

Gerador de código JSP baseado em projeto de banco de dados MySQL

Juliane Menin

Orientanda

Prof. Alexander Roberto Valdameri

Orientador



Roteiro

- Introdução e Objetivos
- Fundamentação Teórica
- Especificação
- Implementação
- Apresentação da Ferramenta
- Conclusão
- Extensões

Introdução

- JSP
- Servlets
- Diferenças JSP e Servlets
- Banco de Dados MySQL
- Geração de Código
- Ferramentas Geração de Código

Objetivos

- Desenvolver uma ferramenta para a geração de código para tecnologia JSP, a partir das definições encontradas em uma base de dados MySQL.

Motivação

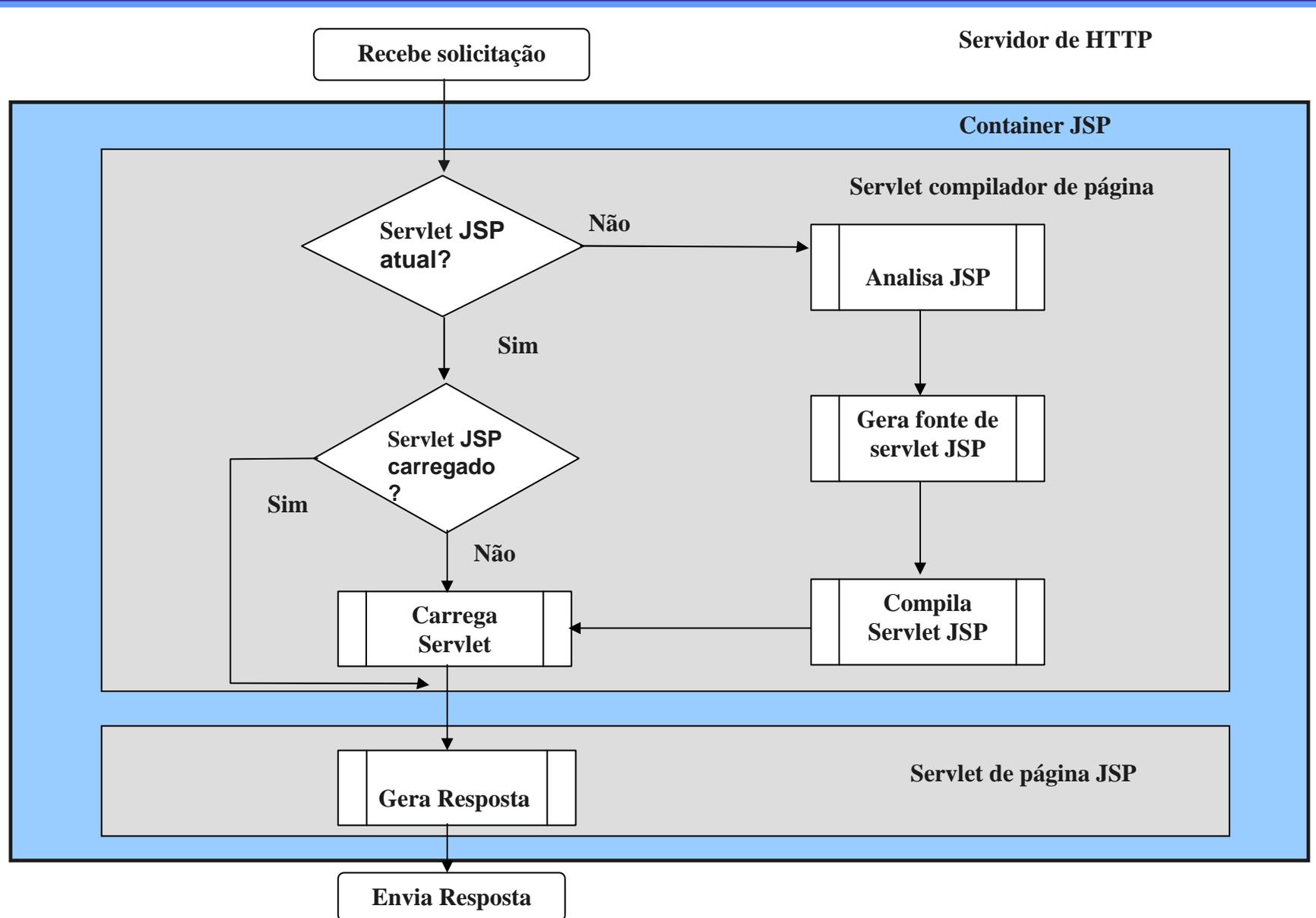
- Auxiliar os programadores
- Diminuir a necessidade de programação
- Evitar código inconsistente

JSP

- Java Server Pages
- Tecnologia baseada em Java
- Simplifica o desenvolvimento de páginas web
- É compilado na forma de um servlet
- Separa a programação da lógica
- Reutilização

Funcionamento do JSP

Fundamentação Teórica



Servlets

- Estende as funcionalidades do servidor
- São independentes de plataforma
- Camada intermediária entre Cliente HTTP e servidor Web
- Possui três métodos : *init*, *destroy*, *service*

JSP X Servlets

Servlets



O HTML é escrito no código Java

JSP



O código Java é escrito no HTML

JSP X Servlets

Servlet = Java

- Fácil de entender
- Controla o sistema
- Classes pequenas
- Lógica pura



Programador

Servlet = Java + HTML

- HTML atrapalha o código Java
- Difícil de entender HTML
- Fácil de entender Java
- Lógica com visualização
- Classes gigantes



Programador

JSP = HTML

- Fácil de entender HTML
- Controla a visualização
- Visualização pura



Designer

JSP = HTML + Java

- Código Java atrapalha o HTML
- Visualização com lógica
- Fácil de entender HTML
- Difícil de entender Java



Designer

Banco de Dados MySQL

- Open Source
- Fácil de usar
- Baixo custo
- Suporta diferentes tipos de tabelas

Tipos de tabelas MySQL

Tabela	Características
MyISAM	Tabela padrão do MySQL; Melhor desempenho para leitura Não possui integridade referencial; Permite o controle de transações; Há limite de tamanhos; São tabelas rápidas;
INNODB	Possui integridade referencial; Tipo de tabela transacional; Armazenamentos de dados em <i>tablespace</i> ; Requer mais espaço em memória;

Tipos de tabelas MySQL

Tabela	Características
MERGE	Divisão de tabelas grandes em tabelas pequenas; Coleção de tabelas MyISAM idênticas;
HEAP	São armazenadas em memória; São rápidas; Conteúdo volátil; Ideal para tabelas que são consultadas com muita frequência;
BDB	Fácil de manipular o controle de transações; Recuperação automática de dados;

