



Escolas e Faculdades



Continuação Editar e Deletar de Usuários

**Andreza Barcaro e Brian
Neukamp
2020**

PHP OO + Composer + PHP Storm + Banco

Para Editar e Deletar Vamos Realizar isso para Os **Usuários** o Restante ficará como Exercício
Vamos selecionar o ID então abra a **classe Usuarios.php**

```
<?php
```

```
__DIR__ . "/../../vendor/autoload.php";
```

```
class Usuarios
```

```
{
```

```
    private $id;
```

```
    private $nome;
```

```
    private $email;
```

```
    private $senha;
```

```
    private $nivel;
```

```
    private $mensagem;
```

Declare o id
e gerar os
get e set

Escolas e Faculdades



PHP OO + Composer + PHP Storm + Banco

Classe Usuarios abaixo do método inserir

```
//Método Selecionar id (Editar)
public function querySelecionaId($dado)
{
    try
    {
        $this->id = $this->objfcn->base64($dado, tipo: 2);
        $cst = $this->con->conectar()->prepare( statement: "SELECT id, nome, email, senha, nivel FROM usuarios WHERE id = :idUser ");
        $cst->bindParam( parameter: ":idUser", &variable: $this->id, data_type: PDO::PARAM_INT);
        $cst->execute();
        return $cst->fetch();
    } catch (PDOException $ex) {
        return 'error '.$ex->getMessage();
    }
}
```

Aqui Chamamos a
Função para criptografar
o id

Escolas e Faculdades



PHP OO + Composer + PHP Storm + Banco

Logo Abaixo vamos ao Método Editar - Passo 1 Usuarios

//Editar

```
public function queryUpdate($dados)
{
    try
    {
        $this->id = $this->objfcn->base64($dados['func'], tipo: 2);
        $this->nome = $dados['nome'];
        $this->email = $dados['email'];
        $this->senha = sha1($dados['senha']);
        $this->nivel = $dados['nivel'];
        $this->mensagem = $dados['mensagem'];

        $cst = $this->con->conectar()->prepare( statement: "UPDATE usuarios SET nome = :nome, email = :email,
        senha = :senha, nivel = :nivel , mensagem = :mensagem WHERE id = :idUser");
```

Escolas e Faculdades



PHP OO + Composer + PHP Storm + Banco

Logo Abaixo vamos ao Método Editar - Passo 2 Usuarios

```
$cst->bindParam( parameter: ":idUser", &variable: $this->id, data_type: PDO::PARAM_INT);
$cst->bindParam( parameter: ":nome", &variable: $this->nome, data_type: PDO::PARAM_STR);
$cst->bindParam( parameter: ":email", &variable: $this->email, data_type: PDO::PARAM_STR);
$cst->bindParam( parameter: ":senha", &variable: $this->senha, data_type: PDO::PARAM_STR);
$cst->bindParam( parameter: ":nivel", &variable: $this->nivel, data_type: PDO::PARAM_INT);
$cst->bindParam( parameter: ":mensagem", &variable: $this->mensagem, data_type: PDO::PARAM_STR);

if($cst->execute())
{
    return 'ok';
}
else
{
    echo '<script type="text/javascript">alert("Não alterou");</script>';
}
}
catch (PDOException $ex)
{
    print_r ($ex->getMessage());
}
```

```
}
```

PHP OO + Composer + PHP Storm + Banco

Logo Abaixo vamos ao Método Deletar - Usuarios

```
//Método para Deletar
public function queryDelete($dado)
{
    try
    {
        $this->id = $this->objfcn->base64($dado, tipo: 2);
        $cst = $this->con->conectar()->prepare( statement: "DELETE FROM usuarios WHERE id = :idUser");
        $cst->bindValue( parameter: ":idUser", $this->id, data_type: PDO::PARAM_INT);
        if($cst->execute())
        {
            return 'ok';
        }
        else
        {
            echo "Não foi possível Deletar";
        }
    }
    catch (PDOException $exc)
    {
        echo $exc->getTraceAsString();
    }
}
```

PHP OO + Composer + PHP Storm + Banco

Agora Vamos ao arquivo **usuarios.php** chamar os métodos e funções então Precisamos Realizar o import das Funções

```
<?php  
require_once "../vendor/autoload.php";  
  
$a = new Usuarios();  
$objfn = new Funcoes();
```


