

Programação Web

Unidade 2 – PHP

Prof. Thiago Waltrik

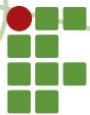
Conhecimentos: Unidade 2

- Arquitetura cliente-servidor
- O que é o PHP e para que é usado?
- Configuração do ambiente (Windows e Linux)
- Introdução a sintaxe básica e ao ambiente
- Variáveis e saída
- Estruturas de Controle (if, else, elseif, switch) e Loops (while, for, foreach)
- Funções
- Operações com formulários (métodos GET e POST)
- Acesso à banco de dados
- Hospedagem de sistemas Web

Objetivos

Aprender o básico da linguagem de programação PHP para que possa prepará-los(as) para a aprendizagem de *frameworks* como o Laravel.





INSTITUTO
FEDERAL
Santa Catarina

Arquitetura Cliente-Servidor



Clients



O servidor hospeda
arquivos: HTML,
CSS, JS, PHP, etc.
O servidor
interpreta o PHP.



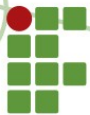
Server



MariaDB

O *browser* recebe
arquivos: HTML,
CSS e JS.
O *browser*
interpreta HTML,
CSS e JS.





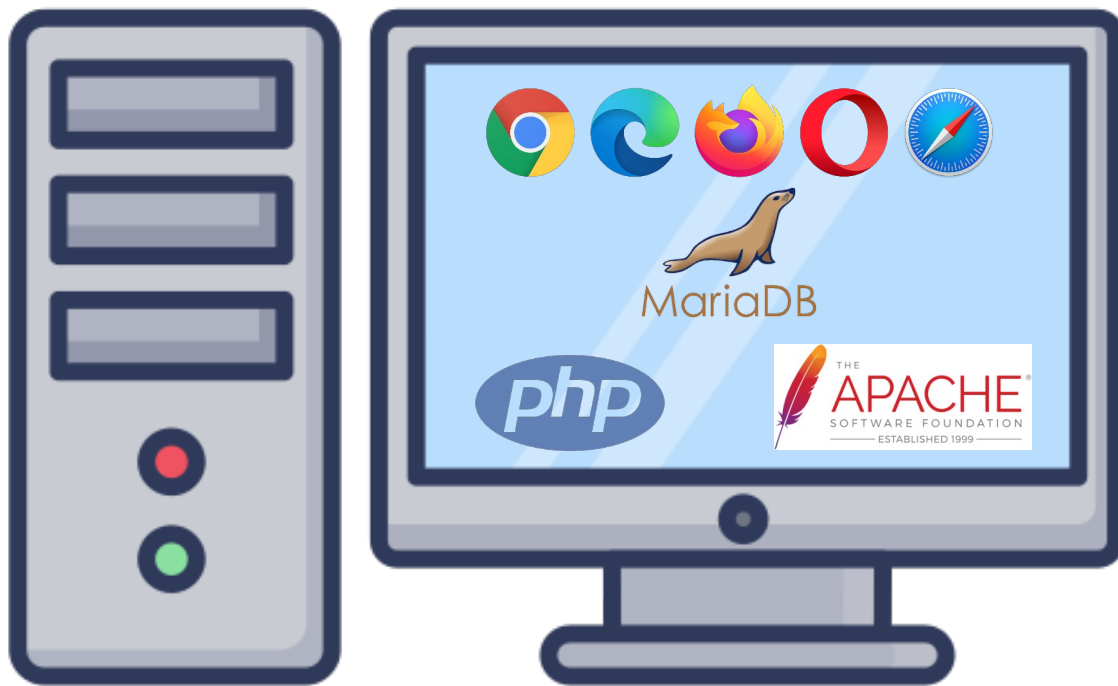
INSTITUTO
FEDERAL
Santa Catarina

Arquitetura Cliente-Servidor

Em um ambiente de desenvolvimento típico...

O servidor hospeda arquivos: HTML, CSS, JS, PHP, etc.
O servidor **interpreta** o PHP.

O *browser* recebe arquivos: HTML, CSS e JS.
O *browser* **interpreta** HTML, CSS e JS.



O que é o PHP e para que é usado?

PHP (Hypertext PreProcessor) é uma linguagem de programação de código aberto, especialmente projetada para desenvolvimento web. Foi criada por Rasmus Lerdorf em 1994, inicialmente como um conjunto de *scripts* para acompanhar o acesso ao seu currículo online, mas evoluiu para se tornar uma das linguagens mais populares para a construção de sites dinâmicos e aplicações web.



INSTITUTO
FEDERAL
Santa Catarina

O que é o PHP e para que é usado?

Quais São as Linguagens de Programação Mais Usadas em 2024? >

1. Python >

2. C# >

3. C++ >

4. JavaScript

5. PHP >

6. Swift >

7. Java >

8. Go >

9. SQL >

10. Ruby >

Fonte: <https://www.hostinger.com.br/tutoriais/linguagens-de-programacao-mais-usadas>
Acesso em: 6/11/2024.

O que é o PHP e para que é usado?

Principais Características do PHP

- **Interpretada pelo Servidor:** PHP é executado no servidor web e gera HTML que é enviado ao navegador do usuário. O usuário final vê apenas o resultado final da execução do script PHP, não o código PHP em si.
- **Integrada com HTML:** PHP é facilmente incorporado em arquivos HTML, permitindo misturar código de servidor com HTML para gerar conteúdo dinâmico.
- **Fácil de Aprender e Usar:** PHP possui uma sintaxe simples, o que facilita o aprendizado e uso para iniciantes em desenvolvimento web.

O que é o PHP e para que é usado?

Principais Características do PHP

- **Código Aberto e Gratuito:** PHP é gratuito e possui uma grande comunidade de desenvolvedores, o que facilita o acesso a recursos e suporte.
- **Suporte a Bancos de Dados:** PHP é altamente compatível com diversos bancos de dados, como MySQL, PostgreSQL, Oracle e muitos outros, o que facilita o desenvolvimento de aplicações que necessitam de armazenamento e gestão de dados.

Configuração do ambiente (Windows)

Para programar em PHP, precisamos de:

- 1) Um editor de textos simples (como o Notepad) ou uma IDE (*Integrated Development Environment* – Ambiente de Desenvolvimento Integrado) como o VsCode;
- 2) Um servidor Web com suporte a PHP.

Configuração do ambiente (Windows)

O servidor Web com suporte a PHP pode ser de vários tipos como:

- WAMP (Windows + Apache + MySQL ou MariaDB + PHP);
- LAMP (Linux + Apache + MySQL ou MariaDB + PHP);
- XAMPP (Windows, Linux ou OS X + Apache + MariaDB + PHP + Perl).



