



Universidade Federal de Santa Catarina
Curso de Tecnologias da Informação e Comunicação
Desenvolvimento de Sistemas Web
Professor: Adriano de Oliveira

Desenvolvimento de Sistemas Web

Prof.: Adriano de Oliveira
email: adriano.inovar@gmail.com
Turma: 05652 - 3-2020-2 e 5-1830-2

PHP





Universidade Federal de Santa Catarina
Curso de Tecnologias da Informação e Comunicação
Desenvolvimento de Sistemas Web
Professor: Adriano de Oliveira

Formulários e MySQL/MariaDB

Formulários

- ✓ O processamento de forms é um dos pontos fortes do PHP.
- ✓ Este processamento é facilitado através de dois arrays chamados de “superglobais”, o `$_GET` e o `$_POST`, que contêm os dados que foram submetidos pelos forms.

Formulários

- ✓ O `$_GET` é um array associativo criado pelo PHP que recebe os dados submetidos para ou outro script PHP, através do método `$_GET`.
- ✓ Para acessar o valor de um par chave/valor submetido via `$_GET`, usamos o nome (chave) que identifica o valor que queremos obter.

Formulários

- ✓ Exemplo:
- ✓ HTML:

```
<input type="text" name="name">
```

- ✓ PHP:

```
<?php  
    $name = $_GET['name'];  
?>
```

Formulários

- ✓ O método POST é geralmente mais usado que o GET. Neste método, os dados introduzidos no form são enviados no header HTTP.
- ✓ A principal diferença é que os dados não são passados pela URL, como no método GET, que usa a query string para passar combinações de par -> valor.
- ✓ No método GET, é usado o '?' para separar a query string do script e o & para separar cada combinação de par -> valor.

Formulários

```
<div class="container">
  <form action="exemplo01_result.php" method="get">
    <label for="name">Nome:</label>
    <input type="text" name="name" id="name" class="form-control">
    <label for="age">Idade:</label>
    <input type="text" name="age" id="age" class="form-control"><br>
    <input type="submit" value="Submeter" class="form-control">
  </form>
</div>
```



```
<body>
  <?php
    echo 'Os valores submetidos foram: <br />';
    echo 'Nome: ' . htmlspecialchars($_GET['name']);
    echo '<br />';
    echo 'Idade: ' . (int)$_GET['age'];
  ?>
  <br />
  <a href="exemplo01_form.php">Voltar ao inicio</a>
</body>
```

Formulários

- ✓ O script exemplo01_get.php recebe os dados submetidos através da variável “superglobal” `$_GET`.
- ✓ Nota: Neste exemplo, foi usado a função `htmlspecialchars()` para obtermos a representação dos eventuais códigos HTML introduzidos pelo utilizador.

Validação de Formulários

- ✓ Vamos ver como validar um formulário do lado servidor.
- ✓ Nem sempre é seguro validar formulários apenas com javascript.
- ✓ Então além da validação javascript é indispensável a validação no lado servidor.

Formulários

- ✓ Criar um formulário que simula o sistema de venda de uma loja de esportes;
- ✓ O formulário enviará os dados via requisição *POST* para a página PHP;
- ✓ Esta página processará a requisição e enviará uma resposta.





Formulários

- ✓ Primeiramente, escreveremos em nossa página uma mensagem informando que o pedido foi processado;
- ✓ Depois pegaremos as quantidades informadas em nosso array `$_POST` (lembre-se que fizemos uma requisição do tipo POST);

Formulários

- ✓ Declararemos uma variável para representar o custo de cada uma das bolas;
- ✓ Multiplicaremos o valor da quantidade de cada bola pelo seu respectivo preço;
- ✓ Mostraremos na tela o valor total de cada item e o valor total da compra.

