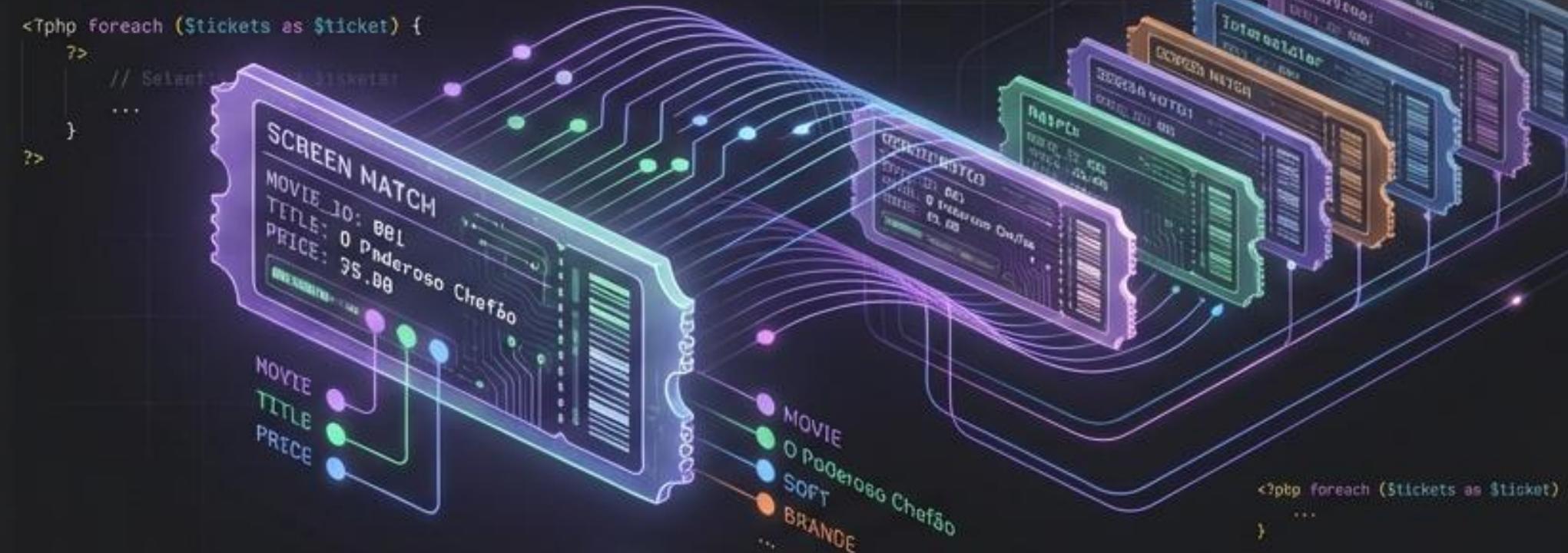


# PHP: Dominando Loops e Coleções

Evoluindo o projeto Screen Match — De dados isolados para estruturas dinâmicas



Curso de Desenvolvimento de Sistemas - 3º Ano

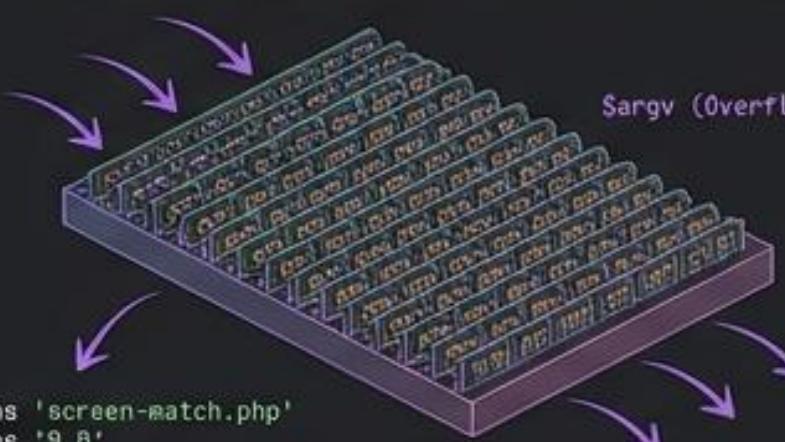
# O Problema da Escala (Contexto Semana 01)