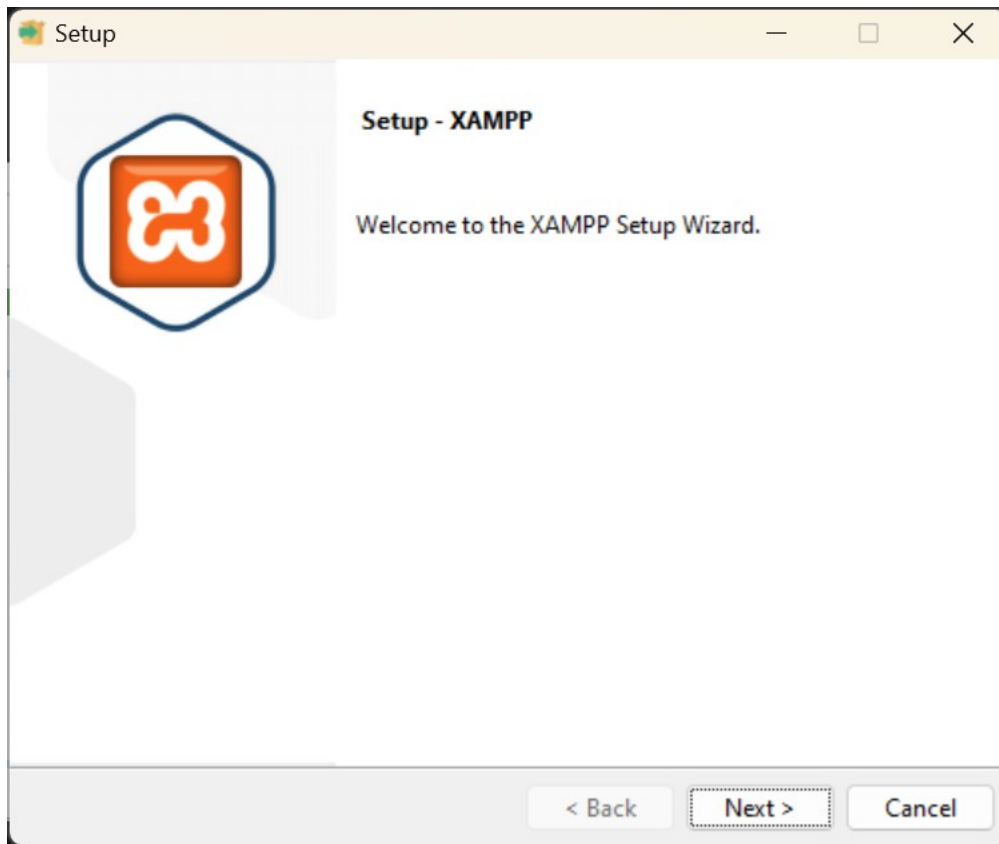
INSTITUTO
FEDERAL
Santa Catarina

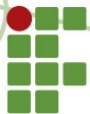
Configuração do ambiente (Windows)

Devido a facilidade de instalação, utilizaremos o XAMPP nas nossas aulas práticas. O XAMPP pode ser baixado em:

https://www.apachefriends.org/pt_br/index.html

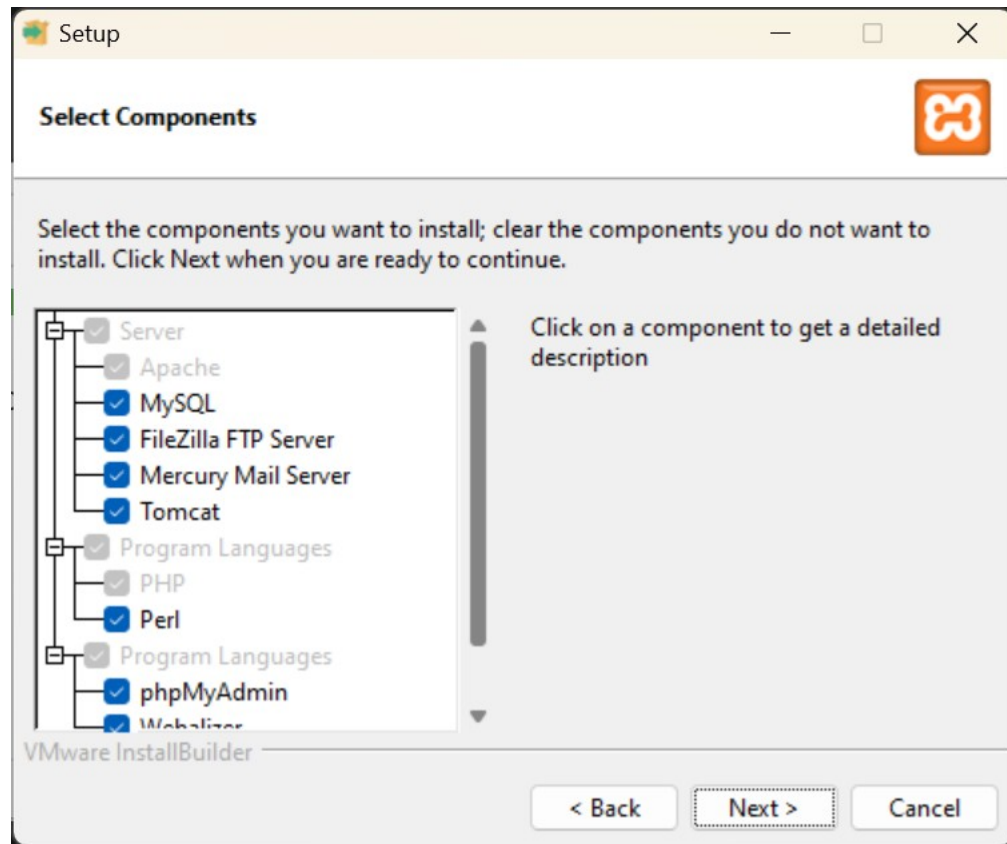
Configuração do ambiente (Windows)