Geração de Código

- Qualidade
- Consistência
- Decisões não previstas
- Mais tempo para o projeto

O Código gerado não possui erro de

Apenas gerar os templates e executar o gerador

Não consome muito tempo com tarefas repetitivas

Ferramentas para Gerar Código

- DBDesigner

Integra Criação, Modelagem, desenvolvimento e manutenção dos dados

- CodeCharge

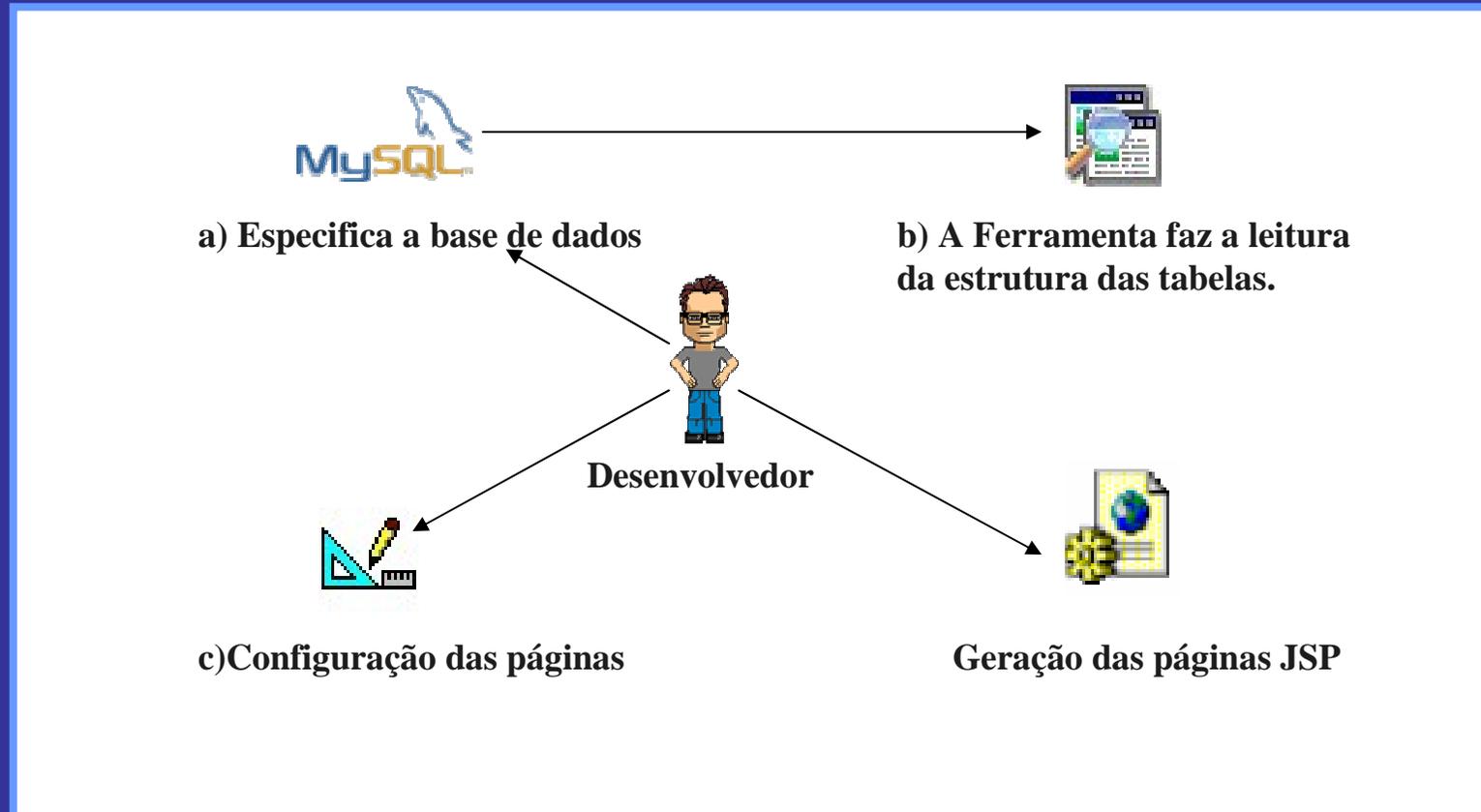
Gera código para várias linguagens

- AspSys

Gera código para as rotinas de cadastros em ASP

Foi desenvolvida em ASP

Visão Geral da Ferramenta



Casos de uso

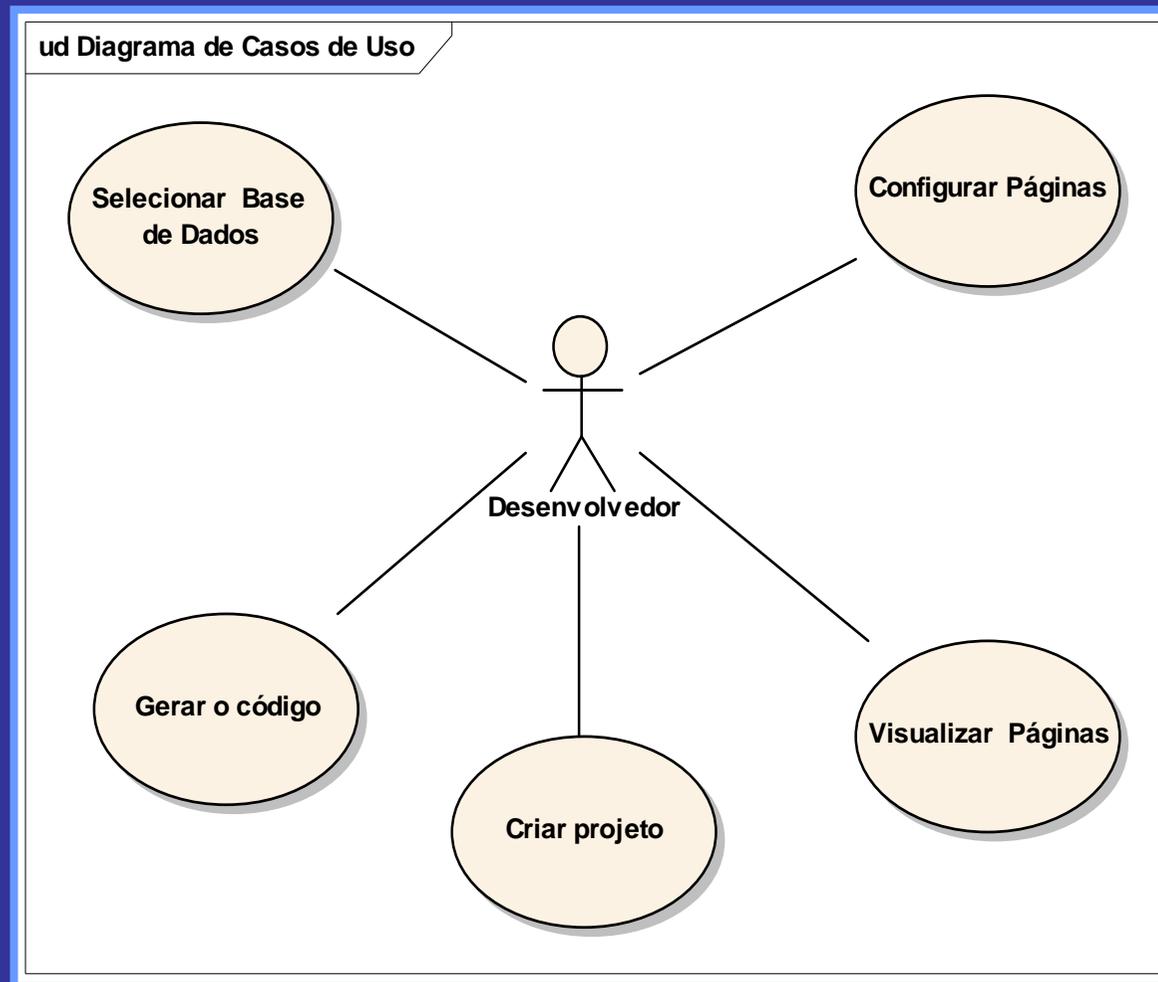
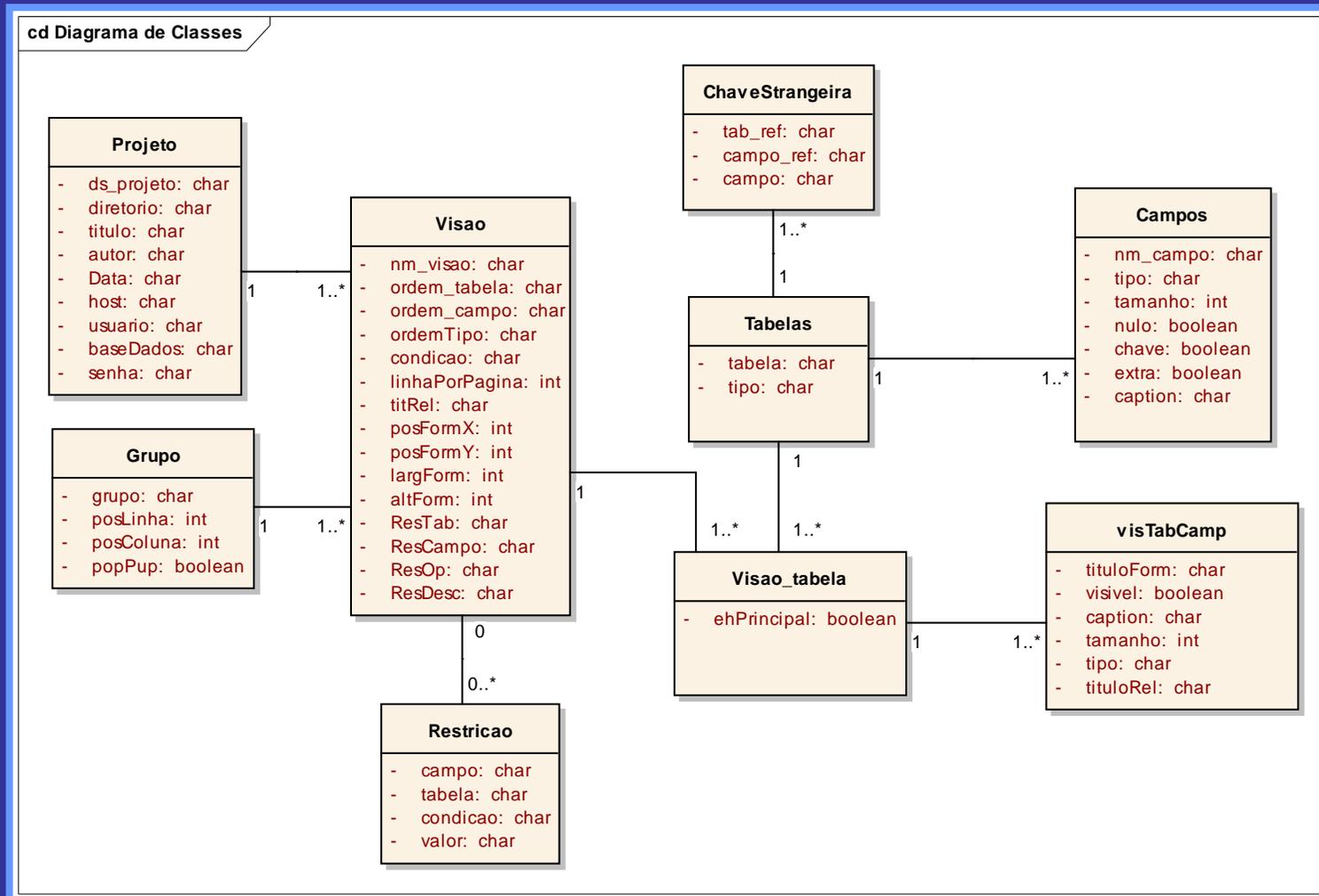


