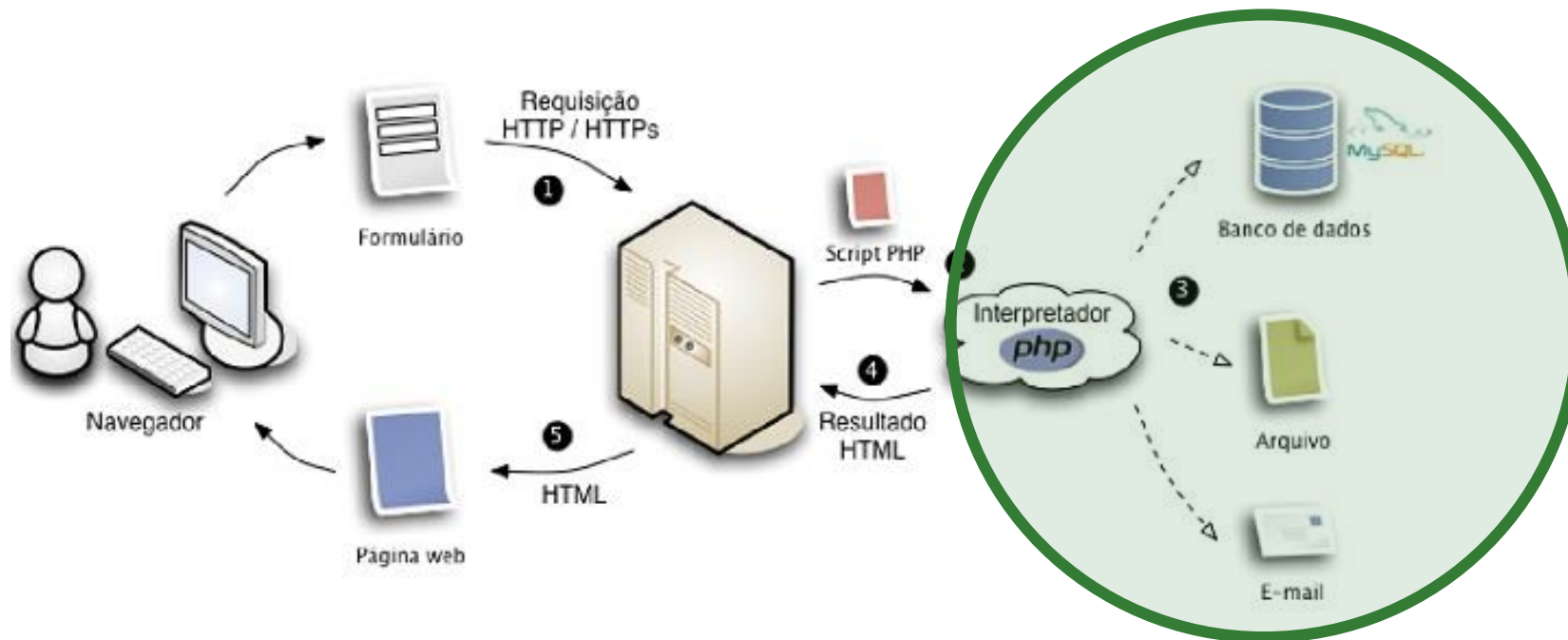
- No servidor está instalado o interpretador PHP. Ele é responsável por “traduzir” o código da página solicitada que está em PHP para HTML.



