Em minha aplicação, tenho uma tela onde o usuario insere registros únicos em uma Lista onde vai somando o total e a quantidade inserida.  
Esses registros são notas fiscais e seus respectivos registros dentro desta lista não podem se repetir.  
Realizei uma velidação em um Bean evitando que se repita ,ok Funciona, o Valor total não incrementa e nem a quantidade, mas no grid é adicionado isso visualmente. Como achei estranho, fiz um teste dando um System.out  na lista para ver se está inserindo repetido, e por surepresa não se repete na lista mas si somente na tela fica duplicado. Cada registro inserido aparece uma imagem X para que o usuario não queira inserir esse registro ou inseriu por engano possa clicar e retirar da lista, esse tambem está funcionando internamente, retira da lista e decrementa o valor, ja como citei acima, se o usuario insere repetido os valores não sao somados a quantidade tambem nao incrementa mas se ele retirar esse repetido porque aparece na tela ele decrementa ficando um valore negativo e o numero negativo.  
  
Vou postar os codigos e imagens passo a passo.   
  
 Bean Controller : O método que adiciona o registro é adicionarNotaAoMapa()  
  
/\*\*  
 \*  
 \* @author victor  
 \*/  
@Controller("mapaController")  
@Scope("session")  
public class MapaController {  
  
    private MapaTransporte mapa;  
    private ItensMapaTransporte itensMapa;  
    private HashMap<Long, ItensMapaTransporte> impt = new HashMap<Long, ItensMapaTransporte>();  
    private ItemMapa itemMapa;  
    private List<MapaTransp> mapaTransp = new ArrayList<MapaTransp>(); //lista onde sera armazena os itens  
    private HashMap<Integer, MapaTransp> mapaTranspMap = new HashMap<Integer, MapaTransp>();  
    private HashMap<Long, ItemMapa> itensMapaMap = new HashMap<Long, ItemMapa>();  
    private HashMap<Integer, ItemMapa> itensValidaGrid = new HashMap<Integer, ItemMapa>();  
    private List<ItemMapa> itemList = new ArrayList<ItemMapa>();  
    private Date dataMapa;  
    private Integer notaF;  
    private Integer vol = null;  
    private String fornecimento;  
    @Resource  
    private DaoGenerico<MapaTransporte, Integer> mapatransporteDao;  
    @Resource  
    private DaoGenerico<ItensMapaTransporte, Integer> itensmapaDao;  
    @Resource  
    private DaoGenerico<NotaFiscal, Integer> notafiscalDao;  
    @Resource  
    private DaoGenerico<Empresa, Integer> empresaDao;  
    @Resource  
    private DaoGenerico<Motorista, Integer> motoristaDao;  
    @Resource  
    private DaoGenerico<Veiculo, Integer> veiculoDao;  
    @Resource  
    private DaoGenerico<Rota, Integer> rotaDao;  
    @Resource  
    private DaoGenerico<Modalidade, Integer> modalidadeDao;  
    @Resource  
    private DaoGenerico<Cliente, Integer> clienteDao;  
    @Resource  
    private DaoGenerico<Area, Integer> areaDao;  
    private Map<String, Object> empresaItem = null;  
    private Map<String, Object> motoristaItem = null;  
    private Map<String, Object> veiculoItem = null;  
    private Map<String, Object> rotaItem = null;  
    private Map<String, Object> modalidadeItem = null;  
    private Map<String, Object> areaItem = null;  
    private Integer qtdMapeo = null;  
    private DataModel model;  
    private boolean exibirTab = false;  
    FacesContext fc = FacesContext.getCurrentInstance();  
    HttpSession session = (HttpSession) fc.getExternalContext().getSession(false);  
  
    public String novoMapa() {  
        this.mapa = new MapaTransporte();  
        this.mapa.setEmpresa(new Empresa());  
        this.mapa.setMotorista(new Motorista());  
        this.mapa.setVeiculo(new Veiculo());  
        this.mapa.setRota(new Rota());  
        this.mapa.setModalidade(new Modalidade());  
        this.mapa.setArea(new Area());  
        this.getEmpresa();  
        this.getMotorista();  
        this.getVeiculo();  
        this.getRota();  
        this.getModalidade();  
        this.getArea();  
  
  
        return "goToformmapa";  
  