Diagrama de classes

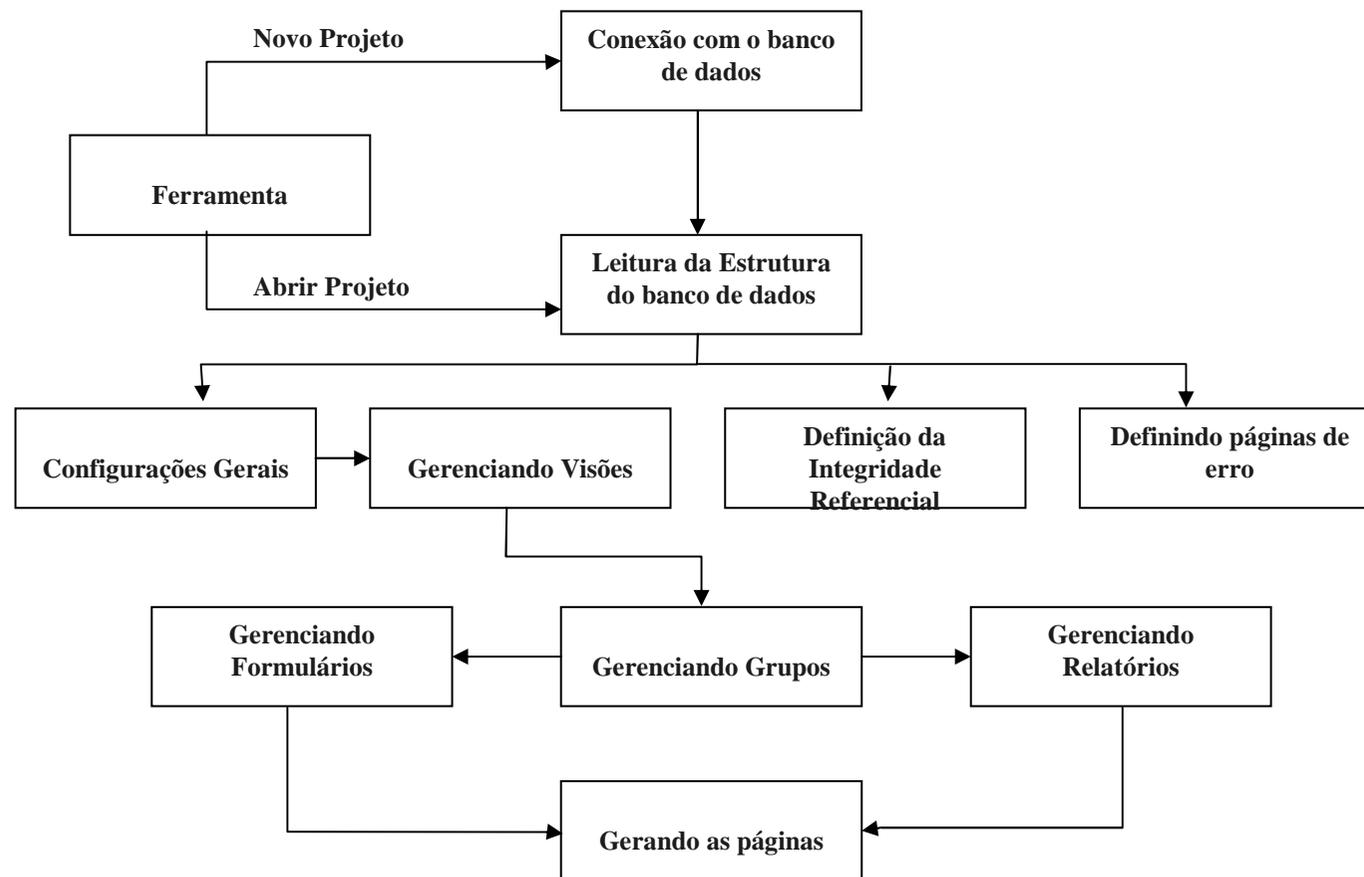
Especificação



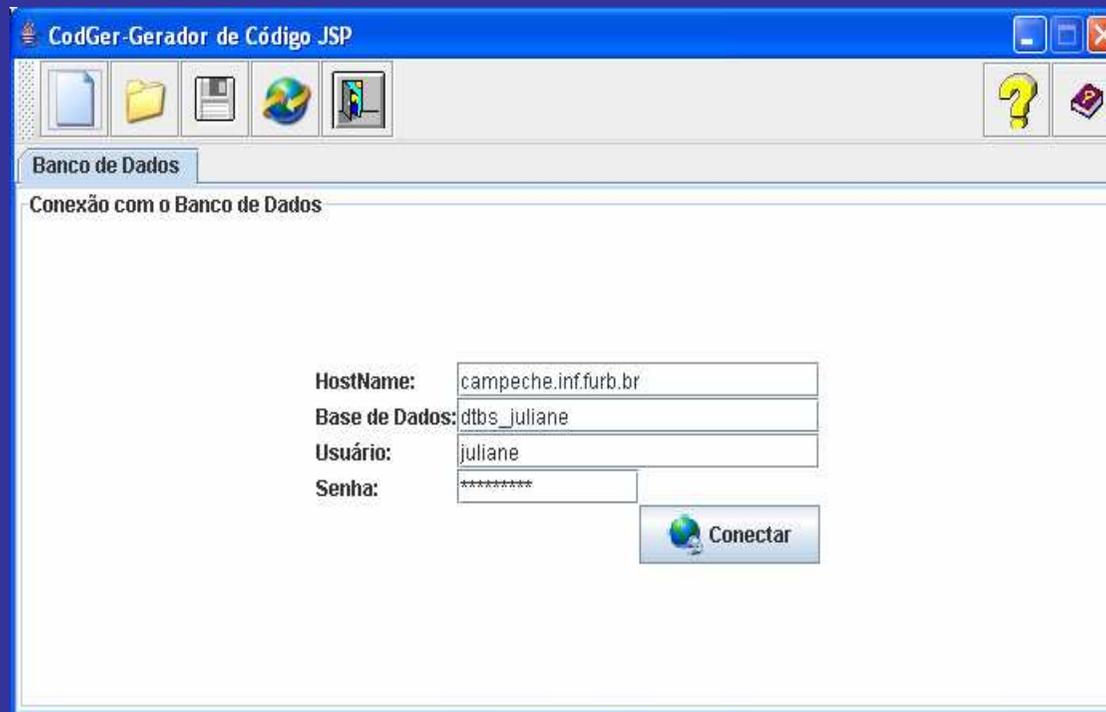
Técnicas e Ferramentas

- Java JDSK 1.5
- Eclipse 3.0
- MySQL
- JBoss

Arquitetura



Conexão com o Banco



The screenshot shows a Windows-style application window titled "CodGer-Generador de Código JSP". The window has a standard toolbar with icons for file operations and help. Below the toolbar, there is a tab labeled "Banco de Dados" and a section titled "Conexão com o Banco de Dados". The main area contains four input fields for database connection details:

HostName:	<input type="text" value="campeche.inf.furb.br"/>
Base de Dados:	<input type="text" value="dtbs_juliane"/>
Usuário:	<input type="text" value="juliane"/>
Senha:	<input type="password" value="*****"/>

Below the input fields is a button labeled "Conectar" with a globe icon.

Leitura da Estrutura das Tabelas

```
private void carregaTabs(){ //lê todas as tabelas da base
ResultSet rsTab;
try {
    rsTab = this.cn.SQLexecuteQuery ("show table status");
    if (rsTab != null){
        while (rsTab.next()){
            Tabela tab = criaTabelas(rsTab);
            tabelas.add(tab);
        }
    }
    catch (Exception e){System.err.println (e.getMessage());
}

private void carregaEstrutura(){ //lê a estrutura dos campos da tabela
    ResultSet rsCamp;
    try {
        Iterator i = tabelas.iterator();
        while ( i.hasNext()) {
            Tabela tp = (Tabela) i.next();
            rsCamp = this.cn.SQLexecuteQuery ("Desc "+tp.getTabela());
            if (rsCamp != null)
                while (rsCamp.next()){
                    Campos camp = criaCampos(rsCamp,tp.getTabela().toString());
                    campos.add(camp);
                }
        }
    }
    catch (SQLException e) { System.err.println (e.getMessage());
}
```

Leitura da Estrutura das Tabelas

```
private Campos criaCampos(ResultSet rsCamp, String tabela){ //instancia os campos {
    try{
        Campos camp = new Campos();

        camp.setTabela(tabela);
        camp.setCampo(rsCamp.getString(1));
        String tipo = rsCamp.getString(2);
        //verificafa o tipo do campo
        camp.setTipo(verificaTipo(tipo));
        //verifica o tamanho
        camp.setTamanho(verificaTam(tipo));
        //verifica se o campo é nulo
        camp.setNulo(verificaNulo(rsCamp.getString(3)));
        //verifica se é chave
        camp.setChave(verificaChave(rsCamp.getString(4)));
        //verifica se é auto_incremento
        camp.setIncremento(verificaIncremento(rsCamp.getString(6)));

        return camp;
    }catch(Exception e){System.out.print(e.getMessage()); return null;}
}
```

Configurações Gerais

Implementação

Projeto: SisGas
Descrição: Este sistema controla a venda de gás
Título Web: SisGas - Sistema para controle de Gás
Data de Criação: 03/06/2005
Autor: Juliane Menin
Diretório: E:\Sistemas

Definindo Integridade

Verificando a Integridade

Tabela Origem: cliente Campos Origem: cd_estado
Tabela Referência: estado Campo Referência: cd_estado