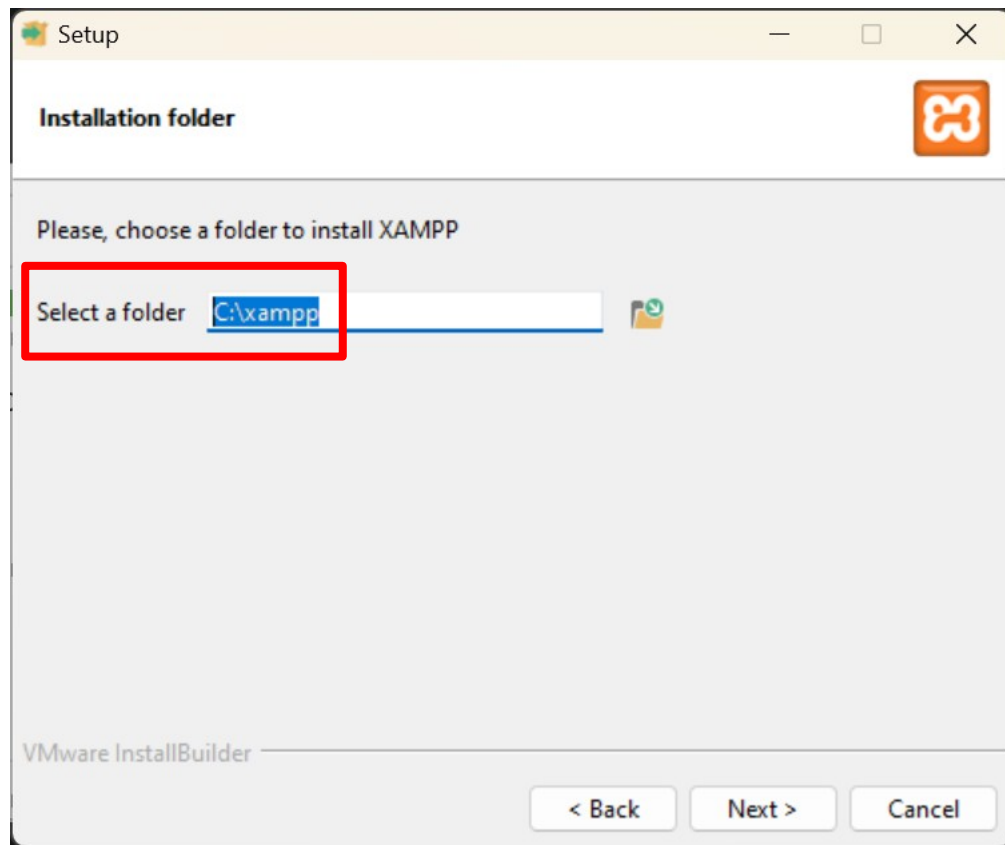


INSTITUTO
FEDERAL
Santa Catarina

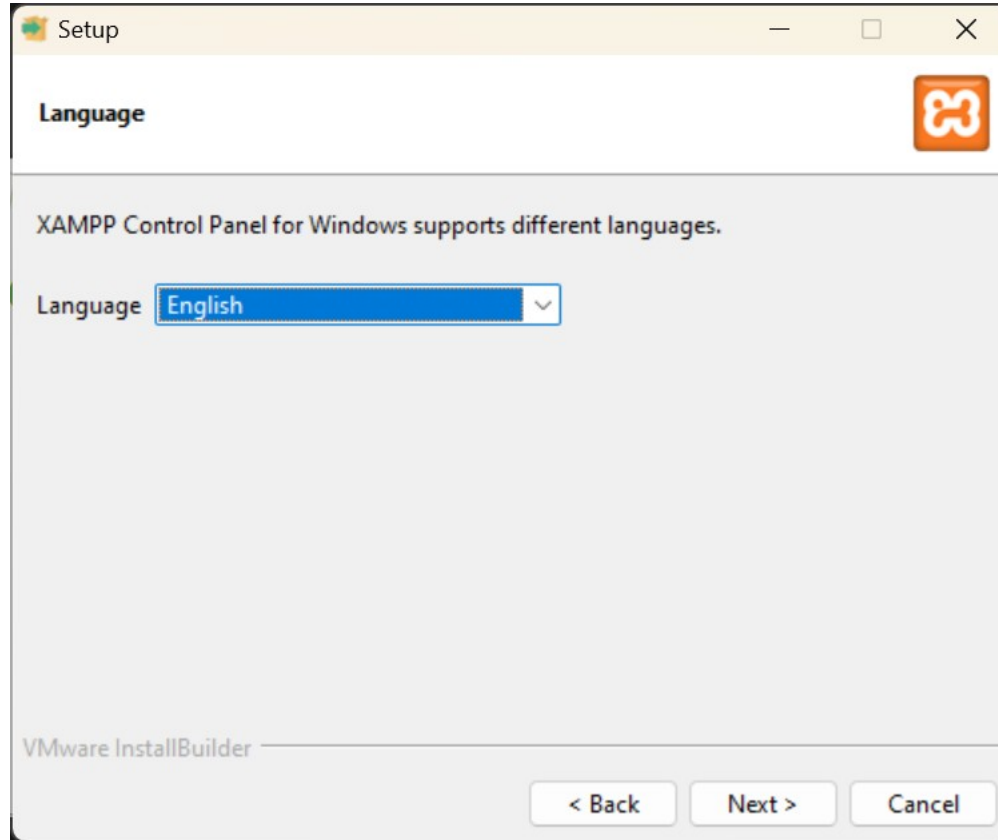
Configuração do ambiente (Windows)



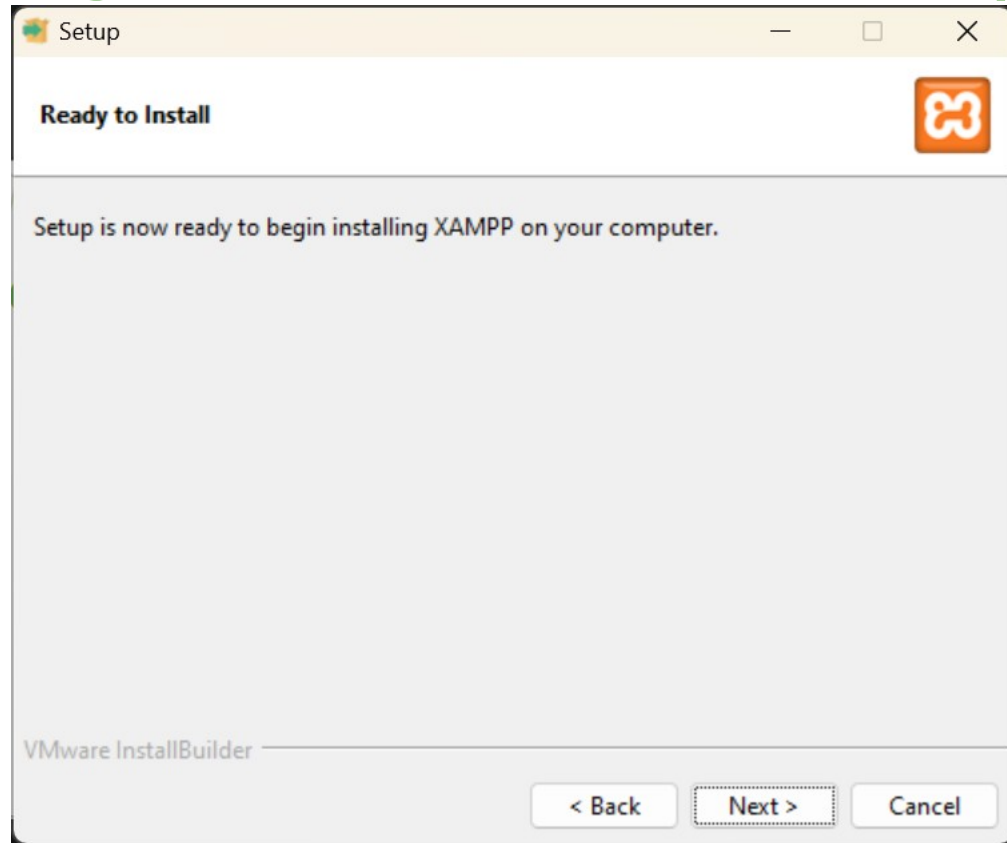
Configuração do ambiente (Windows)



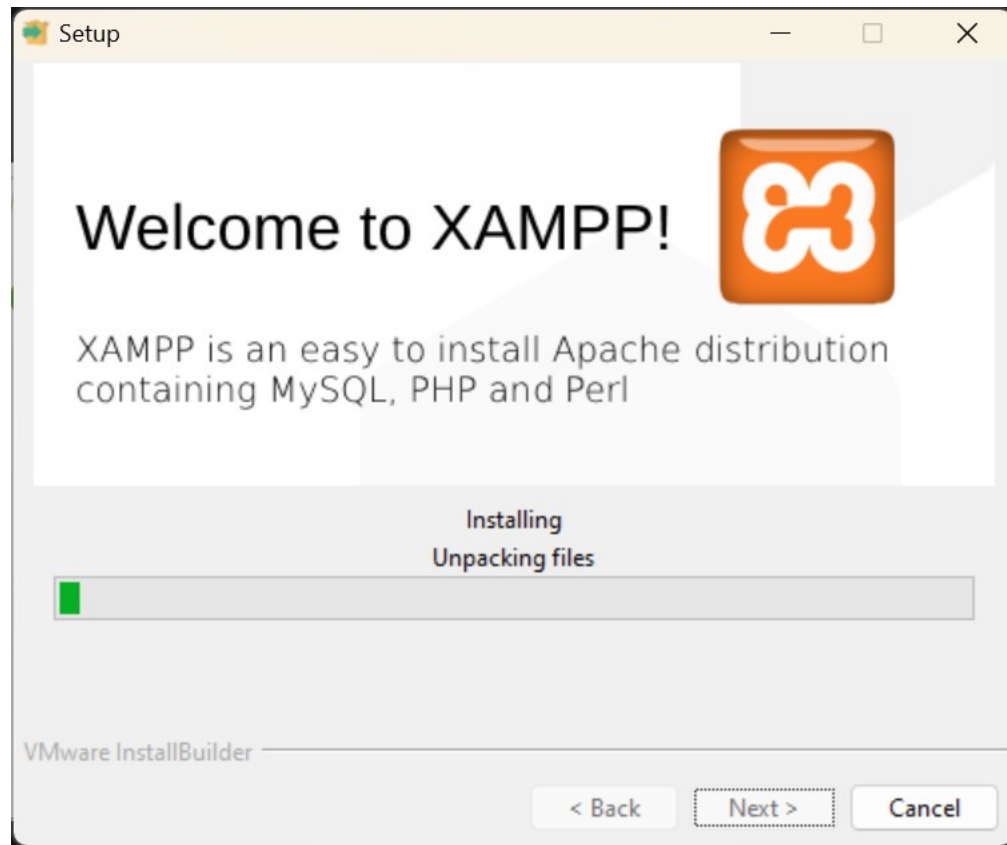
Configuração do ambiente (Windows)