    }  
  
    public Map<String, Object> getEmpresa() {  
        empresaItem = new LinkedHashMap<String, Object>();  
        for (Iterator<?> iter = empresaDao.todos().iterator(); iter.hasNext();) {  
            Empresa e = (Empresa) iter.next();  
            empresaItem.put(e.getDesc\_nome(), e.getCodempr());// label,value  
        }  
        return empresaItem;  
    }  
  
    public Map<String, Object> getMotorista() {  
        motoristaItem = new LinkedHashMap<String, Object>();  
        for (Iterator<?> iter = motoristaDao.todos().iterator(); iter.hasNext();) {  
            Motorista e = (Motorista) iter.next();  
            motoristaItem.put(e.getFirst\_name() + " " + e.getLast\_name(), e.getCodmotorista());// label,value  
        }  
        return motoristaItem;  
    }  
  
    public Map<String, Object> getVeiculo() {  
        veiculoItem = new LinkedHashMap<String, Object>();  
        for (Iterator<?> iter = veiculoDao.todos().iterator(); iter.hasNext();) {  
            Veiculo e = (Veiculo) iter.next();  
            veiculoItem.put(e.getPlaca(), e.getCodveiculo());// label,value  
        }  
        return veiculoItem;  
    }  
  
    public Map<String, Object> getRota() {  
        rotaItem = new LinkedHashMap<String, Object>();  
        for (Iterator<?> iter = rotaDao.todos().iterator(); iter.hasNext();) {  
            Rota e = (Rota) iter.next();  
            rotaItem.put(e.getDesc\_rota(), e.getId());// label,value  
        }  
        return rotaItem;  
    }  
  
    public Map<String, Object> getModalidade() {  
        modalidadeItem = new LinkedHashMap<String, Object>();  
        for (Iterator<?> iter = modalidadeDao.todos().iterator(); iter.hasNext();) {  
            Modalidade e = (Modalidade) iter.next();  
            modalidadeItem.put(e.getDescModalidade(), e.getIdModalidade());// label,value  
        }  
        return modalidadeItem;  
    }  
  
    public Map<String, Object> getArea() {  
        areaItem = new LinkedHashMap<String, Object>();  
        for (Iterator<?> iter = areaDao.todos().iterator(); iter.hasNext();) {  
            Area e = (Area) iter.next();  
            areaItem.put(e.getDesc\_area(), e.getId());// label,value  
        }  
        return areaItem;  
    }  
  
    // Pesquisa a Nota Fiscal  
    private NotaFiscal verificaNF(Integer nf) {  
        String query = "SELECT n from NotaFiscal n WHERE n.notafiscal = :nota";  
        Map<String, Object> params = new HashMap<String, Object>();  
        params.put("nota", nf);  
        return notafiscalDao.pesqParam(query, params);  
    }  
  
    // Verificar a Existencia do Cliente  
    private Cliente verificaCliente(Integer cli) {  
        String query = "SELECT c FROM Cliente c WHERE c.codCli = :codCli";  
        Map<String, Object> params = new HashMap<String, Object>();  
        params.put("codCli", cli);  
        return clienteDao.pesqParam(query, params);  
    }  
  
    // Verificar a Existencia do Cliente  
    private Integer verificaItensMapa(Long it) {  
        String query = "SELECT i FROM ItensMapaTransporte i WHERE i.cod\_nf = :cod";  
        Map<String, Object> params = new HashMap<String, Object>();  
        params.put("cod", it);  
        params =  (Map<String, Object>) itensmapaDao.pesqParam(query, params);  
        return qtdMapeo = params.size();  
  
    }  
  
  
    