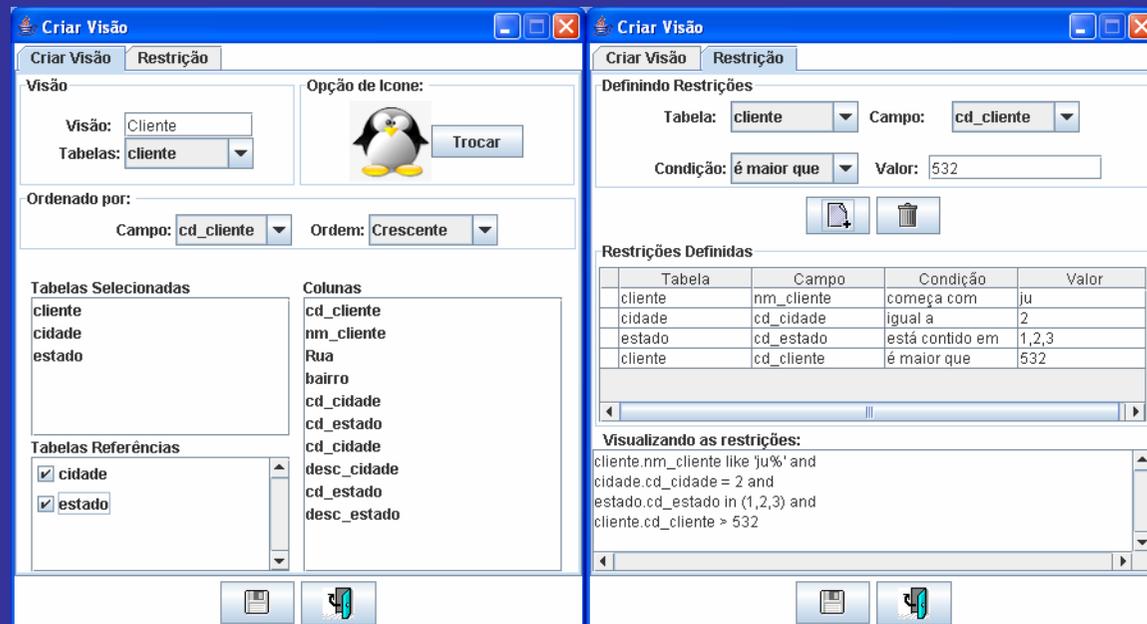
Integridade Referencial

Tabela	Campo	Tabela Referência	Campo Referência
cliente	cd_cidade	cliente	cd_cidade
cliente	cd_estado	estado	cd_estado

Gerenciando visão

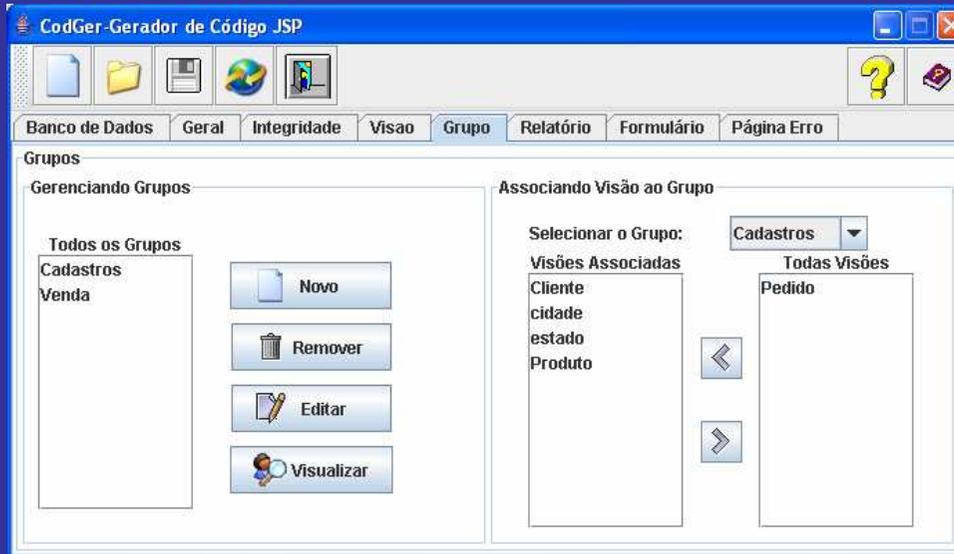


Criando uma visão

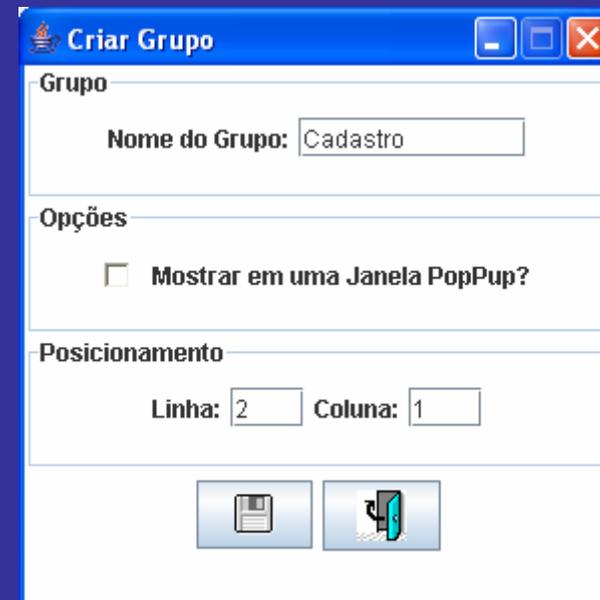


Gerenciando um grupo

Implementação

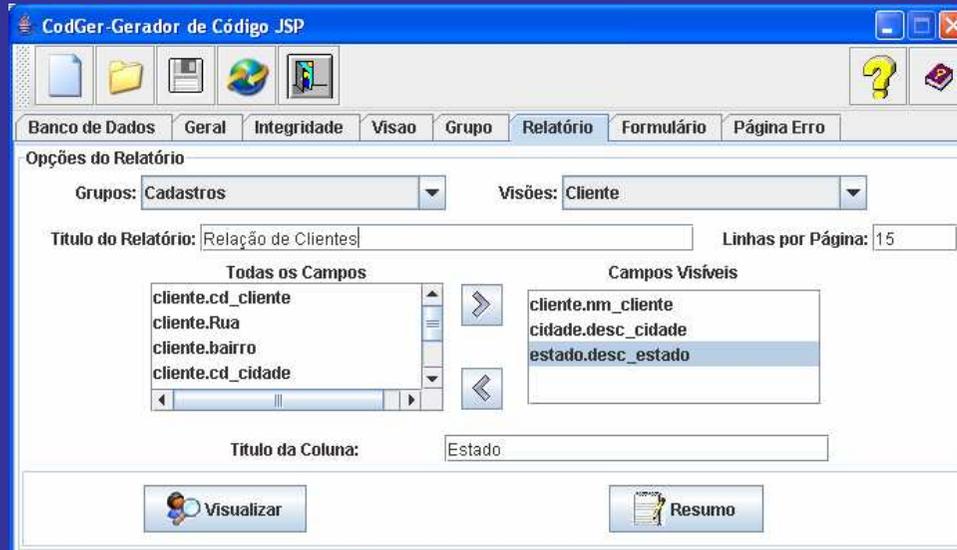


Criando um grupo

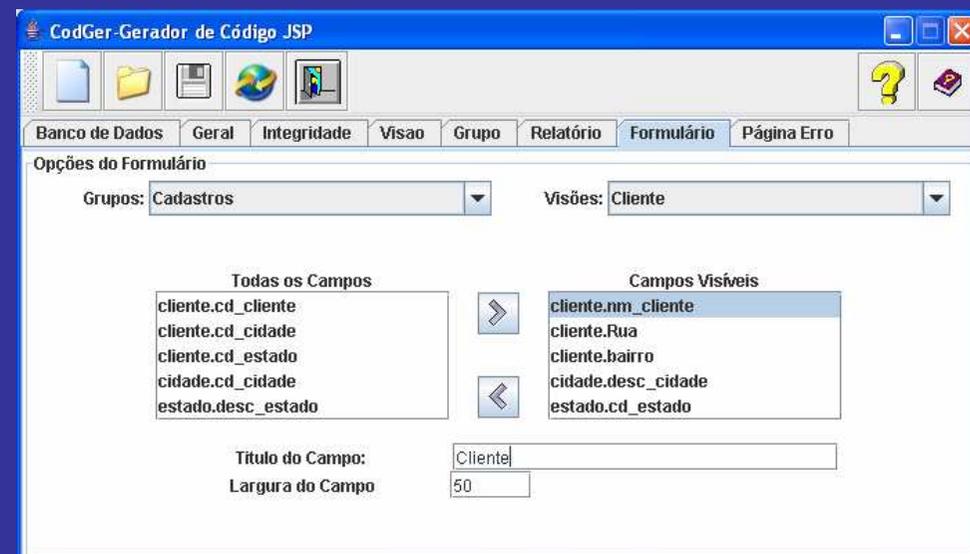


Configurando Relatório

Implementação



Configurando Formulários



Regras para o Formulário

- Se tamanho do campo < 80 = caixa de texto, senão textArea;
- Se campo auto-incremento então campo desabilitado;
- Se campo chave-primária e não visível então oculto
- Se campo chave estrangeira então selected

