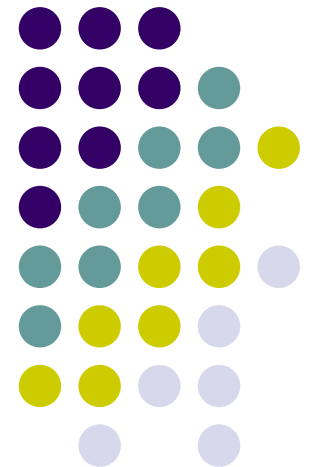


JFreeChart

Diogo Carneiro
Mauricio Ferreira
Savio Sena



Introdução



- O que é JFreeChart ?
- Usabilidades: Applications, servlets, JSP e applets

Vantagens e Funcionalidades



- Licença GNU LGPL (GNU Lesser General Public License).
- Poder construir gráficos facilmente e com poucas linhas de códigos
oder construir gráficos facilmente e com poucas linhas de códigos
s
odem ser exportados para PNG, JPEG, PDF via iText e SVG via
dem ser exportados para PNG, JPEG, PDF via iText e SVG via
teratividade com o usuário do aplicativo: Tool Tips, Zoom, Eventos
suário do aplicativo: Tool Tips, Zoom, Eventos de mouse, anotações
uário do aplicativo: Tool Tips, Zoom, Eventos de mouse, anotações
Tool Tips, Zoom, Eventos de mouse, anotações
gol Tips, Zoom, Eventos de mouse, anotações

Desvantagens



- Não constrói gráficos com domínio em R3
- Esta API não oferece todas as ferramentas necessárias para se fazer gráficos extremamente impressionantes.
(solução: classe Graphics do J2SE)

Instalação



Arquivo	Descrição
jfreechart-1.0.0-rc1.tar.gz	JFreeChart para Linux/Unix
jfreechart-1.0.0-rc1.zip	JFreeChart para Windows

Site: <http://www.jfree.org/jfreechart/index.php>

Javadoc: <http://www.jfree.org/jfreechart/javadoc>

Pacotes do JFreeChart



- **org.jfree.chart**
- **org.jfree.annotations**
- **org.jfree.axis**
- **org.jfree.entity**
- **org.jfree.event**
- **org.jfree.imagemap**
- **org.jfree.labels**
- **org.jfree.needle**
- **org.jfree.plot**
- **org.jfree.renderer**
- **org.jfree.servlet**
- **org.jfree.title**
- **org.jfree.urls**
- **org.jfree.ui**
- **org.jfree.data**
- **org.jfree.data.general**
- **org.jfree.chart.event**

(...)

Principais classes da biblioteca org.jfree.chart



- **Classe ChartUtilities**
- **Classe ChartColor**
- **Classe ChartFrame**
- **Classe ChartPanel**
- **Classe LegendItem**
- **Classe ChartFactory**



Etapas para gerar um gráfico

1. Gerar ou carregar os valores
2. Criar a representação
3. Customizar a aparência e o comportamento
4. Definir a apresentação: Aplicação Swing, salvar o gráfico para um arquivo ou exibir dinamicamente em aplicativos Web



1. Gerar / Carregar valores

Implementar a interface `org.jfree.data.Dataset`

- PieDataset
- CategoryDataset
- XYDataset
- IntervalXYDataset
- HighLowDataset
- IntervalCategoryDataset
- JDBC Datasets

2. Criar representação



- Classe ChartFactory
- Ex.: `JFreeChart chart ChartFactory.createPieChart (`
`“Chart” , dataset, true, true, false);`
- Dataset é passado por referência para a fábrica.