  
  
    public boolean validaItensMapa() {  
        boolean test = false;  
  
        if (listarItensMapaValidaGrid().size() != 0) {  
            Iterator<?> it = listarItensMapaValidaGrid().iterator();  
            Integer n = null;  
            while (it.hasNext()) {  
                ItemMapa i = (ItemMapa) it.next();  
                if (i.getNf() == n) {  
                    test = true;  
  
                    break;  
  
                }  
  
            }  
        }  
        return test;  
    }  
  
  
  
    public void exibirTabPanel() {  
        exibirTab = true;  
    }  
  
    public void ocultarTabPanel() {  
        exibirTab = false;  
    }  
  
    // Adicionar nota ao itens do  Mapa de Transporte  
    public void adicionarNotaAoMapa() {  
  
  
        Iterator<?> i = listarItensMapa().iterator();  
        Integer aux = null;  
        while (i.hasNext()) {  
            ItemMapa it = (ItemMapa) i.next();  
  
            if (it.getNf() == notaF) {  
                aux = it.getNf();  
  
            }  
  
        }  
  
        if (aux != notaF) {  
  
  
            NotaFiscal n = verificaNF(notaF);  
  
            if (n != null) {  
  
                Cliente cli = verificaCliente(n.getCodCli().getCodCli());  
            //    Integer count = verificaItensMapa(n.getCod\_nf());  
  
  
                //   Integer quantMapeo = 0;  
                //  quantMapeo = verificaNfQuant(notaF);  
  
  
                ItemMapa itens = new ItemMapa();  
  
                itens.setCodNf(n.getCod\_nf());  
                itens.setNf(n.getNotafiscal());  
            //    itens.setQuantMapeada(count);  
                itens.setValor(n.getValor());  
                itens.setCliente(cli.getNomeCli());  
                itens.setVolume(vol);  
  
                itensMapaMap.put(itens.getCodNf(), itens);  
                itensValidaGrid.put(itens.getNf(), itens);  
                itemList.add(itens);  
                atualizarQtd();  
                atualizarValor();  
  
            } else {  
                FacesUtils.mensErro("Nota Fiscal não encontrada");  
                String msg = "Nota Fiscal não encontrada";  
                FacesMessage message = new FacesMessage(msg);  
                FacesContext context = FacesContext.getCurrentInstance();  
                context.addMessage("pesquisanfmapa", message);  
            }  
        } else {  
            FacesUtils.mensErro("Nota Fiscal já foi adicionada ao Mapa");  
            String msg = "Nota Fiscal já foi adicionada ao Mapa";  
            FacesMessage message = new FacesMessage(msg);  
            FacesContext context = FacesContext.getCurrentInstance();  
            context.addMessage("pesquisanfmapa", message);  
  
        }  
  
        System.out.println(listarItensMapa().size());  
    }  
  
    public void gravarMapa() {  
  
        if (listarItensMapa().size() > 0) {  
  
            Empresa e = empresaDao.pesquisarPorId(mapa.getEmpresa().getCodempr());  
            Veiculo vei = veiculoDao.pesquisarPorId(mapa.getVeiculo().getCodveiculo());  
            Modalidade mod = modalidadeDao.pesquisarPorId(mapa.getModalidade().getIdModalidade());  
            Rota rota = rotaDao.pesquisarPorId(mapa.getRota().getId());  
            Motorista motorista = motoristaDao.pesquisarPorId(mapa.getMotorista().getCodmotorista());  
  
            mapa.setEmpresa(e);  
            mapa.setVeiculo(vei);  
            mapa.setModalidade(mod);  
            mapa.setRota(rota);  
            mapa.setMotorista(motorista);  
  
            // pegar o usuario  
  
            mapa.setUser((Usuario) session.getAttribute("userlogged"));  
  
            mapa.setData\_lanc(new Timestamp((new Date()).getTime()));  
  
            mapa.setValor\_tot(mapa.getSubTotal());  
            MapaTransporte mapaid = mapatransporteDao.salvar(mapa);  
  
  
            // varre os itens do Mapa para guardar no Banco.  
  