Visualizando os arquivos

Implementação

Relação de Arquivos

Relação de páginas a serem geradas

Relação de Arquivos

Página	Tipo	Caminho
<input type="checkbox"/> cabecalho.jsp	Apresentação	e:\Sistemas\SisGas.war\cabecalho.jsp
Cliente.java	Fonte	e:\Sistemas\SisGas.war\WEB-INF\classes\SisGa...
Cidade.java	Fonte	e:\Sistemas\SisGas.war\WEB-INF\classes\SisGa...
Estado.java	Fonte	e:\Sistemas\SisGas.war\WEB-INF\classes\SisGa...
Produto.java	Fonte	e:\Sistemas\SisGas.war\WEB-INF\classes\SisGa...
CidadeRef.java	Fonte	e:\Sistemas\SisGas.war\WEB-INF\classes\SisGa...
EstadoRef.java	Fonte	e:\Sistemas\SisGas.war\WEB-INF\classes\SisGa...
cliente_relacao.jsp	Apresentação	e:\Sistemas\SisGas.war\cliente_relacao.jsp
cidade_relacao.jsp	Apresentação	e:\Sistemas\SisGas.war\cidade_relacao.jsp
estado_relacao.jsp	Apresentação	e:\Sistemas\SisGas.war\estado_relacao.jsp
produto_relacao.jsp	Apresentação	e:\Sistemas\SisGas.war\produto_relacao.jsp
erro.htm	Apresentação	e:\Sistemas\SisGas.war\erro.htm
pageNaoSalva.htm	Apresentação	e:\Sistemas\SisGas.war\pageNaoSalva.htm
cliente_incluir.jsp	Inclusão	e:\Sistemas\SisGas.war\cliente_incluir.jsp
cidade_incluir.jsp	Inclusão	e:\Sistemas\SisGas.war\cidade_incluir.jsp
estado_incluir.jsp	Inclusão	e:\Sistemas\SisGas.war\estado_incluir.jsp
produto_incluir.jsp	Inclusão	e:\Sistemas\SisGas.war\produto_incluir.jsp
Conn.java	Conexao	e:\Sistemas\SisGas.war\WEB-INF\classes\Conn.j...
Controlador.java	Controle	e:\Sistemas\SisGas.war\WEB-INF\classes\Control...

Alterando o código

Visualizando Código

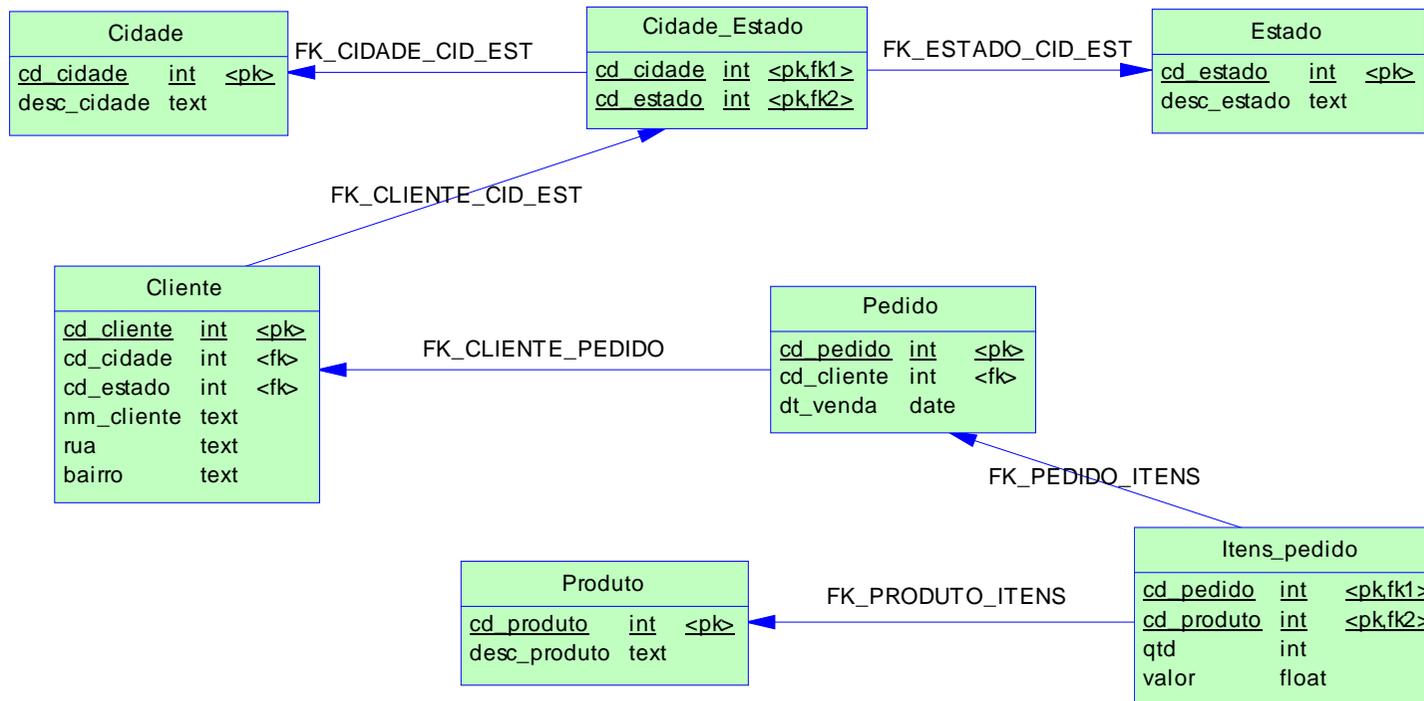
Visualizando a página

Página: cliente_relacao.jsp
Tipo: Apresentação
Caminho: e:\Sistemas\SisGas.war\cliente_relacao.jsp
Codigo:

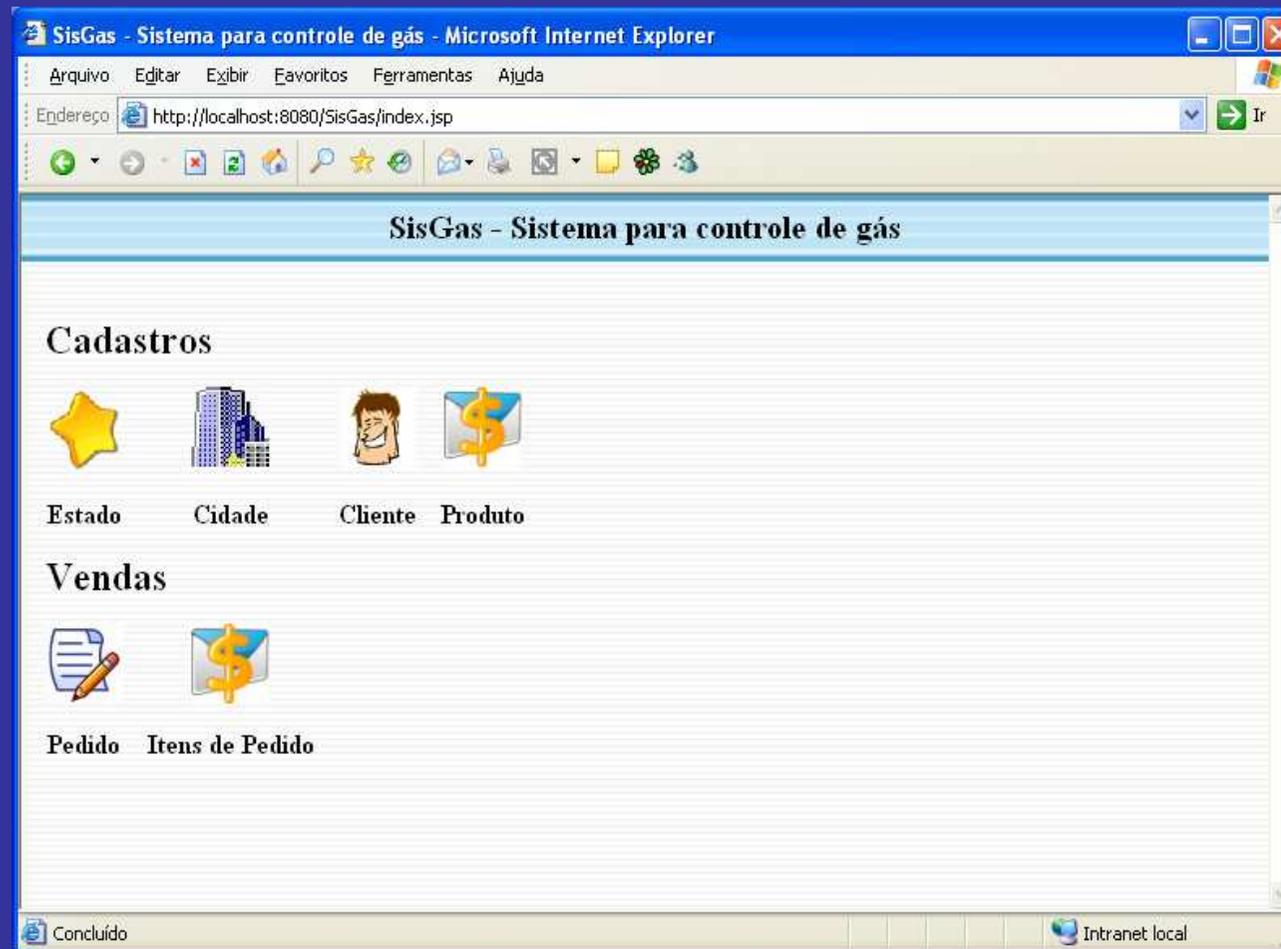
```
<TD id=ca0 COLSPAN=2 WIDTH=2% *&nbsp;sp; *TD *&nbsp;sp; *>
<!-- Dados da Tabela -->
<%
for(int i = 0; i < clientes.size(); i++)
{
    Cliente cliente = (Cliente)clientes.get(i);
    if ((i % 2) == 0)
        out.println("<TR class='cor1' *>");
    else
        out.println("<TR class='cor2' *>");

    out.println("<TD id=dados WIDTH=3% ALIGN=CENTER *><A HREF=altera_cliente.do?cd="+cliente.get
    out.println("<TD id=dados WIDTH=3% ALIGN=CENTER *><A HREF=javascript:excluir("+cliente.getCd
    out.println("<TR *>");
} %>
</TABLE>
```

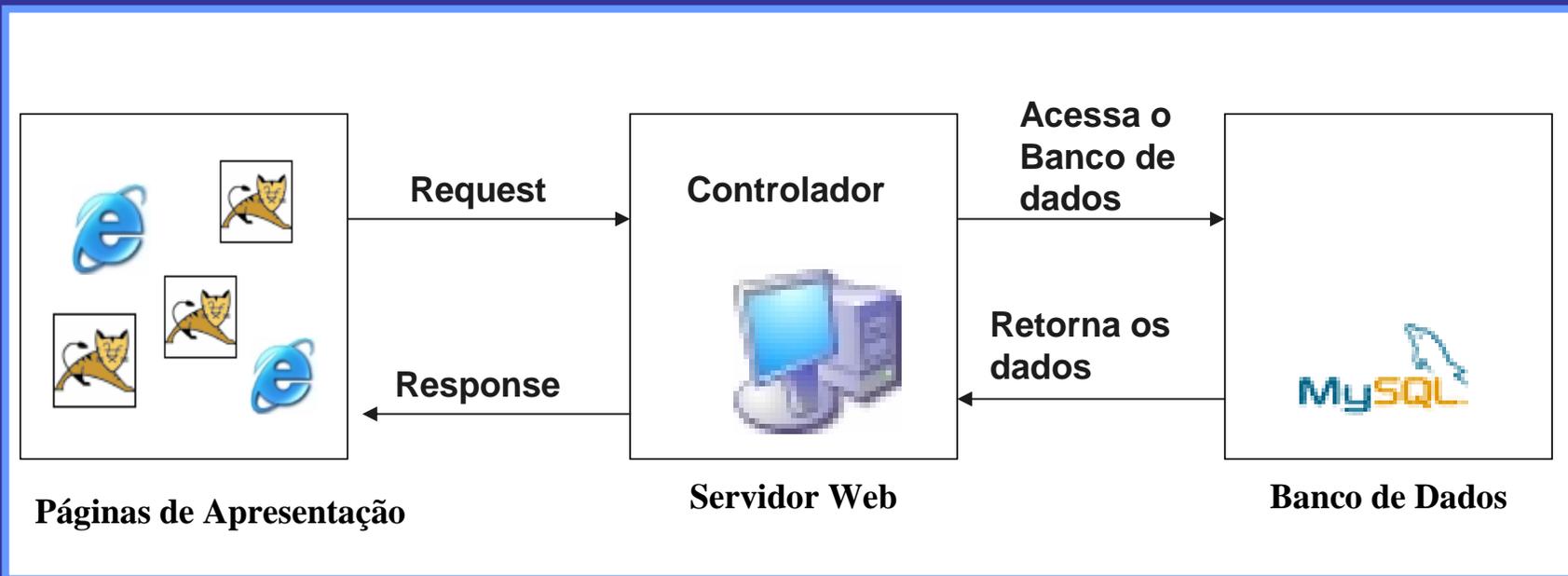
Estudo de caso



Página Inicial do Sistema



Estrutura das páginas geradas



Código do Controlador

Operacionalidade

```
public class Controlador extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, java.io.IOException {
        //decodifica a ação que é para executar de acordo com URL
        String acao = decodificaAcao(request);
        String proximaPagina = "";
        boolean achouPagina = false; //controle para verificar se a página foi encontrada
        if (acao.equals("abrir_Cliente")){
            proximaPagina = abrirCliente(request,response);
            achouPagina = true;
        }
        if(!achouPagina)
            proximaPagina = "/pageNaoSalva.htm";

        getServletContext().getRequestDispatcher(proximaPagina).forward(request, response);
    }
    //Decodifica a ação selecionada
    protected String decodificaAcao(HttpServletRequest request) {
        String caminho = request.getRequestURI();
        String acao = caminho.substring(caminho.lastIndexOf('/')+1, caminho.lastIndexOf(".do"));
        return acao;}
    //Necessário para estender o HttpServlet
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, java.io.IOException { processRequest(request, response); }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, java.io.IOException { processRequest(request, response); }
```