3. Métodos de customização



- Borda
- Título e legendas
- Cor ou imagem de fundo
- Rendering Hints

Métodos de customização

Borda



Desenha uma borda ao redor do gráfico

- Método `setBorderVisible()`
- Controlado pelos métodos: `setBorderPaint()` e `setBorderStroke()`

Métodos de customização

Título



- Método setTitle()
- Posições: top, bottom, left ou right
- Ex.: chart.getTitle().setPosition(RectangleEdge.**BOTTOM**);

Métodos de customização

Legenda



- Método addSubtitle ()
- Ex.: `TextTitle subtitle1 = new TextTitle("A Subtitle");`
`chart.addSubtitle(subtitle1);`

Métodos de customização

Cor de Fundo



- Método setBackgroundPaint()
- Ex.: Paint p = **new** GradientPaint(0, 0, Color.*white*,
1000, 0, Color.*green*));
chart.setBackgroundPaint(p);

Métodos de customização

Imagem de Fundo

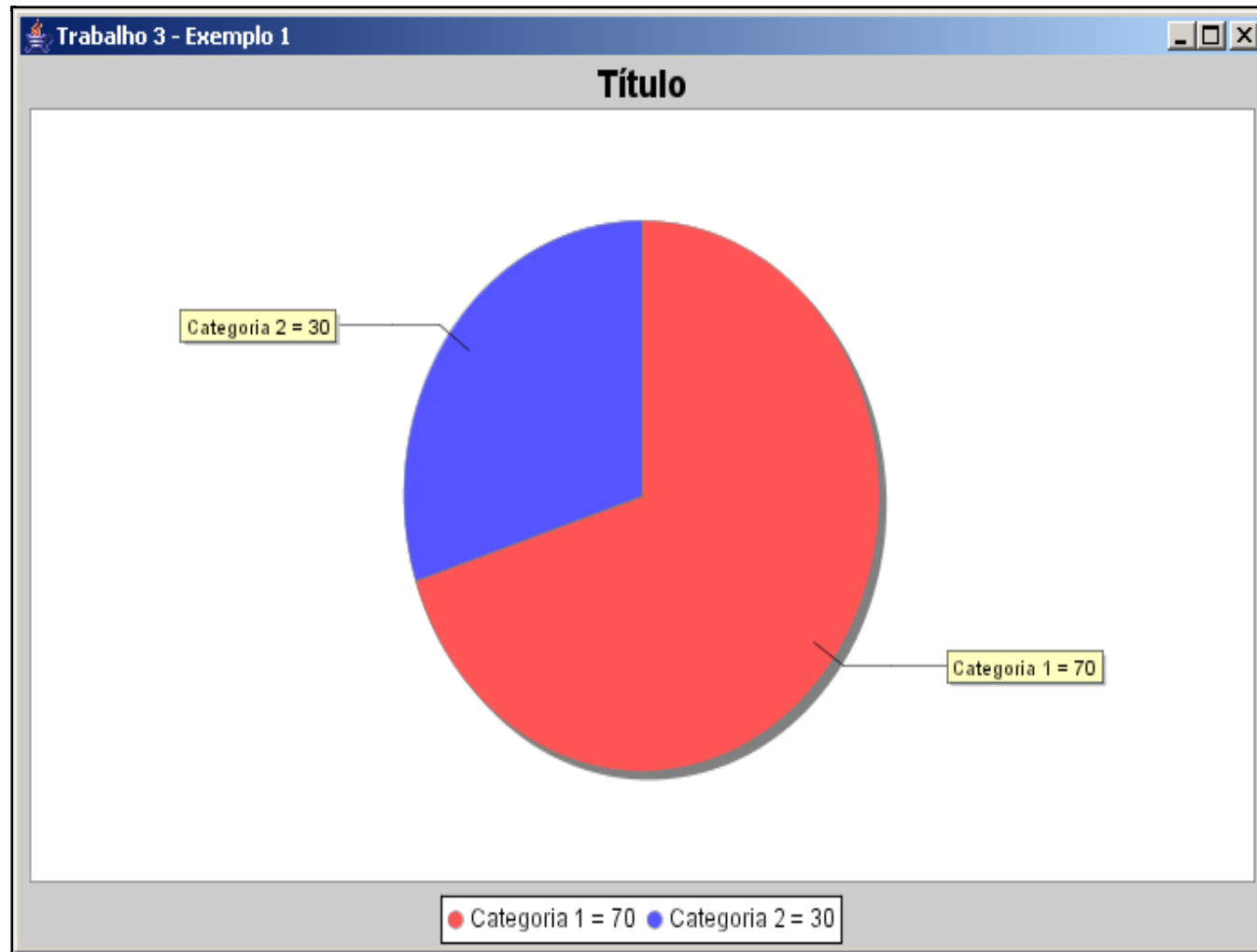
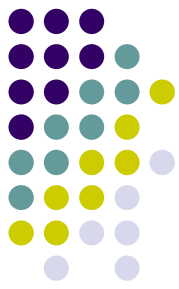


