02

Cadastrando o usuário

Transcrição

Vamos pegar todas as informações que temos no Carrinho e finalizar a compra, usando o botão "Finalizar Compra". Usaremos um formulário simples onde poderemos preencher os dados do usuário e salvar a compra.

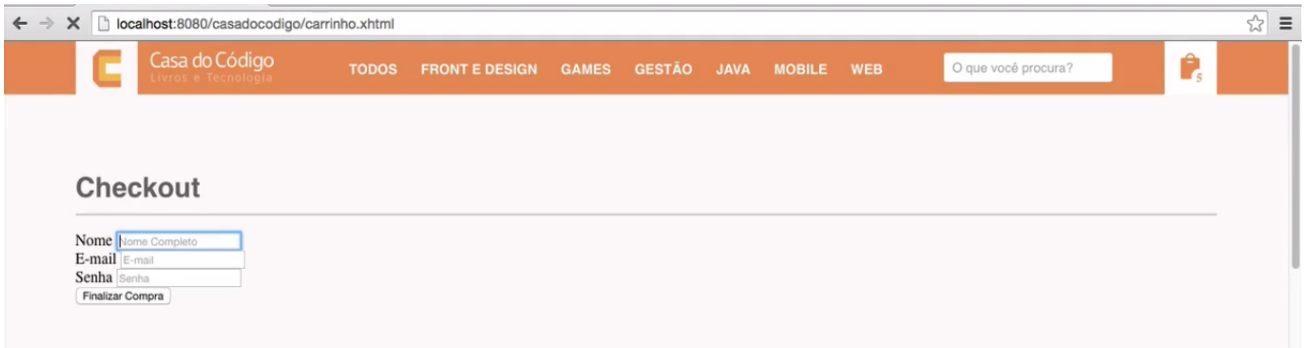
Vamos importar a tela de checkout, a `checkout.xhtml`, para a pasta "webapp". Não nos aprofundaremos em nada específico. A única parte do código de diferente das outras telas é o formulário:

```
<form method="post" jsf:id="formCheckout">
    <div class="form-group">
        <label>Nome</label>
        <input class="form-control" value="" autofocus="autofocus" placeholder="Nome Completo" ,
    </div>
    <div class="form-group">
        <label>E-mail</label>
        <input class="form-control" value="" type="email" placeholder="E-mail" />
    </div>
    <div class="form-group">
        <label>Senha</label>
        <input class="form-control" value="" type="password" placeholder="Senha" />
    </div>
    <button class="btn btn-primary" type="submit">Fechar Compra</button>
</form>
```

Para chegarmos ao formulário, saímos da tela do Carrinho através do botão "Finalizar Compra". O passo é simples:

```
<button class="formularioDoCarrinho-finalizar-botao" jsf:action="checkout"...>
```

Agora, ao clicar em "Finalizar Compra" caímos na tela de formulário:



Não vamos nos ater ao CSS.

O `value`, como sempre, irá nos levar diretamente para o nosso Bean:

```
<input class="form-control"  autofocus="autofocus" placeholder="Nome Completo" jsf:value="#{che

<input class="form-control" type="email" placeholder="E-mail" jsf:value="#{checkoutBean.usuario
```

```
<input class="form-control" type="password" placeholder="Senha" jsf:value="#{checkoutBean.usuar
```

E no botão:

```
<button class="btn btn-primary" type="submit" jsf:action="#{checkoutBean.finalizar}">Fechar Com
```

Não temos ainda as Classes Usuário ou o CheckoutBean. Vamos criá-las, então dentro da pasta de Beans:

```
package br.com.casadocodigo.loja.beans;

import javax.enterprise.inject.Model;

@Model
public class CheckoutBean {

    private Usuario usuario = new Usuario();

    @Inject
    private UsuarioDao usuarioDao;

    public void finalizar() {
        usuarioDao.salvar(usuario)
    }
}
```

Criemos a Classe Usuário com pacote Models:

```
package br.com.casadocodigo.loja.models

@Entity
public class Usuario {

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer id;

    private String nome;

    private String email;

    private String senha;

}
```

Iremos gerar os *getters* e os *setters*:

```
public Integer getId() {
    return id;
}
```

```
public void setId(Integer Id) {
    this.id = id;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getSenha() {
    return senha;
}

public void setSenha(String senha) {
    this.senha = senha;
}
```

Vamos criar o `UsuarioDao.java` com pacote Daos:

```
package br.com.casadocodigo.loja.daos;

import javax.persistence.Entitymanager;

public class UsuarioDao {

    @PersistenceContext
    private EntityManager manager;
}
```

E vamos criar o método `salvar()` :

```
public class UsuarioDao {

    @PersistenceContext
    private EntityManager manager;

    public void salvar(Usuario usuario) {
        manager.persist(ususario);
    }
}
```

Lembrando que toda vez que nós realizamos uma operação que altera o estado do banco de dados, deverá haver uma transação, então no Bean fazemos:

```
@Transactional
public void finalizar() {
    usuarioDao.salvar(usuario)
}
```

Assim teremos uma transação dentro do método `finalizar()`. Também devemos criar o get e o set do Usuário dentro do Bean:

```
public Usuario getUsuario() {
    return ususario;
}

public void seetUsuario(Usuario usuario) {
    this.usuario = usuario;
}
```

Vamos testar se o código está funcionando. Adicionamos um item no Carrinho, clicamos em "Finalizar Compra" e vamos preencher os dados do usuário nos campos do formulário:

No console podemos ver que foi realisado o `insert`:

```
18:51:17,873 INFO  [stdout] (default task-23) Hibernate:
18:51:17,873 INFO  [stdout] (default task-23)     insert
18:51:17,874 INFO  [stdout] (default task-23)     into
18:51:17,874 INFO  [stdout] (default task-23)         Usuario
18:51:17,874 INFO  [stdout] (default task-23)         (email, nome, senha)
18:51:17,874 INFO  [stdout] (default task-23)     values
18:51:17,874 INFO  [stdout] (default task-23)         (?, ?, ?)
```

Entrando no banco de dados para confirmar:

```
mysql> select * from usuario;
+----+--------------------------+------+--------+
| id | email                    | nome | senha  |
+----+--------------------------+------+--------+
```

```
|  1 | paulo.alvez@caelum.com.br | NULL | 123456 |
+----+---------------------------+------+--------+
1 row in set (0,00 sec)
```

O nome saiu nulo, resolveremos depois, mas os dados foram salvos. Veremos a diante como deixar esse checkout totalmente funcional.