            Iterator<?> item = listarItensMapa().iterator();  
  
            while (item.hasNext()) {  
  
                ItemMapa itemMapaGrava = (ItemMapa) item.next();  
  
                // salva os itens no mapa  
  
                itensmapaDao.salvar(new ItensMapaTransporte(  
                        mapaid.getCod\_mapa(),  
                        itemMapaGrava.getCodNf(),  
                        itemMapaGrava.getQuantMapeada(),  
                        itemMapaGrava.getValor(),  
                        itemMapaGrava.getVolume()));  
  
            }  
  
            FacesUtils.mensErro("Gravado com Sucesso");  
            String msg = "Gravado com Sucesso";  
            FacesMessage message = new FacesMessage(msg);  
            FacesContext context = FacesContext.getCurrentInstance();  
            context.addMessage("pesquisanfmapa", message);  
        }  
        limpar();  
    }  
  
    // limpar itens de mapa  
    public void limpar() {  
  
        itemList.clear();  
        itensMapaMap.clear();  
        this.getTodos();  
  
    }  
  
    public DataModel getTodos() {  
  
        return model = new ListDataModel(itemList);  
  
    }  
  
    // lista os pedidos do arquivo importado Itens  
    @SuppressWarnings("unchecked")  
    public List listarItensMapa() {  
        // retorna um ArrayList  
        return new ArrayList(itensMapaMap.values());  
    }  
  
    // lista os itens do mapa para validar se ja existe um  
    @SuppressWarnings("unchecked")  
    public List listarItensMapaValidaGrid() {  
        // retorna um ArrayList  
        return new ArrayList(itensValidaGrid.values());  
    }  
  
    // Atualizar Quantidade e Valor  
    public Integer atualizarQtd() {  
  
        Integer quant = 0;  
  
        Iterator<?> itensMapaNF = listarItensMapa().iterator();  
        while (itensMapaNF.hasNext()) {  
            ItemMapa item = (ItemMapa) itensMapaNF.next();  
  
            quant += 1;  
  
        }  
  
  
        mapa.setQtdTotal(quant);  
  
        return null;  
  
  
  
    }  
  
    // Atualizar e Valor  
    public Double atualizarValor() {  
  
        Double totalGeral = 0.0;  
  
        Iterator<?> itensMapaNF = listarItensMapa().iterator();  
        while (itensMapaNF.hasNext()) {  
            ItemMapa item = (ItemMapa) itensMapaNF.next();  
  
            totalGeral += item.getValor();  
  
        }  
  
  
        mapa.setSubTotal(totalGeral);  
  
        return null;  
  
  
  
    }  
  
    public String remoreItens() {  
        ItemMapa i = getItemMapaParaExcluir();  
        itemList.remove(i);  
        itensMapaMap.remove(i.getCodNf());  
        return null;  
    }  
  
    public ItemMapa getItemMapaParaExcluir() {  
        ItemMapa i = (ItemMapa) model.getRowData();  
        return i;  
    }  
  
    public String removerItensMapa() {  
  
        ItemMapa i = getItemMapaParaExcluir();  
        itemList.remove(i);  
        itensMapaMap.remove(i.getCodNf());  
  
        Integer quant = listarItensMapa().size();  
  
        Double total = mapa.getSubTotal() - i.getValor();  
  
  
        // recalcula  Quantidade  
        mapa.setQtdTotal(quant);  
        mapa.setSubTotal(total);  
  
        return null;  
    }  
  
  
    private Connection getConnection() {  
  
        Connection connection = null;  
  
        try {  
            // Cria a conexao com o banco de dados  
  
            Class.forName("org.postgresql.Driver");  
            String db = "jdbc:postgresql://localhost:5432/webtracking";  
            connection = DriverManager.getConnection(db, "postgres", "viclima");  
  
        } catch (SQLException ex) {  
            ex.printStackTrace();  
        } catch (ClassNotFoundException ex) {  
            ex.printStackTrace();  
        }  
  
        return connection;  
    }  
  
     public void executarRelatorio() throws ParseException, SQLException, IOException{  
  