- Método `chart.setBackgroundImage()`
- Ex.: `chart.setBackgroundImage(JFreeChart.INFO.getLogo());`
`chart.setBackgroundImageAlignment(Align.TOP LEFT);`

XXX Mais métodos a seguir com exemplos

Alguns Tipos de Gráficos

Pie Charts



Pie Charts

Exemplo simples:



```
public static void main(String[ ] args)
{
    /* (1) Gerar valores */
    DefaultPieDataset data = new DefaultPieDataset( );
    data.setValue( "Categoria 1" , new Double(70.0));
    data.setValue( "Categoria 2" , new Double(30.0));

    /* (2) Criar representação*/
    JFreeChart chart;
    chart = ChartFactory.createPieChart( "Título" , data, true, true, false );

    /* (3) Customizando a aparência e comportamento */
    chart.setAntiAlias( true );

    /* (4) Definir a apresentação */
    ChartFrame frame = new ChartFrame( "Trabalho 3 – Exemplo 1" , chart );
    frame.pack( );
    frame.setVisible( true );
}
```

Pie Charts

Exemplo simples:



Etapa 1: Gerar valores

Implementar a Interface PieDataset. Uma opção de classe que implementa essa interface é **Classe DefaultPieDataset** (org.jfree.data.general). Ela adiciona métodos bem práticos como o “setValue” que seta valores para o dataset.

```
DefaultPieDataset data = new DefaultPieDataset( );  
data.setValue( “Categoria 1” , new Double(70.0));  
data.setValue( “Categoria 2” , new Double(30.0));
```

Pie Charts

Exemplo simples:



Etapa 2: Criar representação

Utilizar os métodos da classe ChartFactory para criar uma representação. No caso do PieChart os métodos para criá-lo são createPieChart ou createPieChart3D

```
public static JFreeChart createPieChart ( java.lang.String title, PieDataset dataset,  
                                         boolean legend, boolean tooltips, boolean urls )
```

```
JFreeChart chart;
```

```
chart = ChartFactory.createPieChart( “Exemplo” , data, true, true, false );
```

Pie Charts

Exemplo simples:



Etapa 3: Customizando a aparência e comportamento

Utilizar os métodos de classes como JFreeChart, PiePlot e PiePlot3D para customizar o gráfico.

Exemplos:

```
chart.setBackgroundImage  
chart.setAntiAlias
```

```
plot.setExplodePercent  
plot.setLabelFont  
plot.setCircular  
plot.setForegroundAlpha
```

```
chart.setAntiAlias( true );
```

Pie Charts

Exemplo simples:



Etapa 4: Definir a apresentação

No exemplo usamos Swing. Mais a frente mostraremos outras apresentações usando JFreeChart.

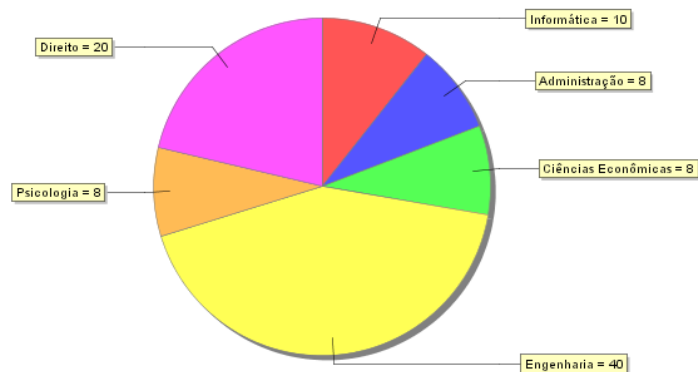
```
ChartFrame frame = new ChartFrame( "Teste" , chart );  
frame.pack( );  
frame.setVisible( true );
```