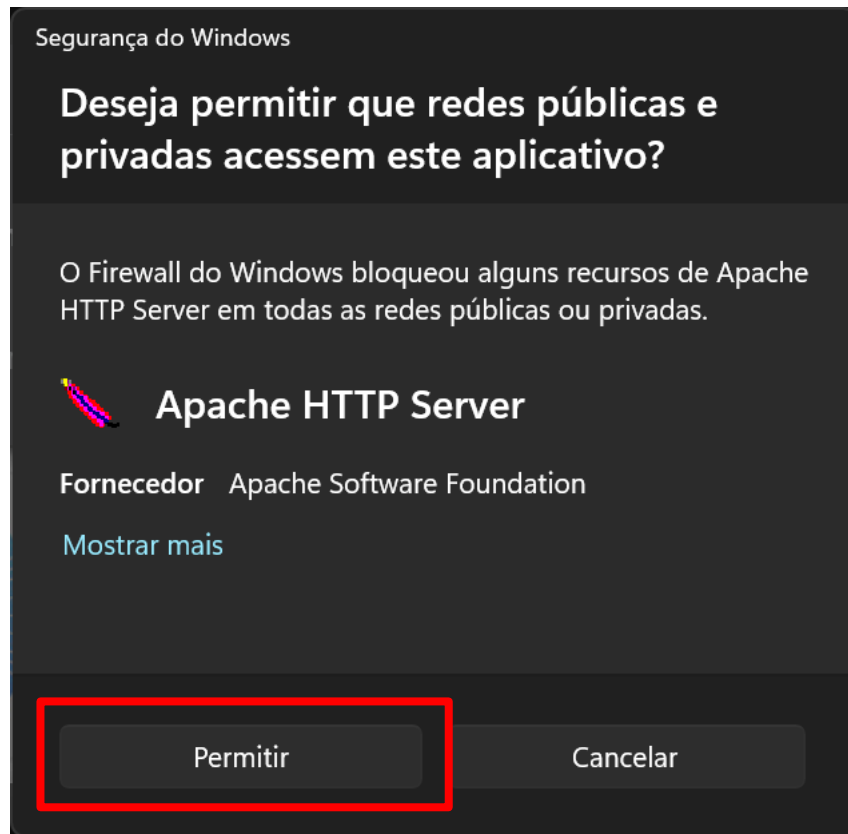
Configuração do ambiente (Windows)



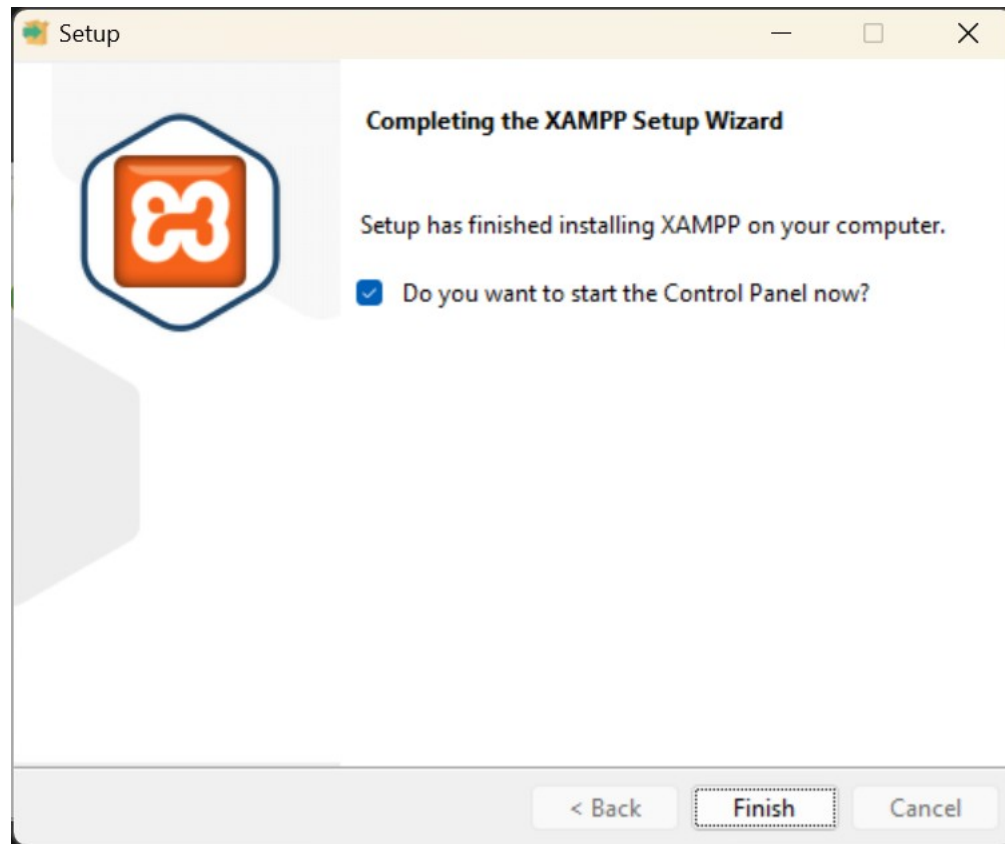
Configuração do ambiente (Windows)



Configuração do ambiente (Windows)



Configuração do ambiente (Windows)



Configuração do ambiente (Windows)

