



**INSTITUTO
FEDERAL**

Sudeste de
Minas Gerais

Campus
Manhuaçu

Desenvolvimento Web III

Controle de acesso

Prof. Leonardo C. R. Soares - leonardo.soares@ifsudestemg.edu.br

Instituto Federal do Sudeste de Minas Gerais

18 de dezembro de 2024





Controle de acesso

Na aula de hoje criaremos um sistema para controle de acesso. Nosso objetivo é criar uma área privada onde apenas usuários autenticados possam interagir. Inicialmente, cada usuário deverá fazer seu próprio cadastro.





Banco de dados

O banco de dados que utilizaremos pode ser criado pelos comandos abaixo:

```
CREATE DATABASE sis_controle;  
CREATE TABLE usuarios ( nome varchar(200) DEFAULT NULL,  
email varchar(200) NOT NULL, senha varchar(250) DEFAULT NULL,  
PRIMARY KEY ('email') );
```





Tela inicial (index.php)

Cadastro

Seu nome

nome

Seu e-mail

contato@larback.com.br

Sua senha

ex. 1234

Confirme a senha

ex. 1234

Cadastrar

Já tem conta?

[Ir para Login](#)





Tela inicial (index.php)

Login

Seu e-mail

ex. contato@larback.com.br

Senha

ex. senha

☐ Manter-me logado

Logar

Ainda não tem conta?

[Cadastre-se](#)





Front-End

Caso deseje utilizar o mesmo layout para os formulários de cadastro e login, o código HTML pode ser baixado aqui e o CSS aqui.





Classe Usuario

```
1  <?php
2      class Usuario{
3          private string $nome;
4          private string $email;
5          private string $senha;
6
7          public function __construct(string $nome = "", string $email = "",string $senha = ""){
8              $this->setNome($nome);
9              $this->setEmail($email);
10             $this->setSenha($senha);
11         }
```





Classe Usuario

```
12 public function salvar(): void{
13     if ($this->liberado()){
14         try{
15             $con = Conexao::conectar();
16             } catch(PDOException $p){
17                 throw new PDOException($p->getMessage());
18             }
19             $SQL = "insert into usuarios (nome,email,senha) values (?,?,md5(?))";
20             try{
21                 $pst = $con->prepare($SQL);
22                 $pst->bindValue(1,$this->getNome());
23                 $pst->bindValue(2,$this->getEmail());
24                 $pst->bindValue(3,$this->getSenha());
25                 $pst->execute();
26             }catch(PDOException $p){
27                 throw new PDOException("Ocorreu um erro ao tentar salvar os dados. <br>Verifique:
28                 <i>" . $p->getMessage(). "</i>");
29             }
30         } else {
31             throw new Exception("O email informado já foi cadastrado.");
32         }
33     }
```





Classe Usuario

```
33 public function liberado():bool{
34     try{
35         $con = Conexao::conectar();
36     } catch(PDOException $p){
37         throw new PDOException($p->getMessage());
38     }
39     $SQL = "select count(*) from usuarios where email=?";
40     try{
41         $pst = $con->prepare($SQL);
42         $pst->bindValue(1,$this->getEmail());
43         $pst->execute();
44         $dado = $pst->fetch(PDO::FETCH_NUM);
45         return $dado[0] == 0;
46     }catch(PDOException $p){
47         throw new PDOException("Ocorreu um erro ao tentar acessar a base de dados.
48         <br>Verifique: <i>" . $p->getMessage() . "</i>");
49     }
```





Classe Usuario

```
50 public static function confere(string $email,string $senha):?Usuario{
51     try{
52         $con = Conexao::conectar();
53     } catch(PDOException $p){
54         throw new PDOException($p->getMessage());
55     }
56     $SQL = "select * from usuarios where email=? and senha=md5(?)";
57     try{
58         $pst = $con->prepare($SQL);
59         $pst->bindParam(1,$email);
60         $pst->bindParam(2,$senha);
61         $pst->execute();
62         $dado = $pst->fetch(PDO::FETCH_OBJ);
63         if ($pst->rowCount()>0){
64             $usuario = new Usuario($dado->nome,$dado->email,$senha,$dado->nivel);
65             return $usuario;
66         }else{
67             return null;
68         }
69     }catch(PDOException $p){
70         throw new PDOException("Ocorreu um erro ao tentar acessar a base de dados.
71         <br>Verifique: <i>" . $p->getMessage(). "</i>");
72     }
73 }
```





