



**INSTITUTO  
FEDERAL**

Sudeste de  
Minas Gerais

Campus  
Manhuaçu

## Desenvolvimento Web III

### Definindo níveis de acesso

Prof. Leonardo C. R. Soares - [leonardo.soares@ifsudestemg.edu.br](mailto:leonardo.soares@ifsudestemg.edu.br)

Instituto Federal do Sudeste de Minas Gerais

15 de janeiro de 2025





## Perfis de usuário

Continuando nosso sistema para controle de acesso, agora vamos definir dois níveis de usuários: Administrador e Visitante.

Para tornar isso possível, iremos criar um campo nível na tabela usuarios. O Administrador terá nível 1 e o visitante nível 0 (que será o padrão).

Para alterar a tabela, utilize o código SQL abaixo:

```
alter table usuarios add column nivel int default 0;
```





## Perfis de usuário

Continuando nosso sistema para controle de acesso, agora vamos definir dois níveis de usuários: Administrador e Visitante.

Para tornar isso possível, iremos criar um campo nível na tabela usuarios. O Administrador terá nível 1 e o visitante nível 0 (que será o padrão).

Para alterar a tabela, utilize o código SQL abaixo:

```
alter table usuarios add column nivel int default 0;
```

Lembre-se de adicionar um usuário administrador para poder efetuar os testes.





# Alterações na classe Usuario

- Altere a classe Usuario para tratar o campo nível. Faça os métodos seletores, adicione-o ao construtor e lembre-se de acrescentá-lo no objeto retornado pelo método confere.





## Alterações na classe Usuario

- Altere a classe Usuario para tratar o campo nível. Faça os métodos seletores, adicione-o ao construtor e lembre-se de acrescentá-lo no objeto retornado pelo método *confere*. Altere o método *livreAcesso* para incluir os dados do nível no objeto retornado.





## Alterações na classe Usuario

- ▶ Altere a classe Usuario para tratar o campo nível. Faça os métodos seletores, adicione-o ao construtor e lembre-se de acrescentá-lo no objeto retornado pelo método *confere*. Altere o método *livreAcesso* para incluir os dados do nível no objeto retornado.
- ▶ Escreva um método estático denominado *getAll()* que retorne um vetor de Usuario contendo todos os usuários cadastrados na tabela em ordem alfabética pelo campo nome.





## Alterações na classe Usuario

- ▶ Altere a classe Usuario para tratar o campo nível. Faça os métodos seletores, adicione-o ao construtor e lembre-se de acrescentá-lo no objeto retornado pelo método *confere*. Altere o método *livreAcesso* para incluir os dados do nível no objeto retornado.
- ▶ Escreva um método estático denominado *getAll()* que retorne um vetor de Usuario contendo todos os usuários cadastrados na tabela em ordem alfabética pelo campo nome.
- ▶ Escreva um método estático denominado *delUser(\$email)*. Este método deverá excluir um usuário da tabela de acordo com o email informado como parâmetro.





# Visualizando todos os usuários

- ▶ Escreva uma página em PHP chamada **verTodos.inc.php** que execute o método *Usuario::getAll()* e imprima o retorno do método em tela.
- ▶ Esta página será incluída pela página principal, assim, faça apenas o conteúdo da página, sem construir um HTML completo.
- ▶ No início da página, faremos uma verificação do nível de acesso necessário:

```
<?php
$NIVEL_EXIGIDO = 1;
// A variável $usuario foi definida na página principal como um objeto
representando o usuário logado
if ( (!isset($usuario)) || ($usuario->getNivel()<$NIVEL_EXIGIDO)) {
    die("<div class=\"alert alert-danger\" role=\"alert\">Seu usuário
    não tem permissão para visualizar este conteúdo.</div>");
}
```







# Visualizando todos os usuários

- Na frente de cada usuário iremos exibir um link para excluí-lo:

```
$usuarios = Usuario::getAll();
?>
<table class="table">
  <thead>
    <tr>
      <th scope="col">Nome</th>
      <th scope="col">Email</th>
      <th scope="col">Nível</th>
    </tr>
  </thead>
  <tbody>
    <?php foreach($usuarios as $linha) { ?>
      <tr>
        <th scope="row"><?=$linha->getNome();?></th>
        <td><?=$linha->getEmail();?></td>
        <td><?=$linha->getNivel()==1 ? "Administrador" : "Usuário";?></td>
        <td><a class="nav-link" href="principal.php?c=<?=base64_encode
("delUser");?>&x=<?=base64_encode($linha->getEmail());?>"
title="Excluir <?=$linha->getNome();?>">Excluir</a></td>
      </tr>
    <?php } ?>
  </tbody>
</table>
```





## Página principal

Todos os links internos a partir da página principal serão redirecionarão para a própria página (com exceção do Logout) informando como parâmetro uma página que deverá ser exibida ou uma ação que deverá ser executada. Os textos chaves da nossa *url* serão criptografados utilizando base64.





# 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>
```





# Página principal

```
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
17         text-decoration-none">
18         <span class="fs-4"><?=$nome;?></span>
19       </a>
20       <ul class="nav nav-pills">
21         <li class="nav-item"><a href="principal.php" class="nav-link">Home</a></li>
22         <?php
23           // Link visível apenas para usuários nível 1 (administradores)
24           if ($usuario->getNivel()==1)
25             echo '<li class="nav-item"><a href="principal.php?pg='.base64_encode("verTodos").'."
26               title="Ver usuários cadastrados" class="nav-link">Usuários</a></li>'
27           <?>
28         <li class="nav-item"><a href="logout.php" class="nav-link">Logout</a></li>
29       </ul>
30     </header>
```





# Página principal

```
29 <?php
30 // Se existir o parâmetro PG, recebemos uma página que será incluída.
31 if(isset($_GET['pg'])){
32     include base64_decode($_GET['pg']).".inc.php";
33 }
34 // Se existir o parâmetro C, recebemos um comando a ser executado.
35 if (isset($_GET['c'])){
36     if (base64_decode($_GET['c'])=="delUser"){
37         try{
38             Usuario::delUser(base64_decode($_GET['x']));
39             echo '<div class="alert alert-success" role="alert">O usuário '. base64_decode($_GET['x']) .'
40                 foi excluído com sucesso. </div>';
41         }catch(PDOException $p){
42             echo '<div class="alert alert-danger" role="alert">Atenção: '. $p->getMessage() .'</div>';
43         }
44     }
45 }
46 ?>
47 </div>
48 </body>
</html>
```





# Página principal

Administrador

[Home](#)

[Usuários](#)

[Logout](#)

Nome	Email	Nível	
Administrador	adm@adm.com	Administrador	<a href="#">Excluir</a>
Leia	princesa@leia.com	Usuário	<a href="#">Excluir</a>
Leo	leo@leo.com	Usuário	<a href="#">Excluir</a>
Pedro	pedro@pedrao.com.br	Usuário	<a href="#">Excluir</a>





# Dúvidas?





# Exercício

- ▶ Implemente o exemplo. Altere o código de forma que o usuário não consiga apagar o próprio usuário, ou seja, o usuário atualmente logado no sistema.
- ▶ Altere o sistema da LibSoft (CEP) para que haja níveis de usuários e para que o usuário administrador possa excluir os endereços cadastrados.