Recomendações



XAMPP Control Panel
Adicionados recentemente

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

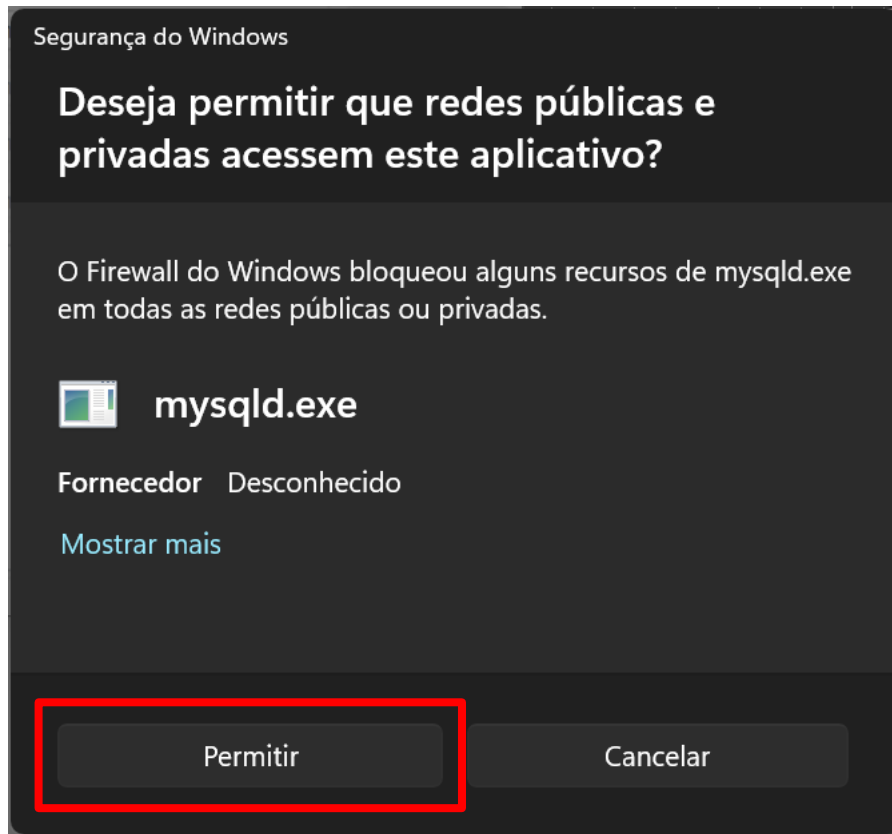
XAMPP Control Panel v3.3.0

Modules	Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	Apache			Start Admin Config Logs
<input type="checkbox"/>	MySQL	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat	Tomcat			Start Admin Config Logs

Config Netstat Shell Explorer Services Help Quit

14:33:58 [main] there will be a security dialogue or things will break! So think about running this application with administrator rights!
14:33:58 [main] XAMPP Installation Directory: "c:\xampp\
14:33:58 [main] Checking for prerequisites
14:33:59 [main] All prerequisites found
14:33:59 [main] Initializing Modules
14:33:59 [main] Starting Check-Timer
14:33:59 [main] Control Panel Ready

Configuração do ambiente (Windows)



Configuração do ambiente (Windows)

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

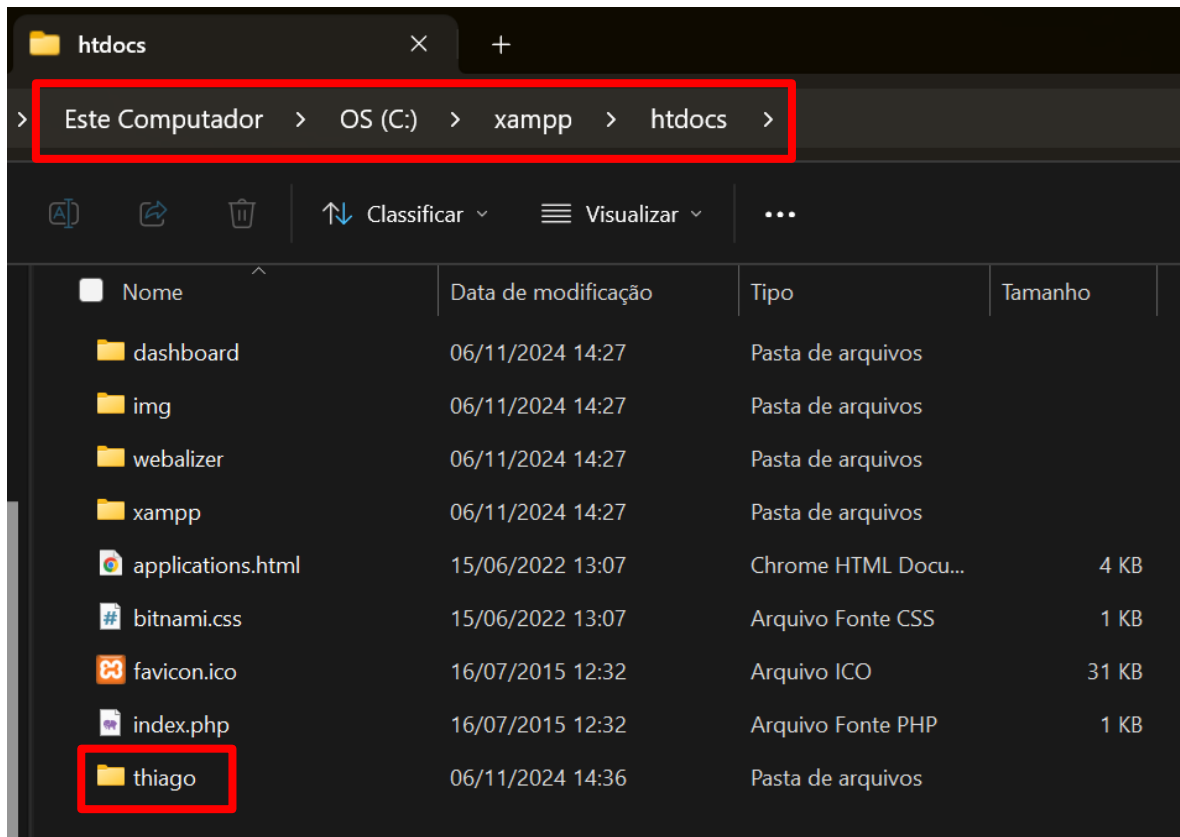
Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	44108 42856	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	28368	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config Netstat Shell Explorer Services Help Quit

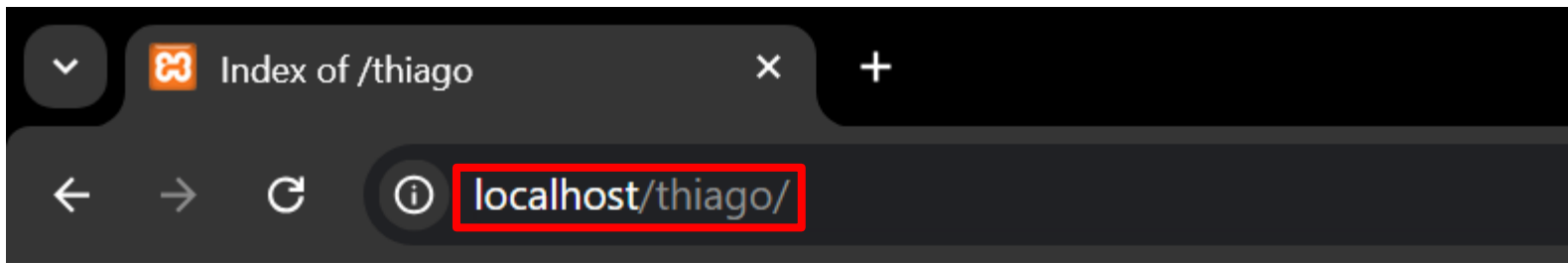
14:33:59 [main] All prerequisites found
14:33:59 [main] Initializing Modules
14:33:59 [main] Starting Check-Timer
14:33:59 [main] Control Panel Ready
14:35:09 [Apache] Attempting to start Apache app...
14:35:10 [Apache] Status change detected: running
14:35:11 [mysql] Attempting to start MySQL app...
14:35:11 [mysql] Status change detected: running

Configuração do ambiente (Windows)

Crie um
diretório
com seu
nome →



Configuração do ambiente (Windows)



Index of /thiago

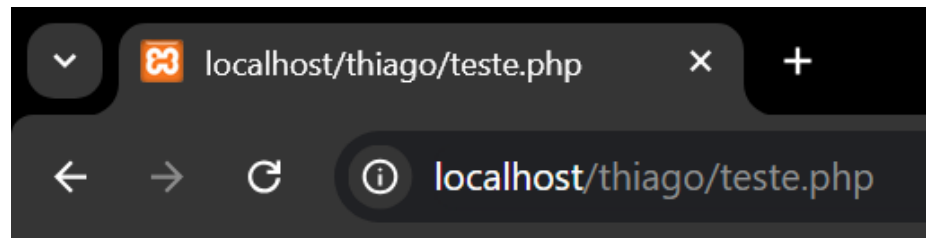
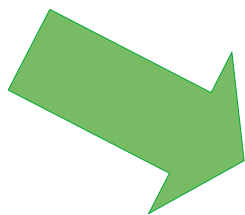
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80

Pronto!
Tudo certo!

