

Exercícios para serem feitos em classe.

Neste exercício cuidaremos de toda a lógica de usuários. Mantenha o padrão do código. Isso é muito importante para os exercícios posteriores!

- 1) Olhe bem todos os arquivos do projeto crud-filmesV05 e observe as mudanças. Havendo dúvidas sobre qualquer parte do código, consulte seu professor.
- 2) Abra o phpMyAdmin e rode o script2.sql onde criamos a tabela de usuários com senhas criptografadas com hash sha256.
- 3) Dê uma boa olhada no arquivo utilJs/funcoesUtil.js. Perceba que agora estamos exportando todas as funções. Arquivos js que importam funções são type="module".
- 4) Em usuarios.html, comente os scripts buscar.js e alterar.js.

Lógica de listar

- 5) Escreva o arquivo usuarioJs/listar.js conforme código abaixo. Se não entender alguma parte, chame seu professor. Ainda não teste o código.

```
1  import { exibirMensagemErro } from "../utilJs/funcoesUtil.js";
2  import { usuarioExcluirFetch } from "../excluir.js";
3  import { usuarioBuscarFetch } from "../buscar.js";
4
5  $("#btn-novo-jquery").click(function(){
6      $("#modal-formulario-inserir").modal({backdrop: 'static'});
7      $("#modal-formulario-inserir").modal('show');
8  });
9  $("#btn-fechar-jquery").click(function(){
10     $("#modal-formulario-inserir").modal({backdrop: 'static'});
11     $("#modal-formulario-inserir").modal('hide');
12 });
13 $("#btn-fechar-jquery-alterar").click(function(){
14     $("#modal-formulario-alterar").modal({backdrop: 'static'});
15     $("#modal-formulario-alterar").modal('hide');
16 });
17 $("#btn-home-jquery").click(function(){
18     window.location.href = "index.html";
19 });
20 window.onload = ()=>{
21     usuarioListarFetch();
22 }
```

```
23
24 let usuarioListarFetch = function(){
25     document.querySelector('tbody').innerHTML="";
26     fetch("../controller/usuarioListar.php")
27     .then(function(resposta){
28         if(! resposta.ok===true){
29             if(resposta.status===401)
30                 window.location.href = "../view/index.html";
31             let msg = resposta.status + " - " + resposta.statusText;
32             throw new Error(msg);
33         }else
34             return resposta.json();
35     })
36     .then(function(respostaJSON){
37         if(respostaJSON.erro===false)
38             fcSucessoListarUsuario(respostaJSON);
39         else
40             fcErroListarUsuario(respostaJSON.msgErro);
41     })
42     .catch(function(erro){
43         fcErroListarUsuario(erro);
44     });
45 };
46
47 function fcSucessoListarUsuario(respostaJSON){
48     montarTabela(respostaJSON.dados);
49 }
50 function fcErroListarUsuario(erro){
51     exibirMensagemErro('#msg',erro);
52 }
```

```

53
54 //Esta função cria as linhas da tabela com os dados recebidos da CONTROLLER
55 function montarTabela(dados){
56     for (const i in dados) {
57         let obj = dados[i];
58         const $tr = document.createElement('tr');
59         //p/ cada atributo faça
60         criarTDePendurar($tr, obj.id , false);
61         criarTDePendurar($tr, obj.nome , false);
62         //Cria os links para as operações editar e excluir
63         let links = "<a href=# '>[Editar]</a>";
64         links+= "<a href=# '>[Excluir]</a>"
65         //true indica que é um código html
66         criarTDePendurar($tr, links , true);
67         //Pendura a linha criada a cada iteração no tbody da tabela
68         document.querySelector('tbody').appendChild($tr);
69     }//Fim do for in
70 }//Fim da função
71
72 //Lógica de exclusão. A função excluirUsuario está em usuariosjs/excluir.js
73 const $corpoTabela = document.querySelector('tbody');
74 $corpoTabela.addEventListener('click',function(event){
75     if(event.target.tagName==='A'){
76         let link = event.target;
77         let objId = obterValorDaColunaId(link);
78         if(objId>0 && link.textContent === "[Excluir]")
79             usuarioExcluirFetch(objId);
80         else if(objId>0 && link.textContent === "[Editar]")
81             usuarioBuscarFetch(objId);
82     }
83 });
84
85 function obterValorDaColunaId(link){
86     //parentNode = nó pai
87     let coluna = link.parentNode;
88     let linha = coluna.parentNode;
89     //firstChild = primeiro filho
90     let idText = linha.firstChild;
91     return parseInt(idText.textContent);
92 }
93
94 function criarTDePendurar(noPai, informacao, ehHtml){
95     let td = document.createElement('td');
96     if(ehHtml)
97         td.innerHTML = informacao;
98     else
99         td.textContent = informacao;
100     noPai.appendChild(td);
101 }
102
103 export {usuarioListarFetch};

