

CURSO DE



PSRs - Padrão de Desenvolvimento

Prof. Júnior Gonçalves



PSR's - Padrão de Desenvolvimento



VAMOS APRENDER:

- Cronologia do PHP
- PHP: The Right Way
- PHP Framework Interop Group
- PSR-1
- PSR-12
- Exercícios

> Introdução PHP



PHP é uma linguagem interpretada de código aberto que atua do lado servidor, e tem seu melhor uso no desenvolvimento de aplicações web dinâmicas.

Ao contrário das linguagens compiladas, onde o código é traduzido antecipadamente em código de máquina, o PHP opera em tempo de execução, lendo e interpretando o código-fonte linha por linha.

> Cronologia

1994/1995 - PHP 1 e PHP 2

Criado por Rasmus Lerdorf para ser usado no desenvolvimento de sua página pessoal, era apenas um conjunto de scripts que foi chamado de Personal Home Page

1997/1998 - PHP 3

Andi Gutmans e Zeev Suraski, fundadores da Zend Technologies, entram no time para reescrever a base do PHP e preparar terreno para evoluções futuras.

2000/2003 - PHP 4

Uma das mudanças mais significativas foi a introdução de recursos da Orientação a Objetos.

> Cronologia

2004/2014 - PHP 5

Uma versão totalmente orientada a objetos. Além disso, trouxe recursos avançados como a biblioteca PDO (PHP Data Objects) para banco de dados, suporte nativo para manipulação de JSON e uma série de outras melhorias.

2014/2020 - PHP 7

Lançada com melhorias significativas de desempenho com a inclusão do novo motor de execução Zend Engine 3.

2020/... - PHP 8

Introduziu recursos modernos como JIT (Just-In-Time Compiler) para melhorias de desempenho.

> Variáveis

Tipos de variáveis:

- string
- int
- float
- bool
- array
- object

PHP é uma linguagem **fracamente tipada**. Isso significa que não é preciso declarar explicitamente o tipo de dados de uma variável.

Você pode simplesmente escrever `$valor = 42;` sem informar que `$valor` é do tipo inteiro.

Padrão de desenvolvimento

PHP é uma linguagem extremamente flexível que permite fazer tarefas de diferentes formas. Evoluiu em muitos aspectos, mas não definiu um padrão de desenvolvimento e uma recomendação oficial de codificação. Dessa forma, os frameworks ganharam força no mercado de desenvolvimento:

- Laravel – reconhecido por adotar o padrão MVC e ser extremamente versátil
- Symfony – focado em regras de negócios e mais indicado para aplicações mais robustas



PHP: The Right Way

"PHP: The Right Way" é uma fonte confiável de informações para desenvolvedores PHP, especialmente aqueles que estão começando ou procurando melhorar suas habilidades.

Ele é mantido por membros ativos da comunidade e é atualizado regularmente para refletir as melhores práticas atuais. É uma referência útil para qualquer desenvolvedor PHP que queira escrever código de qualidade e seguro.

<https://br.phptherightway.com/>

PHP Framework Interop Group

A PHP-FIG (**PHP Framework Interop Group**) é um grupo de trabalho na comunidade PHP que se dedica a criar padrões e recomendações (PSRs) para promover a interoperabilidade entre diferentes frameworks e bibliotecas PHP.

Seu principal objetivo é facilitar a colaboração entre projetos PHP, de modo que os desenvolvedores possam usar componentes de diferentes frameworks de forma mais consistente e sem problemas.