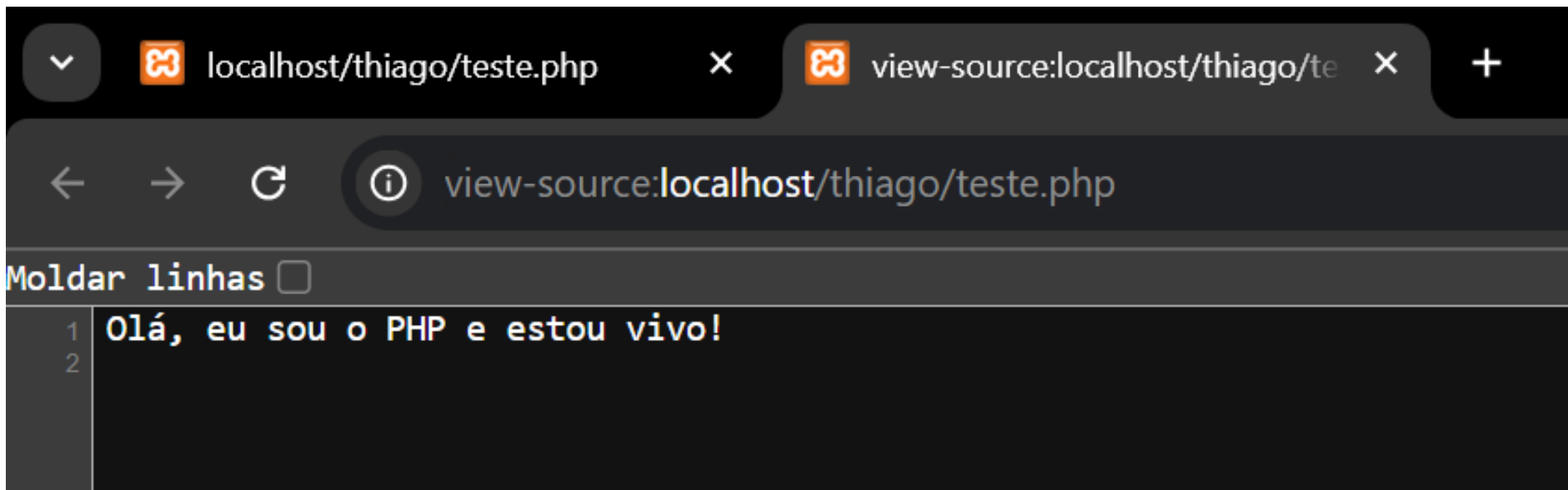
Introdução a sintaxe básica e ao ambiente

```
1  <?php
2
3  echo "Olá, eu sou o PHP e estou vivo!";
4
5  ?>
```



Olá, eu sou o PHP e estou vivo!

Introdução a sintaxe básica e ao ambiente



localhost/thiago/teste.php × view-source:localhost/thiago/te × +

← → ↻ ⓘ view-source:localhost/thiago/teste.php

Moldar linhas ☐

```
1 Olá, eu sou o PHP e estou vivo!
```

Introdução a sintaxe básica e ao ambiente

```
<?php

// Declaração de variáveis e comentários

// Declarando variáveis - nos comentários, os possíveis tipos de dados
$nome = "Maria"; // String
$idade = 30;      // Inteiro
$altura = 1.65;   // Float
$status = true;   // Boolean

// Imprimindo o conteúdo de variáveis
echo "Me chamo $nome, tenho $idade anos e meço $altura m.";

?>
```

Exercícios

Resolva os exercícios 1 ao 5 da Lista 06.



Terceira Nota/Avaliação Final

Para a composição da terceira **nota/avaliação final**, serão utilizados os seguintes critérios:

- Resolução de exercícios durante as aulas (20%);
- Trabalho (30%);
- Prova individual e com consulta (50%).





**INSTITUTO
FEDERAL**
Santa Catarina

Estruturas de Controle



Estruturas de Controle

Estruturas Condicionais – if, else e elseif

Estruturas condicionais permitem que o programa execute blocos de código diferentes com base em determinadas condições. Em PHP, a estrutura if é usada para avaliar uma condição e executar o código correspondente se a condição for verdadeira

Estruturas de Controle

Estruturas Condicionais – if, else e elseif

```
if (condicao) {  
    // Código a ser executado se a condição for verdadeira  
}  
  
// exemplo:  
$idade = 18;  
  
if ($idade >= 18) {  
    echo "Você é maior de idade."  
}
```

Estruturas de Controle

Estruturas Condicionais – if, else e elseif

```
$idade = 16;  
  
if ($idade >= 18) {  
    echo "Você é maior de idade.";  
} else {  
    echo "Você é menor de idade.";  
}
```

Estruturas de Controle

Estruturas Condicionais – if, else e elseif

```
$nota = 7;

if ($nota >= 9) {
    echo "Parabéns, você tirou uma nota excelente!";
} elseif ($nota >= 6) {
    echo "Você tirou uma nota boa!";
} else {
    echo "Infelizmente, você não passou.";
}
```

Estruturas de Controle

Operadores relacionais

Operador	Nome	Função
==	Igual a	Testa se um valor é igual ao outro
!=	Diferente de	Testa se dois valores são diferentes entre si
>	Maior que	Testa se um valor é maior que o outro
>=	Maior ou igual a	Testa se um valor é maior ou igual a outro
<	Menor que	Testa se um valor é menor que o outro
<=	Menor ou igual a	Testa se um valor é menor ou igual a outro

Estruturas de Controle

Operadores lógicos

Operador	Nome	Função
& &	E	Verdadeiro se ambas as condições forem verdadeiras
	OU	Verdadeiro se pelo menos uma das condições for verdadeira
!	Negação	inverte o valor lógico da condição

Exercícios

Resolva os exercícios 6 ao 13 da Lista 06.



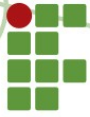
Estruturas de Controle

O que é o **switch**?

- O **switch** é uma estrutura condicional que verifica o valor de uma variável ou expressão e executa um bloco de código correspondente.

Quando usar o **switch**?

- Quando há múltiplas opções baseadas no mesmo valor. Para substituir vários if-elseif-else.



INSTITUTO
FEDERAL
Santa Catarina

Estruturas de Controle

Exemplo: Crie um *script* que receba o dia da semana em uma variável (número de 1 a 3, onde 1 representa domingo) e escreva na tela o nome do dia. Faça com if-elseif-else e com switch e compare os resultados.



```
<?php
// versão com if-elseif-else
$dia = 3;

if ($dia == 1) {
    echo "Hoje é domingo.";
} elseif ($dia == 2) {
    echo "Hoje é segunda-feira.";
} elseif ($dia == 3) {
    echo "Hoje é terça-feira.";
} else {
    echo "Número inválido.";
}
?>
```

```
<?php
// versão com switch-case
$dia = 3;

switch ($dia) {
    case 1:
        echo "Hoje é domingo.";
        break;
    case 2:
        echo "Hoje é segunda-feira.";
        break;
    case 3:
        echo "Hoje é terça-feira.";
        break;
    default:
        echo "Número inválido.";
        break;
}
?>
```

Estruturas de Controle

Vantagens do uso do switch

Legibilidade: O **switch** apresenta uma estrutura mais compacta e visualmente limpa, especialmente em casos com muitas condições. No **if-elseif-else**, o código fica mais extenso, com repetição da variável \$dia em cada condição.