A linguagem PHP

Funcionamento do servidor PHP

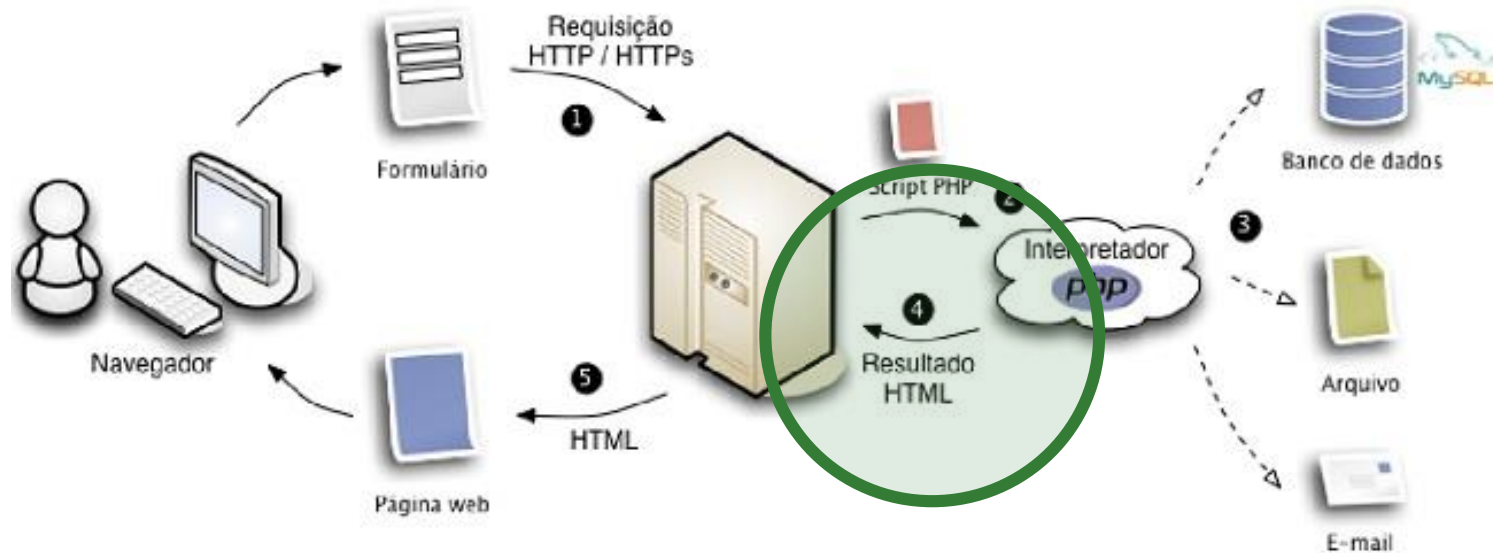
- O servidor PHP pode acessar BDs, ler arquivos, enviar e-mails e efetuar diversas outras tarefas para traduzir o código em HTML.



A linguagem PHP

Funcionamento do servidor PHP

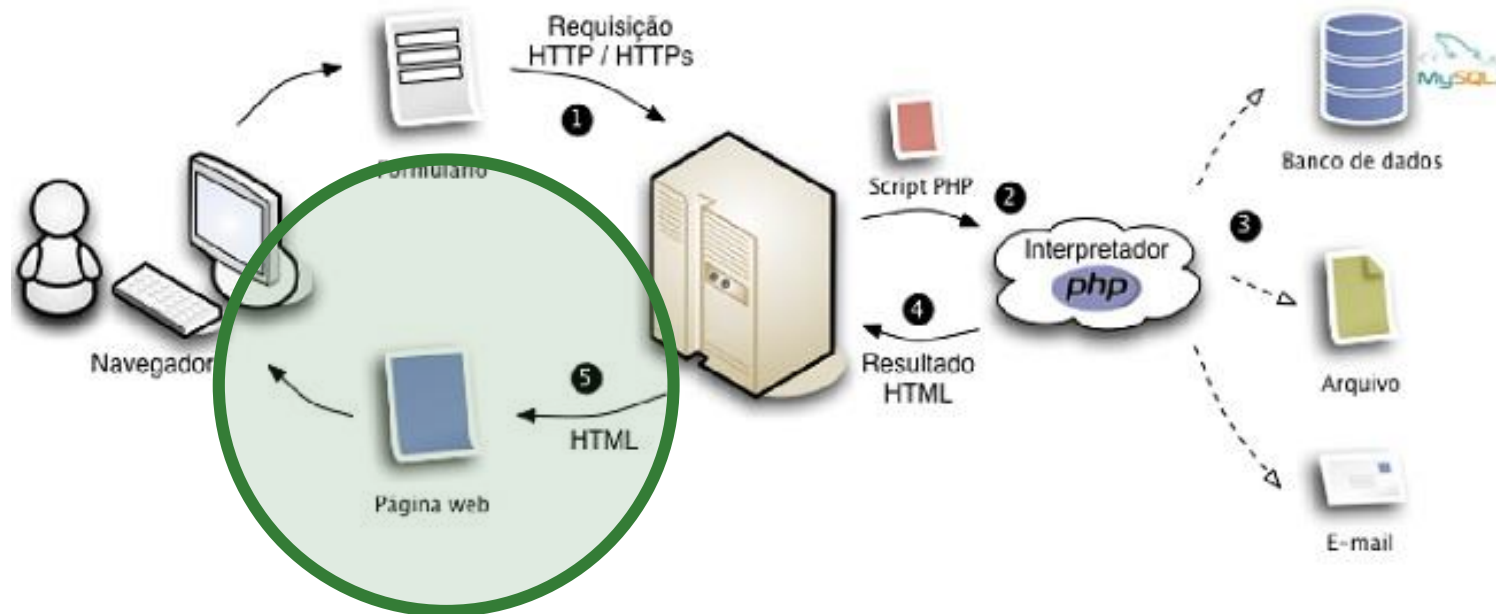
- Após executar todos os procedimentos necessários, o servidor PHP converte o código em HTML.