<https://www.php-fig.org/psr/>

> PSR-1: Basic Coding Standard

- Arquivos: Cada arquivo DEVE declarar apenas uma classe e seu nome DEVE ser igual ao nome do arquivo
- Classes: Os nomes de classes DEVEM ser escritos em **StudlyCaps** (com a primeira letra de cada palavra em maiúscula)
Exemplo: `MinhaClasse`
- Métodos: Os nomes de métodos DEVEM ser escritos em **camelCase** (com a primeira letra minúscula)
Exemplo: `minhaClasse`

> PSR-1: Basic Coding Standard

- Constantes: Os nomes de constantes DEVEM ser escritos em letras maiúsculas separadas por sublinhados
Exemplo: MINHA_CONSTANTE
- Propriedades: Os nomes das propriedades PODEM ser declarados em **StudlyCaps**, **camelCase** ou **underscore**, mas DEVE ser adotado um único padrão em todo o projeto

> PSR-12: Extended Coding Style

- A tag de fechamento “ ?> ” DEVE ser omitida em arquivos contendo apenas PHP
- Linhas em branco PODEM ser adicionadas para melhorar a legibilidade e para indicar blocos de código relacionados, exceto onde for explicitamente proibido
- A forma abreviada de palavras-chave de tipo DEVE ser usada, ou seja, bool em vez de boolean, int em vez de integer, etc
- NÃO DEVE haver mais de uma instrução por linha

> PSR-12: Extended Coding Style

- Ao instanciar uma nova classe os parênteses DEVEM estar sempre presentes, mesmo quando não há argumentos passados ao construtor
- A visibilidade DEVE ser declarada em todas as propriedades e métodos
- Na lista de argumentos NÃO DEVE haver um espaço antes de cada vírgula e DEVE haver um espaço após cada vírgula

Exemplo de codificação PSR

🐘 UsuarioAdmin.php

```
1 <?php
2
3 class UsuarioAdmin
4 {
5     const BDNAME = "sis-escolar";
6
7     public $NomeUsuario;      // Apenas StudlyCaps
8     public $nomeUsuario;     // Apenas camelCase
9     public $nome_usuario;    // Apenas underscore
10
11    public function autenticarUsuario($arg1, $arg2)
12    {
13        return $this->nomeUsuario;
14    }
15 }
16
```

> Recomendação PSR: if...else

```
<?php  
  
if ($expr1) {  
    // if body  
} elseif ($expr2) {  
    // elseif body  
} else {  
    // else body;  
}
```

5.1 if, elseif, else

An `if` structure looks like the following. Note the placement of parentheses, spaces, and braces; and that `else` and `elseif` are on the same line as the closing brace from the earlier body.

index.php

```
3 $idade = 30;
4 if (isset($idade)){
5     if ($idade >= 0 && $idade < 18) {
6         if ($idade < 7) {
7             echo "Infantil";
8         } else {
9             echo "Adolescente";
10        }
11    } else {
12        echo "Adulto";
13    }
14 } else {
15     echo "Erro no carregamento";
16 }
```

> Recomendação PSR: switch

```
switch ($expr) {  
    case 0:  
        echo 'First case, with a break';  
        break;  
    case 1:  
        echo 'Second case, which falls through';  
        // no break  
    case 2:  
    case 3:  
    case 4:  
        echo 'Third case, return instead of break';  
        return;  
    default:  
        echo 'Default case';  
        break;  
}
```

5.2 switch , case

A `switch` structure looks like the following. Note the placement of parentheses, spaces, and braces. The `case` statement MUST be indented once from `switch`, and the `break` keyword (or other terminating keywords) MUST be indented at the same level as the `case` body. There MUST be a comment such as `// no break` when fall-through is intentional in a non-empty `case` body.

O comando `switch` é uma estrutura que simula uma bateria de testes sobre uma variável.

index.php

```
3 $pagamento = "credito";
4 switch ($pagamento) {
5     case "boleto":
6         echo "Boleto";
7         break;
8     case "debito":
9         echo "Cartão de débito";
10    break;
11    case "credito":
12        echo "Cartão de crédito";
13        break;
14    default:
15        echo "Erro ao processar";
16 }
```

➤ Recomendação PSR: while e do...while

```
while ($expr) {  
    // structure body  
}
```

```
do {  
    // structure body;  
} while ($expr);
```

5.3 while, do while

A `while` statement looks like the following. Note the placement of parentheses, spaces, and braces.

```
3 $cont = 1;  
4 do {  
5     echo "Contagem: $cont <br>";  
6     $cont++;  
7 } while ($cont <= 5);
```