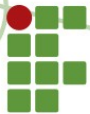
Clareza na Lógica: O **switch** separa claramente os casos possíveis (case) e o caso padrão (default). O **if-elseif-else** mistura as condições e o código resultante, o que pode dificultar a leitura.

Estruturas de Controle

Vantagens do uso do switch

Facilidade de Manutenção: Adicionar um novo dia ou caso no **switch** é simples e não impacta a estrutura. No **if-elseif-else**, adicionar um novo caso exige modificar a lista de condições, aumentando o risco de erro.

Performance: Em cenários com várias condições, o **switch** pode ser mais eficiente, pois evita verificações repetitivas.

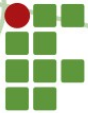


INSTITUTO
FEDERAL
Santa Catarina

Estruturas de Controle

Podemos utilizar o case para “agrupar” condições.
Veja o código no próximo *slide* e tente imaginar a saída dele para as seguintes notas: A, B, C, D e E.

No exemplo, o *script* deve receber uma nota de entrada e dizer o que ela representa.



INSTITUTO
FEDERAL
Santa Catarina

```
<?php
$nota = "B";

switch ($nota) {
    case "A":
    case "B":
        echo "Bom desempenho!";
        break;
    case "C":
        echo "Desempenho médio.";
        break;
    default:
        echo "Desempenho insuficiente.";
        break;
}
?>
```

Estruturas de Controle

Pode-se utilizar o **switch** para verificações mais complexas como, por exemplo, a verificação de intervalos.

Veja no próximo slide um exemplo, atente para a definição do **switch**: **switch(true)**.

Estruturas de Controle

Exemplo: Crie um programa em PHP que classifique a faixa etária de uma pessoa com base na sua idade. O programa deve receber a idade armazenada em uma variável e, utilizando um switch-case, exibir uma mensagem indicando a categoria correspondente:

- Idades menores que 13: "Você é uma criança."
- Idades entre 13 e 17: "Você é um adolescente."
- Idades entre 18 e 59: "Você é um adulto."
- Idades a partir de 60: "Você é um idoso."



INSTITUTO
FEDERAL
Santa Catarina

```
<?php
// Usando o switch-case para verificar intervalos
$idade = 20;

switch (true) {
    case ($idade < 13):
        echo "Você é uma criança.";
        break;
    case ($idade < 18):
        echo "Você é um adolescente.";
        break;
    case ($idade < 60):
        echo "Você é um adulto.";
        break;
    default:
        echo "Você é um idoso.";
        break;
}
?>
```


Exercícios

Resolva os exercícios 1 ao 11 da Lista 07 - Switch.



“O que são *loops*?”

Um *loop* é uma estrutura de repetição que executa um bloco de código várias vezes enquanto uma condição for verdadeira. Ele é usado para tarefas repetitivas, como percorrer listas ou somar valores.”

Em PHP temos quatro tipos de *loops*:

- 1) **while**;
- 2) **do..while**;
- 3) **for**;
- 4) **foreach**.

O **while** e **do..while** possuem um comportamento semelhante: enquanto faça ou faça enquanto.

Loops

Para tentar perceber a diferença entre **while** e **do..while**, analise os dois códigos abaixo (são contadores até 5, ou deveriam ser):

```
$contador = 5;  
while ($contador < 5) {  
    echo "Contagem: $contador <br>";  
    $contador++;  
}
```

```
$contador = 5;  
do {  
    echo "Contagem: $contador <br>";  
    $contador++;  
} while ($contador < 5);
```

Qual é a saída de cada um dos dois algoritmos?

O **for** também pode ser usado para contagem e possui uma sintaxe mais prática, dependendo do contexto:

```
for (início; condição; incremento) {  
    // Código a ser executado em cada iteração  
}
```

Veja um comparativo de contadores de 1 a 5 (reais-oficiais) com **while**, **do..while** e **for**:

```
$contador = 1;
while ($contador <= 5) {
    echo "Contagem: $contador <br>";
    $contador++;
}
```

```
$contador = 1;
do {
    echo "Contagem: $contador <br>";
    $contador++;
} while ($contador <= 5);
```

```
for ($i = 1; $i <= 5; $i++) {
    echo "Contagem: $i <br>";
}
```

Exercícios

Resolva os exercícios 1 ao 8 da Lista 08 – *Loops* – Parte 1.



“O que é o **foreach** (para cada)?

O **foreach** é uma estrutura de repetição em PHP usada exclusivamente para percorrer elementos de **arrays** e **objetos**. Ele simplifica o acesso a cada elemento de uma coleção, evitando a necessidade de gerenciar índices manualmente.”

Sintaxe básica do **foreach**

```
# básica  
foreach ($array as $valor) {  
    // Código a ser executado  
}
```

Comparativo: percorrendo um **array** de frutas com **for** e com **foreach**

```
$frutas = ["Maça", "Pera", "Uva"];
```

```
for($i = 0; $i < sizeof($frutas); $i++) {  
    print "Fruta: $frutas[$i] <br>";  
}
```

```
foreach ($frutas as $fruta) {  
    print "Fruta: $fruta <br>";  
}
```

Sintaxe com chave e valor do **foreach**

```
# com chave e valor
foreach ($array as $chave => $valor) {
    // Código a ser executado
}
```

Exemplo com chave e valor:

```
# exemplo: percorrer um array com chave e valor
# que associa nomes a idades
$ pessoas = ["Ana" => 25, "João" => 30, "Carlos" => 35];

foreach ($pessoas as $nome => $idade) {
    echo "$nome tem $idade anos.<br>";
}
```

Exercícios

Resolva os exercícios 1 ao 7 da Lista 09 – *Loops* – Parte 2.



Manipulação de Formulários

Para que possamos dar “vida” para nossos sistemas Web, é necessário manipular os dados informados pelos usuários.

Quando um usuário preenche um formulário, clica em um botão, navega entre as páginas, etc., ele envia dados para o servidor Web que devem ser **processados**.

Manipulação de Formulários

É possível manipular dados enviados via GET ou POST.
Vamos iniciar pelo método GET...

“O que é o Método GET?

O método GET é usado para enviar dados de um formulário ao servidor como parte da URL. É ideal para:

- Recuperar informações que não precisam ser privadas.
- Pesquisas e filtros.
- Situações onde os dados enviados não excedem o limite de tamanho do URL (geralmente 2000 caracteres).”

Manipulação de Formulários

Veremos um exemplo prático de uso do GET.
Criaremos um formulário bem simples com dados que
serão enviados para o servidor (arquivo processar.html):

Nome: Idade:

Manipulação de Formulários

Veremos um exemplo prático de uso do GET.
Criaremos um formulário bem simples com dados que
serão enviados para o servidor (arquivo processar.html):

```
<form action="processar.php" method="get">  
  <label for="nome">Nome:</label>  
  <input type="text" id="nome" name="nome">  
  
  <label for="idade">Idade:</label>  
  <input type="number" id="idade" name="idade">  
  
  <button type="submit">Enviar</button>  
</form>
```

Manipulação de Formulários

Agora é necessário criar o arquivo que processará os dados enviados (arquivo processar.php – **versão 1**):

```
<?php
$nome = $_GET['nome'];
$idade = $_GET['idade'];
echo "Olá, $nome! Você tem $idade anos.";
?>
```

Há algum problema nesse código?