## Código Legado (Fixo)

```
1 <?php  
2  
3     $nota1 = 9.0;  
4     $nota2 = 8.5;  
5     $nota3 = 7.0;  
6  
7 ;
```

## Terminal (Escala Caótica)

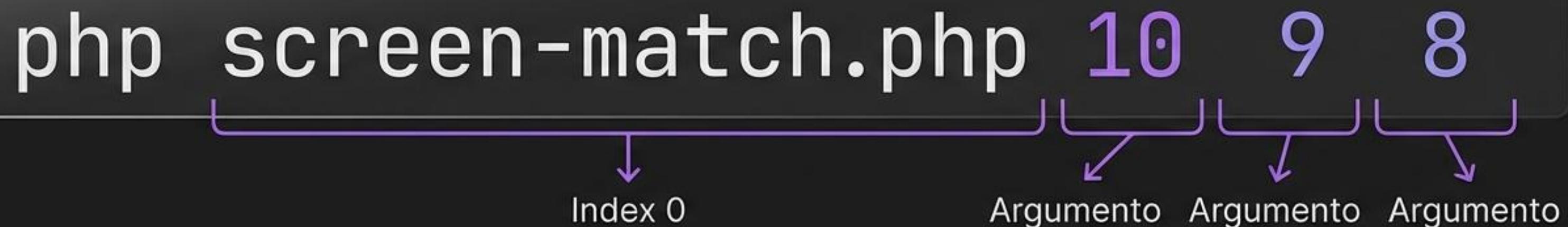
```
>_ php screen-match.php 9.0 8.3 7.0 6.5 9.2 8.8 7.9 9.5 8.1 7.3 9.9 8.4 7.1 6.8  
9.7 8.6 8.6 7.8 9.3 8.0 7.5 9.1 8.7 7.2 9.8 8.3 7.4 9.6 8.2 7.7 9.6 8.9 7.6  
10.0 6.7 9.8 8.5 7.0 6.5 9.2 8.8 7.9 9.5 8.1 7.3 9.9 8.4 7.1 6.8 9.7 8.6...  
  
Index [8] contains 'screen-match.php'  
Index [1] contains '9.0'  
Index [2] contains '8.5'  
Index [3] contains '7.0'  
Index [4] contains '6.5'  
Index [5] contains '9.2'
```



O código atual é rígido. Se o filme receber 100 avaliações, precisaremos criar 100 variáveis? **O objetivo de hoje é tornar o código dinâmico.**