➤ Recomendação PSR: for e foreach

5.4 for

A `for` statement looks like the following. Note the placement of parentheses, spaces, and braces.

```
for ($i = 0; $i < 10; $i++) {  
    // for body  
}
```

5.5 foreach

A `foreach` statement looks like the following. Note the placement of parentheses, spaces, and braces.

```
foreach ($iterable as $key => $value) {  
    // foreach body  
}
```

O foreach é uma estrutura de controle que é usada para iterar (percorrer) elementos de um array ou objetos iteráveis, como listas e arrays associativos. Ele facilita a iteração sobre cada elemento dessas estruturas de dados sem a necessidade de controle manual de índices.

> Recomendação PSR: for e foreach

Neste exemplo, o foreach itera sobre o array associativo \$aluno, onde \$chave representa a chave (nome, idade, curso) e \$valor representa o valor associado a essa chave.

```
2 $aluno = array(  
3     "nome" => "João",  
4     "idade" => 25,  
5     "curso" => "Engenharia"  
6 );  
7  
8 foreach ($aluno as $chave => $valor) {  
9     echo "$chave: $valor <br>";  
10 }
```

← → C ⓘ localhost/aulasPHP

nome: João
idade: 25
curso: Engenharia



Exercícios

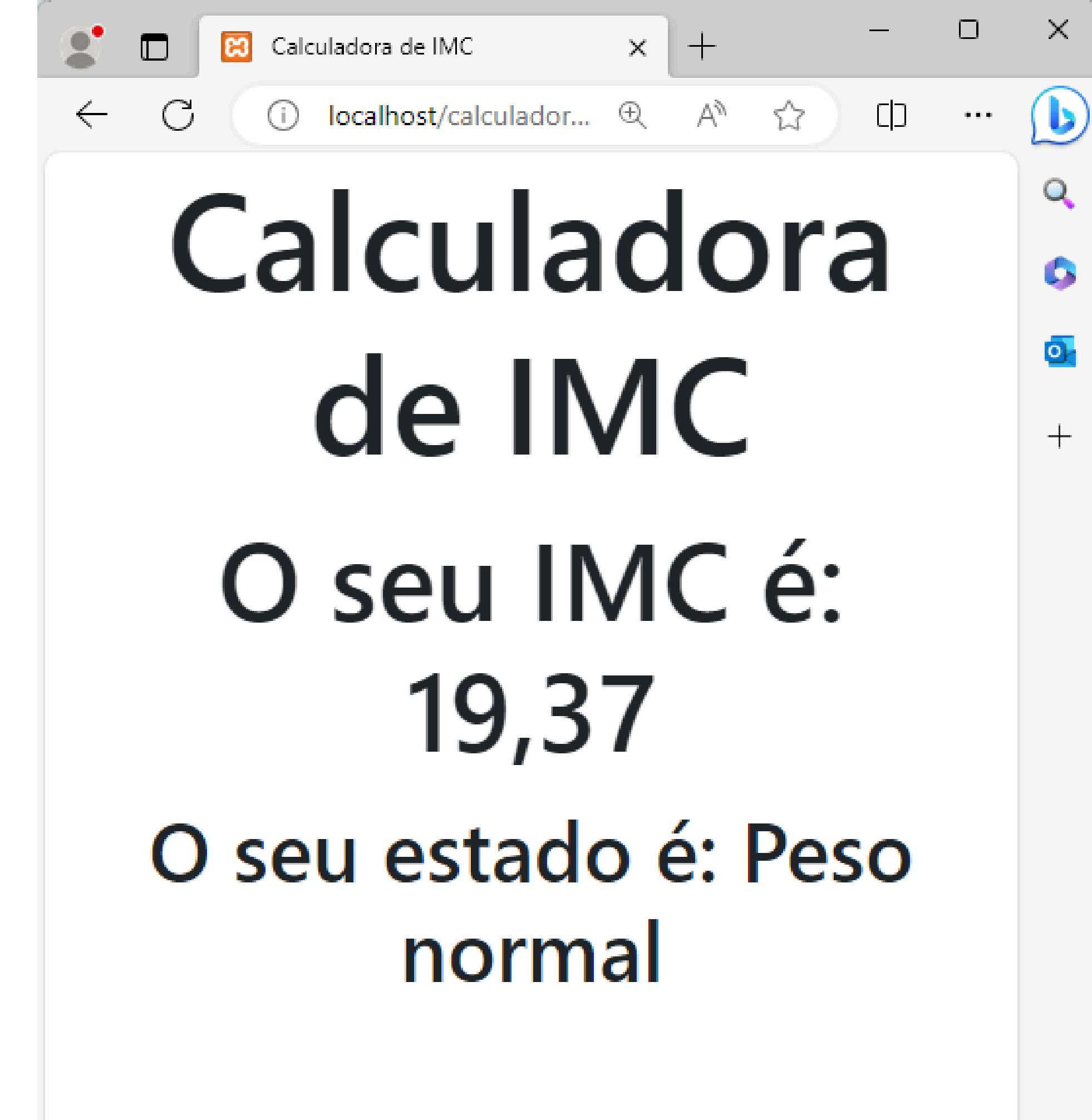
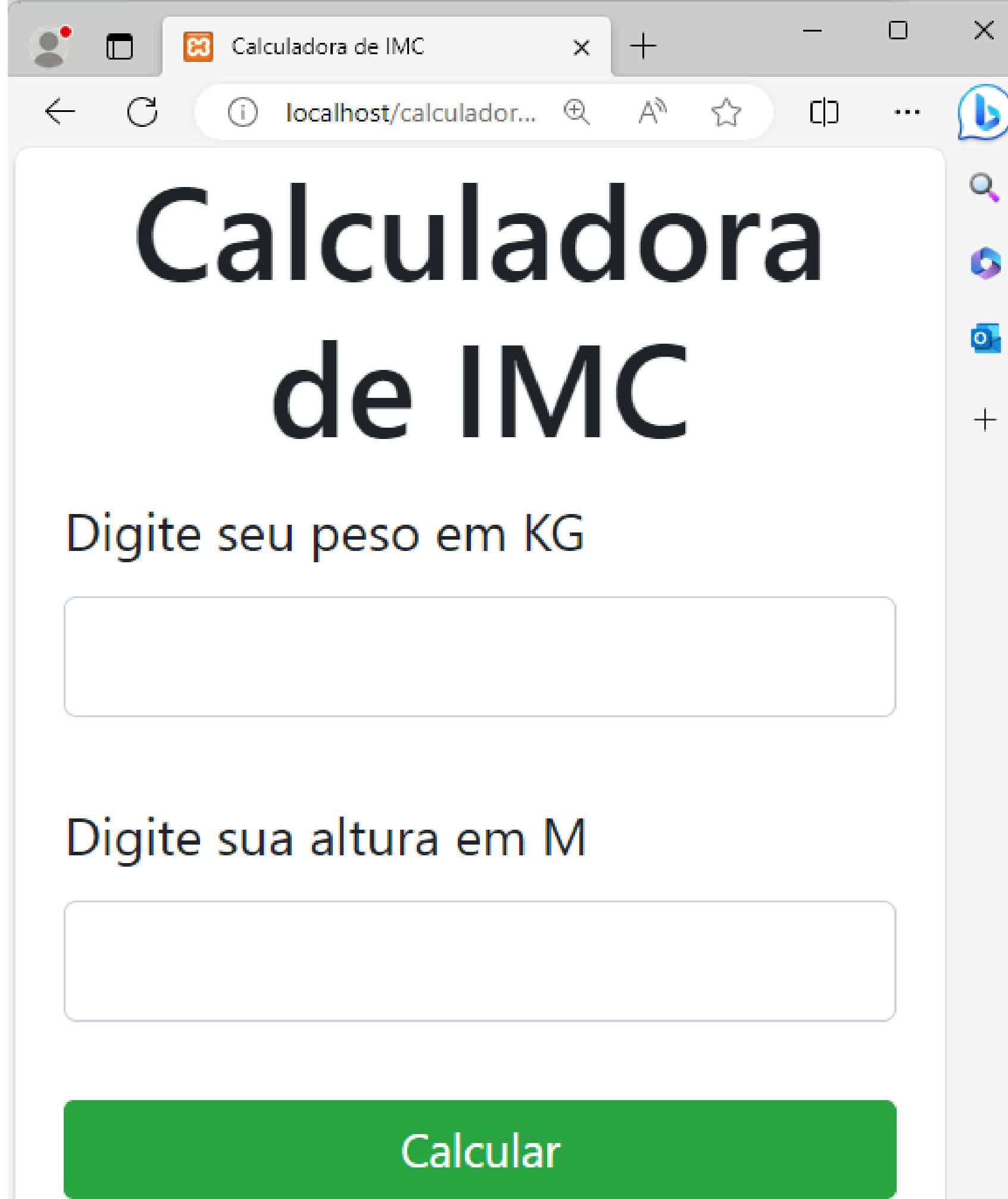
1 - Crie um programa em PHP que calcule o Índice de Massa Corporal (IMC) de uma pessoa com base em sua altura e peso. O IMC é uma medida usada para avaliar se uma pessoa está com peso adequado em relação à sua altura.