Manipulação de Formulários

Agora é necessário criar o arquivo que processará os dados enviados (arquivo processar.php – **versão 2**):

```
<?php
if (isset($_GET['nome']) && isset($_GET['idade'])) {
    $nome = $_GET['nome'];
    $idade = $_GET['idade'];
    echo "Olá, $nome! Você tem $idade anos.";
} else {
    echo "Por favor, preencha o formulário.";
}
?>
```

Manipulação de Formulários

Podemos “misturar” HTML e PHP em um único arquivo, facilitando a implementação. Veja o exemplo abaixo (arquivo processar-mix1.php):

```
<h1>Minha página única</h1>

<?php
if (isset($_GET['nome']) && isset($_GET['idade'])) {
    $nome = $_GET['nome'];
    $idade = $_GET['idade'];
    echo "<p>Olá, $nome! Você tem $idade anos.</p>";
} else {
    echo "<p>Erro: Preencha o formulário!</p>";
}
?>

<form action="" method="get">
    <label for="nome">Nome:</label>
    <input type="text" id="nome" name="nome" required>
    <label for="idade">Idade:</label>
    <input type="number" id="idade" name="idade" required>
    <button type="submit">Enviar</button>
</form>
```

Manipulação de Formulários

Podemos, ainda, “misturar” HTML e PHP em um único arquivo mais completo e elegante.

Veja o exemplo no SIGAA utilizando heredoc (**um nível um pouco mais avançado**).

Arquivo: processar-mix2.php

Manipulação de Formulários

Veremos agora como manipular dados de formulários com o POST.

Veja no slide, a seguir, as principais diferenças entre o GET e o POST.

Manipulação de Formulários

Diferenças entre o GET e o POST

Aspecto	GET	POST
Visibilidade	Os dados aparecem na URL.	Os dados não aparecem na URL.
Segurança	Menos seguro, pois os dados são visíveis.	Mais seguro, adequado para dados sensíveis.
Tamanho	Limite de tamanho (depende do navegador).	Não há limite prático.
Uso Comum	Filtros, buscas, navegação.	Login, formulários longos.

Manipulação de Formulários

Como ficaria nosso primeiro exemplo com o uso do POST?

Veja abaixo (arquivo processar-post.html):

```
<form action="processar-post.php" method="post">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome" required>

  <label for="idade">Idade:</label>
  <input type="number" id="idade" name="idade" required>

  <button type="submit">Enviar</button>
</form>
```


Manipulação de Formulários

Como ficaria nosso primeiro exemplo com o uso do POST?

Veja abaixo (arquivo processar-post.php):

```
<?php
if (isset($_POST['nome']) && isset($_POST['idade'])) {
    $nome = $_POST['nome'];
    $idade = $_POST['idade'];
    echo "Olá, $nome! Você tem $idade anos.";
} else {
    echo "Por favor, preencha o formulário.";
}
?>
```

Manipulação de Formulários

Crie e teste os arquivos `processar-post.html` e `processar-post.php` e verifique o comportamento da URL.



Exercícios

Resolva os exercícios 1 ao 6 da Lista 10.



Acesso a Banco de Dados

Onde os sistemas Web armazenam seus dados como logins, senhas, informações dos produtos, etc?

O armazenamento é realizado em banco de dados.



Acesso a Banco de Dados

O PHP permite que possamos, facilmente, fazer consultas, alterações, inclusões e exclusões em SGBDs como o mariaDB (antigo MySQL).

Veremos, a seguir, um exemplo de inclusão (cadastro) de usuário em banco de dados.

As outras operações como consultas, alterações e exclusões são feitas de maneira semelhante.

Acesso a Banco de Dados

Vamos supor que desejamos cadastrar o nome completo e o endereço de e-mail de algum usuário.

Para isso, é necessário criarmos uma tabela no banco de dados para receber esses dados:

usuarios		
id	nome	email

Acesso a Banco de Dados

Utilizaremos, como base, o *script* SQL abaixo que cria um database chamado **meu_banco** no mariaDB, uma tabela chamada **usuarios** e adiciona 3 usuários fictícios nessa tabela. Você pode utilizar o phpMyAdmin para executar o script.

```
CREATE DATABASE meu_banco;
USE meu_banco;
CREATE TABLE usuarios (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL
);
INSERT INTO usuarios (nome, email) VALUES
('João Silva', 'joao.silva@example.com'),
('Maria Oliveira', 'maria.oliveira@example.com'),
('Carlos Souza', 'carlos.souza@example.com');
```

Acesso a Banco de Dados

Depois, criaremos um “módulo” em PHP que será responsável por fazer a conexão no SGBD (arquivo conectar.php).

Veja o conteúdo no próximo slide.



Acesso à Banco de Dados

```
<?php
// Configurações de conexão ao banco de dados mariaDB
$servername = "localhost"; // Endereço do servidor
$username = "root"; // Usuário do banco de dados
$password = ""; // Senha do banco de dados - o padrão é em branco
$dbname = "meu_banco"; // Nome do banco de dados

// Cria a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifica se a conexão deu certo
if ($conn->connect_error) {
    die("Conexão falhou: " . $conn->connect_error);
}
?>
```

Acesso à Banco de Dados

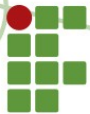
Com o “módulo” que faz a conexão no banco (arquivo conecta.php), podemos dar sequência ao formulário de cadastro (arquivo cadastrar.php).

Podemos realizar o cadastro de diversas maneiras, aqui apresentarei uma que utiliza o seguinte esquema:

- 1) conecta no banco (include “conectar.php”);
- 2) verifica os dados enviados via formulário e insere no banco;
- 3) parte em HTML do formulário.

Acesso à Banco de Dados

Todo o conteúdo, a seguir, faz parte de um mesmo arquivo: `cadastrar.php`



INSTITUTO
FEDERAL
Santa Catarina

Acesso à Banco de Dados

```
<?php
// 1ª parte
// inclui o arquivo que faz a conexão no SGBD
include "conectar.php";
```

```
// 2ª parte
// Verifica se o formulário foi enviado via método POST
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Recebe os dados do formulário
    $nome = $_POST['nome'];
    $email = $_POST['email'];

    // Valida os dados - verifica se os dois não estão vazios
    if (!empty($nome) && !empty($email)) {
        // Prepara a SQL para evitar SQL Injection
        $stmt = $conn->prepare("INSERT INTO usuarios (nome, email) VALUES (?, ?)");
        $stmt->bind_param("ss", $nome, $email);

        // executa o SQL e verifica se o retorno foi positivo
        if ($stmt->execute()) {
            echo "<p>Usuário cadastrado com sucesso!</p>";
        } else {
            echo "<p>Erro ao inserir registro: " . $stmt->error . "</p>";
        }
        // fecha a consulta
        $stmt->close();
    } else {
        echo "<p>Erro: Por favor, preencha todos os campos.</p>";
    }
}

// Fecha a conexão com o mariaDB - importante!
$conn->close();
?>
```



INSTITUTO
FEDERAL
Santa Catarina

Acesso a Banco de Dados

```
<!--
```

3ª parte - Nosso formulário de cadastro, aqui ele está em uma versão minimalista, porém deve estar em um arquivo HTML com todos os cabeçalhos, estilos, etc.

```
-->
```

```
<form action="cadastrar.php" method="post">
```

```
    Nome: <input type="text" name="nome"><br>
```

```
    Email: <input type="email" name="email"><br>
```

```
    <input type="submit" value="Cadastrar">
```

```
</form>
```

Acesso a Banco de Dados

Ao submeter os dados no formulário, esse deve ser o resultado obtido:

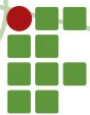
Usuário cadastrado com sucesso!

Nome:

Email:

Cadastrar

Também deve ser possível consultar os dados no phpMyAdmin!



**INSTITUTO
FEDERAL**
Santa Catarina

Duvidas?