# Conceito - argc

Definindo o Limite da Repetição



\$argc = 4

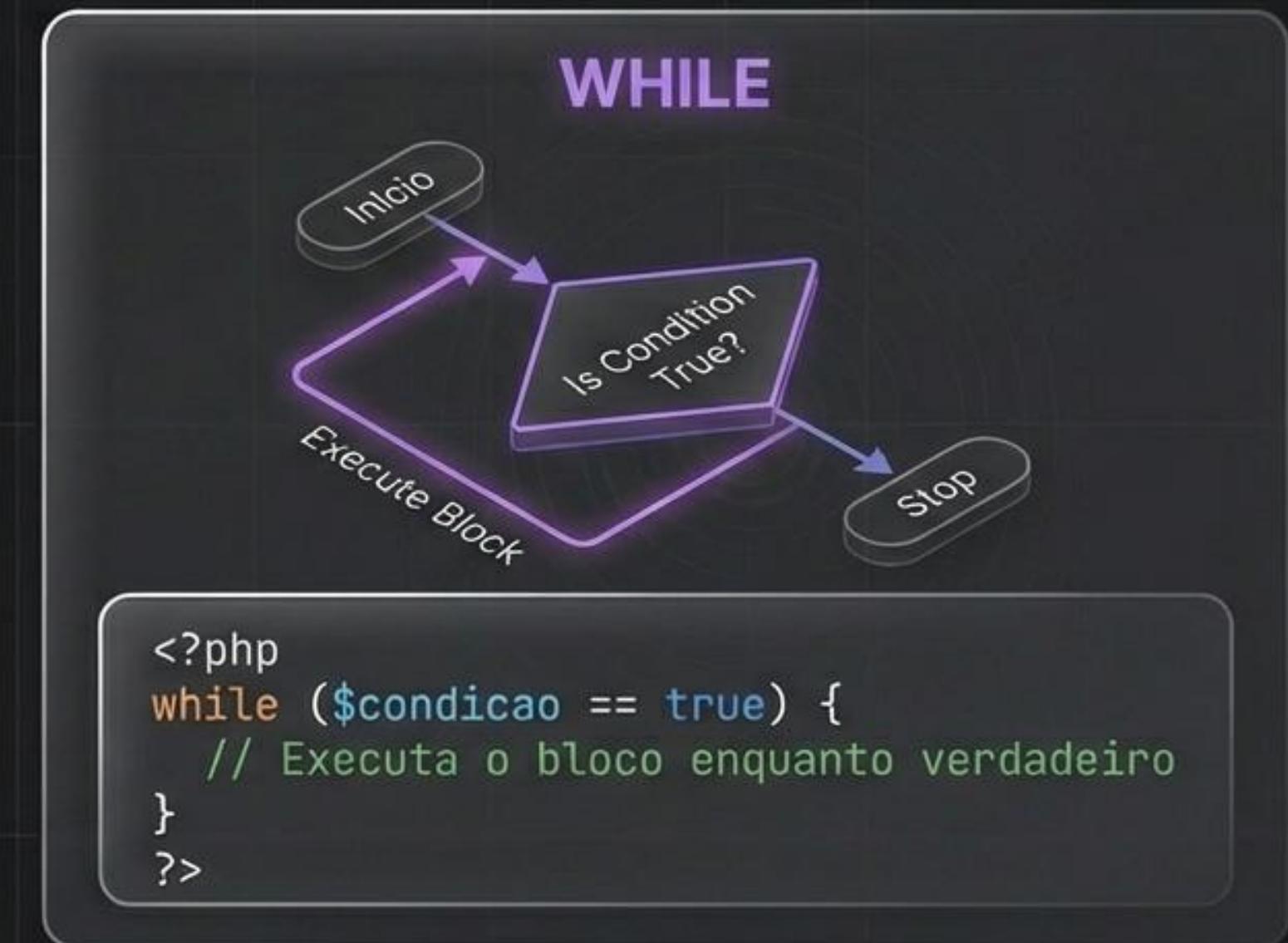