```

6) Escreva o arquivo controller/usuarioListar.php. Havendo dúvidas sobre o código, consulte seu professor. Ainda não teste o código.

```

1  <?php
2      require_once('../model/autenticar.php');
3      require_once('../model/conexao.php');
4      $resposta["erro"] = false;
5      $resposta["dados"] = null;
6      $resposta["msgErro"] = "";
7      $resposta["msgSucesso"] = "";
8      if($autenticado===true){
9          try{
10             $sql = "SELECT id,nome FROM usuarios ORDER BY nome";
11             $stmt = $conexao->prepare($sql);
12             $stmt->execute();
13             $resposta["dados"] = $stmt->fetchAll(PDO::FETCH_ASSOC);
14             $resposta["msgSucesso"] = "Usuários listados com sucesso!";
15         }catch(PDOException $e){
16             $resposta["erro"] = true;
17             $resposta["msgErro"] = "Erro ao listar usuários. ".$e->getMessage();
18         }finally{
19             echo json_encode($resposta);
20         }
21     }
22 }>

```

Lógica de excluir

7) Escreva o arquivo usuarioJs/excluir.js conforme código abaixo. Se não entender alguma parte, chame seu professor. Ainda não teste o código.

```

1  import { limparSpan, exibirMensagem, exibirMensagemErro } from "../utils/funcoesUtil.js";
2  import { usuarioListarFetch } from "../listar.js";
3  function usuarioExcluirFetch(id){
4      if(confirm('Confirma a exclusão do usuário de id '+id+'?')){
5          let usuario = {"id": id};
6          let configMetodo = {
7              method: "DELETE"
8              ,body: JSON.stringify(usuario)
9              ,headers:{"Content-Type": "application/json;charset=UTF-8"}
10         };
11         //fetch enviando o id do usuario a ser excluído
12         fetch("../controller/usuarioExcluir.php", configMetodo)
13             .then(function(resposta){
14                 if(!resposta.ok===true){
15                     if(resposta.status===401)
16                         window.location.href = "../view/index.html";
17                     let msg = resposta.status + " - " + resposta.statusText;
18                     throw new Error(msg);
19                 }else
20                     return resposta.json();
21             })
22             .then(function(respostaJSON){
23                 if(respostaJSON.erro===false)
24                     fcSucessoExcluirUsuario(respostaJSON);
25                 else
26                     fcErroExcluirUsuario(respostaJSON.msgErro);
27             })
28             .catch(function(erro){
29                 fcErroExcluirUsuario(erro);
30             })
31     }
32 }

```



```

33
34 function fcSucessoExcluirUsuario(respostaJSON){
35     exibirMensagem('#msg',respostaJSON.msgSucesso);
36     setTimeout(function(){
37         //Agora apenas aguarda 1,5 segundos para a mensagem ser exibida e recarregar a lista
38         limparSpan('#msg');
39         usuarioListarFetch();
40     },1500);
41 }
42
43 function fcErroExcluirUsuario(erro){
44     exibirMensagemErro('#msg',erro);
45     setTimeout(function(){
46         limparSpan('#msg');
47     },1500);
48 }
49
50 export {usuarioExcluirFetch};

```

8) Escreva o arquivo controller/usuarioExcluir.php. Havendo dúvidas sobre o código, consulte seu professor.

```

1  <?php
2  require_once('../model/autenticar.php');
3  require_once('../model/conexao.php');
4  $usuarioDelete = file_get_contents('php://input');
5  $usuarioMatriz = json_decode($usuarioDelete, true);
6  $id = (isset($usuarioMatriz["id"]) && $usuarioMatriz["id"] != null) ? $usuarioMatriz["id"] : null;
7  $resposta["erro"] = false;
8  $resposta["msgErro"] = "";
9  $resposta["msgSucesso"] = "";
10 $resposta["dados"] = null;
11 if( $id != null && $autenticado===true){
12     try {
13         $sql = "DELETE FROM usuarios WHERE id=?";
14         $stmt = $conexao->prepare($sql);
15         $stmt->bindParam(1,$id);
16         $stmt->execute();
17         $resposta["msgSucesso"] = "Usuário de id $id excluído com sucesso!";
18     }catch(PDOException $e) {
19         $resposta["erro"] = true;
20         $resposta["msgErro"] = "Erro ao excluir usuário. ".$e->getMessage();
21     }finally{
22         echo json_encode($resposta);
23     }
24 }
25 ?>