Pie Charts - Exemplo 2:



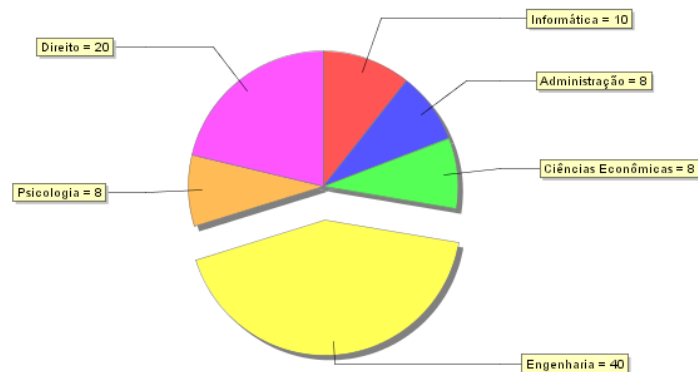
Trabalho 3 - Exemplo 2

Alunos de graduação (Flat Pie Chart)



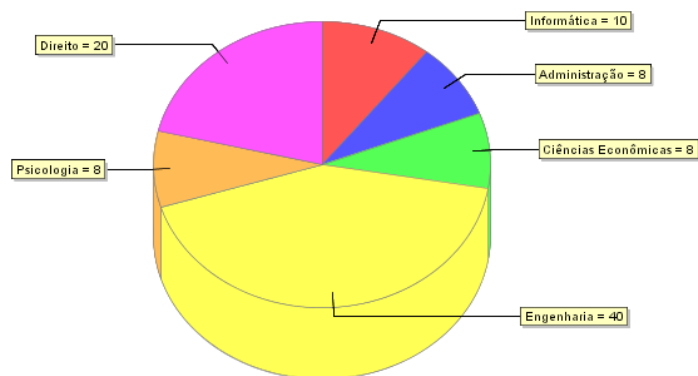
● Informática = 10 ● Administração = 8 ● Ciências Econômicas = 8 ● Engenharia = 40 ● Psicologia = 8
● Direito = 20

Alunos de graduação (Exploded Pie Chart)



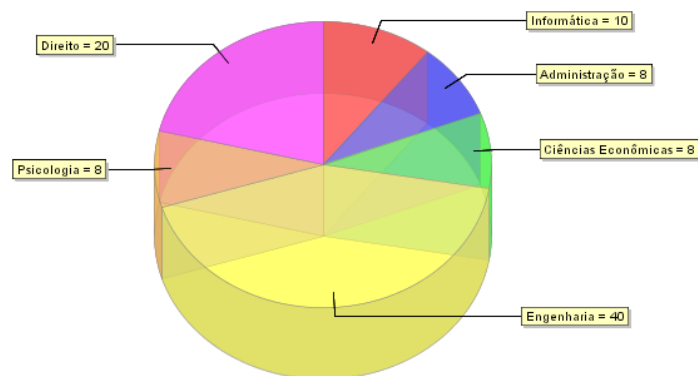
● Informática = 10 ● Administração = 8 ● Ciências Econômicas = 8 ● Engenharia = 40 ● Psicologia = 8
● Direito = 20

Alunos de graduação (3D Pie Chart)



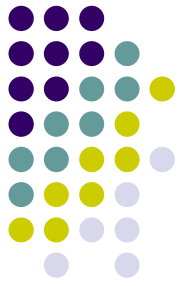
● Informática = 10 ● Administração = 8 ● Ciências Econômicas = 8 ● Engenharia = 40 ● Psicologia = 8
● Direito = 20

Alunos de graduação (3D com transparência)



● Informática = 10 ● Administração = 8 ● Ciências Econômicas = 8 ● Engenharia = 40 ● Psicologia = 8
● Direito = 20