A linguagem PHP

Funcionamento do servidor PHP

- O servidor web envia o código HTML com as informações geradas para o usuário que o solicitou.



A linguagem PHP

Primeiros passos

O que vamos utilizar

WAMP



`\wamp64\www\pagina.php`

ou

XAMPP



`\xampp\htdocs\pagina.php`

A linguagem PHP

Primeiros passos

Testando os arquivos no servidor

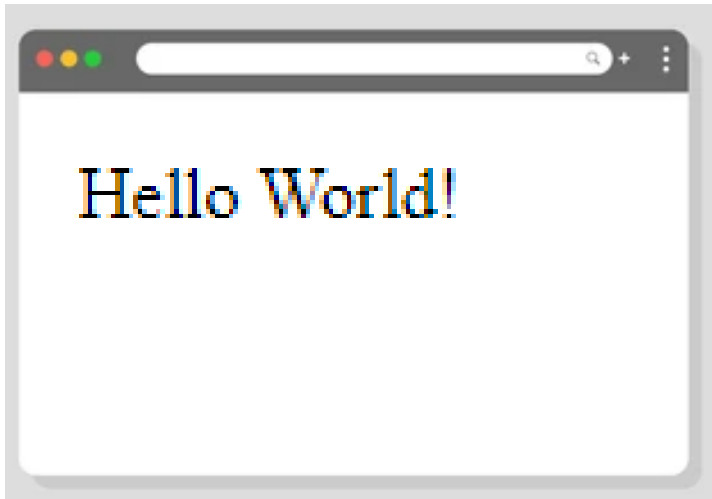


Acessando o servidor local

- **localhost/arquivo.php**
- **127.0.0.1/arquivo.php**

A linguagem PHP

```
<?php
    // meu primeiro programa php
    echo "Hello World!";
?>
```



O PHP imprime informações em um **documento**, que será acessado pelo navegador.

A linguagem PHP

Estrutura e sintaxe

Delimitando o código

```
<?php
```

```
    echo "Olá Mundo!";
```

```
?>
```

```
<script language="PHP">
```

```
    echo "Olá Mundo!";
```

```
</script>
```

```
<?
```

```
    echo "Olá Mundo!";
```

```
?>
```

```
<%
```

```
    echo "Olá Mundo!";
```

```
%>
```

OBS: Similar à sintaxe de ASP. Necessário habilitá-lo no **php.ini**.

