

Nome: _____ Data: ____/____/____

Projeto – Utilizando Prime Faces (Loja Virtual)

1. Alterar o projeto LojaVirtual, adicionando uma nova classe bean chamada FormaPgto.

```
1 package beans;
2
3 import javax.persistence.*;
4
5 @Entity
6 @Table(name = "forma_pgto")
7 public class FormaPgto {
8
9     @Id
10    @GeneratedValue(strategy = GenerationType.IDENTITY)
11    @Column(name = "fpg_id")
12    private int id;
13    @Column(name = "fpg_descricao", length = 20, nullable = true)
14    private String descricao;
15    @Column(name = "fpg_num_max_parco", nullable = true)
16    private int numMaxParc;
17    @Column(name = "fpg_num_padrao_parco", nullable = true)
18    private int numPadraoParc;
19    @Column(name = "fpg_intervalo_dias", nullable = true)
20    private int intervaloDias;
21    @Column(name = "fpg_percentual_acres", nullable = true)
22    private float percentualAcres;
23
24    public int getId() {
25        return id;
26    }
27    public void setId(int id) {
28        this.id = id;
29    }
30    public String getDescricao() {
31        return descricao;
32    }
33    public void setDescricao(String descricao) {
34        this.descricao = descricao;
35    }
36    public int getNumMaxParc() {
37        return numMaxParc;
38    }
39    public void setNumMaxParc(int numMaxParc) {
40        this.numMaxParc = numMaxParc;
41    }
42    public int getNumPadraoParc() {
43        return numPadraoParc;
44    }
45    public void setNumPadraoParc(int numPadraoParc) {
46        this.numPadraoParc = numPadraoParc;
47    }
48    public int getIntervaloDias() {
49        return intervaloDias;
50    }
51    public void setIntervaloDias(int intervaloDias) {
52        this.intervaloDias = intervaloDias;
53    }
54    public float getPercentualAcres() {
55        return percentualAcres;
56    }
57    public void setPercentualAcres(float percentualAcres) {
58        this.percentualAcres = percentualAcres;
59    }
60 }
61 }
```

Nome: _____ Data: ____/____/____

2. Criar a classe de Persistência das Formas de Pagamento:

```
1 package persistencia;
2
3 import java.io.Serializable;
4
5 public class FormaPgtoDAO implements Serializable {
6
7     private static final long serialVersionUID = 1L;
8
9     public static void inserir(FormaPgto formaPgto) {
10         Session sessao = HibernateUtil.getSessionFactory().openSession();
11         Transaction t = sessao.beginTransaction();
12         sessao.save(formaPgto);
13         t.commit();
14         sessao.close();
15     }
16
17     public static void alterar(FormaPgto formaPgto) {
18         Session sessao = HibernateUtil.getSessionFactory().openSession();
19         Transaction t = sessao.beginTransaction();
20         sessao.update(formaPgto);
21         t.commit();
22         sessao.close();
23     }
24
25     public static void excluir(FormaPgto formaPgto) {
26         Session sessao = HibernateUtil.getSessionFactory().openSession();
27         Transaction t = sessao.beginTransaction();
28         sessao.delete(formaPgto);
29         t.commit();
30         sessao.close();
31     }
32
33     public static List<FormaPgto> listagem(String filtro) {
34         Session sessao = HibernateUtil.getSessionFactory().openSession();
35         Query consulta;
36         List<FormaPgto> lista = null;
37         if (filtro.trim().length() == 0) {
38             consulta = sessao.createQuery("from FormaPgto order by fpg_descricao");
39         }
40         else {
41             consulta = sessao.createQuery("from FormaPgto where fpg_descricao like :parametro order by fpg_descricao");
42             consulta.setString("parametro", "%" + filtro + "%");
43         }
44         lista = consulta.list();
45         sessao.close();
46         return lista;
47     }
48
49     public static FormaPgto pesqId(int valor) {
50         Session sessao = HibernateUtil.getSessionFactory().openSession();
51         Query consulta = sessao.createQuery("from FormaPgto where fpg_id = :parametro");
52         consulta.setInteger("parametro", valor);
53         sessao.close();
54         return (FormaPgto) consulta.uniqueResult();
55     }
56 }
57
58
59
60 }
```