Classe Usuario

```
72     public function getNome() :string{
73         return $this->nome;
74     }
75     public function getEmail() :string{
76         return $this->email;
77     }
78     public function getSenha() :string{
79         return $this->senha;
80     }
81
82     public function setSenha(string $senha) :void{
83         $this->senha = $senha;
84     }
85     public function setEmail(string $email) :void{
86         $this->email = $email;
87     }
88     public function setNome(string $nome) :void{
89         $this->nome = $nome;
90     }
91 }
92 ?>
```





Cadastrando usuários (cadUsuario.php)

```
1  <?php
2  require_once "autoload.php";
3  $usuario = new Usuario($_POST['nome_cad'], $_POST['email_cad'],
4  $_POST['senha_cad']);
5  try{
6      $usuario->salvar();
7      echo "<script>window.alert('Cadastro efetuado com sucesso.');"
8      window.location.href='index.php#paralogin';</script>";
9  } catch(PDOException $p){
10     echo "<div class='erro'>" . $p->getMessage() . "</div>";
11 } catch(Exception $e){
12     echo "<script>window.alert('". $e->getMessage() . "')";
13     window.location.href='index.php';</script>";
14 }
15 ?>
```





Fazendo Login (login.php)

```
1  <?php
2  session_start();
3  require_once "autoload.php";
4  try {
5      $usuario = Usuario::confere($_POST['email_login'],$_POST['senha_login']);
6      if (!is_null($usuario)){
7          $_SESSION['logado'] = true;
8          $_SESSION['usuario'] = serialize($usuario);
9          header('Location: principal.php');
10     } else {
11         $_SESSION['logado'] = false;
12         echo "<script>window.alert('Usuário ou senha incorretos.');
```





Controle de acesso

Para garantir que nenhum acesso não autorizado ocorra, todas as páginas restritas deverão conferir se as variáveis de sessão destinadas ao controle de acesso estão configuradas corretamente. Para facilitar, criaremos um arquivo (*session.php*) somente para isso.

```
1  <?php
2  session_start();
3  require_once "autoload.php";
4  if((!isset($_SESSION['logado'])) || ($_SESSION['logado']!=true)){
5  |    header('Location: index.php');
6  }
7  ?>
```





Página principal (principal.php)

```
1  <?php
2  require_once "session.php";
3  $usuario = unserialize($_SESSION['usuario']);
4  $nome = $usuario->getNome();
5  ?>
6  <!doctype html>
7  <html lang="pt-br">
8      <head>
9          <meta charset="utf-8">
10         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
            rel="stylesheet"
            integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
            crossorigin="anonymous">
11         <title>Menu principal</title>
12     </head>
13     <body>
14         <div class="container">
15             <header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
16                 <a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark
                    text-decoration-none">
17                     <span class="fs-4"><?=$nome;?></span>
18                 </a>
19                 <ul class="nav nav-pills">
20                     <li class="nav-item"><a href="principal.php" class="nav-link">Home</a></li>
21                     <li class="nav-item"><a href="logout.php" class="nav-link">Logout</a></li>
22                 </ul>
23             </header>
24         </div>
25     </body>
26 </html>
```





Encerrando a sessão (logout.php)

```
1  <?php
2  session_start();
3  $_SESSION = array();
4  session_destroy();
5  header('Location: index.php');
6  ?>
```





Encerrando a sessão (logout.php)

```
1  <?php
2  session_start();
3  $_SESSION = array();
4  session_destroy();
5  header('Location: index.php');
6  ?>
```

Caso prefira, o PHP possui a função `session_unset()` que pode ser utilizada para apagar todas as variáveis de sessão (em substituição à linha 3).





Dúvidas?