```

Hora de testar as lógicas de listar e excluir antes de prosseguir com o exercício. Só continue se tudo estiver funcionando corretamente.

Lógica de inserir

9) Escreva o arquivo usuarioJs/inserir.js conforme código abaixo. Se não entender alguma parte, chame seu professor. Ainda não teste o código.

```

1  import { limparSpan, exibirMensagem, exibirMensagemErro } from "../utils/funcoesUtil.js";
2  import { usuarioListarFetch } from "../listar.js";
3
4  //Recupera o elemento input:submit(Também poderíamos usar o form e o mesmo evento submit)
5  const $btnEnviar = document.querySelector('#enviar');
6  $btnEnviar.addEventListener('click', function(event){
7      event.preventDefault();
8      usuarioInserirFetch();
9      $("#modal-formulario-inserir").modal('hide');
10 });
11
12 let usuarioInserirFetch = function(){
13     let usuario = {
14         "nome": document.querySelector('#form-inserir').querySelector('#nome').value,
15         "login": document.querySelector('#form-inserir').querySelector('#login').value,
16         "senha": document.querySelector('#form-inserir').querySelector('#senha').value
17     };
18     let configMetodo = {
19         method: "POST"
20         ,body: JSON.stringify(usuario)
21         ,headers: {"Content-Type":"application/json;charset=UTF-8"}
22     };
23     fetch("../controller/usuarioInserir.php", configMetodo)
24     .then(function(resposta){
25         if(!resposta.ok===true){
26             if(resposta.status===401)
27                 window.location.href = "../view/index.html";
28             let msg = resposta.status + " - " + resposta.statusText;
29             throw new Error(msg);
30         }else
31             return resposta.json();
32     })
33     .then(function(respostaJSON){
34         if(respostaJSON.erro===false)
35             fcSucessoInserirusuario(respostaJSON);
36         else
37             fcErroInserirusuario(respostaJSON.msgErro);
38     })
39     .catch(function(erro){
40         fcErroInserirusuario(erro);
41     });
42 };
43
44 function fcSucessoInserirusuario(respostaJSON){
45     exibirMensagem('#msg', respostaJSON.msgSucesso);
46     setTimeout(function(){
47         //Agora apenas aguarda 1,5 segundos para a mensagem ser exibida e recarregar a lista
48         limparSpan('#msg');
49         usuarioListarFetch();
50     },1500);
51 }
52
53 function fcErroInserirusuario(erro){
54     exibirMensagemErro('#msg',erro);
55     setTimeout(function(){
56         limparSpan('#msg');
57     },1500);
58 }

```

10) Escreva o arquivo controller/usuarioInserir.php. Havendo dúvidas sobre o código, consulte seu professor.

```

2   require_once('../model/autenticar.php');
3   require_once('../model/conexao.php');
4   $usuarioPost = file_get_contents('php://input');
5   $usuarioMatriz = json_decode($usuarioPost, true);
6   $nome = (isset($usuarioMatriz["nome"]) && $usuarioMatriz["nome"] != null)
7   ? $usuarioMatriz["nome"] : "";
8   $login = (isset($usuarioMatriz["login"]) && $usuarioMatriz["login"] != null)
9   ? $usuarioMatriz["login"] : "";
10  $senha = (isset($usuarioMatriz["senha"]) && $usuarioMatriz["senha"] != null)
11  ? $usuarioMatriz["senha"] : "";
12  $senha = hash('sha256', $senha);
13  $resposta["erro"] = false;
14  $resposta["msgErro"] = "";
15  $resposta["msgSucesso"] = "";
16  $resposta["dados"] = null;
17  if( $nome != "" && $login != "" && $senha != "" && $autenticado===true){
18      try {
19          $sql = "INSERT INTO usuarios(nome,login,senha) VALUES(?,?,?)";
20          $stmt = $conexao->prepare($sql);
21          $stmt->bindParam(1, $nome );
22          $stmt->bindParam(2, $login );
23          $stmt->bindParam(3, $senha);
24          $stmt->execute();
25          $resposta["msgSucesso"] = "{ $stmt->rowCount() } usuario inserido com sucesso! O id inserido
26          foi { $conexao->lastInsertId() }";
27      } catch(PDOException $e) {
28          $resposta["erro"] = true;
29          $resposta["msgErro"] = "Erro: Não foi possível efetuar a inserção no BD". $e->getMessage();
30      } finally{
31          echo json_encode($resposta);
32      }
33  }