Nome: _____ Data: ____/____/____

3. Criar a classe de Controle:

```
1 package negocio;
2
3 import java.io.Serializable;
4
5 @ManagedBean
6 @SessionScoped
7 public class FormaPgtoCtrl implements Serializable{
8
9     private static final long serialVersionUID = 1L;
10    private FormaPgto formaPgto = new FormaPgto();
11    private String filtro = "";
12
13    public FormaPgto getFormaPgto() {
14        return formaPgto;
15    }
16    public void setFormaPgto(FormaPgto formaPgto) {
17        this.formaPgto = formaPgto;
18    }
19
20    public String getFiltro() {
21        return filtro;
22    }
23    public void setFiltro(String filtro) {
24        this.filtro = filtro;
25    }
26
27    public List<FormaPgto> getListagem() {
28        return FormaPgtoDAO.listagem(filtro);
29    }
30
31    public String actionGravar() {
32        FacesContext context = FacesContext.getCurrentInstance();
33        if (formaPgto.getId() == 0) {
34            FormaPgtoDAO.inserir(formaPgto);
35            context.addMessage(null, new FacesMessage("Sucesso", "Inserido com sucesso!"));
36        }
37        else {
38            FormaPgtoDAO.alterar(formaPgto);
39            context.addMessage(null, new FacesMessage("Sucesso", "Alterado com sucesso!"));
40        }
41        return "/admin/lista_formaPgto";
42    }
43
44    public String actionInserir() {
45        formaPgto = new FormaPgto();
46        return "/admin/lista_formaPgto";
47    }
48
49    public String actionExcluir() {
50        FormaPgtoDAO.excluir(formaPgto);
51        return "/admin/lista_formaPgto";
52    }
53
54    public void onRowSelect(SelectEvent event) {
55        FacesMessage msg = new FacesMessage("Forma de Pagamento Selecionada",
56            String.valueOf(((FormaPgto) event.getObject()).getId()));
57        FacesContext.getCurrentInstance().addMessage(null, msg);
58    }
59 }
```

4. Não esquecer de mapear a classe bean de Formas de Pagamento no arquivo: hibernate.cfg.xml.

```
<!-- Mapeamento das Classes -->
<mapping class="beans.Produto"/>
<mapping class="beans.Cliente"/>
<mapping class="beans.Fone"/>
<mapping class="beans.FormaPgto"/>
```

Nome: _____ Data: ____/____/____

5. Adicionar ao arquivo pom.xml, a dependência do prime faces.

```
<dependency>
  <groupId>org.primefaces</groupId>
  <artifactId>primefaces</artifactId>
  <version>5.3</version>
</dependency>
```

6. Criar a página xhtml para gerenciar as Formas de Pagamento, conforme o código a seguir (salvar o arquivo na pasta admin):

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3
4 <html xmlns="http://www.w3.org/1999/xhtml"
5       xmlns:ui="http://java.sun.com/jsf/facelets"
6       xmlns:h="http://java.sun.com/jsf/html"
7       xmlns:f="http://java.sun.com/jsf/core"
8       xmlns:p="http://primefaces.org/ui">
9
10 <h:head>
11   <title>Lista de Formas de Pagamento</title>
12 </h:head>
13 <h:body>
14   <h:form id="form">
15
16     <p:growl id="msgs" showDetail="true" />
17
18     <div align="right">
19       <h:commandLink>
20         <p:graphicImage value="/publico/imagens/excel.png" width="24"/>
21         <p:dataExporter type="xls" target="tbl" fileName="f" />
22       </h:commandLink>
23
24       <h:commandLink>
25         <p:graphicImage value="/publico/imagens/pdf.png" width="24"/>
26         <p:dataExporter type="pdf" target="tbl" fileName="f"/>
27       </h:commandLink>
28
29       <h:commandLink>
30         <p:graphicImage value="/publico/imagens/csv.png" width="24"/>
31         <p:dataExporter type="csv" target="tbl" fileName="f" />
32       </h:commandLink>
```

Nome: _____ Data: ____/____/____