  
        FacesContext context = FacesContext.getCurrentInstance();  
        HttpServletResponse response = (HttpServletResponse) context.getExternalContext().getResponse();  
  
        //pega o caminho do arquivo .jasper da aplicação  
       InputStream reportStream = context.getExternalContext().getResourceAsStream("/reporter/mapa/mapatransporte.jasper");  
  
        try {  
  
            Connection connection = getConnection();  
            ServletOutputStream servletOutputStream = response.getOutputStream();  
  
                HashMap<String, Integer> map = new HashMap<String, Integer>();  
                map.put("PAR\_CODMAPA", 18);  
  
                 //envia a resposta com o MIME Type PDF  
                response.setContentType("application/pdf");  
                //envia para o navegador o PDF gerado  
                JasperRunManager.runReportToPdfStream(reportStream, servletOutputStream, map, connection);  
  
  
  
            servletOutputStream.flush();  
            servletOutputStream.close();  
  
            connection.close();  
  
        } catch (JRException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        } catch (IOException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        } finally {  
            //evita erro do JSF após completar  
            //a geração do relatório  
            //avisando o FacesContext que a resposta está completa  
            context.responseComplete();  
  
        }  
  
     }  
  
   .... getes e Seters ...  
  
  
Codigo da Tela ...   
  
<?xml version='1.0' encoding='UTF-8' ?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml"  
      xmlns:ui="http://java.sun.com/jsf/facelets"  
      xmlns:h="http://java.sun.com/jsf/html"  
      xmlns:f="http://java.sun.com/jsf/core"  
      xmlns:a4j="http://richfaces.org/a4j"  
      xmlns:rich="http://richfaces.org/rich"  
      xmlns:ajax="https://ajax4jsf.dev.java.net/ajax">  
  
    <style>  
        .scrolls{  
            width:300px;  
            height:200px;  
            overflow:auto;  
        }  
    </style>  
    <body>  
        <ui:composition template="../layouts/template.xhtml">  
  
            <ui:define name="content">  
  
                <rich:tabPanel switchType="ajax" id="tabmapa">  
                    <rich:tab label="Mapeamento de Transporte" id="mapatransporte">  
  
                        <h:form id="mapa">  
                            <rich:panel style="width:800px;height:300px">   
                                <f:facet name="header">  
                                    <h:outputText value="Dados para Mapeamento de Carga"/>  
                                </f:facet>  
                               <h:commandLink value="Imprimir"  action="#{mapaController.executarRelatorio}" />  
  
                                <h:panelGrid columns="2">  
                                    <h:outputText  styleClass="rich-label"  value="Data do Mapa" />  
                                    <rich:calendar required="true" value="#{mapaController.mapa.data\_mapa}"  datePattern="dd/MM/yyyy"/>  
  
  
                                    <h:outputText  styleClass="rich-label" value="Trasnportadora:" />  
                                    <h:selectOneMenu  id="empresa"  required="true" default="Selecione Empresa" value="#{mapaController.mapa.empresa.codempr}"  >  
                                        <f:selectItems length="30" value="#{mapaController.empresa}"/>  
                                    </h:selectOneMenu>  
  
  
  
                                    <h:outputText  styleClass="rich-label" value="Area:" />  
                                    <h:selectOneMenu  id="rota"  required="true" default="Selecione Rota" value="#{mapaController.mapa.area.id}"  >  
                                        <f:selectItems length="30" value="#{mapaController.area}"/>  
                                    </h:selectOneMenu>  
  
                                    <h:outputText  styleClass="rich-label" value="Motorista:" />  
                                    <h:selectOneMenu  id="motorista" required="true"  default="Selecione Motorista" value="#{mapaController.mapa.motorista.codmotorista}"  >  
                                        <f:selectItems length="30" value="#{mapaController.motorista}"/>  
                                    </h:selectOneMenu>  
  