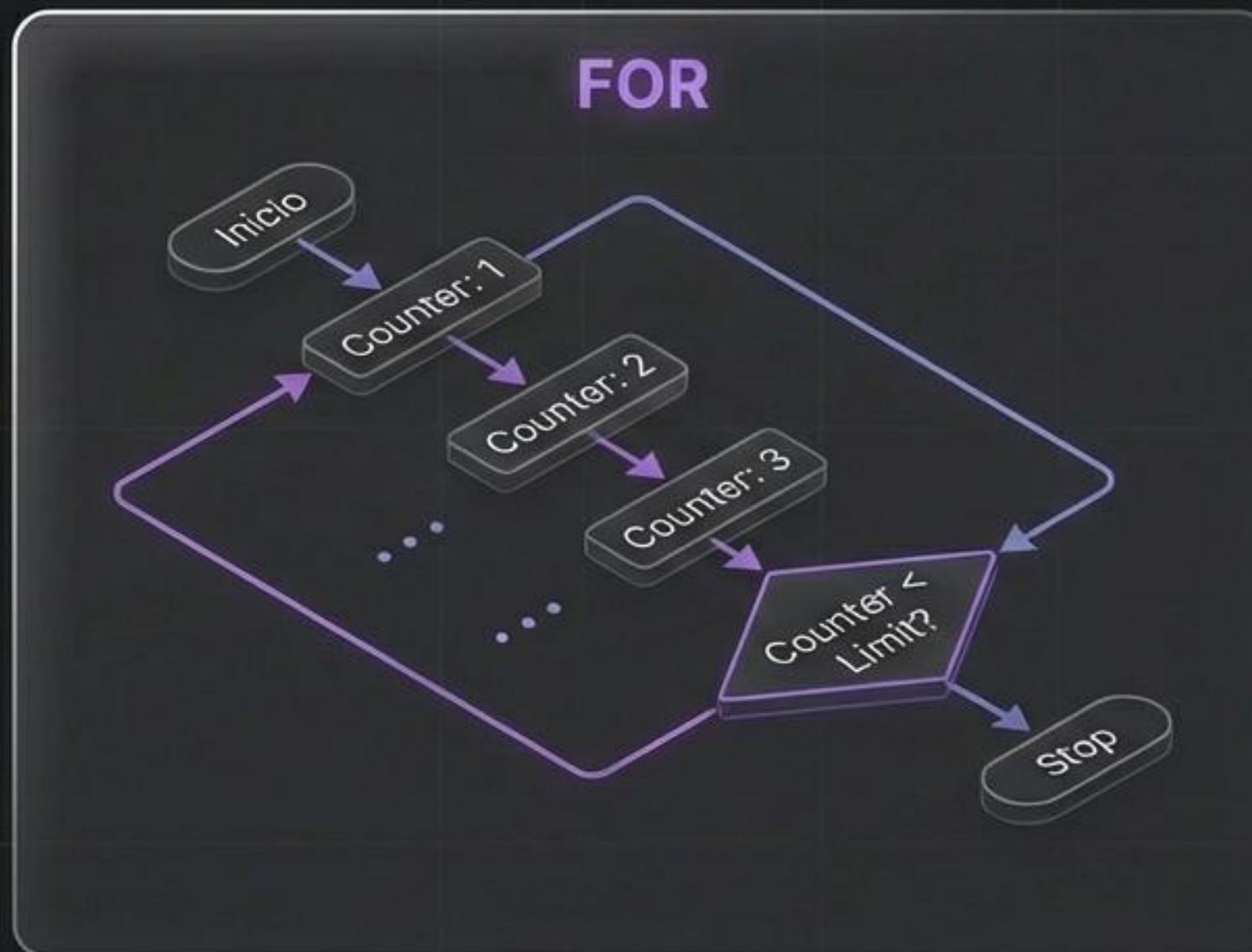
Argument Count

```
var_dump($argc); // int(4)
```