```
33
34<h:commandLink>
35    <p:graphicImage value ="/publico/imagens/xml.png" width="24"/>
36    <p:dataExporter type="xml" target="tbl" fileName="f" />
37</h:commandLink>
38</div>
39
40<p:dataTable id="tbl" var="f" value="#{formaPgtoCtrl.listagem}"
41    paginatorTemplate="{CurrentPageReport} {FirstPageLink} {PreviousPageLink}
42    {PageLinks} {NextPageLink} {LastPageLink}"
43    paginator="true" rows="10" style="margin-bottom:20px">
44
45    <f:facet name="header">
46        <h:outputLabel value="Lista de Formas de Pagamento"/> |
47        <p:commandButton value="Inserir" icon="icon-novo"
48            actionListener="#{formaPgtoCtrl.actionInserir}"
49            oncomplete="PF('cadDialog').show()"
50            update=":form:cadDetail">
51            <p:commandButton>
52    </f:facet>
53
54    <p:column headerText="Código" sortBy="f.id">
55        <h:outputText value="#{f.id}" />
56    </p:column>
57
58    <p:column headerText="Descrição" sortBy="f.descricao">
59        <h:outputText value="#{f.descricao}" />
60    </p:column>
61
62    <p:column headerText="Nº Máx. Parc." sortBy="f.numMaxParc">
63        <h:outputText value="#{f.numMaxParc}" />
64    </p:column>
65
66    <p:column style="width:32px;text-align: center">
67        <p:commandButton update=":form:cadDetail" oncomplete="PF('cadDialog').show()"
68            icon="ui-icon-search" title="View">
69            <f:setPropertyActionListener value="#{f}" target="#{formaPgtoCtrl.formaPgto}" />
70        </p:commandButton>
71    </p:column>
72</p:dataTable>
73
74<p:dialog header="Cadastro de Forma de Pagamento" widgetVar="cadDialog" modal="true"
75    showEffect="fade" hideEffect="fade" resizable="false">
76    <p:outputPanel id="cadDetail" style="text-align:center;">
77        <h:panelGrid columns="2" style="text-align: left" >
78            <h:outputLabel value="Código"/>
79            <p:inputText value="#{formaPgtoCtrl.formaPgto.id}" size="6" maxlength="10" readonly="true" />
80            <h:outputLabel value="Nome"/>
81            <p:inputText value="#{formaPgtoCtrl.formaPgto.descricao}" size="30" maxlength="20"/>
82            <h:outputLabel value="Nº Máx. Parcelas"/>
83            <p:inputText value="#{formaPgtoCtrl.formaPgto.numMaxParc}" size="6" maxlength="3"/>
84            <h:outputLabel value="Nº Padrão de Parcelas"/>
85            <p:inputText value="#{formaPgtoCtrl.formaPgto.numPadraoParc}" size="6" maxlength="3"/>
86            <h:outputLabel value="Intervalo (dias)"/>
87            <p:inputText value="#{formaPgtoCtrl.formaPgto.intervaloDias}" size="6" maxlength="3"/>
88        </h:panelGrid>
89        <h:commandButton value="Excluir" action="#{formaPgtoCtrl.actionExcluir}" update="growl"/> |
90        <h:commandButton value="Gravar" action="#{formaPgtoCtrl.actionGravar}" update="growl"/> |
91        <h:commandButton value="Cancelar" oncomplete="cadDialog.hide()"/>
92    </p:outputPanel>
93</p:dialog>
94
95</h:form>
96</h:body>
97</html>
```

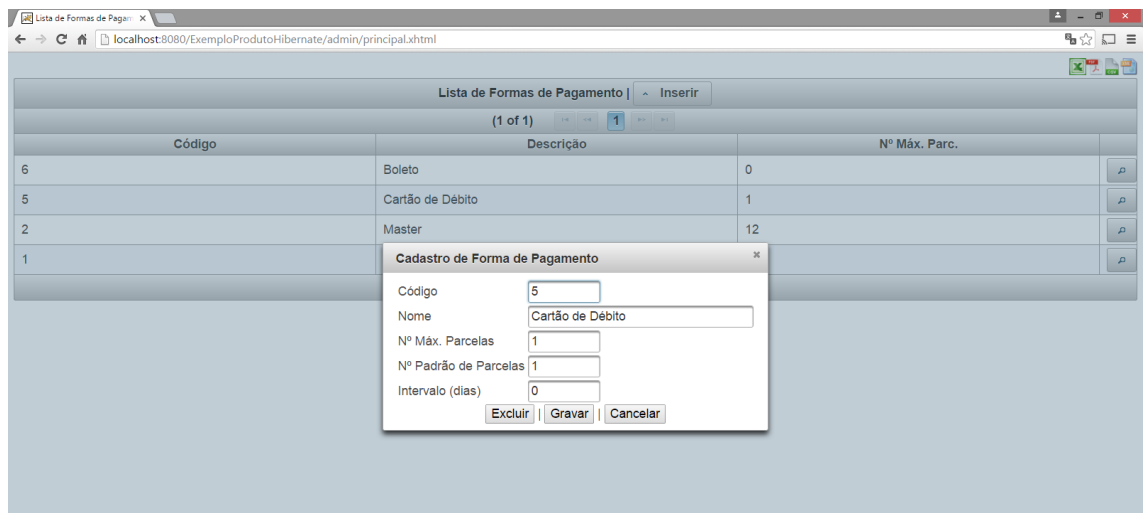
Nome: _____ Data: ____/____/____

Desta forma o cadastro ficará com o seguinte layout:



Código	Descrição	Nº Máx. Parc.
6	Boleto	0
5	Cartão de Débito	1
2	Master	12
1	Visa	8

E ao clicar no botão inserir ou alterar, aparecerá uma janela modal, com o seguinte layout:



Cadastro de Forma de Pagamento

Código:

Nome:

Nº Máx. Parcelas:

Nº Padrão de Parcelas:

Intervalo (dias):

7. Não esquecer de rodar o script da criação da tabela de Formas de Pagamento no banco de dados.

```
create table forma_pgto
(
    fpg_id          serial primary key,
    fpg_descricao   varchar(20),
    fpg_num_max_parce integer,
    fpg_num_padrao_parce integer,
    fpg_intervalo_dias integer,
    fpg_percentual_acres float
);
```

8. Executar e fazer os testes no projeto da Loja Virtual.