

SW-I

SISTEMAS WEB I

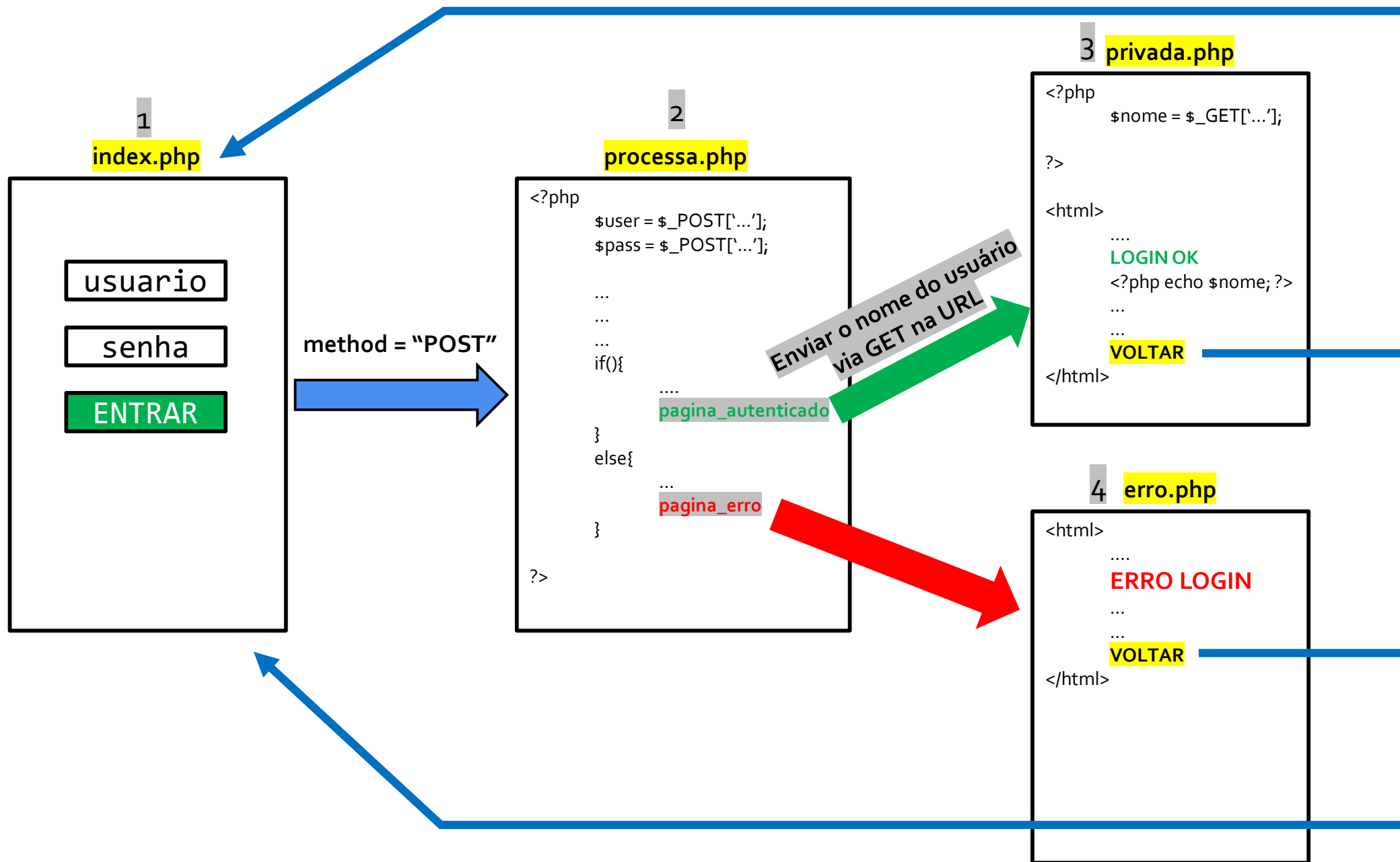
Prof. Anderson Vanin

AULA 05

INTEGRANDO HTML, CSS e JS com PHP

Projeto

Vamos criar um exemplo de autenticação de usuário sem o uso de banco de dados somente para exemplificar o processo de navegação entre páginas de um sistema utilizando PHP, uso prático dos métodos POST e GET, além de entender como inserir instruções PHP em páginas mescladas com HTML, CSS e JS.



Projeto

1. Página **index.php**: Página contendo somente código html com o formulário para captar as informações de nome de usuário e senha. Ao clicar em ENTRAR, os dados devem ser submetidos para a página **processa.php** via método POST.
2. Página **processa.php**: Página com somente códigos php que irá receber as informações do formulário da página **index.php**. Nesta página, os dados serão verificados através de um comando if (nome de usuário e senha no próprio script). Se o if retornar verdadeiro, encaminhar o nome do usuário via GET (na URL) para a página **privada.php**, caso contrário, encaminhar o usuário para a página **erro.php**.

Projeto

3. Página **privada.php**: Página que irá ser exibida se as informações de usuário e senha estiverem corretas. Esta página irá conter inicialmente um trecho de código php no inicio da página para recuperar a informação do nome de usuário que será enviada pela URL vindo da página **processa.php**. Depois deste trecho, código html e css para exibir as informações relativas à página restrita. No meio do código html, você pode inserir php para exibir o nome do usuário recuperado no inicio desta página, além disso, um botão que permitirá o usuário retornar para a página **index.php** e iniciar o processo novamente.

Projeto

4. Página **erro.php**: Página que irá ser exibida se as informações de usuário e senha estiverem incorretas. Esta página irá conter código html mostrando que houve um erro de login (usuário e/ou senha incorretos) e um botão que permitirá o usuário retornar para a página **index.php** e iniciar o processo novamente. Nesta página não será necessário nenhuma lógica a mais com códigos php.

ESTRUTURA DO PROJETO

Estrutura do projeto

Neste projeto, iremos utilizar alguns *templates* prontos com uso do framework Bootstrap para as páginas: **index.php**, **privada.php** e **erro.php**.

A página **index.php** irá conter somente um formulário que pode ser obtido em:
<https://mdbootstrap.com/docs/standard/forms/overview/>.

Estrutura do projeto – index.php

Da página <https://mdbootstrap.com/docs/standard/forms/overview/> vamos utilizar o seguinte formulário (clique em **SHOW CODE** e copie o código):

The screenshot displays the MDB Bootstrap documentation page for the 'Login form' component. The page features a sidebar with navigation links and a main content area with the form example. The form includes the following elements:

- Search bar: Search (ctrl + "/" to focus)
- Navigation links: Docs, Pricing, Support, Ser
- Form title: Login form
- Form description: Typical login form with additional register buttons.
- Form fields: Email address, Password
- Form controls: ☒ Remember me, [Forgot password?](#)
- Form button: SIGN IN
- Form text: Not a member? [Register](#)
- Form text: or sign up with:
- Form social media links: [f](#), [G](#), [t](#), [i](#)
- Form footer: [SHOW CODE](#), [EDIT IN SANDBOX](#)

Red arrows highlight the 'SHOW CODE' button and the 'Forgot password?' link.

Estrutura do projeto – index.php

Clique em **SHOW CODE** e copie o código **HTML**

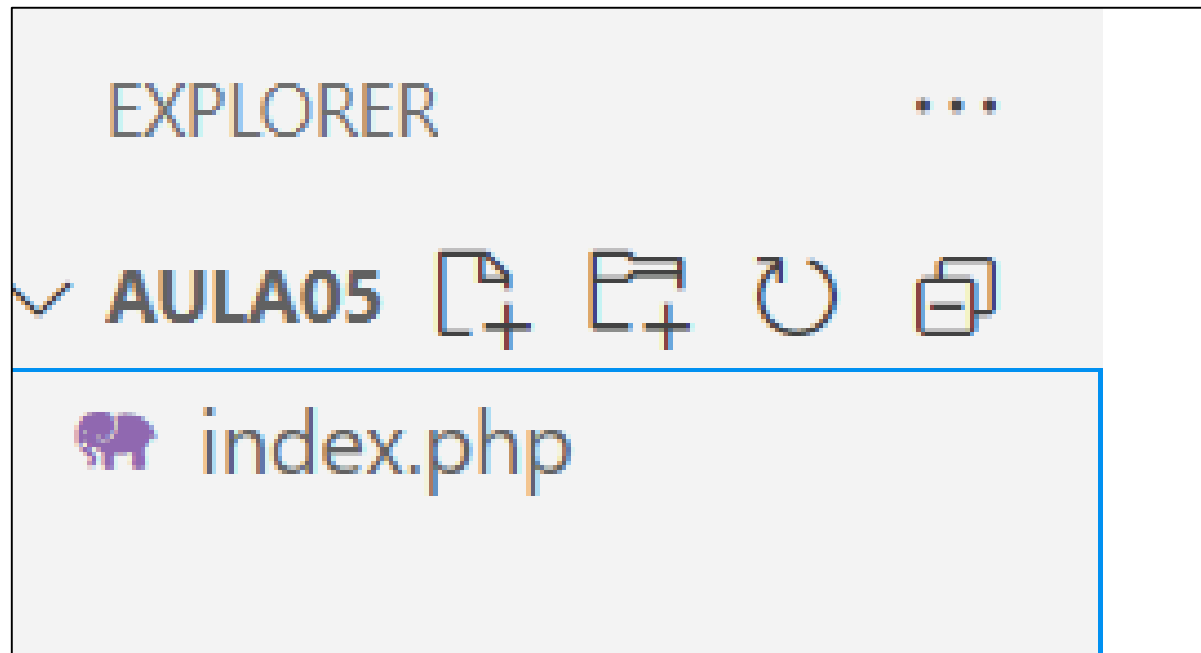
```
HTML  JAVASCRIPT  COPY

1  <form>
2    <!-- Email input -->
3    <div data-mdb-input-init class="form-outline mb-4">
4      <input type="email" id="form2Example1" class="form-control" />
5      <label class="form-label" for="form2Example1">Email address</label>
6    </div>
7
8    <!-- Password input -->
9    <div data-mdb-input-init class="form-outline mb-4">
10     <input type="password" id="form2Example2" class="form-control" />
11     <label class="form-label" for="form2Example2">Password</label>
12   </div>
13
14   <!-- 2 column grid layout for inline styling -->
15   <div class="row mb-4">
16     <div class="col d-flex justify-content-center">
17       <!-- Checkbox -->
18       <div class="form-check">
19         <input class="form-check-input" type="checkbox" value="" id="form2Example34" checked />
20         <label class="form-check-label" for="form2Example34">Remember me </label>
```

Estrutura do projeto – index.php

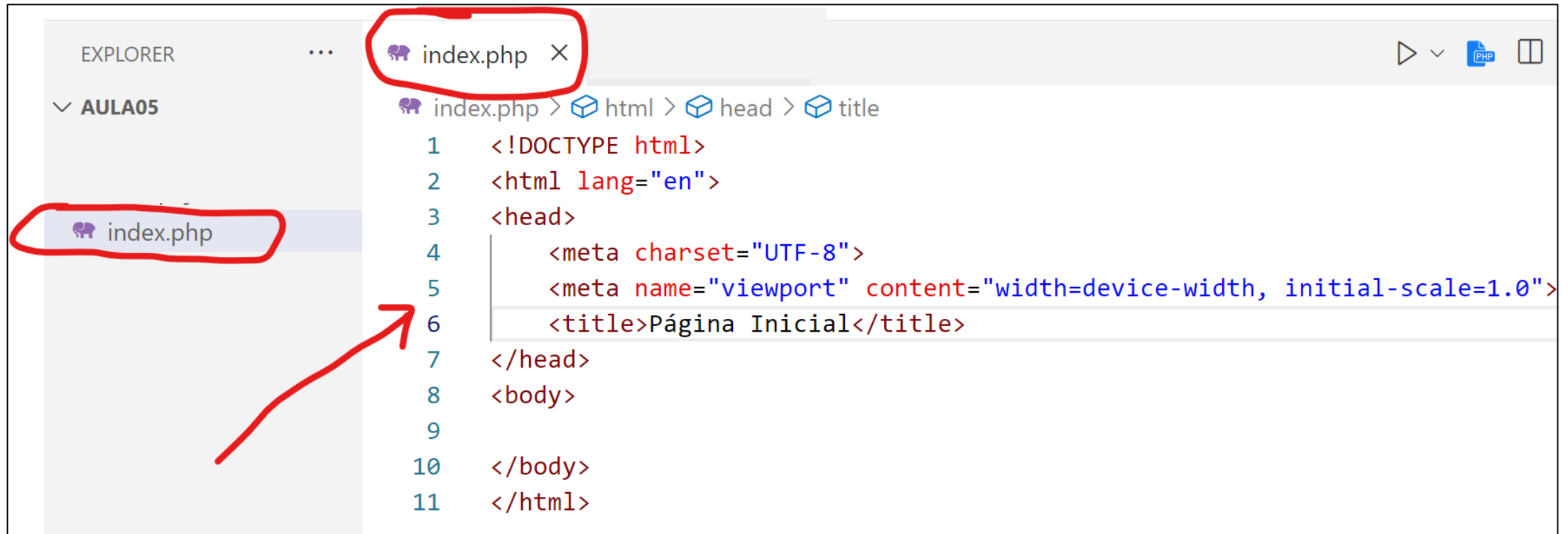
Crie uma pasta chamada **aula05** e crie o arquivo **index.php**