                                    <h:outputText  styleClass="rich-label" value="Modalidade:" />  
                                    <h:selectOneMenu  id="modalidade"  required="true" default="Selecione Modalidade" value="#{mapaController.mapa.modalidade.idModalidade}"  >  
                                        <f:selectItems length="30" value="#{mapaController.modalidade}"/>  
                                    </h:selectOneMenu>  
  
  
  
                                    <h:outputText  styleClass="rich-label" value="Veiculo:" />  
                                    <h:selectOneMenu  id="veiculo"  default="Selecione Veiculo" value="#{mapaController.mapa.veiculo.codveiculo}"  >  
                                        <f:selectItems length="30" value="#{mapaController.veiculo}"/>  
                                    </h:selectOneMenu>  
  
                                    <h:outputText  styleClass="rich-label" value="Obeservação:" />  
                                    <h:inputText label="Observação"  id="obs" size="80"  value="#{mapaController.mapa.observacao}">  
  
                                    </h:inputText>  
  
                                    <h:panelGrid columns="1">  
                                        <br/>  
  
                                        <a4j:commandButton value="Confirmar" action="#{mapaController.exibirTabPanel}" ajaxSubmit="true"  
                                                           reRender="itensmapa"/>  
  
  
                                    </h:panelGrid>  
  
                                </h:panelGrid>  
  
                            </rich:panel>  
                        </h:form>  
                    </rich:tab>  
  
                    <rich:tab label="Adicionar Pedidos" id="itensmapa" disabled="#{mapaController.exibirTab}">  
                        <h:form>  
                            <rich:panel style="width:800px;height:70px" id="paneladd"  >  
                                <f:facet name="header">  
                                    <h:outputText value="Pesquisa de Notas e Pedidos"/>  
                                </f:facet>  
                                <h:panelGrid columns="3">  
                                    <h:panelGrid columns="3">  
                                        <h:outputText  styleClass="rich-label" value="Volume:" />  
                                        <h:inputText label="Volume"  id="volume" size="11" required="false"  maxlength="11" value="#{mapaController.vol}" >  
                                            <f:validateLength minimum="1" maximum="11" />  
                                        </h:inputText>  
                                    </h:panelGrid>  
                                    <h:panelGrid columns="2">  
                                        <h:outputText  styleClass="rich-label" value="Nota Fiscal:" />  
                                        <h:inputText label="Nota Fiscal"  id="nfadd" size="6" required="false" maxlength="6" value="#{mapaController.notaF}" class="mantemFocus" >  
                                            <f:validateLength minimum="1" maximum="6" />  
                                        </h:inputText>  
                                    </h:panelGrid>  
                                    <a4j:commandButton value="Adicionar" action="#{mapaController.adicionarNotaAoMapa}" ajaxSubmit="true"  
                                                       reRender="listnotas,valortotal,qtdtotal"/>  
                                </h:panelGrid>  
                                <rich:spacer height="0px"/>  
                            </rich:panel>  
  
                            <rich:panel style="width:800px;height:1000px;backgroundColor=#F1F1F1" id="panelGrid"  >  
                                <f:facet name="header">  
                                    <h:outputText value="Informações adicionadas ao Mapa"/>  
                                </f:facet>  
  
                                <h:panelGrid columns="2">  
  
                                    <h:outputText  styleClass="rich-label" value="Valor Total Notas:" />  
                                    <h:outputText styleClass="rich-label" id="valortotal" value="#{mapaController.mapa.subTotal}">  
                                        <f:convertNumber type="currency" />  
                                    </h:outputText>  
                                </h:panelGrid>  
                                <h:panelGrid columns="2">  
  
                                    <h:outputText  styleClass="rich-label" value="Quantidade de Notas:" />  
                                    <h:outputText styleClass="rich-label" id="qtdtotal" value="#{mapaController.mapa.qtdTotal}"/>  
                                </h:panelGrid>  
                                <h:panelGrid columns="2">  
                                    <a4j:commandButton value="Gravar" action="#{mapaController.gravarMapa}" ajaxSubmit="true"  
                                                       reRender="listnotas,valortotal,qtdtotal"/>  
                                </h:panelGrid>  
   