Linguagem PHP

Podemos imprimir tags html dentro do php

```
<?php
    echo "<h1>Título</h1>";
    echo "<p>Parágrafo</p>";
    echo '<a href="#">Link</a>';
?>
```



E até mesmo misturar com conteúdo HTML

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <h1>Título</h1>
    <p>Parágrafo</p>
    <?php
        echo "<p>Outro parágrafo</p>";
        echo "<a href='#>link</a>";
    ?>
    <a href='#>Outro link</a>;
</body>
</html>
```



Linguagem PHP

Variáveis em PHP

- Não possui tipo! (dinamicamente tipada)
- Nome deve ser precedido pelo \$
- Atribuição com operador =

```
<?php
    $nome = "João";
    $idade = 25;
    echo "<p>$nome possui $idade anos</p>";
?>
```

A linguagem PHP

Estrutura e sintaxe

Comentários

- Comentário de única linha (`//` e `#`).

```
<?php

    echo "Olá Pessoal!"; // Isso é um comentário!

    # A linha abaixo imprime uma quebra de linha
    echo "<br>";

?>
```

A linguagem PHP

Estrutura e sintaxe

Comentários

- Comentário de várias linhas (`/* ... */`).

```
<?php

    echo "Olá Pessoal!";
    /*
     Este é um comentário com mais de uma linha
     echo "<br>"; Este comando é ignorado!
    */

?>
```

A linguagem PHP

Variáveis

Regras básicas

```
<?php

$var = "Olá mundo";
$var2 = 50;
$outra_var = 4.1;
$varNova = "Fulano de Tal";

?>
```

- **OBS1:** Não esqueça de colocar o \$ no início do identificador de variável.
- **OBS2:** O PHP é *case sensitive*, por isso, \$nome, \$Nome, \$NOME e \$NoMe são variáveis diferentes

A linguagem PHP

Expressões e operadores

Operador de atribuição

- O operador básico de atribuição é o sinal de **igualdade** “=”;

```
<?php  
  
    $a = 4;  
    $b = "101 Dálmatas";  
  
?>
```

A linguagem PHP

Expressões e operadores

Operador de atribuição

- O operador básico de atribuição é o sinal de **igualdade** “=”;

Operador	Função	Exemplo	Equivalente a
<code>+=</code>	Atribuição e adição	<code>\$a += \$b</code>	<code>\$a = \$a + \$b</code>
<code>-=</code>	Atribuição e Subtração	<code>\$a -= \$b</code>	<code>\$a = \$a - \$b</code>
<code>*=</code>	Atribuição e Multiplicação	<code>\$a *= \$b</code>	<code>\$a = \$a * \$b</code>
<code>/=</code>	Atribuição e Divisão	<code>\$a /= \$b</code>	<code>\$a = \$a / \$b</code>
<code>%=</code>	Atribuição e Módulo	<code>\$a %= \$b</code>	<code>\$a = \$a % \$b</code>
<code>.=</code>	Atribuição e concatenação	<code>\$a .= \$b</code>	<code>\$a = \$a . \$b</code>

A linguagem PHP

Expressões e operadores

Operador de concatenação

- O operador básico de concatenação é o **ponto** “.”;

```
<?php

$a = "Bom ";
$b = "dia!";
$c = $a . $b;

echo $c; // Imprime Bom dia!

?>
```

A linguagem PHP

Expressões e operadores

Aspas 'simples' x "duplas"

- O que será impresso em cada uma das linhas a seguir?

```
<?php  
  
    $nome = "Fulaninho";  
  
    echo "Meu nome é $nome."  
  
    echo "Meu nome é " . $nome . "."  
  
    echo 'Meu nome é $nome.'  
?>
```


A linguagem PHP

Expressões e operadores

Operadores aritméticos

Exemplo	Nome	Resultado
$-\$a$	Negação	Oposto de $\$a$.
$\$a + \b	Adição	Soma de $\$a$ e $\$b$.
$\$a - \b	Subtração	Diferença entre $\$a$ e $\$b$.
$\$a * \b	Multiplicação	Produto de $\$a$ e $\$b$.
$\$a / \b	Divisão	Quociente de $\$a$ por $\$b$.
$\$a \% \b	Módulo	Resto de $\$a$ dividido por $\$b$.

A linguagem PHP

Expressões e operadores

Operadores aritméticos