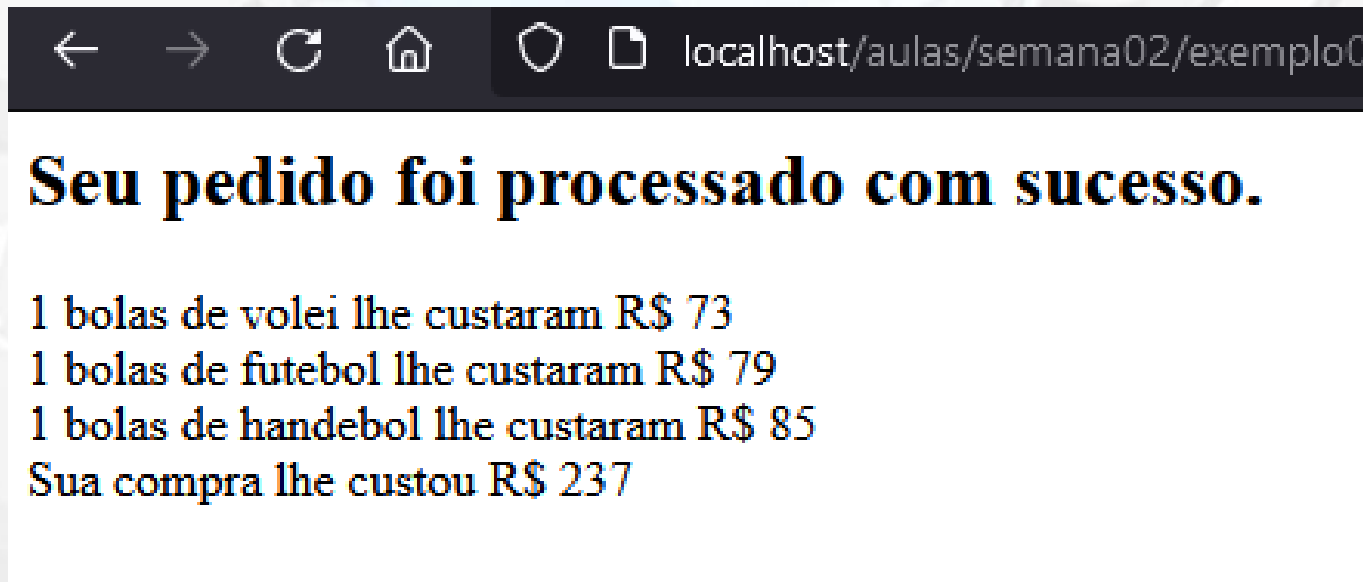
Formulários

 localhost/aulas/semana02/exemplo03_

| Itens | Quantidades |
|--------------------|----------------------|
| Bolas de volei: | <input type="text"/> |
| Bolas de futebol: | <input type="text"/> |
| Bolas de basquete: | <input type="text"/> |

Enviar pedido

Formulários



Formulários

```
<?php
    $volei = $_POST['edtvolei'];
    $futebol = $_POST['edtfutebol'];
    $basquete = $_POST['edtbasquete'];

    $precovolei = 73;
    $precofutebol = 79;
    $precobasquete = 85;

    $valorBV = $volei * $precovolei;
    $valorBF = $futebol * $precofutebol;
    $valorBB = $basquete * $precobasquete;
    $valorTotal = $valorBV + $valorBF + $valorBB;

    echo "$volei bolas de volei lhe custaram R$ $valorBV<br />";
    echo "$futebol bolas de futebol lhe custaram R$ $valorBF<br />";
    echo "$basquete bolas de handebol lhe custaram R$ $valorBB<br />";
    echo "Sua compra lhe custou R$ $valorTotal<br />";
```



Universidade Federal de Santa Catarina
Curso de Tecnologias da Informação e Comunicação
Desenvolvimento de Sistemas Web
Professor: Adriano de Oliveira

Acessando Banco de Dados MySQL

PHP & MySQL

- ✓ PHP e MySQL formam uma dupla extremamente famosa da internet;
- ✓ Acessar um banco de dados MySQL utilizando PHP não é difícil;

PHP & MySQL

- ✓ Utilizamos uma função que abre uma conexão com um servidor e a partir deste momento podemos lidar facilmente com esta conexão, inserindo, atualizando, apagando ou obtendo dados do banco.