Código do Controlador

Implementação

```
protected String abrirCliente(HttpServletRequest request, HttpServletResponse response) throws
ServletException, java.io.IOException{

    String paginaErro= "/erro.htm";
    String paginaSucesso= "/cliente_relacao.jsp";
    String proximaPagina = paginaErro;
    try {
        ...
    ArrayList cidades = Conn.getListaCliente(param, limitePorPagina);
        request.getSession().setAttribute("listaCliente", clientes);
        proximaPagina = paginaSucesso;
    }catch(Exception e){ System.out.print(e.toString());}
    return proximaPagina; }
```

```
public static ArrayList getListaCliente(int inicio, int qtd){
try{
    String sql = "Select CLIENTE.NM_CLIENTE as CLIENTE, CLIENTE.RUA, CLIENTE.BAIRRO,
CIDADE.DESC_CIDADE as CIDADE, ESTADO.DESC_ESTADO as ESTADO from CLIENTE, CIDADE, ESTADO where
CLIENTE.CD_CIDADE = CIDADE.CD_CIDADE and CLIENTE.CD_ESTADO = ESTADO.CD_ESTADO order by
CLIENTE.CD_CLIENTE asc LIMIT "+inicio+", "+qtd;

    connecta();
    rs = st.executeQuery(sql);
    ArrayList cliente = new ArrayList();
    while (rs.next()){
        Cliente umCliente = criaCliente(rs);
        cliente.add(umCliente); }

    return cliente; }
catch ( Exception e ){ System.out.println(e.getMessage()); return null; } }
```

Código da Página de Relatório

```
<% ArrayList clientes = (ArrayList)session.getAttribute("listaCliente");

...

<!-- Dados da Tabela -->
<% for(int i = 0; i < clientes.size(); i++){
    Cliente cliente = (Cliente)clientes.get(i);
    if ((i % 2) == 0)
        out.println("<TR class='cor1'>");
    else
        out.println("<TR class='cor2'>");
    out.println("<TD id=dados>"+cliente.getNm_cliente()+"</TD>");
    out.println("<TD id=dados>"+cliente.getDesc_cidade()+"</TD>");
    out.println("<TD id=dados>"+cliente.getDesc_estado()+"</TD>");

    out.println("<TD id=dados WIDTH='3%' ALIGN=CENTER><A
    HREF=altera_cliente.do?cd="+cliente.getCd_cliente()+"><IMG SRC='img/editar.png' BORDER=0
    ALT='Editar Cliente'></IMG></A></TD>");

    out.println("<TD id=dados WIDTH='3%' ALIGN=CENTER><A
    HREF=javascript:excluir("+cliente.getCd_cliente()+"><IMG SRC='img/excluir.png' BORDER=0
    ALT='Excluir Cliente'></IMG></A></TD>");

    out.println("</TR>");
} %>
```

Apresentação do Relatório

The screenshot shows a Microsoft Internet Explorer window titled 'SisGas - Sistema para controle de gás'. The address bar shows the URL 'http://localhost:8080/SisGas/abrir_Cliente.do?pr=10'. The main content area displays a table with the following data:

Cliente	Cidade	Estado	
Anderson Benetti	Brusque	Santa Catarina	 
Caroline Alexandra	Blumenau	Santa Catarina	 
Cristiano Menin	São José do Cedro	Santa Catarina	 
Denilson Kruger	Canoinhas	Santa Catarina	 
Gabriel Matias	Blumenau	Santa Catarina	 
Joice Martins	Blumenau	Santa Catarina	 
Márcia Benetti	São José do Cedro	Santa Catarina	 
Robson Luiz Reguim	Bauru	São Paulo	 
Ruan Carlos	Guarujá do Sul	Santa Catarina	 
Willian Roberto	Curitiba	Paraná	 

Below the table, it states: **Quantidade total de Clientes: 25**

At the bottom of the page, there are navigation buttons: 'Menu', 'Novo', a search icon, and four arrow icons (back, forward, home, refresh).

The status bar at the bottom shows 'Concluído' on the left and 'Intranet local' on the right.

Apresentação do Formulário

The screenshot shows a Microsoft Internet Explorer window titled "SisGas - Sistema para controle de gás". The address bar contains "http://localhost:8080/SisGas/chama_Cliente.do". The form is titled "SisGas - Sistema para controle de gás" and contains the following fields:

Código	<input type="text" value="26"/>
Cliente	<input type="text" value="Juliane Menin"/>
Rua	<input type="text" value="Rua Guilherme Poerner"/>
bairro	<input type="text" value="Velha"/>
Cidade	<input type="text" value="Blumenau"/>
Estado	<input type="text" value="Santa Catarina"/>

At the bottom of the form are two buttons: "Salvar" and "Cancelar". The status bar at the bottom of the browser window shows "Concluído" and "Intranet local".

Conclusões

O objetivo foi alcançado gerando código para as rotinas básicas de um sistema como:

- Inclusão
- Exclusão
- Alteração
- Pesquisas
- Relatórios

Nos relatórios é permitida a customização, como exemplo, definir atributos de ordenação e paginação de registros.

Conclusões

- A utilização da IDE eclipse mostrou-se adequada a construção e depuração do software desenvolvido;
- Execução multiplataforma.

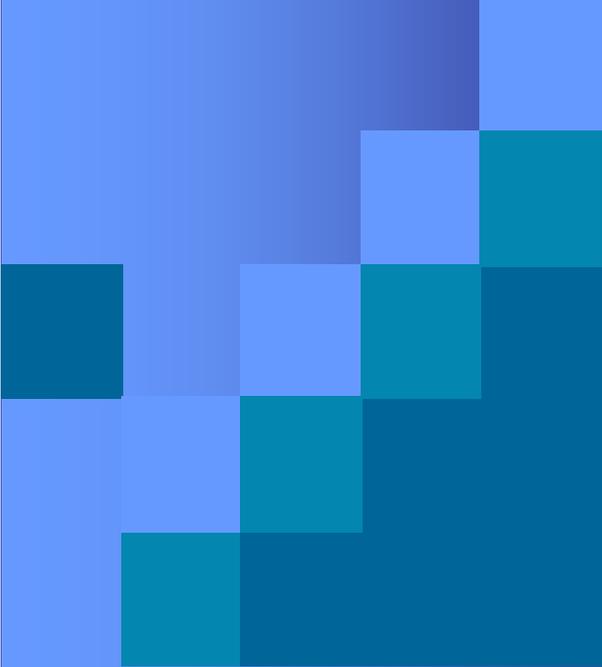
Extensões

- permitir a utilização de outros bancos de dados como PostgreSQL, SQLServer ou Oracle;
- permitir ao desenvolvedor a definição de seu próprio *layout* do aplicativo, através de folhas de estilos;
- permitir a geração de relatórios mais complexos baseados nas funções de grupos.

Relevância

Este trabalho foi muito importante para mim através dele :

- Aumentei o meu conhecimento em Java, que é uma linguagem poderosa;
- Aprendi a desenvolver interfaces em Java;
- Aprendi JSP, que é um tecnologia baseada em Java;



Gerador de código JSP baseado em projeto de banco de dados MySQL

Juliane Menin

Orientanda

Prof. Alexander Roberto Valdameri

Orientador