A variável **\$argc** (Argument Count) armazena a quantidade total de argumentos. O nome do script conta como o primeiro item.

# Conceito - while

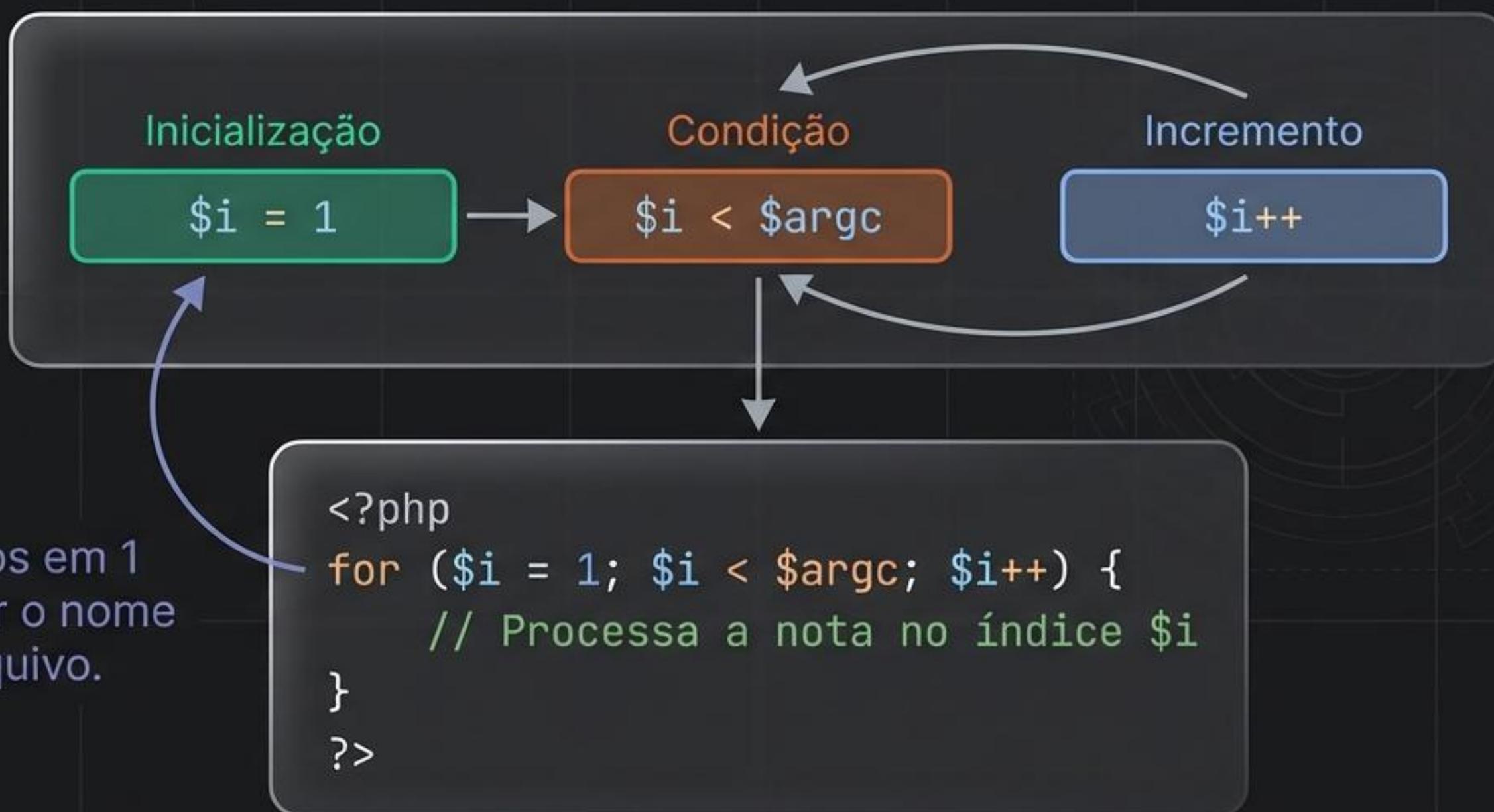
Repetição Baseada em Condição



Ideal para quando não sabemos o número exato de repetições (ex: ler um arquivo até o fim).

# Conceito - for

A Estrutura de Repetição Contada



# Conceito - Array Associativo

Dando Nome aos Índices



```
$filme = [  
    'nome' => 'Top Gun: Maverick',  
    'ano'  => 2022,  
    'nota' => 8.8  
];
```

Substituímos índices numéricos confusos ([0]) por chaves de texto ('nome') para dar significado aos dados.

# Conceito - var\_dump

O Raio-X da Variável

Tipo de dado  
(array)

Índice

```
array(3) {  
    ["nome"]=>  
        string(17) "Top Gun: Maverick"  
    ["ano"]=>  
        int(2022)  
    ["nota"]=>  
        float(8.8)  
}
```



Tamanho

Valor e Tipo

# Conceito - Sintaxe de Array

O Container de Dados



```
// Ruim (Difícil de escalar)
$nota1 = 10;
$nota2 = 8;
```

```
// Bom (Moderno e Dinâmico)
$notas = [10, 8, 9, 7.5];
```

Utilizamos a sintaxe curta [ ] moderna. O índice sempre começa em 0.

# Conceito - Cast & Append

Sanitização e Inserção Dinâmica



```
// Conversão e Inserção automática  
$notas[] = (float) $argv[$i];
```

O PHP converte o tipo (Cast) e gerencia automaticamente o próximo índice vazio do array.

# Conceito - `foreach()`

## A Elegância da Iteração

### Before and After

```
for ($i=0; $i < count($notas); $i++) {  
    $nota = $notas[$i];  
    // Lógica complexa de índices  
}
```