Pie Charts - Exemplo 2:



```
public class Exemplo2 extends JPanel
```

```
{
```

```
    private JFreeChart chart1, chart2, chart3, chart4;
```

```
    private ChartPanel panel1, panel2, panel3, panel4;
```

```
    private DefaultPieDataset dataset = new DefaultPieDataset( );
```

```
    public PieChartExample( )
```

```
{
```

```
        dataset.setValue( "Informática", new Double( 10.0 ) );
```

```
        dataset.setValue( "Administração", new Double( 8.0 ) );
```

```
        dataset.setValue( "Ciências Econômicas", new Double( 8.0 ) );
```

```
        dataset.setValue( "Engenharia", new Double( 40.0 ) );
```

```
        dataset.setValue( "Psicologia", new Double( 8.0 ) );
```

```
        dataset.setValue( "Direito", new Double( 20.0 ) );
```

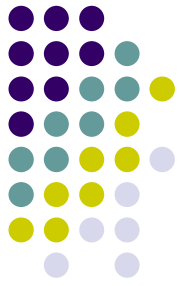
```
        chart1 = ChartFactory.createPieChart ( "Alunos de graduação (Flat)",  
                                             dataset, true, true, false);
```

```
        chart2 = ChartFactory.createPieChart ( "Alunos de graduação (Exploded)",  
                                             dataset, true, true, false);
```

```
        PiePlot plot = ( PiePlot )chart2.getPlot( );
```

```
        plot.setExplodePercent( 3, 0.25 );
```


Pie Charts - Exemplo 2:



```
chart3 = ChartFactory.createPieChart3D ( "Alunos de graduação (3D)" ,  
                                         dataset, true, true, false);
```

```
chart4 = ChartFactory.createPieChart3D ( "Alunos de graduação (3D transp.)" ,  
                                         dataset, true, true, false);
```

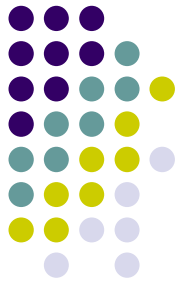
```
PiePlot3D plot4 = ( PiePlot3D )chart4.getPlot( );  
plot4.setForegroundAlpha( 0.6f );
```

```
this.setLayout( new GridLayout( 2, 2 ) );  
this.panel1 = new ChartPanel( chart1 );  
this.panel2 = new ChartPanel( chart2 );  
this.panel3 = new ChartPanel( chart3 );  
this.panel4 = new ChartPanel( chart4 );
```

```
this.add( panel1 );  
this.add( panel2 );  
this.add( panel3 );  
this.add( panel4 );
```

```
}
```