OBS.: ESTE PROJETO DEVE SER CRIADO DENTRO DA PASTA C:/XAMPP/HTDOCS



Estrutura do projeto – index.php

O arquivo **index.php** deve conter uma estrutura html que irá ter o formulário que vamos utilizar do template de formulário. Crie a estrutura html:5.



```
EXPLORER
...
index.php
AULA05
index.php

index.php > html > head > title
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Página Inicial</title>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

Estrutura do projeto – index.php

No arquivo **index.php** insira um link **cdn** para o framework bootstrap.


Os links podem ser obtidos em:

<https://getbootstrap.com/docs/5.2/getting-started/introduction/#cdn-links>

CDN links

As reference, here are our primary CDN links.

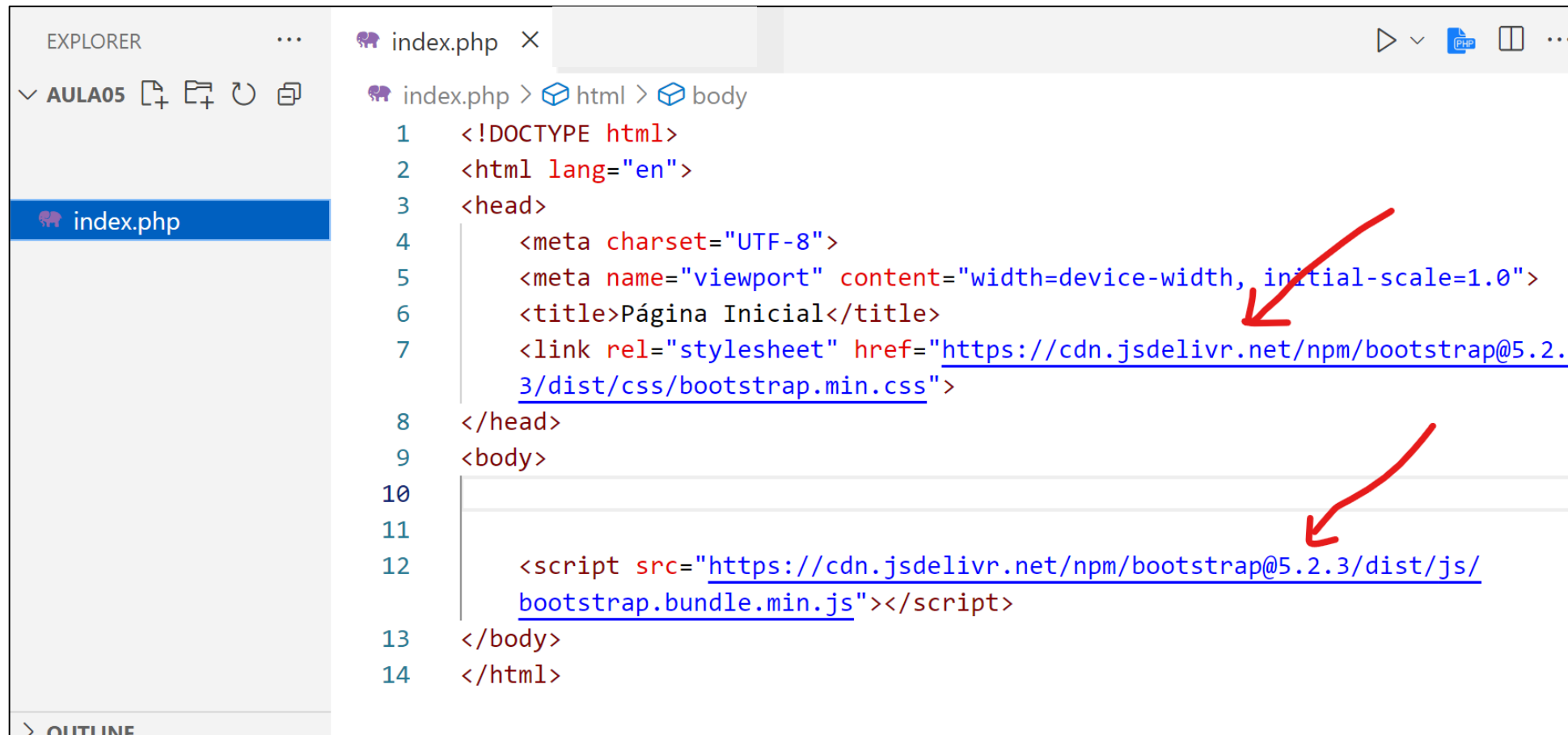
Description	URL
CSS	https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css
JS	https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js



You can also use the CDN to fetch any of our [additional builds listed in the Contents page](#).

Estrutura do projeto – index.php

No arquivo **index.php** insira um link **cdn** para o framework bootstrap.



```
EXPLORER
  AULA05
    index.php

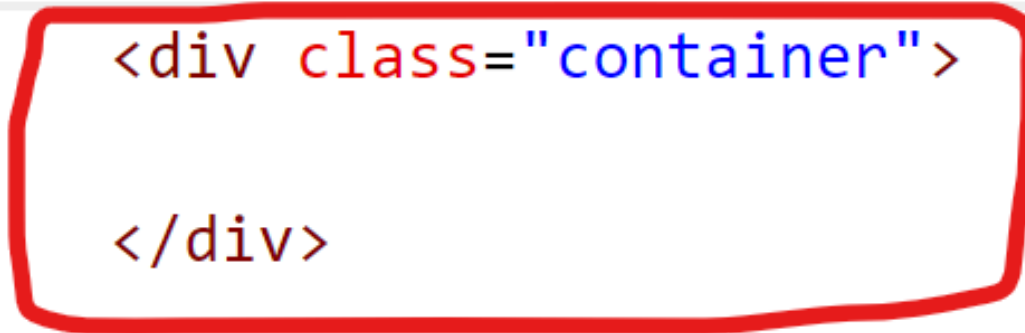
index.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Página Inicial</title>
7      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
8  </head>
9  <body>
10
11
12      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
13  </body>
14  </html>
```

Estrutura do projeto – index.php

No arquivo **index.php** insira uma div com a classe container para inserir o código do formulário do template.

```
</head>
<body>
  <div class="container">
  </div>

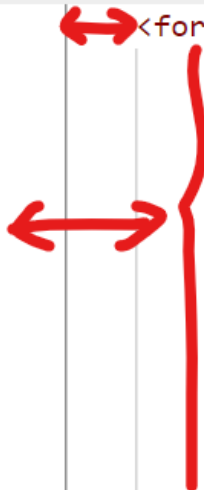
  <script src="https://cdn.jsdelivr.net/gh/twitter/tw-bootstrap@4.0.0-alpha.6/dist/js/bootstrap.bundle.min.js"></script>
</body>
```



Estrutura do projeto – index.php

Na div com a classe container cole o código do formulário do template. Ao colar, faça a indentação do código. Deixe seu código **LEGÍVEL**.

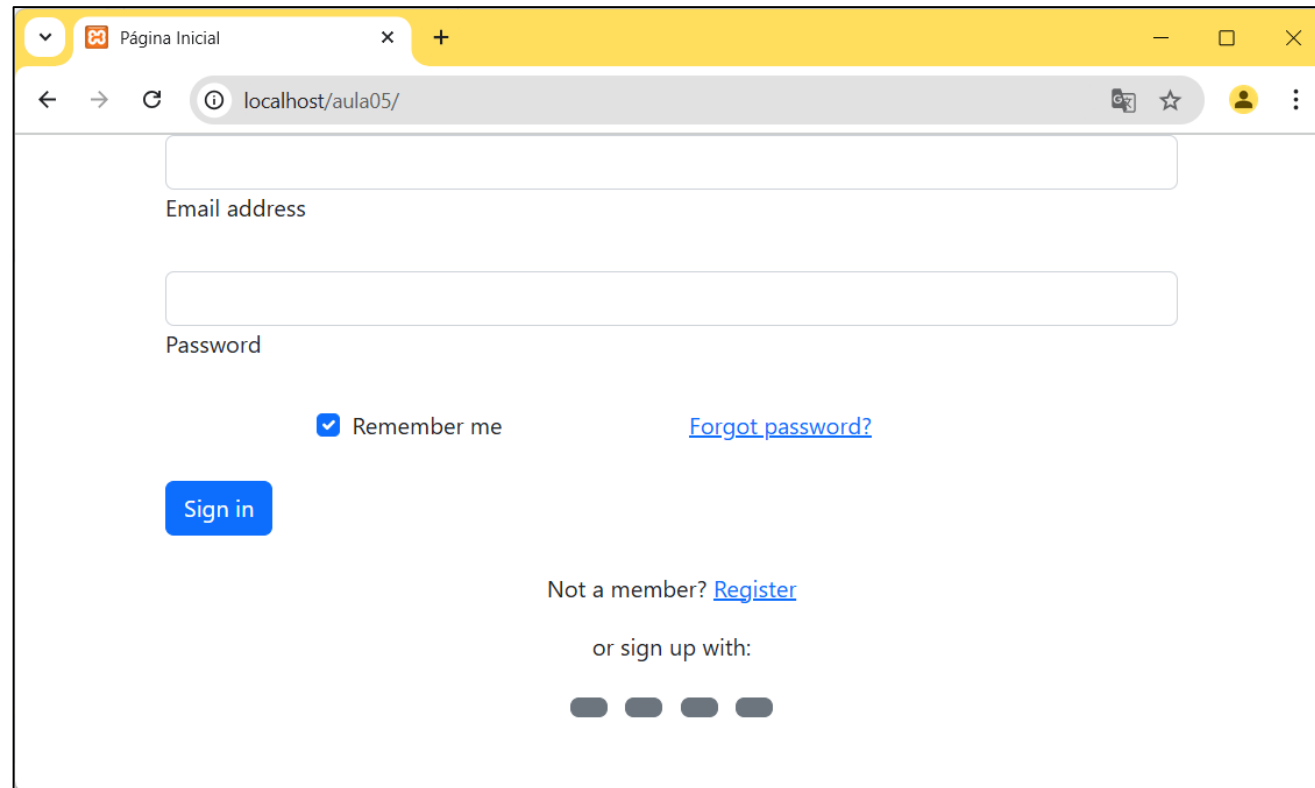
```
7 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/
8 </head>
9 <body>
10 <div class="container">
11   <form>
12     <!-- Email input -->
13     <div data-mdb-input-init class="form-outline mb-4">
14       <input type="email" id="form2Example1" class="form-control" />
15       <label class="form-label" for="form2Example1">Email address</label>
16     </div>
17
18     <!-- Password input -->
19     <div data-mdb-input-init class="form-outline mb-4">
20       <input type="password" id="form2Example2" class="form-control"
21       <label class="form-label" for="form2Example2">Password</label>
22     </div>
23
24     <!-- 2 column grid layout for inline styling -->
```

A diagram illustrating code indentation. A horizontal red double-headed arrow is positioned next to the opening <form> tag on line 11. A vertical red line with a bracket on its left side spans from line 12 to line 22, indicating the scope of the nested form elements. A horizontal red double-headed arrow is positioned next to the first nested element, the email input div, on line 13.

Estrutura do projeto – index.php

Agora abra o arquivo no navegador. Digite na URL:

localhost/aula05

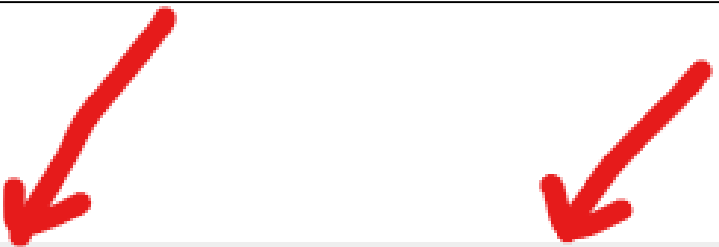


A screenshot of a web browser window with a yellow header. The address bar shows 'localhost/aula05/'. The page contains a login form with two input fields: 'Email address' and 'Password'. Below the 'Password' field is a checked checkbox labeled 'Remember me' and a blue link 'Forgot password?'. A blue 'Sign in' button is positioned to the left of the 'Remember me' checkbox. Below the 'Sign in' button, the text 'Not a member? Register' is displayed, with 'Register' as a blue link. At the bottom, the text 'or sign up with:' is followed by four gray, rounded rectangular buttons.

Estrutura do projeto – index.php

Agora precisamos configurar as propriedades **action** e **method** do formulário.

```
<body>
  <div class="container">
    <form action="processa.php" method="POST">
      <!-- Email input -->
      <div data-mdb-input-init class="form-outline mb-4">
        <input type="email" id="form2Example1" class="form
```




Estrutura do projeto – index.php

Precisamos, também, configurar os campos *input* e suas propriedades *name* do formulário.

```
<div class="container">
  <form action="processa.php" method="POST">
    <!-- Email input -->
    <div data-mdb-input-init class="form-outline mb-4">
      <input type="email" id="form2Example1" class="form-control" name="email" />
      <label class="form-label" for="form2Example1">Email address</label>
    </div>

    <!-- Password input -->
    <div data-mdb-input-init class="form-outline mb-4">
      <input type="password" id="form2Example2" class="form-control" name="senha" />
      <label class="form-label" for="form2Example2">Password</label>
    </div>

    <!-- 2 column grid layout for inline styling -->
    <div class="row mb-4">
      <div class="col d-flex justify-content-center">
```



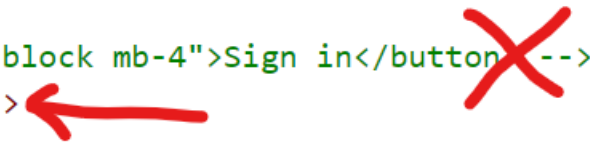
Estrutura do projeto – index.php

Neste template de formulário, vamos alterar o botão de submit, para uma tag de input.

```
<div class="col">
  <!-- Simple link -->
  <a href="#">Forgot password?</a>
</div>

<!-- Submit button -->
<!-- <button data-mdb-ripple-init type="button" class="btn btn-primary btn-block mb-4">Sign in</button> -->
<input type="submit" class="btn btn-primary btn-block mb-4" value="Sign in">