  
                                <rich:dataTable  onRowMouseOver="this.style.backgroundColor='#F1F1F1'"  
                                                 onRowMouseOut="this.style.backgroundColor='#{a4jSkin.tableBackgroundColor}'"  
                                                 cellpadding="0"  border="0" var="itemmapa" frozenColCount="1"  
                                                 height="10px" width="780px" id="listnotas"  columnClasses="col"  
                                                 value="#{mapaController.todos}"   sortMode="single" >  
                                    <rich:column id="numnf">  
                                        <f:facet name="header"><h:outputText styleClass="headerText" value="Numero da Nota" /></f:facet>  
                                        <h:outputText value="#{itemmapa.nf}"  />  
                                    </rich:column>  
                                    <rich:column id="cliente">  
                                        <f:facet name="header"><h:outputText  styleClass="headerText" value="Cliente" /></f:facet>  
                                        <h:outputText value="#{itemmapa.cliente}"  />  
                                    </rich:column>  
                                    <rich:column id="quant">  
                                        <f:facet name="header"><h:outputText  styleClass="width:20px" value="Cont. Mapeamento" /></f:facet>  
                                        <h:outputText value="#{itemmapa.quantMapeada}"   />  
                                    </rich:column>  
                                    <rich:column id="volume">  
                                        <f:facet name="header"><h:outputText   value="Vol." /></f:facet>  
                                        <h:outputText value="#{itemmapa.volume}"   />  
                                    </rich:column>  
                                    <rich:column id="valor">  
                                        <f:facet name="header"><h:outputText  styleClass="headerText" value="Valor da Nota" /></f:facet>  
                                        <h:outputText value="#{itemmapa.valor}"   />  
                                    </rich:column>  
                                    <rich:column id="action">  
                                        <f:facet name="header"><h:outputText  styleClass="headerText" value="Excluir" /></f:facet>  
                                        <center><a4j:commandLink  reRender="listnotas,valortotal,qtdtotal"  action="#{mapaController.removerItensMapa}">  
                                        <h:graphicImage value="/img/error.gif" border="0" /></a4j:commandLink></center>  
  
                                    </rich:column>  
  
                                </rich:dataTable>  
  
  
                            </rich:panel>  
                             
                        </h:form>  
  
                    </rich:tab>  
  
                </rich:tabPanel>  
            </ui:define>  
        </ui:composition>  
  
    </body>  
</html>  
  
Tela onde estou inserindo os Itens com um exemplo de uma nota 596088 duplicada.  
  
  
  
  
Tela onde clico no botao vermelho do registro Coluna(Excluir), removendo os registros.  
  
  
   
  
Atenciosamente   
  
Victor

[**Voltar ao topo**](http://www.devmedia.com.br/forum/adicionar-dados-em-um-grid-sem-duplicar-jsf-richfaces/10732#topoforum)

ANDREI INOUE HIRATA  
ANDREI INOUE HIRATADevMaster

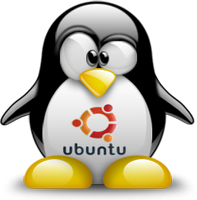
Star

País: Brasil  
Estado: SP  
Cidade: andradina  
Mensagens: 219

Ícone Post  Postado em: 29/10/2009 13:17:27

Eu recomendaria assim.Apenas levando em consideração a lógica e não o código fonte para seu problema  
Cada vez que vc clicar em adicionar, pegue o valor o codigo e compare com a grid na linha de codigo da nota.  
  
Para ser mais exato, vc pega o codigo do seu objeto  
Depois da um loop na grid e veja se na coluna 1 até o final das linhas existe este codigo  
se existir da alerta  
se nao existir permita isto  
  
para saber a coluna da grid = col e linha = row  
  
Assim funcionaria legal..