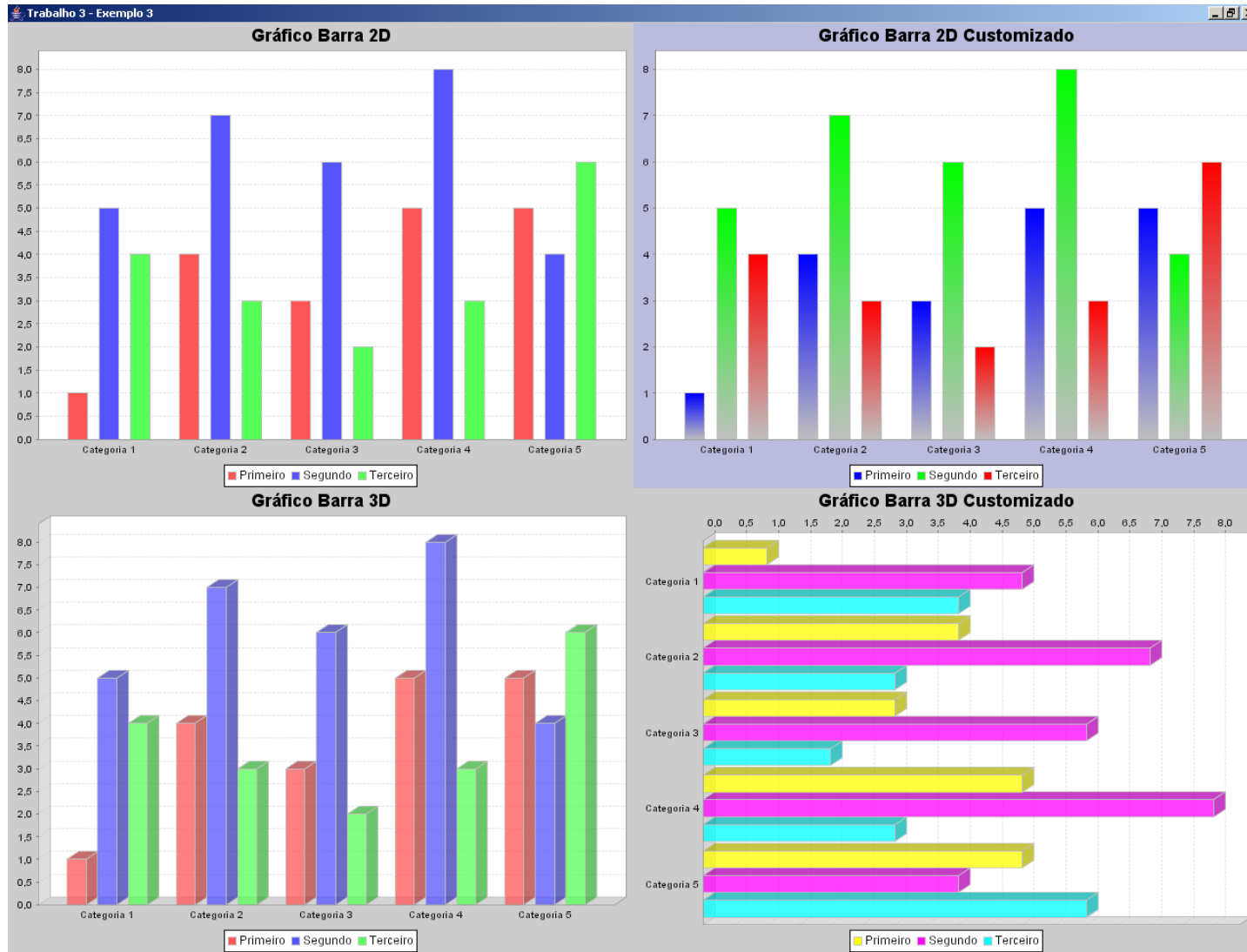
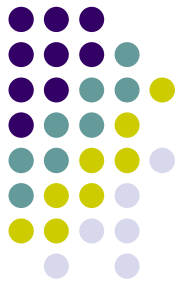
Pie Charts - Exemplo 2:



```
public static void main( String[ ] args )
{
    JFrame frame = new JFrame( "Trabalho 3 - Exemplo 2" );
    PieChartExample chart = new PieChartExample( );
    frame.getContentPane().add( chart, BorderLayout.CENTER );
    frame.setSize( 800, 800 );
    frame.setVisible( true );
    frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
}
}
```

Alguns Tipos de Gráficos

Bar Charts



Alguns Tipos de Gráficos

Bar Charts



- 2D e com efeito 3D
- Interface CategoryDataset
- Zoom Interativo
- Customizações: Cor de fundo, mudar cores das barras, cores gradientes, espaçamento, etc.

Bar Charts - Exemplo:



```
public class GraficosBarra extends JPanel
{
    private JFreeChart chart1, chart2, chart3, chart4;
    private ChartPanel panel1, panel2, panel3, panel4;
    private DefaultCategoryDataset dataset = new DefaultCategoryDataset ( );

    public GraficosBarra( )
    {
        // Linhas
        String series1 = "Primeiro" ; String series2 = "Segundo" ; String series3 = "Terceiro" ;

        // Colunas
        String category1 = "Categoria 1" ; String category2 = "Categoria 2" ;
        String category3 = "Categoria 3" ; String category4 = "Categoria 4" ;
        String category5 = "Categoria 5" ;

        // Criando Valores
        dataset.addValue(1.0, series1, category1); dataset.addValue(4.0, series1, category2);
        dataset.addValue(3.0, series1, category3); dataset.addValue(5.0, series1, category4);
        dataset.addValue(5.0, series1, category5); dataset.addValue(5.0, series2, category1);
        dataset.addValue(7.0, series2, category2); dataset.addValue(6.0, series2, category3);
        dataset.addValue(8.0, series2, category4); dataset.addValue(4.0, series2, category5);
        dataset.addValue(4.0, series3, category1); dataset.addValue(3.0, series3, category2);
        dataset.addValue(2.0, series3, category3); dataset.addValue(3.0, series3, category4);
        dataset.addValue(6.0, series3, category5);
```