```

Hora de testar a lógica de inserir antes de prosseguir com o exercício. Só continue se tudo estiver funcionando corretamente.

Lógica de buscar

- 11) Em usuários.html, descomente a linha do script usuariosJs/buscar.js
- 12) Escreva o arquivo usuarioJs/buscar.js conforme código abaixo. Se não entender alguma parte, chame seu professor. Ainda não teste o código.

```

1  import { exibirMensagem, exibirMensagemErro, limparSpan } from "../utils/funcoesUtil.js";
2  function usuarioBuscarFetch(id){
3      //fetch enviando o id do usuario a ser recuperado
4      fetch("../controller/usuarioBuscar.php?id="+id+"")
5          .then(function(resposta){
6              if(!resposta.ok===true){
7                  if(resposta.status===401)
8                      window.location.href = "../view/index.html";
9                  let msg = resposta.status + " - " + resposta.statusText;
10                 throw new Error(msg);
11             }
12             else
13                 return resposta.json();
14         })
15         .then(function(respostaJSON){
16             if(respostaJSON.erro===false)
17                 fcSucessoBuscarusuario(respostaJSON);
18             else
19                 fcErroBuscarusuario(respostaJSON.msgErro);
20         })
21         .catch(function(erro){
22             exibirMensagemErro("#msg",erro);
23         });
24     }
25
26     function fcSucessoBuscarusuario(respostaJSON){
27         $("#modal-formulario-alterar").modal({backdrop: 'static'});
28         $("#modal-formulario-alterar").modal('show');
29         let formAlterar = document.querySelector('#form-alterar');
30         let usuario = respostaJSON.dados;
31         formAlterar.querySelector('#id').value = usuario.id;
32         formAlterar.querySelector('#nome').value = usuario.nome;
33     }
34
35     function fcErroBuscarusuario(erro){
36         exibirMensagemErro("#msg",erro);
37         setTimeout(function(){
38             limparSpan('#msg');
39         },1500);
40     }
41
42     export {usuarioBuscarFetch};

```

13) Escreva o arquivo controller/usuarioBuscar.php. Havendo dúvidas sobre o código, consulte seu professor. Ainda não teste o código.


```

1  <?php
2      require_once('../model/autenticar.php');
3      require_once('../model/conexao.php');
4      $id = (isset($_GET['id']) && $_GET['id'] > 0) ? $_GET['id'] : null;
5      $resposta["erro"] = false;
6      $resposta["dados"] = null;
7      $resposta["msgErro"] = "";
8      $resposta["msgSucesso"] = "";
9      if($id!=null && $autenticado==true){
10         try{
11             $sql = "SELECT id,nome FROM usuarios WHERE id = ?";
12             $stmt = $conexao->prepare($sql);
13             $stmt->bindParam(1, $id);
14             $stmt->execute();
15             $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
16             $resposta["dados"] = $result[0];
17             $resposta["msgSucesso"] = "usuario de id {$id} retornado com sucesso!";
18         }catch(PDOException $e){
19             $resposta["erro"] = true;
20             $resposta["msgErro"] = "Erro ao retornar usuario de id {$id}. ".$e->getMessage();
21         }finally{
22             echo json_encode($resposta); exit();
23         }
24     }
25 }>

```

Hora de testar a lógica de buscar antes de prosseguir com o exercício. IMPORTANTE: Ainda não é possível submeter o formulário. Basta verificar se ele está carregando com as informações do usuário solicitado. Só continue se tudo estiver funcionando corretamente.