PHP OO + Composer + PHP Storm + Banco

usuarios.php

Após termos o método para inserir abaixo do método inserir vamos a ação de deletar e editar

```
<?php
    require_once "../vendor/autoload.php";

    $a = new Usuarios();
    $objfn = new Funcoes();
```


PHP OO + Composer + PHP Storm + Banco

usuarios.php editar

Após temos o método para inserir abaixo do método inserir vamos a ação de editar e deletar
Abaixo do if do inserir

```
if(isset($_POST["btAlterar"]))
{
    if($a->queryUpdate($_POST) == "ok")
    {
        header( string: "Location: ?acao=edit?func". $objfn->base64($_POST["func"], tipo: 1));
    }
    else
    {
        echo "erro ao alterar";
    }
}
```

PHP OO + Composer + PHP Storm + Banco

usuarios.php deletar

Após temos o método para inserir abaixo do método inserir vamos a ação de editar e deletar
Abaixo do if do editar

```
if(isset($_GET["acao"]))
{
    switch ($_GET["acao"])
    {
        case "edit" : $func = $a->querySelecioneid($_GET["func"]); break;
        case "delet" :
            if($a->queryDelete($_GET["func"]) == "ok")
            {
                header( string: 'http://localhost/phpstorm/sistema/usuarios.php');
            }
            else
            {
                echo "erro ao deletar";
            }
            break;
    }
}
```

PHP OO + Composer + PHP Storm + Banco

Agora vamos ao html e ações nos botões

O que vamos dizer aqui: Quando For Cadastrar o botão é Cadastrar quando for Editar o Botão será Editar, dessa maneira não precisamos criar outro formulário para editar os valores o name="func" é o que chamamos de parâmetro de URL

```
<input type="submit" class="btn btn-primary" name="<?= (isset($_GET["acao"])) == "edit" ? ("btAlterar") : ("btCadastrar") ) ?>"
<input type="hidden" name="func" value="<?= (isset($func["id"]) ? $objfn->base64($func["id"], tipo: 1) : " ") ?>">
```

Nessa parte para facilitar vamos e evitar erros de sintaxe vamos copiar o código

```
<input type="submit" class="btn btn-primary" name="<?= (isset($_GET["acao"])) == "edit" ?
("btAlterar") : ("btCadastrar") ) ?>" value="<?= (isset($_GET["acao"])) == "edit" ? ("Alterar")
: ("Cadastrar") ) ?>">
```

```
<input type="hidden" name="func" value="<?= (isset($func["id"])) ? $objfn->base64($func["id"],
1) : " " ) ?>">
```

PHP OO + Composer + PHP Storm + Banco

Agora vamos ao select dar ação para cada ide selecionado botão editar e deletar

```
<td><a href="?acao=edit&func=<?= $objfn->base64($rst["id"], tipo: 1) ?>">Editar</a></td>  
<td><a href="?acao=delet&func=<?= $objfn->base64($rst["id"], tipo: 1) ?>">Deletar</a></td>
```

Nessa parte para facilitar vamos e evitar erros de sintaxe vamos copiar o código

```
<td><a href="?acao=edit&func=<?= $objfn->base64($rst["id"], 1) ?>">Editar</a></td>  
<td><a href="?acao=delet&func=<?= $objfn->base64($rst["id"], 1) ?>">Deletar</a></td>
```

PHP OO + Composer + PHP Storm + Banco

E para mostrar os valores no input quando usuário clica em editar basta adicionar ao value= " " de cada input

```
exampleFormControlInput1">Nome do Usuário:</label>  
"text" name="nome" class="form-control" id="nome" value="<?=(isset($func['nome'])) ?($func['nome']): ('')?>">
```

Pode Copiar Será o mesmo para todos inputs para o select já temos no caso de clientes aí não é necessário

```
<input type="text" name="nome" class="form-control" id="nome" value="<?=(isset($func['nome']))  
?($func['nome']): ('')?>">
```

Escolas e Faculdades



PHP OO + Composer + PHP Storm + Banco

Agora Pode Testar o CRUD para usuários

Escolas e Faculdades



Dúvidas



Escolas e Faculdades

QI

Referência Bibliográfica

- **Desenvolvimento Web com HTML, CSS e JavaScript.** Caelum, Disponível via web em <www.caelum.com.br>. 2015.
- GLAZAR, Jean Eduardo. **Programação para Web.** IFES, 2011.
- **Aprender PHP.** <http://aprenderphp.com.br>