**Lidando com Selects e Checkboxes**

Nosso sistema está cada vez mais completo. Já aprendemos a criar telas de adição, remoção e listagem de produtos. Isso quer dizer que poderíamos replicar a mesma coisa para outras entidades do sistema, como por exemplo, cadastro de usuários, ou mesmo cadastro de categorias.

Vamos agora lidar com outros elementos HTML. Nesse momento, a escolha de uma categoria é dada por meio de radio buttons. Mas temos outras opções. Podemos usar, por exemplo, uma caixa de seleção, um select. Vamos alterar nosso código para tal. Veja que aqui usamos as tags select e a tag option:

<tr>

<td>Categoria</td>

<td>

<select name="categoria\_id">

<?php foreach($categorias as $categoria) : ?>

<option value="<?=$categoria['id']?>"><?=$categoria['nome']?></option>

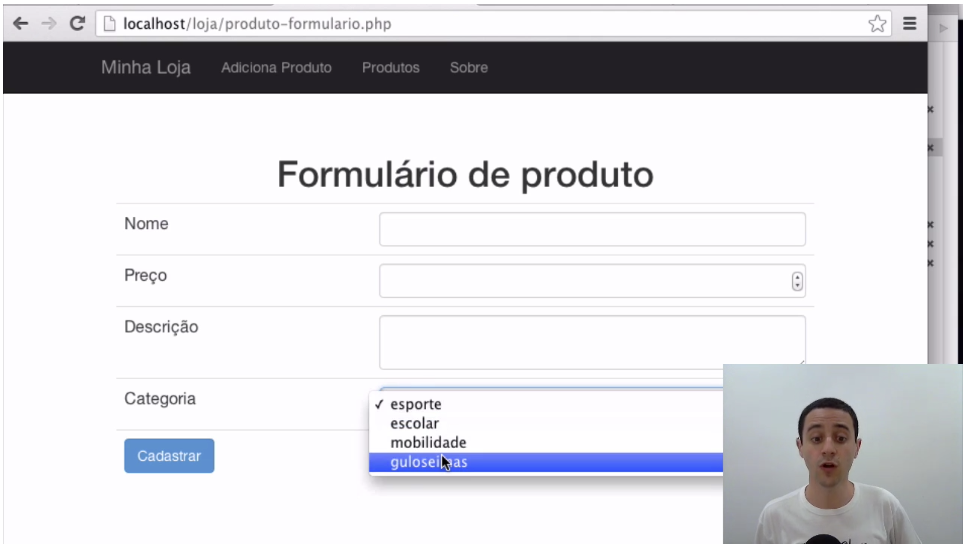
<?php endforeach ?>

</select>

</td>

</tr>

Veja como ficou:



Da mesma forma, o que é mostrado pro usuário é diferente do que é postado. O select post o valor que está na tagvalue, do option.

O próximo passo agora é dar a possibilidade do usuário dizer se aquele produto é usado ou não. Para isso, colocaremos um checkbox, aquele componente visual, que você seleciona ou desseleciona. Logo acima da tr da categoria, colocamos mais uma:

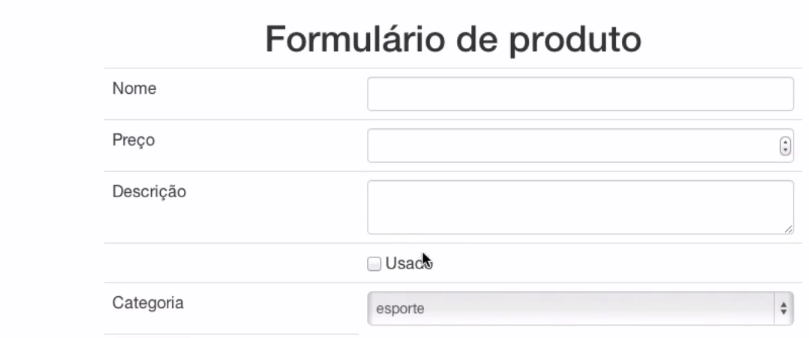
<tr>

<td></td>

<td><input type="checkbox" name="usado" value="true"> Usado

</tr>

Veja como ficou:



Precisamos criar a coluna no nosso banco de dados. Ela se chamará usado e terá valor padrão falso (ou seja, o produto é novo por padrão):

alter table produtos add column usado boolean default false;

Precisamos agora capturar o valor do checkbox que foi postado em nosso adiciona-produto.php, igual fizemos com todos os outros campos:

$usado = $\_POST['usado'];

if(insere\_produto($conexao,$nome,$preco,$descricao,$categoria\_id, $usado)) {

// ...

}

Agora, no banco-produto.php, vamos mudar a função insere\_produto para que ela passe esse valor na SQL:

function insereProduto($nome, $preco, $descricao, $categoria\_id, $usado) {

$query = "insert into produtos (nome, preco, descricao, categoria\_id, usado)

values ('{$nome}', {$preco}, '{$descricao}', {$categoria\_id}, {$usado})";

$resultadoDaInsercao = mysqli\_query($conexao, $query);

}

Se tentarmos adicionar um produto usado, tudo funciona. Mas e se adicionarmos um produto novo, o que acontece? Tomaremos um erro! A nossa query não vai funcionar. E nesse caso, ainda teremos mais de um erro, já que o PHP não para de executar seu programa caso um erro aconteça; ele vai até o fim.

Veja na aba Network, que quando o checkbox não é checado, o formulário não posta a informação! Esse é o funcionamento: checkbox não selecionado, não é enviado. Então precisaremos tratar isso no código. Lá noadiciona-produto.php:

if(array\_key\_exists('usado', $\_POST)) {

$usado = true;

} else {

$usado = false;

}

Se testarmos de novo, tomaremos outro erro. Agora não tem jeito, precisamos descobrir o que está acontecendo. Uma maneira para isso, é exibir a query que está sendo executada. Vamos colocar um echo no insereProduto:

function insereProduto($nome, $preco, $descricao, $categoria\_id, $usado) {

$query = "insert into produtos (nome, preco, descricao, categoria\_id, usado)

values ('{$nome}', {$preco}, '{$descricao}', {$categoria\_id}, {$usado})";

echo $query;

$resultadoDaInsercao = mysqli\_query($conexao, $query);

}

Veja que o usado ainda está indo vazio.

O que acontece é que quando a variável usado é false, na hora da concatenação, o PHP não escreve a palavra "false", mas sim põe um espaço vazio. Precisamos tomar bastante cuidado com isso. Para resolver, vamos colocar a string false dentro da variável (repare nas aspas):

if(array\_key\_exists('usado', $\_POST)) {

$usado = "true";

} else {

$usado = "false";

}

Agora sim, tudo funciona. Não esqueça de tirar a mensagem de log que ficou dentro do insereProduto.