[**Voltar ao topo**](http://www.devmedia.com.br/forum/adicionar-dados-em-um-grid-sem-duplicar-jsf-richfaces/10732#topoforum)

Victor Lima  


País: Brasil  
Estado: RJ  
Cidade: Rio de Janeiro  
Mensagens: 80

Ícone Post  Postado em: 29/10/2009 13:47:53

Boa Tarde,  
  
Eu fiz isso mas no codigo do controller, o que vc fala e fazer isso na tela tambem, eu entendi realemente,  
agora fazer e complicado ate porque sou novato e eh a primeira vez que faco isso.   
  
Voce poderia me dar um exemplo ?  
  
obrigado  
  
Victor

[**Voltar ao topo**](http://www.devmedia.com.br/forum/adicionar-dados-em-um-grid-sem-duplicar-jsf-richfaces/10732#topoforum)

ANDREI INOUE HIRATA  
ANDREI INOUE HIRATADevMaster

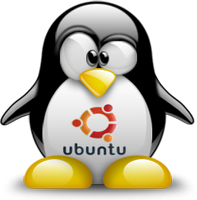
Star

País: Brasil  
Estado: SP  
Cidade: andradina  
Mensagens: 219

Ícone Post  Postado em: 29/10/2009 16:34:28

Então amigo eu vou facilitar seu trabalho.. Toda vez que a pessoa clicar no BOTAO INSERIR VALOR. Adicione o valor em uma lista. Assim atualize sua tabela,se usar ajax = comando reRender..algo assim quando o usuario for adicionar o mesmo valor..E agora oq devo fazer?   SIMPLES..   pegue o valor digitado percorra a lista e veja se este valor é igual, se for igual vc incrimenta o valor..SE posse produto,vc incrementaria a quantidade ouuu se for igual vc nao deixa fazer nada..       VEJA QUE É MAIS FACIL TRABALHAR COM LISTA.ISTO SERVE PARA DATA MODEL..VOU TE PASSAR UM EXEMPLO BABA DA LISTA.Eu acho que ligar uma tabela em jsf com a lista vc ja sabe ne? NA DUVIDA>>.   rich:dataTable rowClasses="classOne,classTwo"  
                                id = "tblLocais"  
                                rows = "10"  
                                columnClasses="col"  
                                **value="#{listaBean.adicionado}"**                                var='item'  
                                border="1"  
                                cellpadding="2"  
                                cellspacing="0">                     <f:facet name="header">  
                        <h:outputText value="#{msg.langTblLocais}"/>  
                    </f:facet>   aee a sua lista.Aqui exemplo de percorrer.       List lista = new ArrayList();     
... preencha a lista com seus objetos;     
for (Iterator iterator = lista.iterator(); iterator.hasNext(); ) {     
    SuaClasse obj = (SuaClasse) iterator.next();     
    System.out.println (obj.x1 + "," + obj.x2);     
}  OU EXEMPLO List<Pessoa> ps = new ArrayList<Pessoa>();  
ps.add(new Pessoa("Maria 1.5"));  
for (Pessoa p : ps)  
  System.out.println(p.getNome());     assim seu codigo seria   List<Pessoa> ps = new ArrayList<Pessoa>();  
for (Pessoa p : ps)  
if ValorCapturadoNoJedit ==p.codigo { faca algo } else { faca nada }  
  DEU PARA ENTENDER?  

[**Voltar ao topo**](http://www.devmedia.com.br/forum/adicionar-dados-em-um-grid-sem-duplicar-jsf-richfaces/10732#topoforum)

Victor Lima  


País: Brasil  
Estado: RJ  
Cidade: Rio de Janeiro  
Mensagens: 80

Ícone Post  Postado em: 09/11/2009 19:24:30

boa tarde favor fechar chamado  
  
atenciosamente Victor

[**Voltar ao topo**](http://www.devmedia.com.br/forum/adicionar-dados-em-um-grid-sem-duplicar-jsf-richfaces/10732#topoforum)

[Responder](http://www.devmedia.com.br/forum/add/?p=10732)