Simplificação

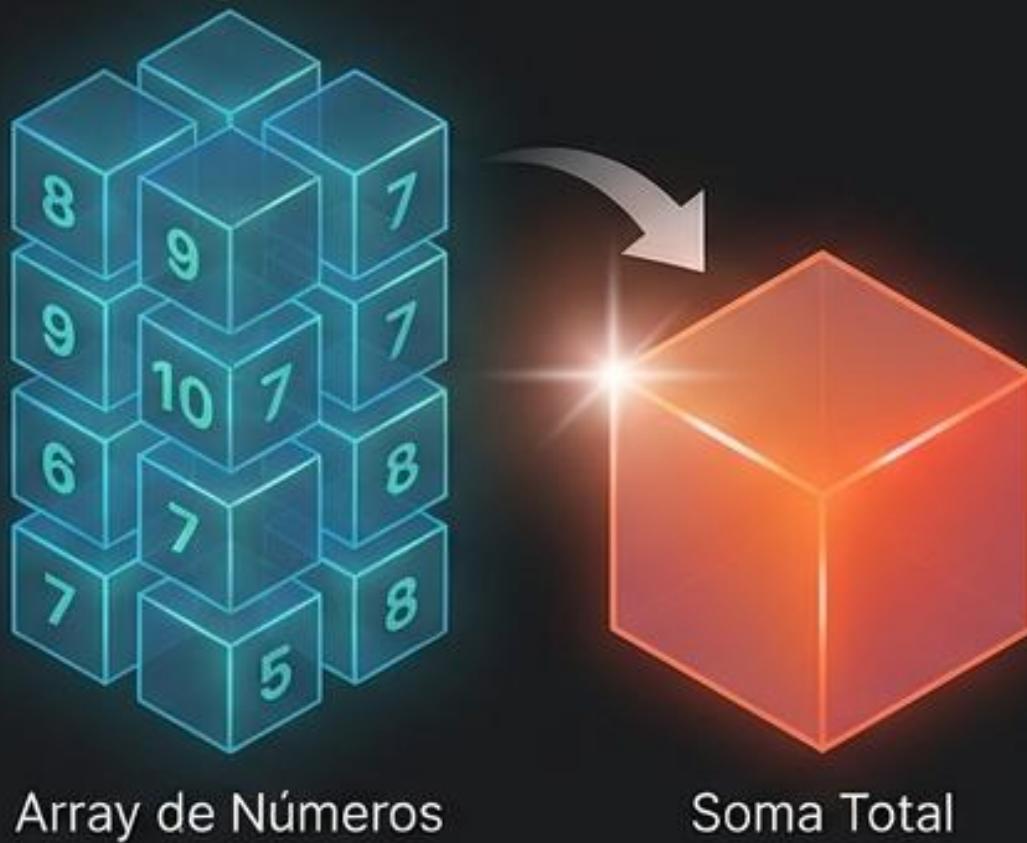
```
foreach ($notas as $nota) {  
    // Foco total no dado ($nota)  
}
```

Eliminamos o gerenciamento manual de contadores.  
"Para cada nota na lista de notas..."



# Conceito - array\_sum

O Poder da Biblioteca Padrão



De/Para

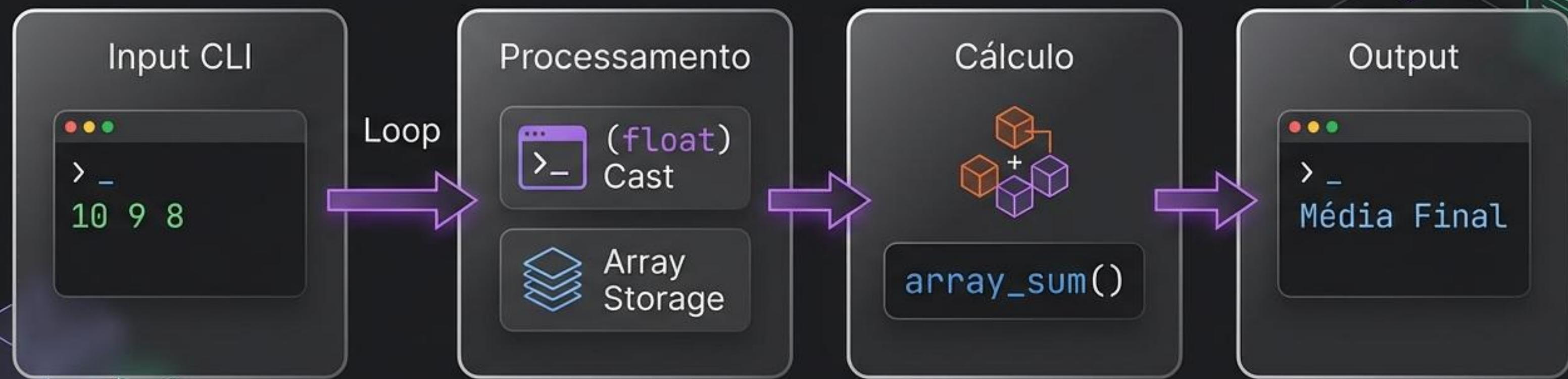
```
// De:  
foreach ($notas as $nota) {  
    $soma += $nota;  
}
```



```
// Para:  
$soma = array_sum($notas);
```

Conhecer as funções nativas separa o código júnior do código sênior.  
Menos linhas, mais legibilidade.