Lógica de alterar

14) Em usuarios.html, descomente a linha do script usuariosJs/alterar.js

15) Escreva o arquivo usuarioJs/alterar.js conforme código abaixo. Se não entender alguma parte, chame seu professor. Ainda não teste o código.

```

1  import { exibirMensagem,exibirMensagemErro,limparSpan } from "../utilJs/funcoesUtil.js";
2  import { usuarioListarFetch } from "../listar.js";
3  //Recupera o botão alterar (Também poderíamos usar o form e o evento submit)
4  const $btnAlterar = document.querySelector('#alterar');
5  $btnAlterar.addEventListener('click', function(event){
6      event.preventDefault();
7      usuarioAlterarFetch();
8      $("#modal-formulario-alterar").modal('hide');
9  });

```

```

10 let usuarioAlterarFetch = function(){
11     let formAlterar = document.querySelector('#form-alterar');
12     let usuario = {
13         "id": formAlterar.querySelector('#id').value,
14         "nome": formAlterar.querySelector('#nome').value
15     };
16     let configMetodo = {
17         method: "PUT"
18         ,body: JSON.stringify(usuario)
19         ,headers: {"Content-Type":"application/json;charset=UTF-8"}
20     };
21     fetch("../controller/usuarioAlterar.php", configMetodo)
22         .then(function(resposta){
23             if(!resposta.ok===true){
24                 if(resposta.status===401)
25                     window.location.href = "../view/index.html";
26                 let msg = resposta.status + " - " + resposta.statusText;
27                 throw new Error(msg);
28             }
29             else
30                 return resposta.json();
31         })
32         .then(function(respostaJSON){
33             if(respostaJSON.erro===false)
34                 fcSucessoAlterarusuario(respostaJSON);
35             else
36                 fcErroAlterarusuario(respostaJSON.msgErro);
37         })
38         .catch(function(erro){
39             exibirMensagemErro('#msg',erro);
40         });
41 }

42
43 const $btnCancelar = document.querySelector('#cancelar');
44 $btnCancelar.addEventListener('click',function(){
45     if(confirm('Deseja mesmo cancelar a alteração?'))
46         window.location.href = "../view/usuarios.html";
47 })
48
49 function fcSucessoAlterarusuario(respostaJSON){
50     exibirMensagem('#msg',respostaJSON.msgSucesso);
51     setTimeout(function(){
52         limparSpan('#msg');
53         usuarioListarFetch();
54     },1500);
55 }
56
57 function fcErroAlterarusuario(erro){
58     exibirMensagemErro('#msg',erro);
59     setTimeout(function(){
60         limparSpan('#msg');
61     },1500);
62 }

```

16) Escreva o arquivo `controller/usuarioAlterar.php`. Havendo dúvidas sobre o código, consulte seu professor. Ainda não teste o código.

```

1  <?php
2      require_once('../model/autenticar.php');
3      require_once('../model/conexao.php');
4      $usuarioPut = file_get_contents('php://input');
5      $usuarioMatriz = json_decode($usuarioPut, true);
6      $id = (isset($usuarioMatriz["id"]) && $usuarioMatriz["id"] > 0) ? $usuarioMatriz["id"] : null;
7      $nome = (isset($usuarioMatriz["nome"]) && $usuarioMatriz["nome"] != null)
8      ? $usuarioMatriz["nome"] : "";
9      $resposta["erro"] = false;
10     $resposta["msgErro"] = "";
11     $resposta["msgSucesso"] = "";
12     $resposta["dados"] = null;
13     if( $nome != "" && $id != "" && $autenticado===true){
14         try {
15             $sql = "UPDATE usuarios SET nome=? WHERE id=?";
16             $stmt = $conexao->prepare($sql);
17             $stmt->bindParam(1, $nome );
18             $stmt->bindParam(2, $id);
19             $stmt->execute();
20             $resposta["msgSucesso"] = "{ $stmt->rowCount() } usuario alterado com sucesso!
21             O id do usuario alterado foi { $id }";
22         } catch(PDOException $e) {
23             $resposta["erro"] = true;
24             $resposta["msgErro"] = "Erro: Não foi possível efetuar a alteração no BD". $e->getMessage();
25         } finally{
26             echo json_encode($resposta);
27         }
28     }
29     ?>

```

Hora de testar a lógica de alterar. Se tudo estiver funcionando corretamente, nos próximos exercícios cuidaremos da autenticação de usuários e da refatoração do código que ainda pode ficar muito mais simples!

17) Renomeie o projeto para crud-filmesV05_