PHP & MySQL

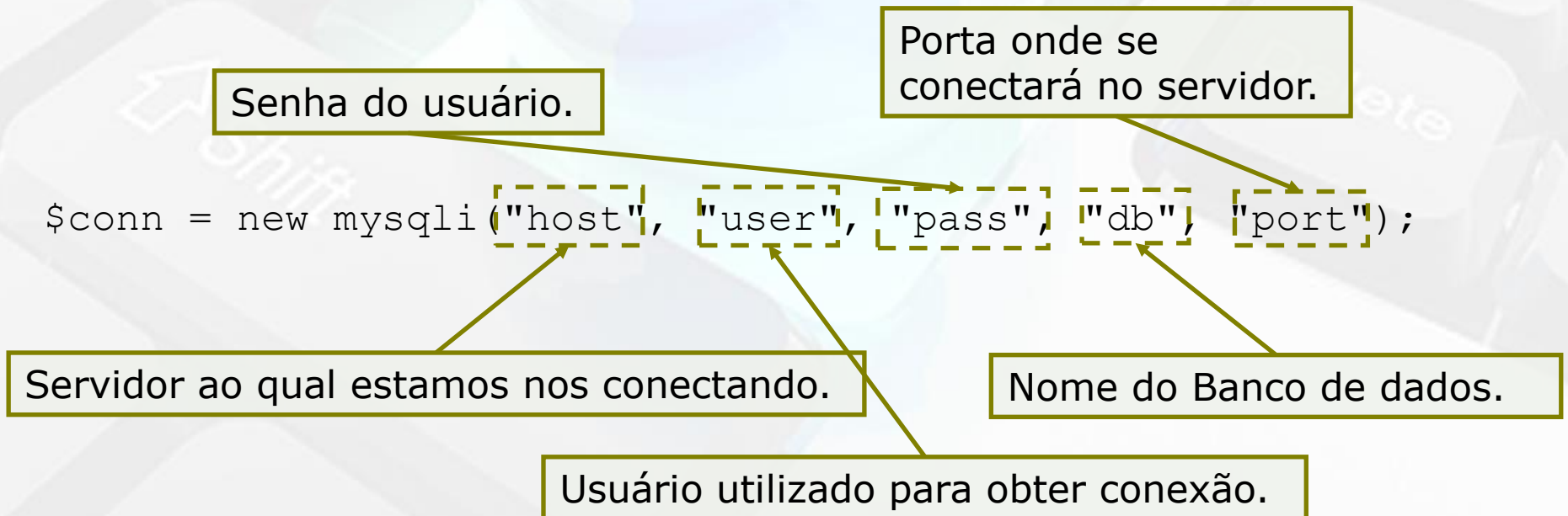
- ✓ Vamos conhecer as funções do PHP para trabalhar com MySQL, sem entrar em detalhes sobre os scripts utilizados para acessar o banco de dados.

Bibliotecas

- ✓ Utilizaremos a biblioteca *mysqli* (MySQL Improved), ela trabalha com objetos, mas possui apelidos para a maioria dos métodos;
- ✓ Esta biblioteca para conexões possui melhor desempenho e maior segurança;

Conectando-se ao banco de dados

- ✓ Para conectar-se ao banco de dados, por OO, usando o construtor *new mysqli*;



Testando a conexão

- ✓ A função *mysqli_connect* retorna um identificador ou *false*;
- ✓ Por OO, usando o construtor *new mysqli*

```
<?php
    $host = "localhost";
    $user = "aulasufsc";
    $password = "ufsc";
    $db = "aulas";
    $port = "3307";

    $conn = new mysqli($host, $user, $password, $db, $port);

    if ($conn->connect_error) {
        die("Conexão falhou: " . $conn->connect_error);
    }
    echo "Conexão bem sucedida";
```

Exemplo 04



Conexão bem sucedida

CRUD

Tabela do Banco de Dados

```
CREATE TABLE client (  
    CLI_ID INT(11) NOT NULL AUTO_INCREMENT,  
    CLI_NAME VARCHAR(50) NULL DEFAULT NULL,  
    CLI_LASTNAME VARCHAR(50) NULL DEFAULT NULL,  
    CLI_ADDRESS VARCHAR(50) NULL DEFAULT NULL,  
    PRIMARY KEY (CLI_ID)  
) ENGINE=InnoDB;
```

O que faremos?

- ✓ Criaremos um pequeno aplicativo com as funções:
- ✓ Inserção de clientes;
- ✓ Listar os clientes no banco de dados;
- ✓ Alteração de cliente;
- ✓ Exclusão de clientes;

CRUD em PHP com MySQL

localhost/aulas/semana02/crud/index.php

CRUD em PHP com MySQL

Nome

Sobrenome

Endereço

Inserir

| Id | Nome/Sobrenome | Endereço | Opções |
|----|---------------------|---------------------|---|
| 6 | Adriano de Oliveira | Jardim das avenidas |   |

Exemplo CRUD