<!-- Register buttons -->
<div class="text-center">
  <p>Not a member? <a href="#">Register</a></p>
  <p>or sign up with:</p>
  <button data-mdb-ripple-init type="button" class="btn btn-secondary btn-floating mx-1">
    <i class="fab fa-facebook-f"></i>
```



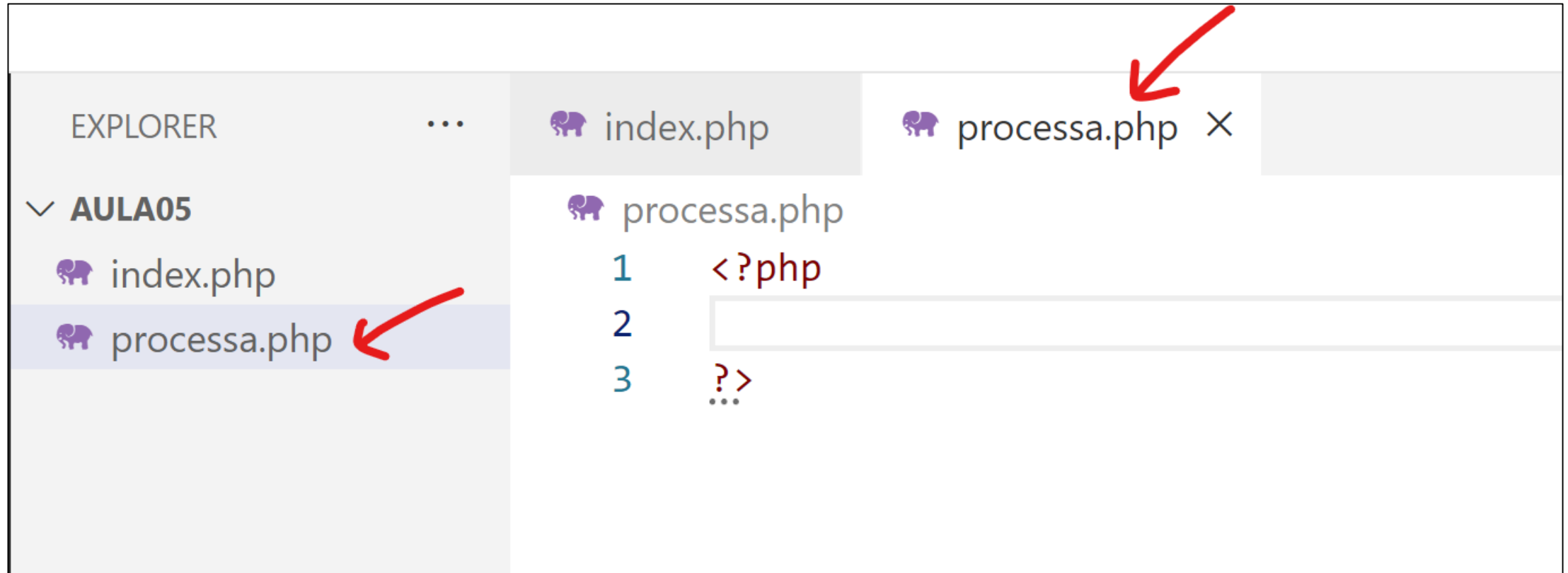
Estrutura do projeto – index.php

Neste template de formulário, há vários componentes que não vamos utilizar nesta aula. Caso queira remova-os para deixar mais limpa a tela (**OPCIONAL**).

Estrutura do projeto – processa.php

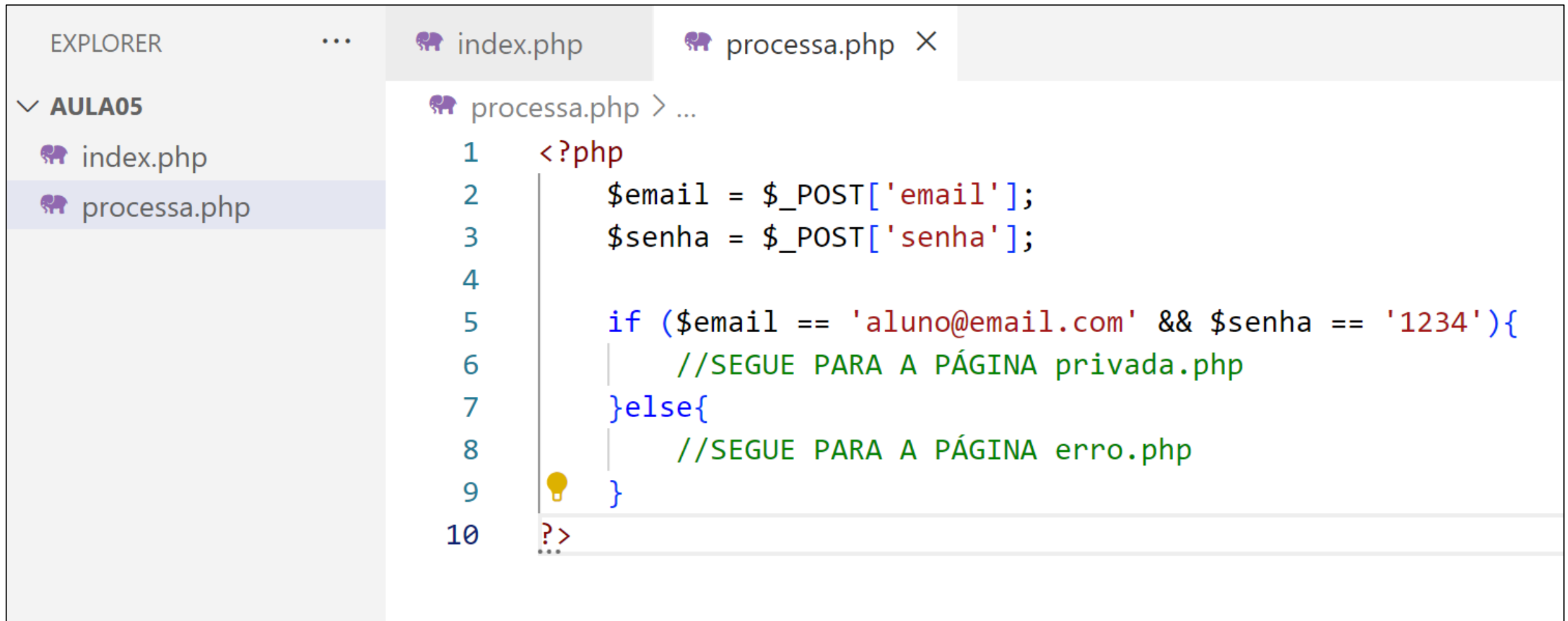
Agora vamos fazer a programação da página processa.php.

Crie o arquivo **processa.php**.



Estrutura do projeto – processa.php

Crie o arquivo **processa.php**.



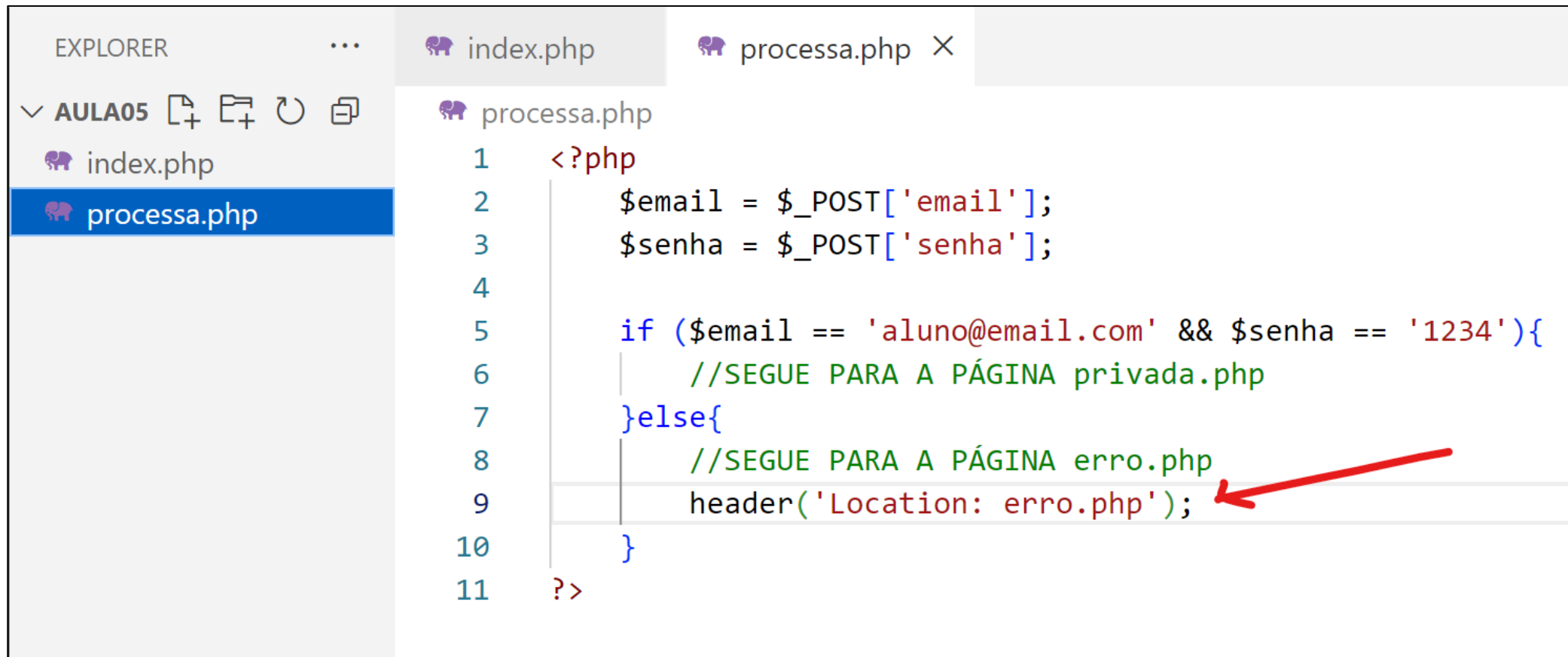
The screenshot shows a code editor with two tabs: 'index.php' and 'processa.php'. The 'EXPLORER' sidebar on the left shows a folder named 'AULA05' containing 'index.php' and 'processa.php'. The 'processa.php' tab is active, displaying the following PHP code:

```
1  <?php
2      $email = $_POST['email'];
3      $senha = $_POST['senha'];
4
5      if ($email == 'aluno@email.com' && $senha == '1234'){
6          //SEGUE PARA A PÁGINA privada.php
7      }else{
8          //SEGUE PARA A PÁGINA erro.php
9      }
10  ?>
```


Estrutura do projeto – processa.php

Encaminhamento para a página **privada.php** e **erro.php**.

O encaminhamento para a página **erro.php** é bem simples:

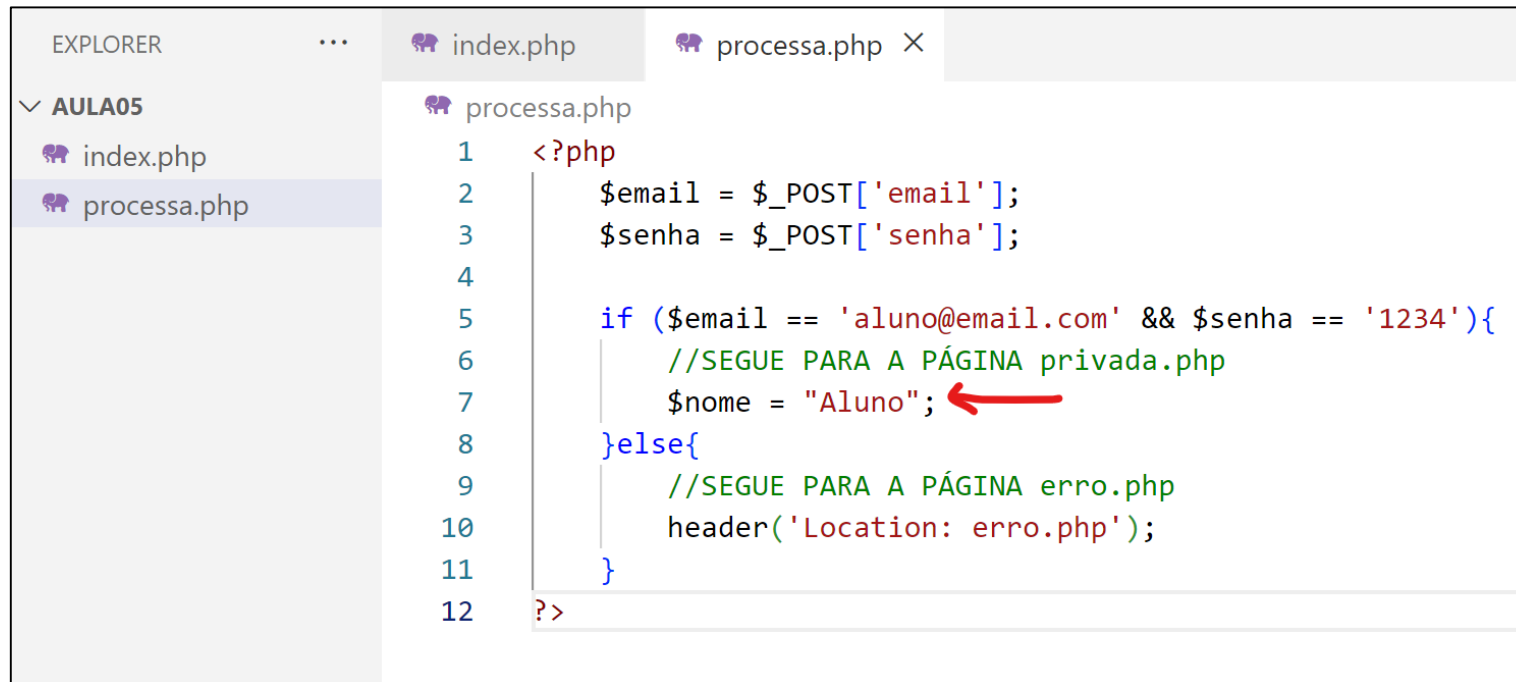


The screenshot shows a code editor with two tabs: 'index.php' and 'processa.php'. The 'EXPLORER' sidebar on the left shows the project structure with 'AULA05' as the root, containing 'index.php' and 'processa.php'. The 'processa.php' file is selected and its code is displayed in the main editor area. The code is a PHP script that checks if the email is 'aluno@email.com' and the password is '1234'. If both conditions are met, it redirects to 'privada.php'. Otherwise, it redirects to 'erro.php' using the 'header()' function. A red arrow points to the 'header()' function call on line 9.

```
1  <?php
2      $email = $_POST['email'];
3      $senha = $_POST['senha'];
4
5      if ($email == 'aluno@email.com' && $senha == '1234'){
6          //SEGUE PARA A PÁGINA privada.php
7      }else{
8          //SEGUE PARA A PÁGINA erro.php
9          header('Location: erro.php');
10     }
11  ?>
```

Estrutura do projeto – processa.php

Já no encaminhamento para a página **privada.php** vamos enviar por GET (via URL) um nome do usuário. Nesta página temos somente a informação de email e senha do usuário. Então vamos criar um nome fictício para o nosso usuário caso o login seja realizado com sucesso.



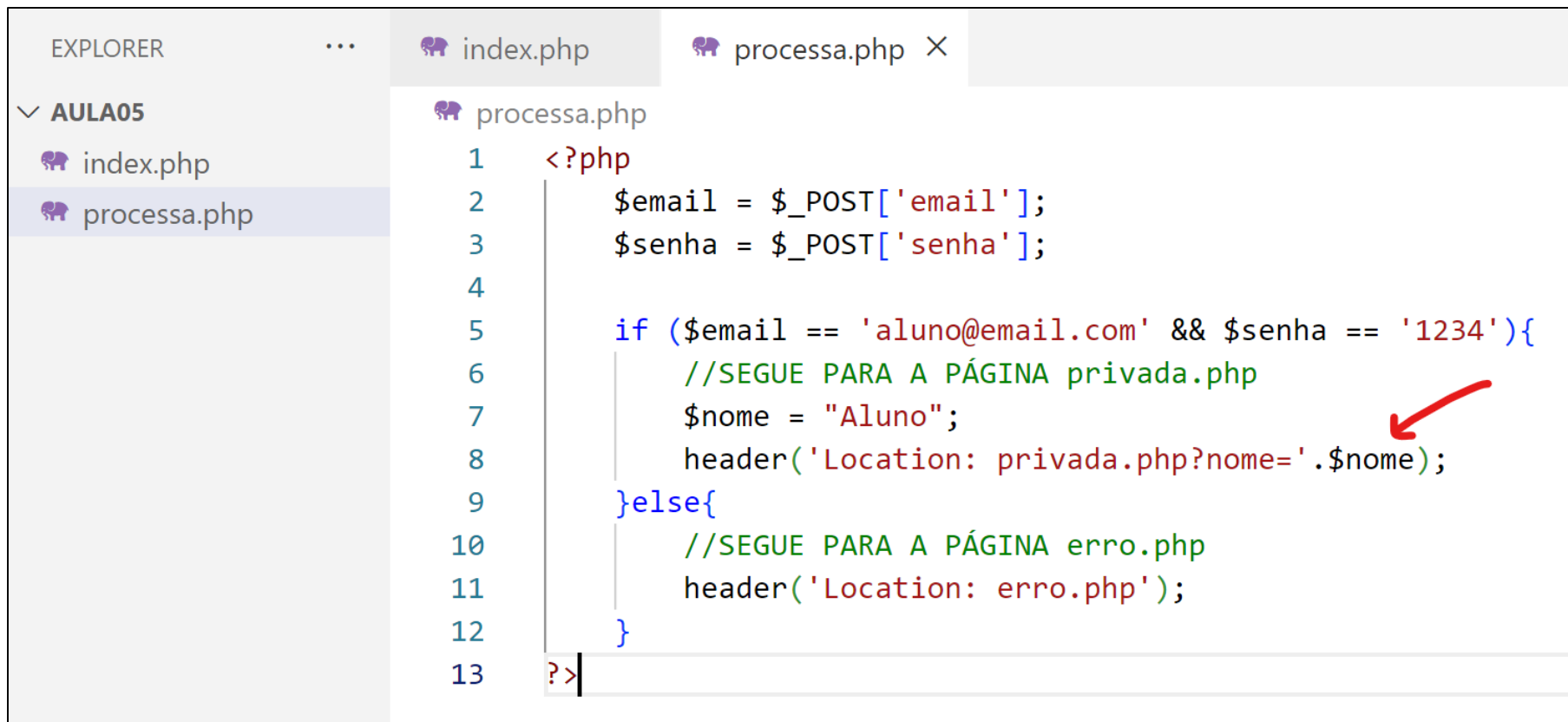
The screenshot shows a code editor with two tabs: 'index.php' and 'processa.php'. The 'EXPLORER' sidebar on the left shows a folder named 'AULA05' containing 'index.php' and 'processa.php'. The 'processa.php' file is open in the main editor area, displaying the following PHP code:

```
1  <?php
2      $email = $_POST['email'];
3      $senha = $_POST['senha'];
4
5      if ($email == 'aluno@email.com' && $senha == '1234'){
6          //SEGUE PARA A PÁGINA privada.php
7          $nome = "Aluno";
8      }else{
9          //SEGUE PARA A PÁGINA erro.php
10         header('Location: erro.php');
11     }
12  ?>
```

A red arrow points to the assignment of the variable `$nome` on line 7.

Estrutura do projeto – processa.php

Agora vamos encaminhar para a página **privada.php** o nome do usuário via GET pela URL.



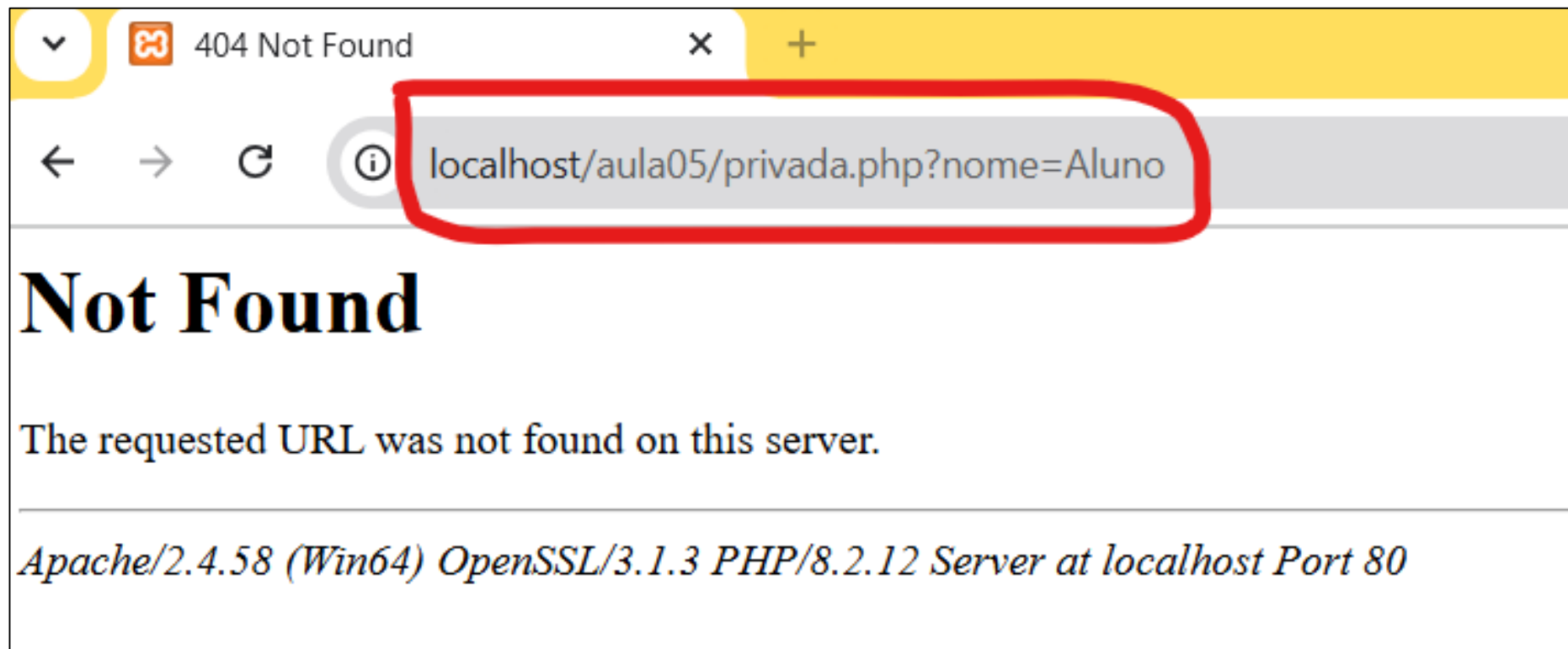
The screenshot shows a code editor with two tabs: 'index.php' and 'processa.php'. The 'EXPLORER' sidebar on the left shows a folder named 'AULA05' containing 'index.php' and 'processa.php'. The 'processa.php' file is open in the main editor area. The code in the file is as follows:

```
1  <?php
2      $email = $_POST['email'];
3      $senha = $_POST['senha'];
4
5      if ($email == 'aluno@email.com' && $senha == '1234'){
6          //SEGUE PARA A PÁGINA privada.php
7          $nome = "Aluno";
8          header('Location: privada.php?nome='.$nome);
9      }else{
10         //SEGUE PARA A PÁGINA erro.php
11         header('Location: erro.php');
12     }
13  ?>
```

A red arrow points to the `header('Location: privada.php?nome='.$nome);` line in the code.

Estrutura do projeto – processa.php

Faça um teste iniciando pela página **index.php**. Insira as informações corretas (email: aluno@email.com e senha: **1234**). Clique em *Sign In* e verifique a *url* da requisição feita pela página **processa.php**.



Estrutura do projeto – processa.php

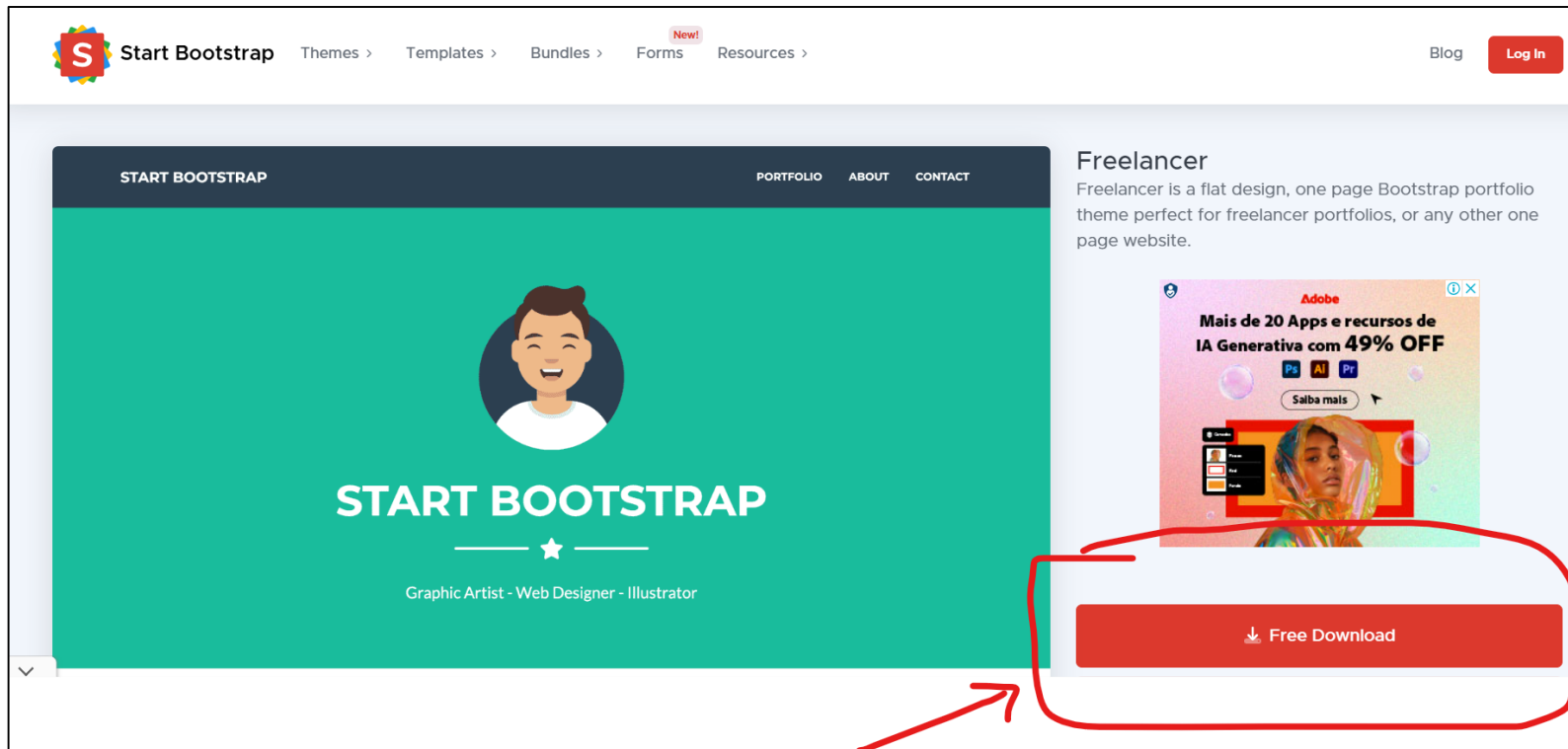
Neste ponto estamos enviando via *get* uma informação para a página **privada.php** que está contida dentro de um parâmetro chamado nome.

Agora vamos criar a página **privada.php**.

Estrutura do projeto – privada.php

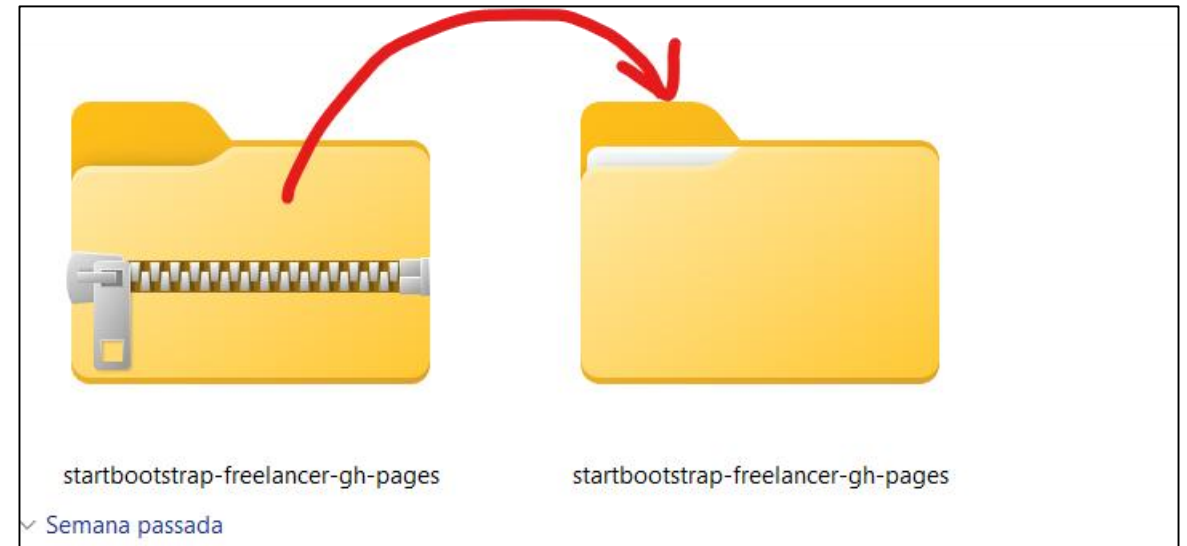
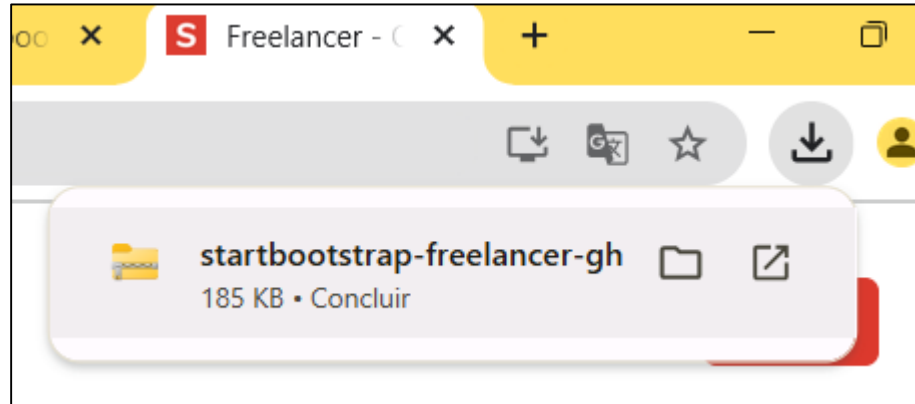
Nesta página podemos utilizar algum template de exibição html e css. Vamos utilizar o seguinte template:

<https://startbootstrap.com/theme/freelancer>



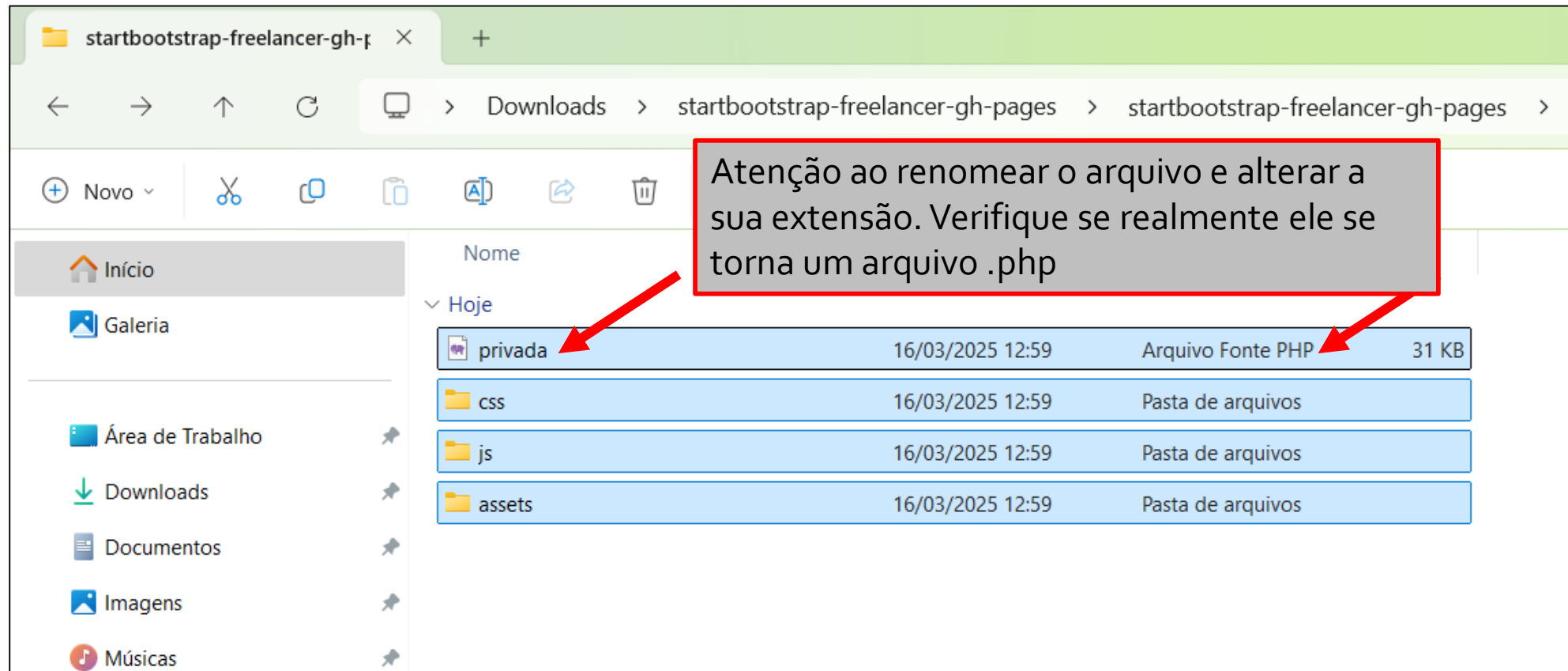
Estrutura do projeto – privada.php

Extraia os arquivos baixados.



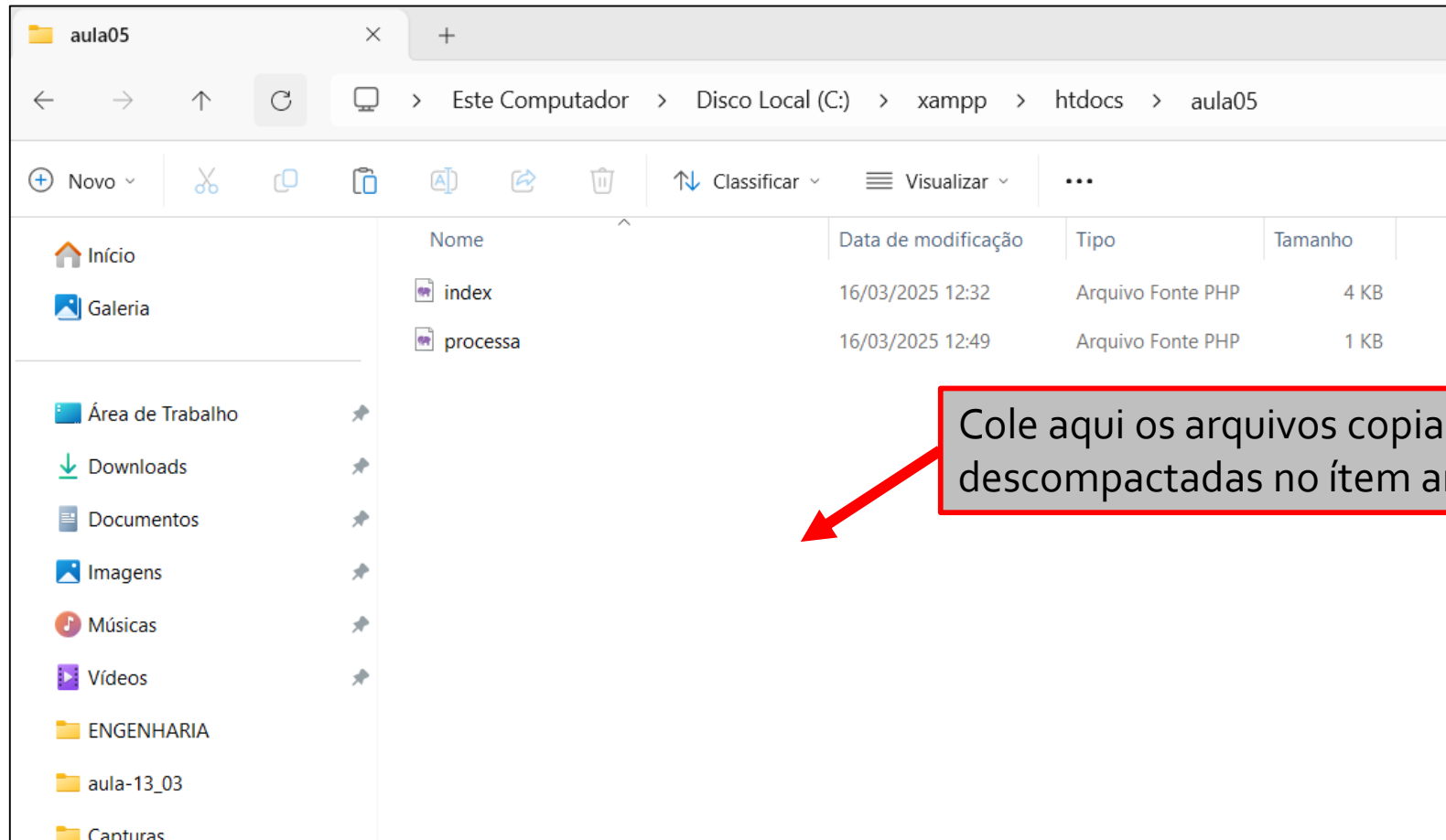
Estrutura do projeto – privada.php

Abra a pasta com os arquivos extraídos **altere o nome do arquivo index.html** para **privada.php**. **Copie todos os arquivos.**



Estrutura do projeto – privada.php

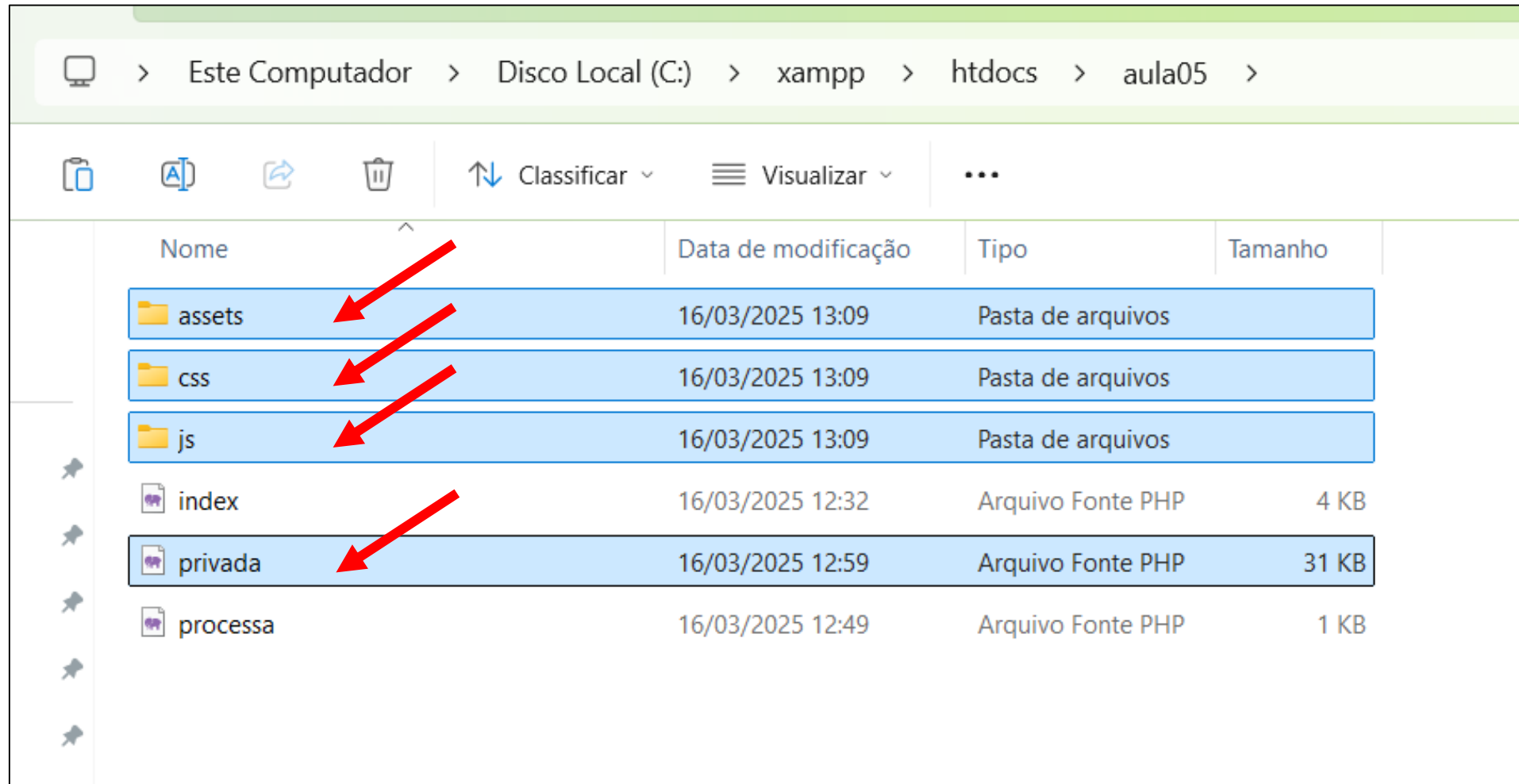
Abra a pasta aula05 e cole os arquivos.



Cole aqui os arquivos copiados que foram descompactados no ítem anterior

Estrutura do projeto – privada.php

Abra a pasta aula05 e cole os arquivos.



Estrutura do projeto – privada.php

Edite o início da página **privada.php** para que ela recupere da *url* o nome do usuário enviado da página **processa.php**.

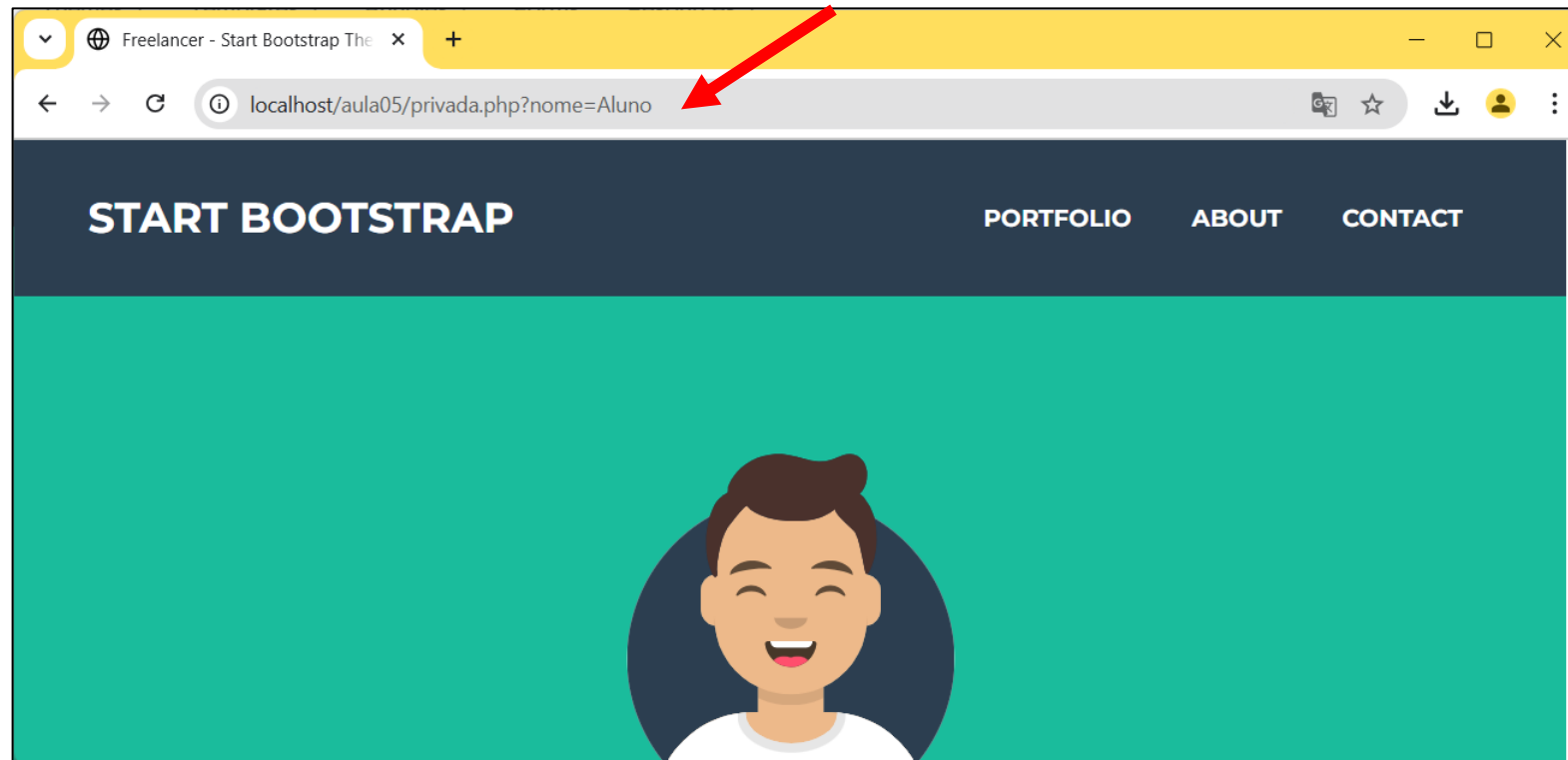