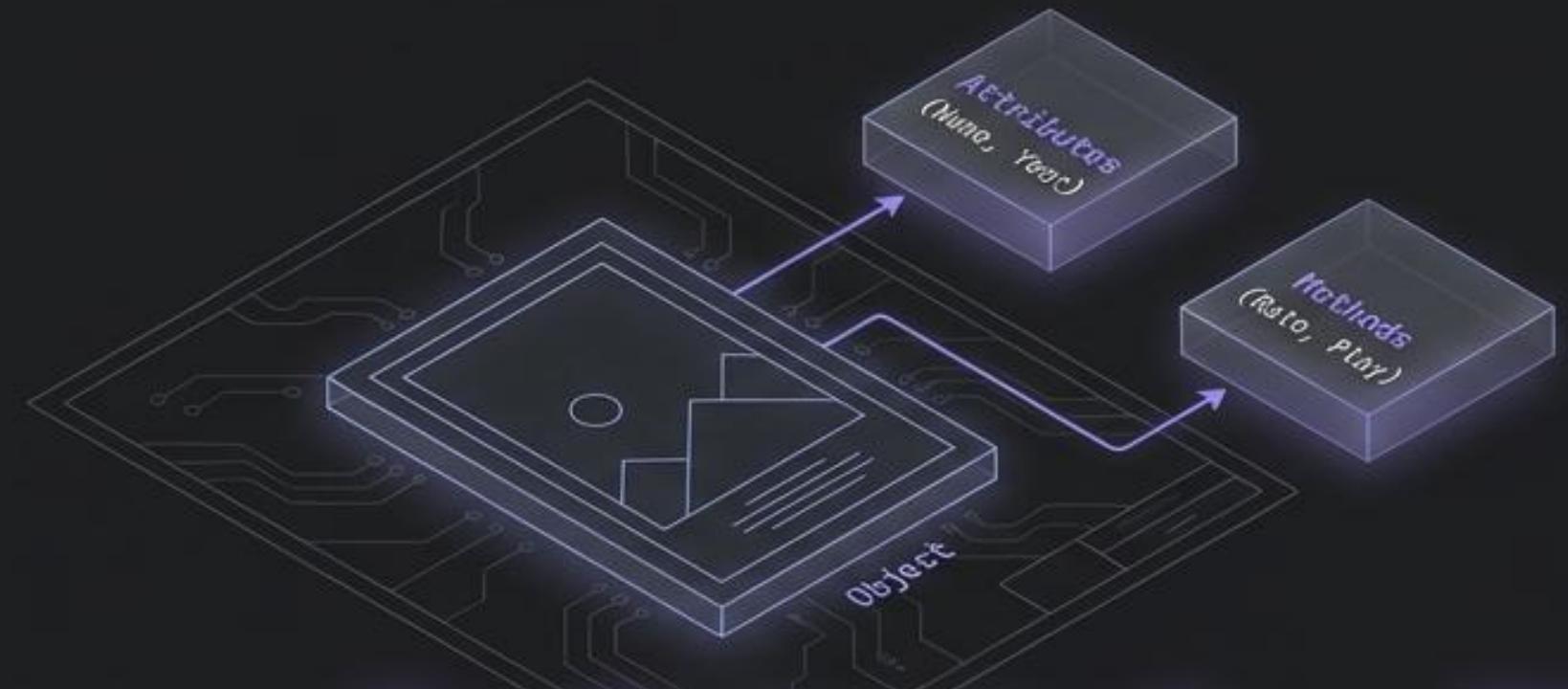
# O Fluxo Completo: Screen Match v2



Saímos de variáveis soltas para uma estrutura capaz de processar milhares de notas com o mesmo código.

# Próximo Nível: Paradigmas

Modelando o Mundo Real



## Programação Orientada a Objetos

- Classes
- Objetos
- Atributos e Métodos

Dominamos a lógica estruturada. Agora vamos estruturar as entidades.