```
$r = $a + $b; // adição  
$r = $a - $b; // subtração  
$r = $a * $b; // multiplicação  
$r = $a / $b; // divisão  
$r = $a % $b; // resto da divisão
```

```
$a = 10;  
$b = 5;  
  
$r = $a + $b; // $r = 10 + 5 = 15  
$r = $a - $b; // $r = 10 - 5 = 5  
$r = $a * $b; // $r = 10 * 5 = 50  
$r = $a / $b; // $r = 10 / 5 = 2  
$r = $a % $b; // $r = 10 % 5 = 0
```

A linguagem PHP

Expressões e operadores

Operadores de incremento/decremento

Operador	Nome	Função
<code>++\$a</code>	Pré-incremento	Incrementa \$a em um, e então retorna \$a.
<code>\$a++</code>	Pós-incremento	Retorna \$a, e então incrementa \$a em um.
<code>--\$a</code>	Pré-decremento	Decrementa \$a em um, e então retorna \$a.
<code>\$a--</code>	Pós-decremento	Retorna \$a, e então decrementa \$a em um.
<code>-\$a</code>	Inverter o sinal	Inverte o sinal de \$a e retorna \$a.

Linguagem PHP

Operadores

Operações com acumuladores

```
$cont = 0;  
  
$cont++; // $cont = $cont + 1  
  
$cont += 5; // $cont = $cont + 5  
  
$cont -= 5; // $cont = $cont - 5  
  
$cont *= 5; // $cont = $cont * 5  
  
$cont /= 5; // $cont = $cont / 5
```

Concatenação de strings

```
$str = "Oi.";   
  
$str = $str . " Olá.";   
  
$str .= " Tudo bem?";   
  
echo $str; // "Oi. Olá. Tudo bem?"
```

A linguagem PHP

Expressões e operadores

Operadores de comparação

Exemplo	Nome	Resultado
<code>\$a == \$b</code>	Igual	True se \$a é igual a \$b.
<code>\$a === \$b</code>	Idêntico	True se \$a é igual a \$b e são do mesmo tipo.
<code>\$a != \$b</code>	Diferente	True se \$a não é igual a \$b.
<code>\$a <> \$b</code>	Diferente	True se \$a não é igual a \$b.
<code>\$a !== \$b</code>	Não idêntico	True se \$a não é igual a \$b e/ou não são do mesmo tipo.
<code>\$a < \$b</code>	Menor que	True se \$a é menor que \$b.
<code>\$a > \$b</code>	Maior que	True se \$a é maior que \$b.
<code>\$a <= \$b</code>	Menor ou igual	True se \$a é menor ou igual a \$b.
<code>\$a >= \$b</code>	Maior ou igual	True se \$a é maior ou igual a \$b.

A linguagem PHP

Operadores de comparação

```
$a == $b // igual  
$a === $b // mesmo valor e tipo  
$a != $b // diferente  
$a !== $b // tipo ou valor diferente  
$a < $b // menor  
$a <= $b // menor ou igual  
$a > $b // maior  
$a >= $b // maior ou igual
```

Operadores lógicos

```
$a && $b // TRUE se $a E $b = TRUE  
$a || $b // TRUE se $a OU $b = TRUE  
!$a // TRUE se $a = FALSE
```

A linguagem PHP

Expressões e operadores

Operadores lógicos

Exemplo	Nome	Resultado
<code>\$a and \$b</code>	E	True se \$a e \$b são true.
<code>\$a or \$b</code>	OU	True se \$a ou \$b são true.
<code>\$a xor \$b</code>	XOR	True se \$a ou \$b forem true, mas não os dois.
<code>!\$a</code>	NÃO	True se \$a é false.
<code>\$a && \$b</code>	E	True se \$a e \$b são true.
<code>\$a \$b</code>	OU	True se \$a ou \$b são true.

Links Úteis

- **Página oficial do PHP:**
- <https://www.php.net/>
- **WAMP:**
- <https://sourceforge.net/projects/wampserver/>
- **XAMPP:**
- https://www.apachefriends.org/pt_br/index.html