```
processa.php  privada.php X
privada.php
1  <?php
2  |      $nome = $_GET['nome'];
3  ?>
4
5  <!DOCTYPE html>
6  <html lang="en">
7  |     <head>
8  |         <meta charset="utf-8" />
9  |         <meta name="viewport" content="width=device-width, initial-scale=1">
10 |         <meta name="description" content="" />
11 |         <meta name="author" content="" />
12 |         <title>Freelancer - Start Bootstrap Themes</title>
```

Estrutura do projeto – privada.php

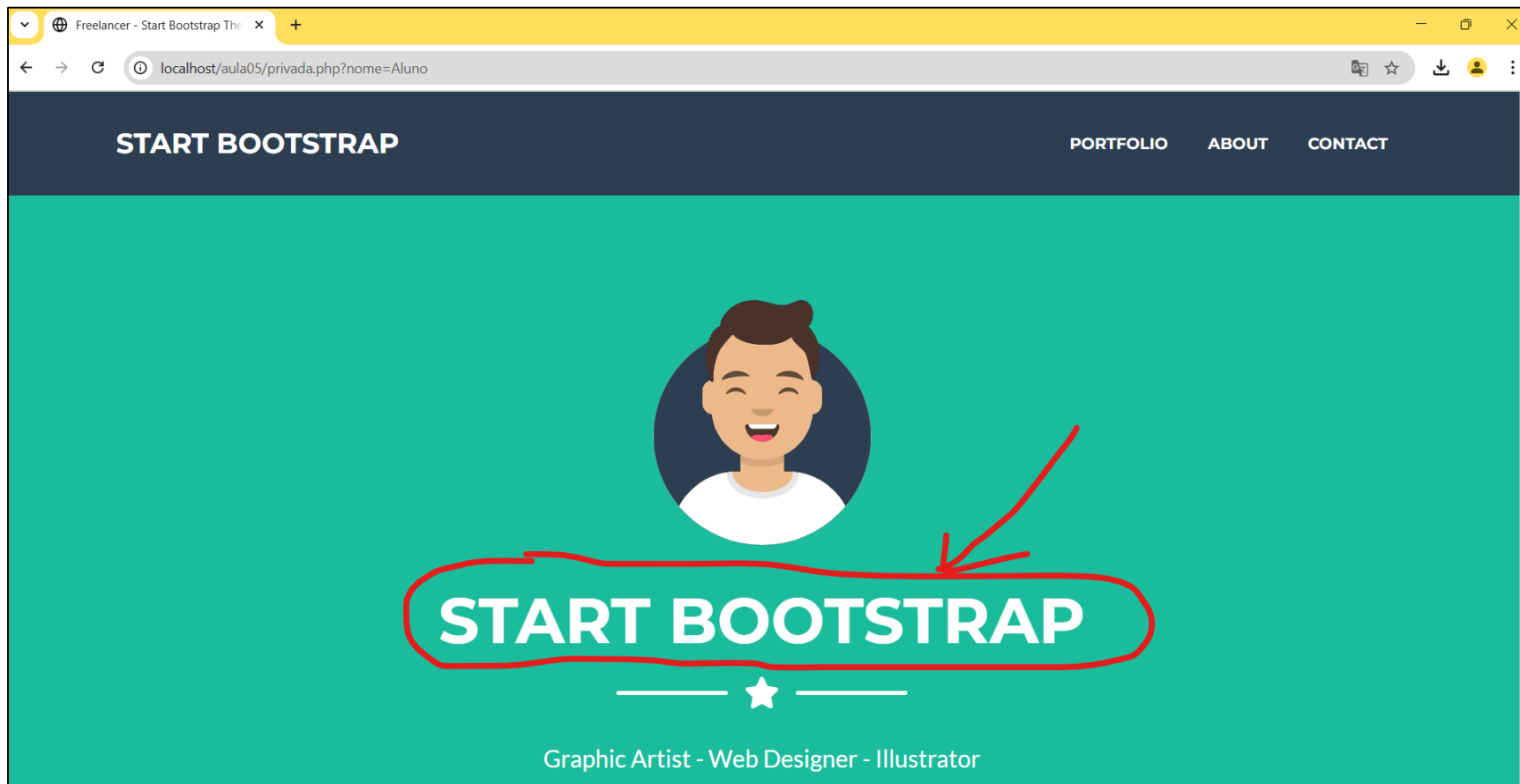
Vamos fazer um novo teste a partir da página **index.php**.

OBS.: SEMPRE QUE FOR INICIAR UM NOVO TESTE, FAÇA UM REFRESH EM SUA PÁGINA (F5).



Estrutura do projeto – privada.php

Vamos escolher um local para que o nome do usuário, que agora está na variável `$nome`, seja exibido nesta página.



Estrutura do projeto – privada.php

Vamos fazer com que apareça: **BEM VINDO FULANO....**, onde FULANO será substituído pelo conteúdo da variável `$nome`.

```
<header class="masthead bg-primary text-white text-center">
  <div class="container d-flex align-items-center flex-column">
    <!-- Masthead Avatar Image-->
    
    <!-- Masthead Heading-->
    <!-- <h1 class="masthead-heading text-uppercase mb-0">Start Bootstrap</h1> -->
    <h1 class="masthead-heading text-uppercase mb-0">BEM VINDO FULANO</h1>
    <!-- Icon Divider-->
    <div class="divider-custom divider-light">
      <div class="divider-custom-line"></div>
```

Estrutura do projeto – privada.php

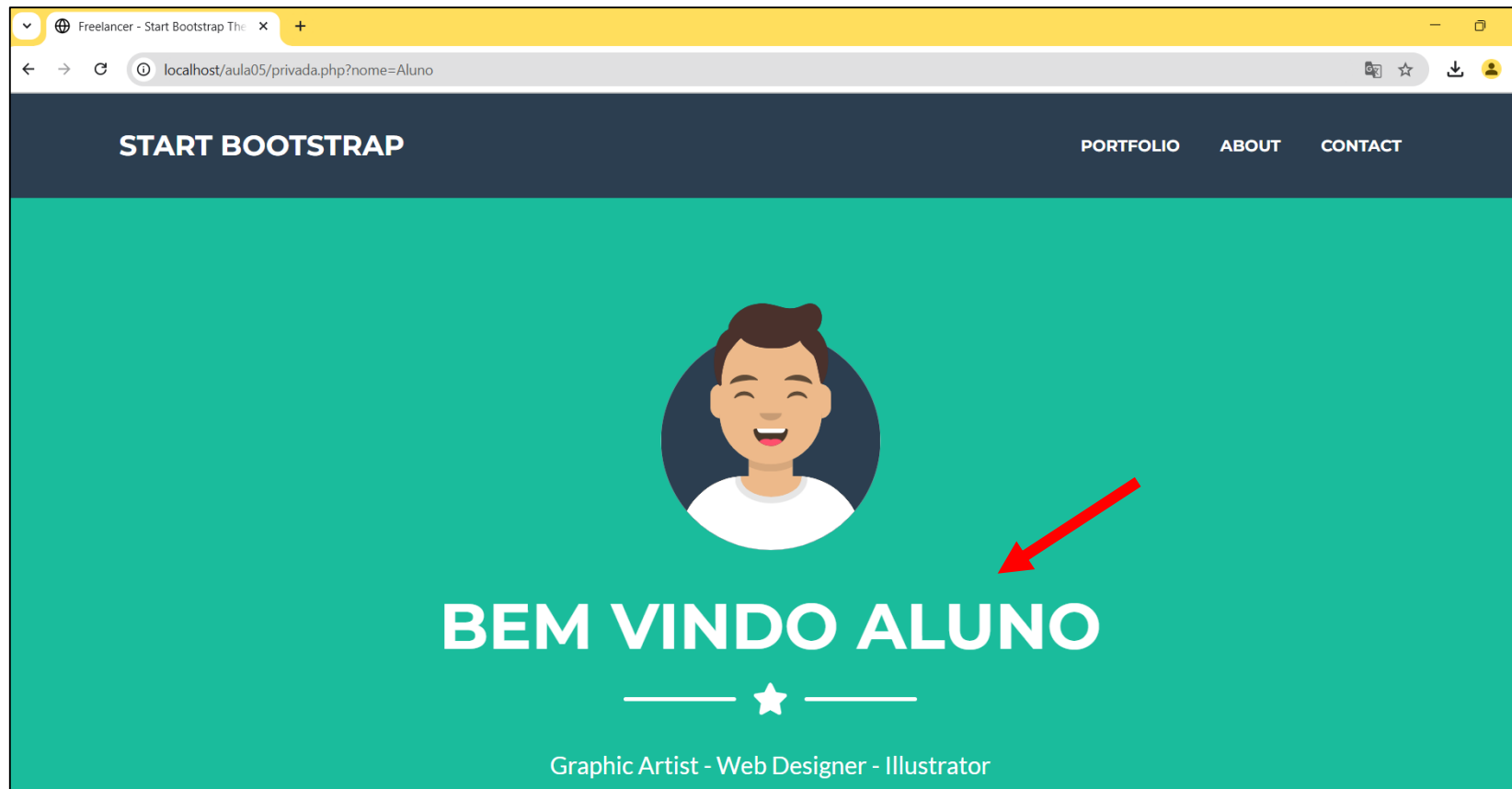
Vamos fazer com que apareça: **BEM VINDO FULANO....**, onde FULANO será substituído pelo conteúdo da variável **\$nome**.

```

<!-- Masthead Heading-->
<!-- <h1 class="masthead-heading text-uppercase mb-0">Start Bootstrap</h1> -->
<h1 class="masthead-heading text-uppercase mb-0">BEM VINDO <?php echo $nome; ?> </h1>
<!-- Icon Divider-->
<div class="divider-custom divider-light">
  <div class="divider-custom-line"></div>
  <div class="divider-custom-icon"><i class="fas fa-star"></i></div>
  <div class="divider-custom-line"></div>
```

Estrutura do projeto – privada.php

Testando novamente, agora teremos o nome aparecendo conforme o script que está na página **processa.php**.



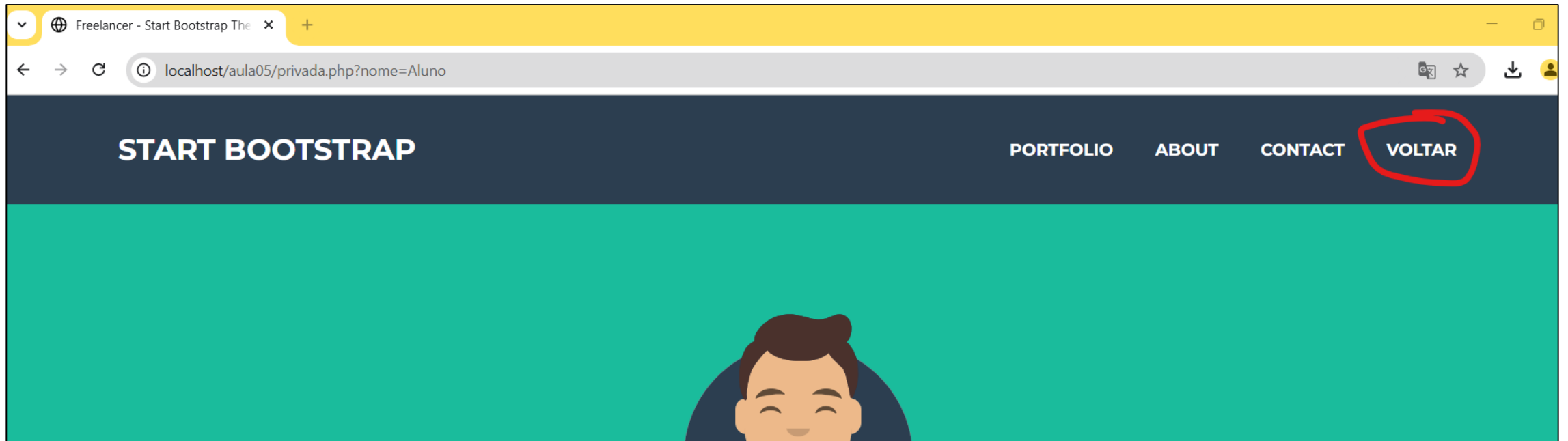
Estrutura do projeto – privada.php

Agora vamos criar um botão para voltar para a página **index.php**. Vamos criar este botão no menu por exemplo.

```
<body id="page-top">
  <!-- Navigation-->
  <nav class="navbar navbar-expand-lg bg-secondary text-uppercase fixed-top" id="mainNav">
    <div class="container">
      <a class="navbar-brand" href="#page-top">Start Bootstrap</a>
      <button class="navbar-toggler text-uppercase font-weight-bold bg-primary text-white rounded" type="button" data-bs-toggle="collaps
        Menu
        <i class="fas fa-bars"></i>
      </button>
      <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#portfolio">Portfolio</a></li>
          <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#about">About</a></li>
          <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="#contact">Contact</a></li>
          <li class="nav-item mx-0 mx-lg-1"><a class="nav-link py-3 px-0 px-lg-3 rounded" href="index.php">Voltar</a></li>
        </ul>
      </div>
    </div>
  </nav>
```



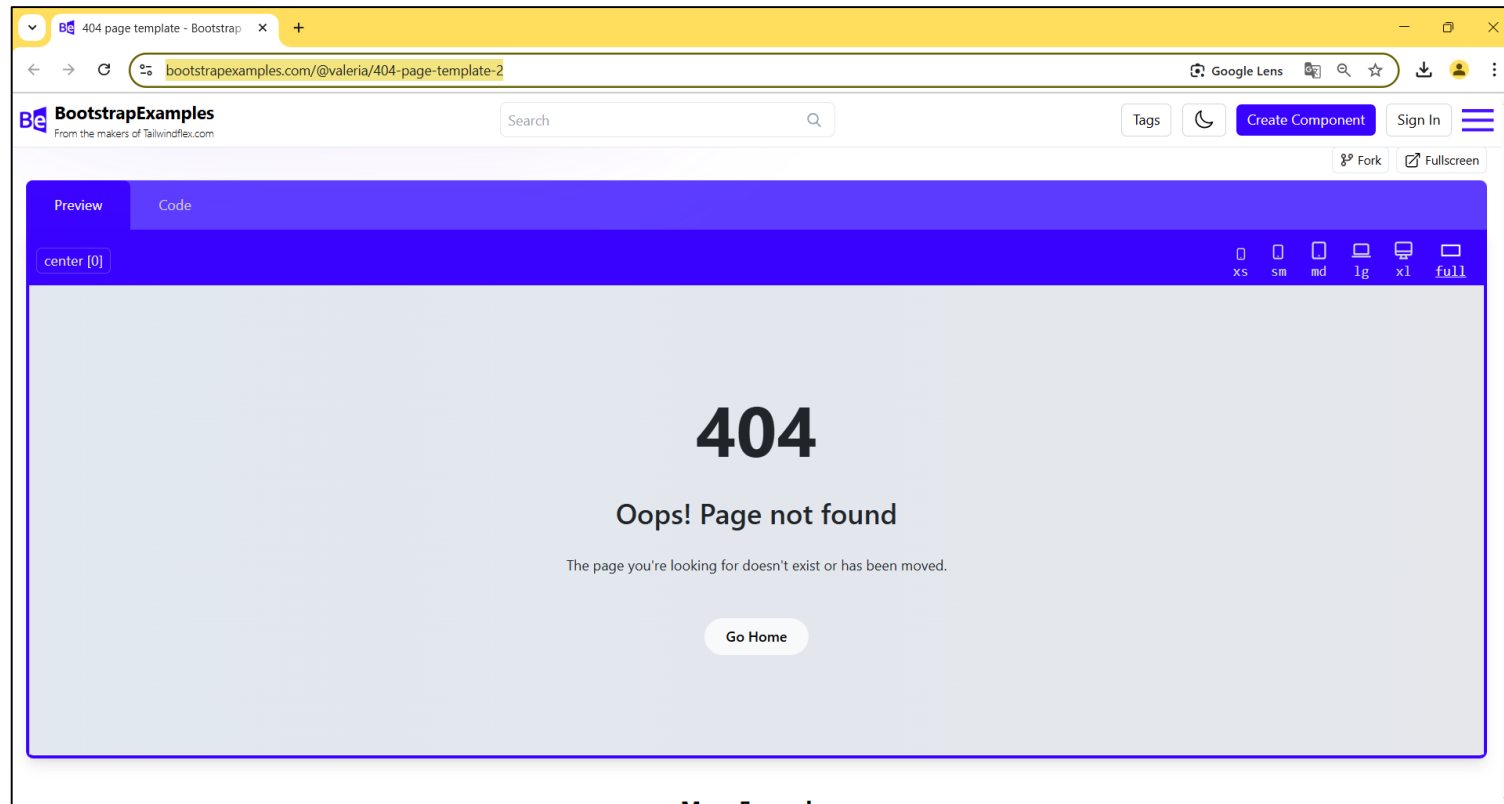
Estrutura do projeto – privada.php



Estrutura do projeto – erro.php

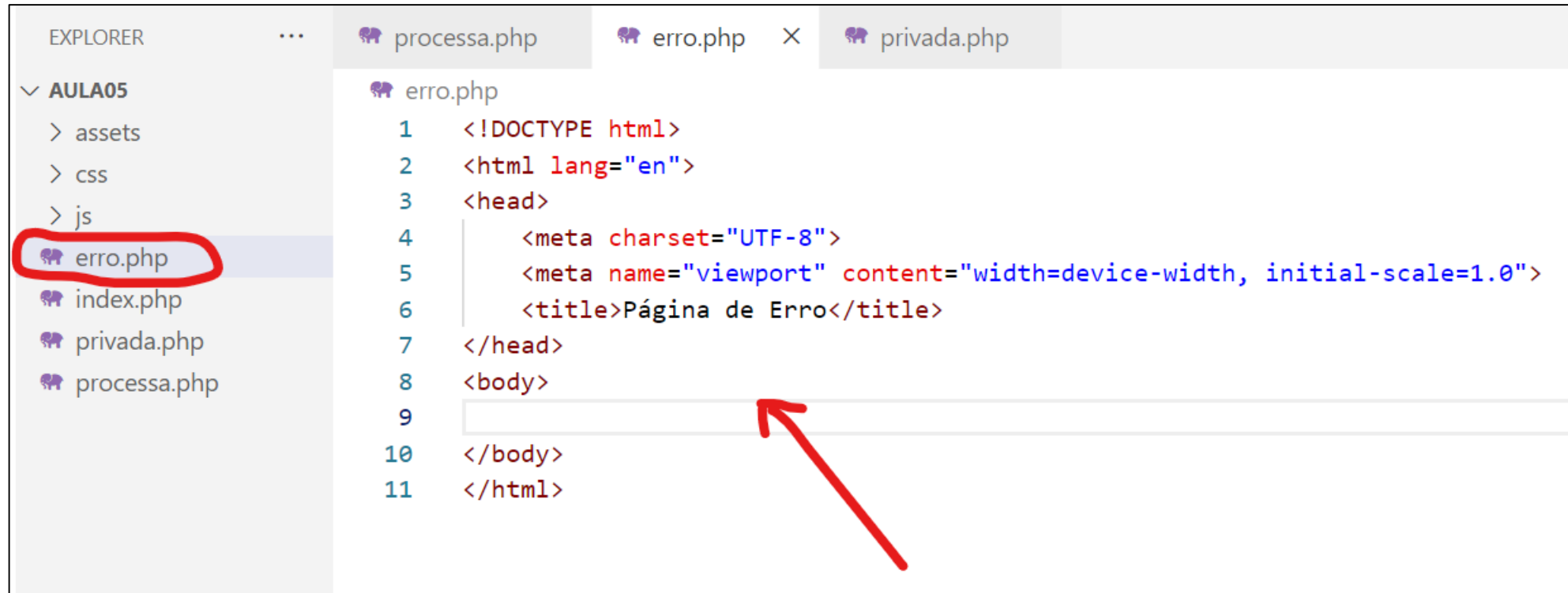
Assim como fizemos para a página **privada.php**, precisamos também de um template para a página **erro.php**. Vamos utilizar por exemplo:

<https://bootstrapexamples.com/@valeria/404-page-template-2>



Estrutura do projeto – erro.php

Veja que este template possui uma aba de código. Há o código *HTML* e o *CSS* que são necessários para esta página funcionar corretamente. Crie a página ***erro.php***.



Estrutura do projeto – erro.php

Insira o *cdn* do **bootstrap 5**, também, nesta página.

erro.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Página de Erro</title>
7      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
8  </head>
9  <body>
10
11 </body>
12 </html>
```