- Crie um formulário HTML com campos para inserir o peso (em quilogramas) e a altura (em metros) de uma pessoa.
- Crie um script PHP que receba os valores do peso e da altura do formulário
- Calcule o IMC usando a fórmula $MC = \text{peso} \text{ (em quilogramas)} / (\text{altura em metros}^2)$



Exercícios

- Com base no valor do IMC calculado, exiba uma mensagem que indique a categoria de peso da pessoa de acordo com a tabela abaixo:
 - Menos de 18.5: Abaixo do peso
 - De 18.5 a 24.9: Peso normal
 - De 25 a 29.9: Sobre peso
 - De 30 a 34.9: Obesidade Grau 1
 - De 35 a 39.9: Obesidade Grau 2
 - 40 ou mais: Obesidade Grau 3 (Obesidade Mórbida)
- Exiba o resultado na página, incluindo o valor do IMC calculado e a categoria de peso correspondente.





Exercícios

2 - Crie um programa em PHP que permita a um aluno do ensino superior selecionar sua área de estudo e, com base nessa escolha, calcular e exibir a mensalidade correspondente usando a estrutura switch-case.

- Crie um formulário HTML que permita ao aluno selecionar sua área de estudo a partir de um menu suspenso.
- As opções podem ser carregados a partir de: \$areasEstudo = array("Ciência da Computação", "Engenharia", "Ciências Sociais", "Artes");
- Utilize a estrutura foreach para percorrer o array de áreas de estudo e exibir cada opção em uma lista no formato de HTML



Exercícios

- Crie um script PHP que receba a escolha do aluno do formulário.
- Use a estrutura switch-case para determinar a mensalidade com base na área de estudo escolhida. Defina diferentes valores de mensalidade para cada área de estudo. Por exemplo:
 - Ciência da Computação: R\$ 800,00
 - Engenharia: R\$ 900,00
 - Ciências Sociais: R\$ 750,00
 - Artes: R\$ 700,00
 - Se a área de estudo não estiver listada, exiba uma mensagem de erro.
- Exiba o resultado na página, mostrando a área de estudo escolhida e o valor da mensalidade correspondente ou a mensagem de erro, conforme aplicável.

Áreas de Estudo

localhost/areas_estu...

Áreas de Estudo

Selecione a área de estudo:

Ciência da Computação

Selecionar

Áreas de Estudo

localhost/areas_estu...

Áreas de Estudo

O estudo escolhido é:
Ciência da Computação

O valor da mensalidade é: R\$ 800,00