

PROGRAMAÇÃO  
EM NOVAS  
TECNOLOGIAS  
(WEB)

VINICIUS  
MARQUES



# INTRODUÇÃO AO PHP



PHP é uma linguagem de script open-source amplamente utilizada no desenvolvimento web.



Criada por Rasmus Lerdorf em 1994.



Mantida pelo PHP Group e usada em milhões de sites.

## POR QUE USAR PHP?

Facilidade de uso,  
flexibilidade e forte  
comunidade de  
suporte.

Compatível com  
diversos bancos de  
dados, incluindo  
MySQL, PostgreSQL e  
SQLite.

# INSTALAÇÃO DO PHP

Instalação no Windows (usando XAMPP), macOS (usando MAMP) e Linux (usando comandos do terminal).

Configuração básica do servidor web.

# ESTRUTURA DO PHP

Exemplo de  
código PHP  
básico.

Tags de  
abertura e  
fechamento.

Variáveis,  
constantes e  
tipos de dados.

## CÓDIGO BÁSICO PHP

```
<?php
    // Este é um comentário
    echo "Olá, Mundo!"; // Imprime uma mensagem
    $variavel = 10; // Declara uma variável
    const CONSTANTE = 20; // Declara uma constante
    echo $variavel + CONSTANTE; // Soma e imprime
?>
```

## CONDICIONAIS E LOOPS

Exemplos de  
uso de if,  
else, switch.

Loops for,  
while, do-  
while.

# CONDICIONAIS E LOOPS

```
<?php
// Estrutura if-else
$numero = 10;
if ($numero > 5) {
    echo "Maior que 5";
} else {
    echo "Menor ou igual a 5";
}

// Loop for
for ($i = 0; $i < 5; $i++) {
    echo $i;
}

// Loop while
$contagem = 0;
while ($contagem < 5) {
    echo $contagem;
    $contagem++;
}
?>
```



# CRIANDO E USANDO FUNÇÕES

- Como declarar funções em PHP.
- Passando parâmetros e retornando valores.
- Exemplo de uma função simples.

```
<?php
// Função simples
function saudacao($nome) {
    return "Olá, " . $nome . "!";
}

// Chamar a função
echo saudacao("Maria"); // Imprime "Olá, Maria!"
?>
```

# MANIPULAÇÃO DE ARRAYS



CRIAÇÃO E  
MANIPULAÇÃO DE  
ARRAYS.



TIPOS DE ARRAYS  
(ASSOCIATIVOS,  
MULTIDIMENSIONAIS).



FUNÇÕES ÚTEIS  
(ARRAY\_PUSH,  
ARRAY\_MERGE).

# MANIPULAÇÃO DE ARRAYS

```
<?php
// Array simples
$frutas = array("Maçã", "Banana", "Laranja");

// Adicionar elemento ao array
array_push($frutas, "Abacaxi");

// Array associativo
$idade = array("Maria" => 25, "João" => 30);

// Array multidimensional
$peessoas = array(
    array("nome" => "Carlos", "idade" => 28),
    array("nome" => "Ana", "idade" => 22)
);

// Exibir elementos do array
foreach ($frutas as $fruta) {
    echo $fruta . " ";
}

?>
```

# RECEBENDO DADOS DE FORMULÁRIOS



MÉTODOS GET E POST.



CAPTURANDO E PROCESSANDO  
DADOS ENVIADOS POR  
FORMULÁRIOS HTML.

Html

```
<!DOCTYPE html>
<html>
<body>

<h2>Formulário HTML</h2>
<form method="post" action="processa.php">
  Nome: <input type="text" name="nome">
  <input type="submit">
</form>

</body>
</html>
```

## Arquivo 2: processa.php

Php

```
<?php
// Captura o dado enviado pelo formulário
$nome = $_POST['nome'];
echo "Olá, " . $nome;
?>
```

# BOAS PRÁTICAS DE PROGRAMAÇÃO

- Dicas para escrever código limpo e reutilizável.
- Importância de comentários e documentação.
- Uso de controle de versão (Git).