Estrutura do projeto – erro.php

Crie uma div com a classe container.

erro.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Página de Erro</title>
7      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
8  </head>
9  <body>
10     <div class="container">
11
12     </div>
13 </body>
14 </html>
```



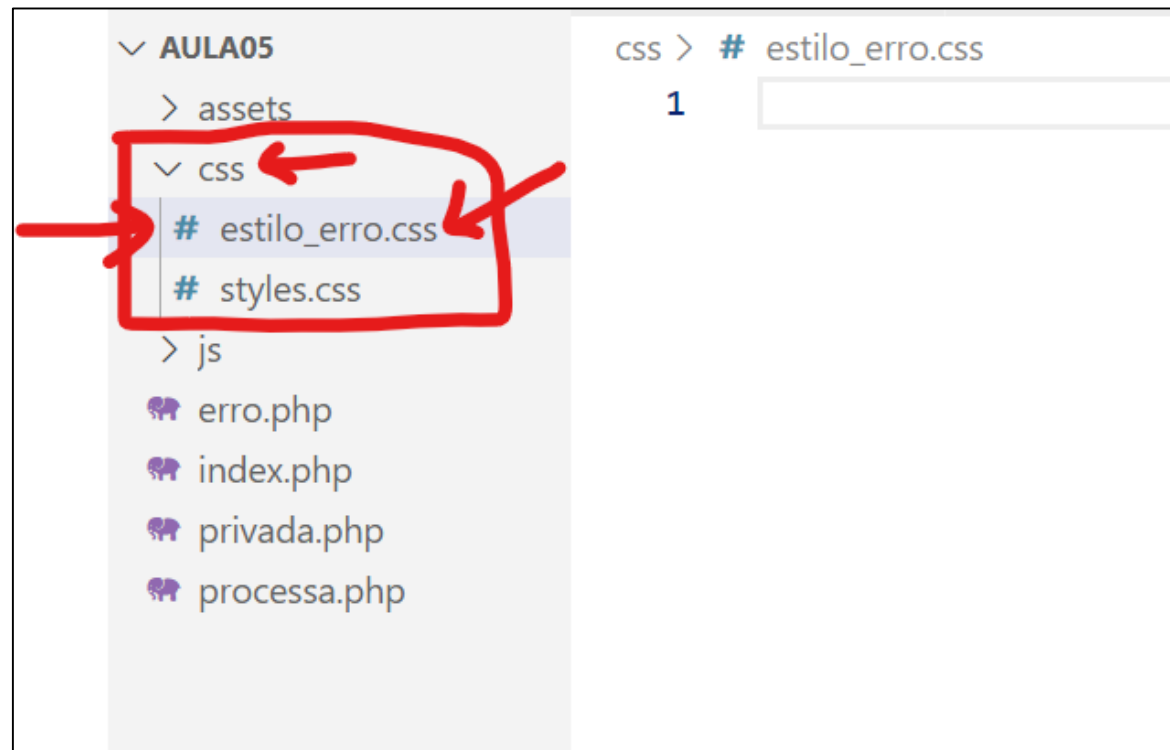
Estrutura do projeto – erro.php

Cole o código html do template da página de erro dentro da div criada.

```
erro.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Página de Erro</title>
7      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
8  </head>
9  <body>
10     <div class="container">
11         <div class="custom-bg text-dark">
12             <div class="d-flex align-items-center justify-content-center min-vh-100 px-2">
13                 <div class="text-center">
14                     <h1 class="display-1 fw-bold">404</h1>
15                     <p class="fs-2 fw-medium mt-4">Oops! Page not found</p>
16                     <p class="mt-4 mb-5">The page you're looking for doesn't exist or has been moved.</p>
17                     <a href="/" class="btn btn-light fw-semibold rounded-pill px-4 py-2 custom-btn">
18                         Go Home
19                     </a>
20                 </div>
21             </div>
22         </div>
23     </div>
24 </body>
25 </html>
```

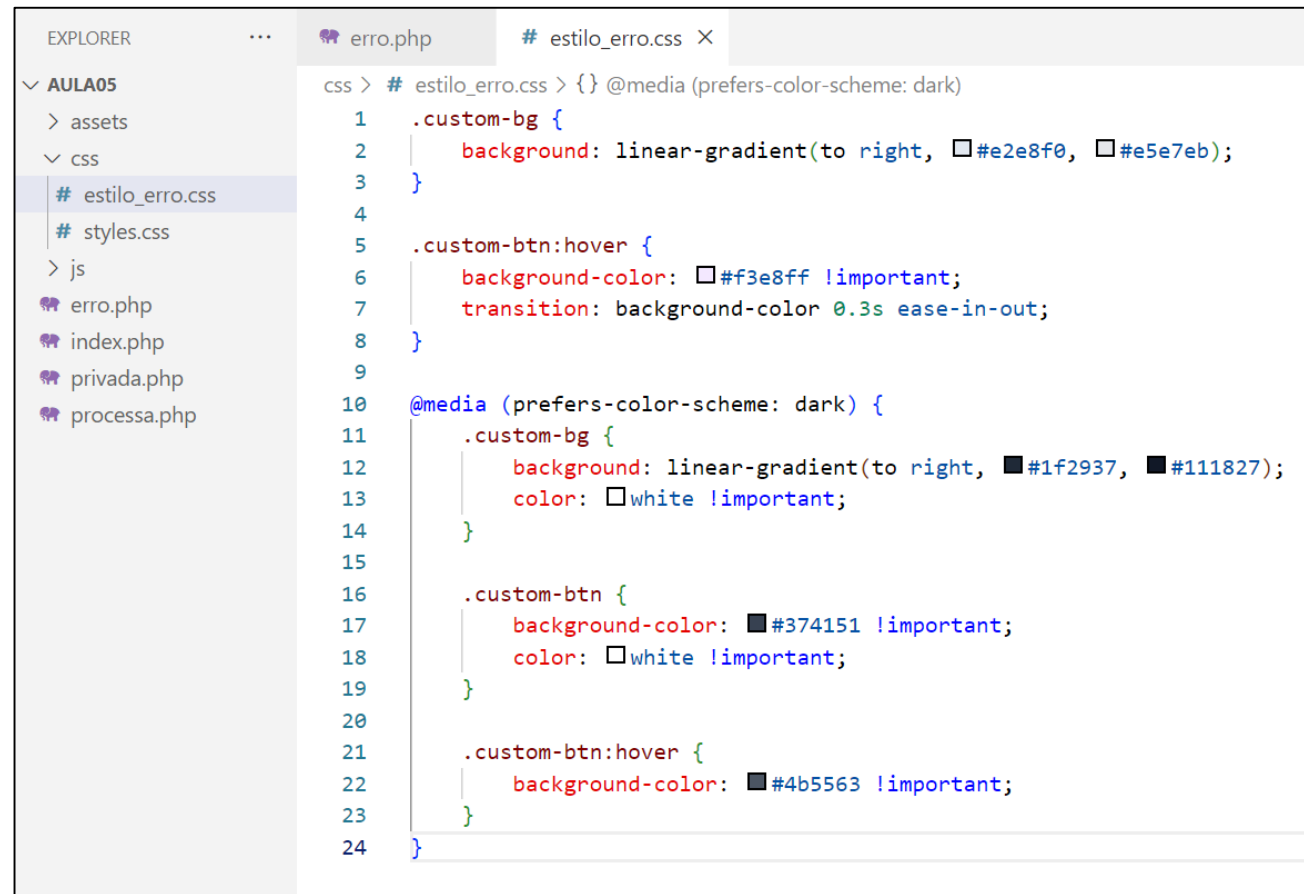
Estrutura do projeto – erro.php

Na estrutura do projeto aula05, já possuímos uma pasta chamada `css`. Nesta pasta crie um novo arquivo chamado **estilo_erro.css** que será utilizado pela página **erro.php**.



Estrutura do projeto – erro.php

No arquivo **estilo_erro.css** cole o código css do template que utilizamos para a página **erro.php**.

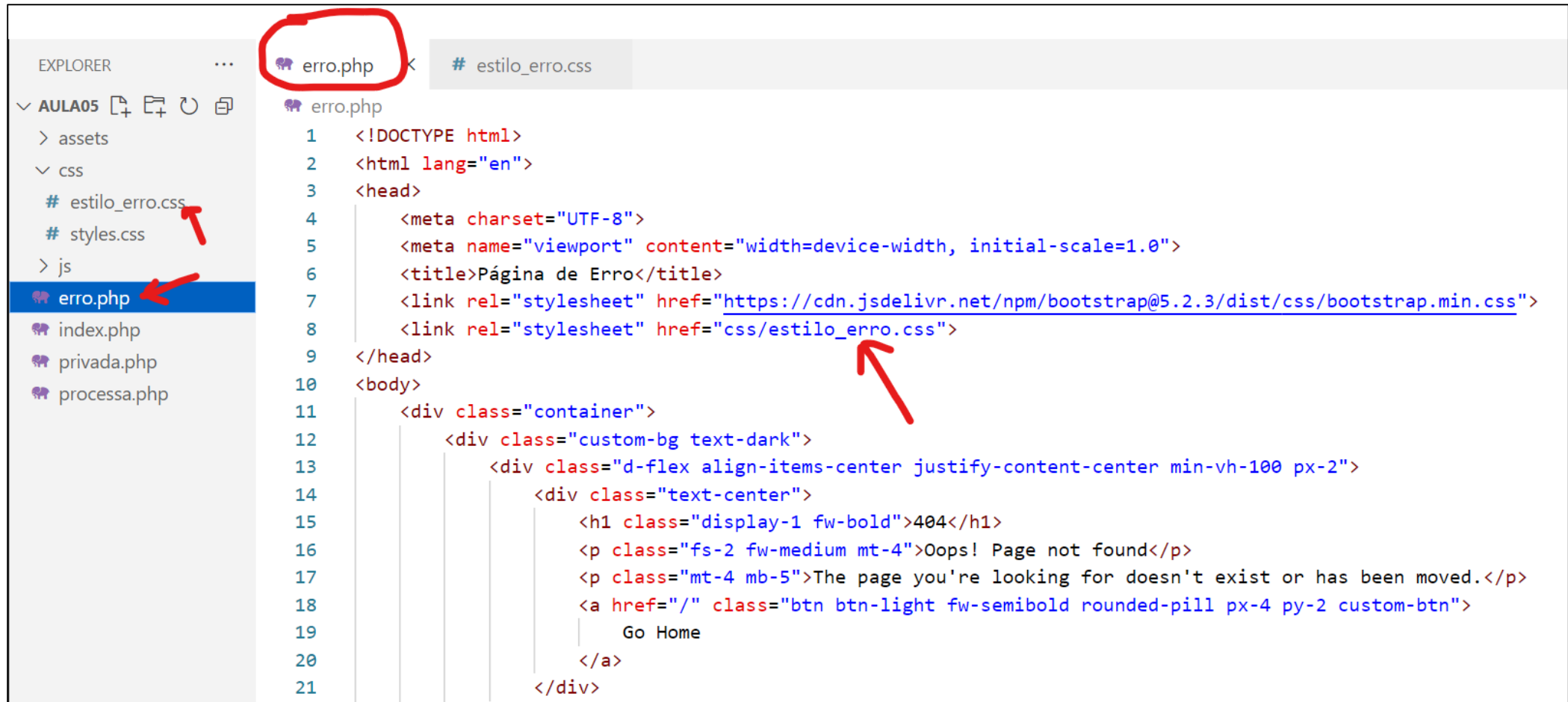


The screenshot shows a code editor with two panels. The left panel, titled 'EXPLORER', displays the project structure under 'AULA05'. It includes folders 'assets' and 'css', and files 'erro.php', 'index.php', 'privada.php', and 'processa.php'. The 'css' folder is expanded, showing 'estilo_erro.css' and 'styles.css'. The right panel shows the content of 'estilo_erro.css' with the following CSS code:

```
css > # estilo_erro.css > {} @media (prefers-color-scheme: dark)
1  .custom-bg {
2      background: linear-gradient(to right, #e2e8f0, #e5e7eb);
3  }
4
5  .custom-btn:hover {
6      background-color: #f3e8ff !important;
7      transition: background-color 0.3s ease-in-out;
8  }
9
10 @media (prefers-color-scheme: dark) {
11     .custom-bg {
12         background: linear-gradient(to right, #1f2937, #111827);
13         color: white !important;
14     }
15
16     .custom-btn {
17         background-color: #374151 !important;
18         color: white !important;
19     }
20
21     .custom-btn:hover {
22         background-color: #4b5563 !important;
23     }
24 }
```

Estrutura do projeto – erro.php

No arquivo **erro.php** precisamos criar a referência para o arquivo **estilo_erro.css**.

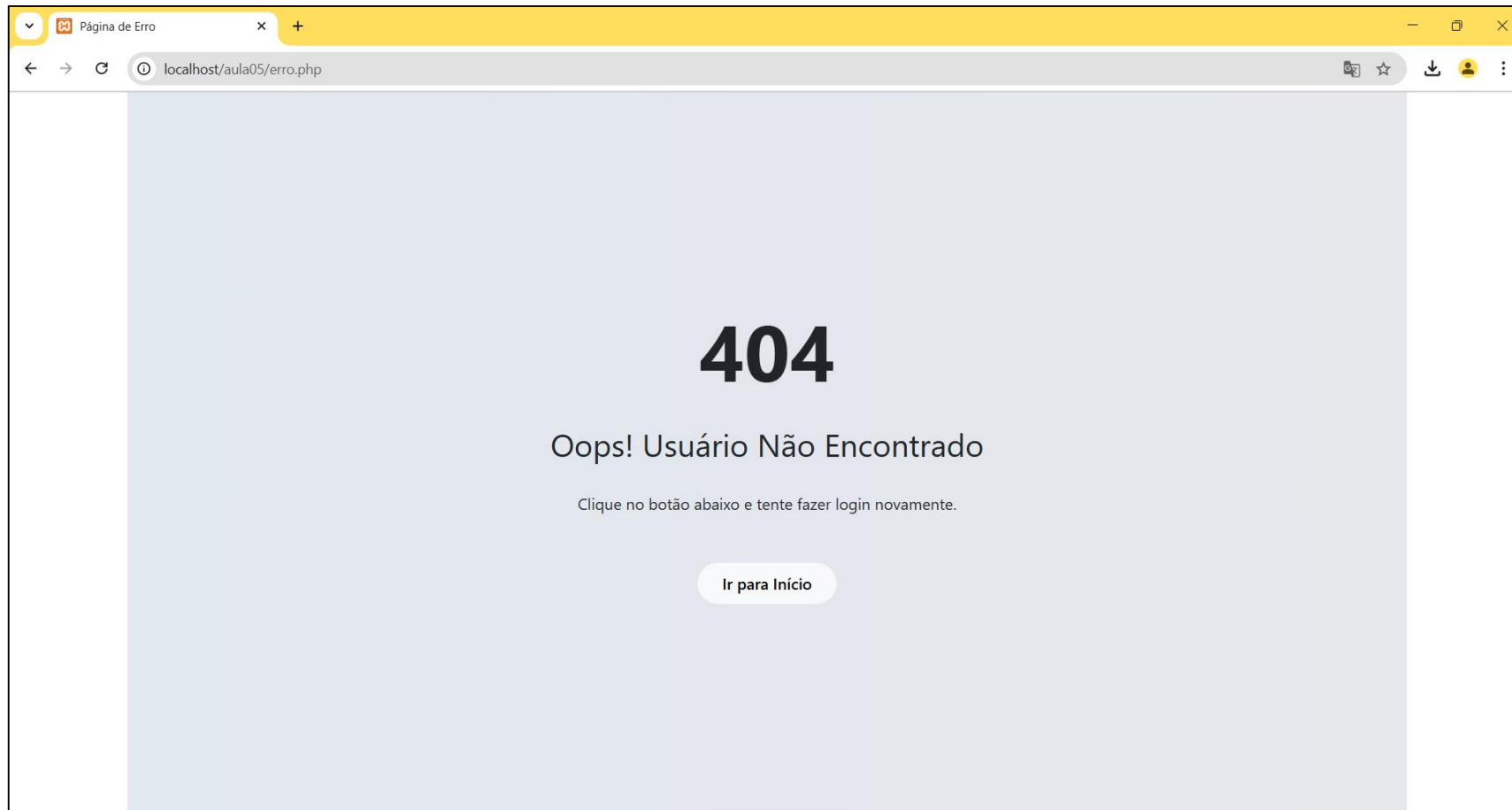


The screenshot shows a code editor with two panels. The left panel, titled 'EXPLORER', displays a file tree for a project named 'AULA05'. The files listed are 'assets', 'css', 'js', 'erro.php', 'index.php', 'privada.php', and 'processa.php'. The 'css' folder is expanded, showing 'estilo_erro.css' and 'styles.css'. Red arrows point from 'estilo_erro.css' and 'erro.php' in the file tree to their respective locations in the code. The right panel shows the content of 'erro.php', which is an HTML file. A red circle highlights the tab 'erro.php' at the top. A red arrow points from the 'estilo_erro.css' file in the file tree to the `href="css/estilo_erro.css"` attribute in the `<link>` tag on line 8 of the code.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Página de Erro</title>
7   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
8   <link rel="stylesheet" href="css/estilo_erro.css">
9 </head>
10 <body>
11   <div class="container">
12     <div class="custom-bg text-dark">
13       <div class="d-flex align-items-center justify-content-center min-vh-100 px-2">
14         <div class="text-center">
15           <h1 class="display-1 fw-bold">404</h1>
16           <p class="fs-2 fw-medium mt-4">Oops! Page not found</p>
17           <p class="mt-4 mb-5">The page you're looking for doesn't exist or has been moved.</p>
18           <a href="/" class="btn btn-light fw-semibold rounded-pill px-4 py-2 custom-btn">
19             Go Home
20           </a>
21         </div>
22       </div>
23     </div>
24   </div>
```

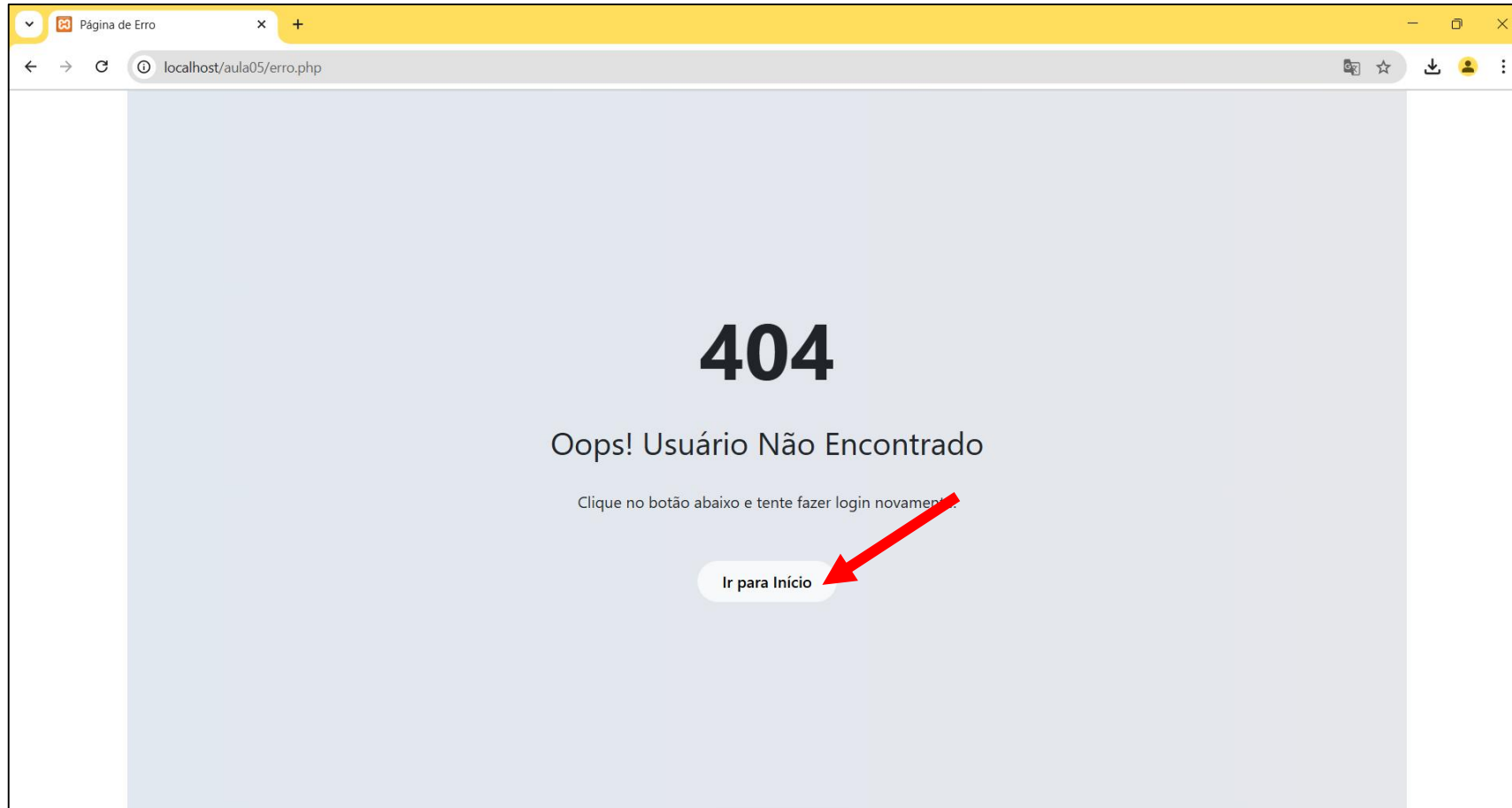
Estrutura do projeto – erro.php

Faça as edições necessárias no código html desta página.



Estrutura do projeto – erro.php

Vamos configurar o botão para voltar para a página **index.php**.



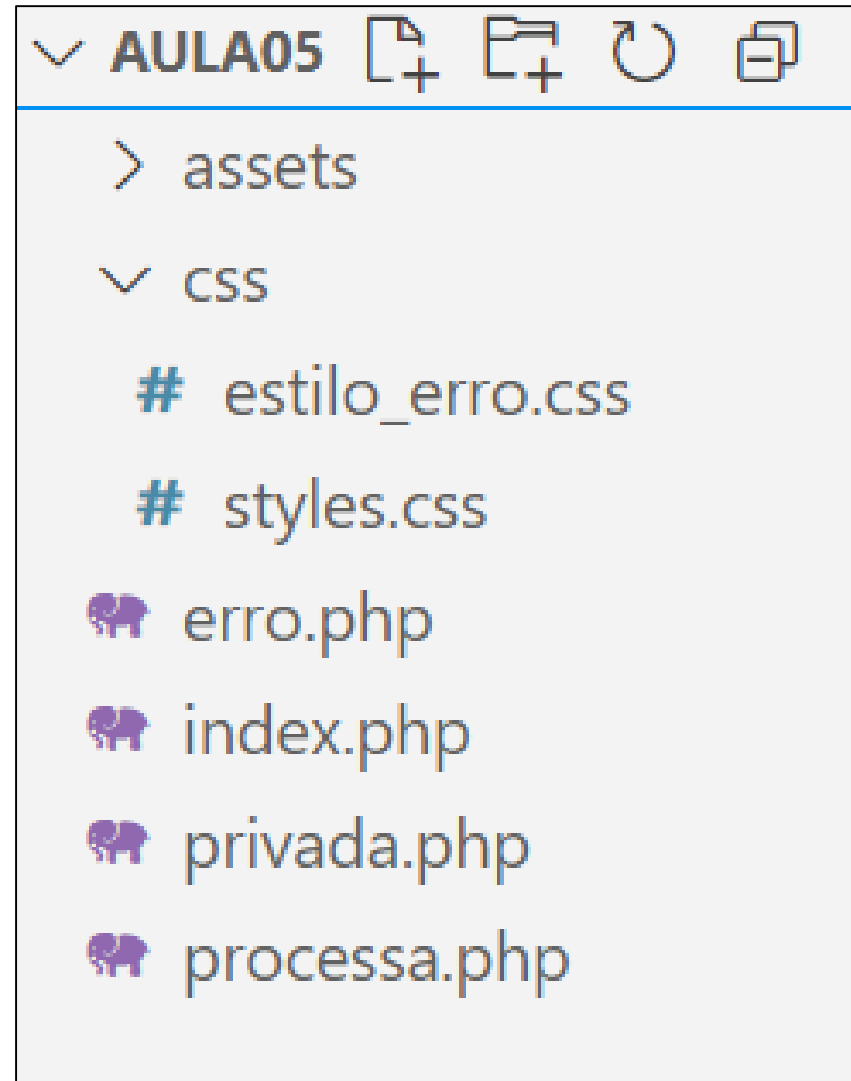
Estrutura do projeto – erro.php

Vamos configurar o botão para voltar para a página **index.php**.

```
erro.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Página de Erro</title>
7      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/boots
8      <link rel="stylesheet" href="css/estilo_erro.css">
9  </head>
10 <body>
11     <div class="container">
12         <div class="custom-bg text-dark">
13             <div class="d-flex align-items-center justify-content-center min-vh-100 px-2">
14                 <div class="text-center">
15                     <h1 class="display-1 fw-bold">404</h1>
16                     <p class="fs-2 fw-medium mt-4">Oops! Usuário Não Encontrado</p>
17                     <p class="mt-4 mb-5">Clique no botão abaixo e tente fazer login novament
18                     <a href="index.php" class="btn btn-light fw-semibold rounded-pill px-4 p
19                         Ir para Início
20                     </a>
21                 </div>
22             </div>
23         </div>
24     </div>
25 </body>
```



Estrutura final de arquivos do projeto



Teste todo o projeto desde o início.

Faça testes para login correto e para incorreto.

Demonstração de funcionamento

Página Inicial

localhost/aula05/index.php

PÁGINA INICIAL - PROJETO AULA 05

Endereço de Email

Senha

☒ Lembre-me [Esqueceu a senha?](#)

Não é membro? [Registro](#)

ou entre com:

Crie uma pasta no seu github e salve este projeto lá.

Faça todas as personalizações necessárias nos arquivos (**html, css, js e php**).

Para esta atividade é

OBRIGATÓRIA A PERSONALIZAÇÃO

DO FRONT-END.