Bar Charts - Exemplo:



// Criando representação

```
chart1 = ChartFactory.createBarChart( "Gráfico Barra 2D" , null, null,  
dataset, PlotOrientation.VERTICAL, true, false, false);
```

```
chart2 = ChartFactory.createBarChart( " Gráfico Barra 2D Customizado" , null, null,  
dataset, PlotOrientation.VERTICAL, true, false, false);
```

// Customizações do Chart 2

```
chart2.setBackgroundPaint( new Color(0xBBBBDD));
```

// Pegando a referência do Plot para customização

```
CategoryPlot plot = chart2.getCategoryPlot( );
```

Pegando a referência do Number Axis

```
NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis( );  
rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits( ));
```

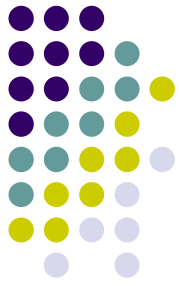
// Pegando a referência do renderer

```
BarRenderer renderer = (BarRenderer) plot.getRenderer();
```

```
GradientPaint gp0 = new GradientPaint( 0.0f , 0.0f , Color.blue, 0.0f , 0.0f , Color.lightGray );  
GradientPaint gp1 = new GradientPaint( 0.0f , 0.0f , Color.green, 0.0f , 0.0f , Color.lightGray );  
GradientPaint gp2 = new GradientPaint( 0.0f , 0.0f , Color.red, 0.0f , 0.0f , Color.lightGray );
```

```
renderer.setSeriesPaint(0, gp0);  
renderer.setSeriesPaint(1, gp1);  
renderer.setSeriesPaint(2, gp2);
```

Bar Charts - Exemplo:



```
chart3 = ChartFactory.createBarChart3D( "Gráfico Barra 3D" , null, null,  
dataset, PlotOrientation.VERTICAL, true, false, false);
```

```
chart2 = ChartFactory.createBarChart3D( " Gráfico Barra 3D Customizado" , null, null,  
dataset, PlotOrientation.HORIZONTAL, true, false, false);
```

// Customizações do Chart 4

```
CategoryPlot plot4 = chart4.getCategoryPlot();
```

```
BarRenderer renderer4 = (BarRenderer) plot4.getRenderer();  
renderer.setDrawBarOutline( true );
```

```
CategoryAxis axis4 = plot4.getDomainAxis();  
axis4.setLowerMargin( 0.02 );  
axis4.setCategoryMargin( 0.10 );  
axis4.setUpperMargin( 0.02 );
```

```
renderer4.setSeriesPaint(0, Color.yellow);  
renderer4.setSeriesPaint(1, Color.magenta);  
renderer4.setSeriesPaint(2, Color.cyan);
```

Bar Charts - Exemplo:

// Criando a apresentação

```
this.setLayout( new GridLayout( 2, 2 ) );
this.panel1 = new ChartPanel( chart1 );
this.panel2 = new ChartPanel( chart2 );
this.panel3 = new ChartPanel( chart3 );
this.panel4 = new ChartPanel( chart4 );
this.add( panel1 );
this.add( panel2 );
this.add( panel3 );
this.add( panel4 );
}

public static void main( String[] args )
{
    JFrame frame = new JFrame( "Trabalho 3 - Exemplo 3" );

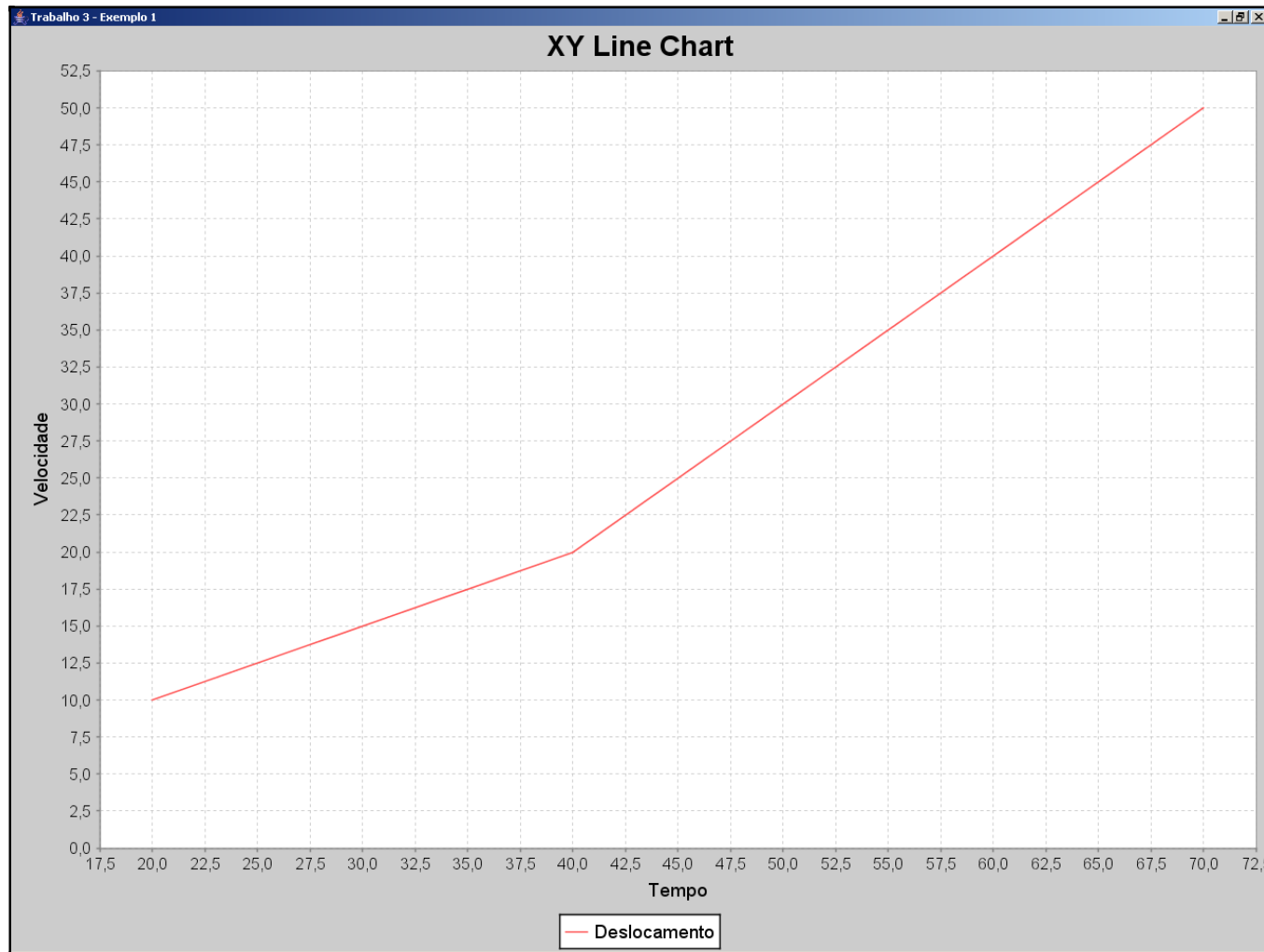
    GraficosBarra chart = new GraficosBarra();

    frame.getContentPane().add( chart, BorderLayout.CENTER );
    frame.setSize( 800, 800 );
    frame.setVisible( true );
    frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
}
```



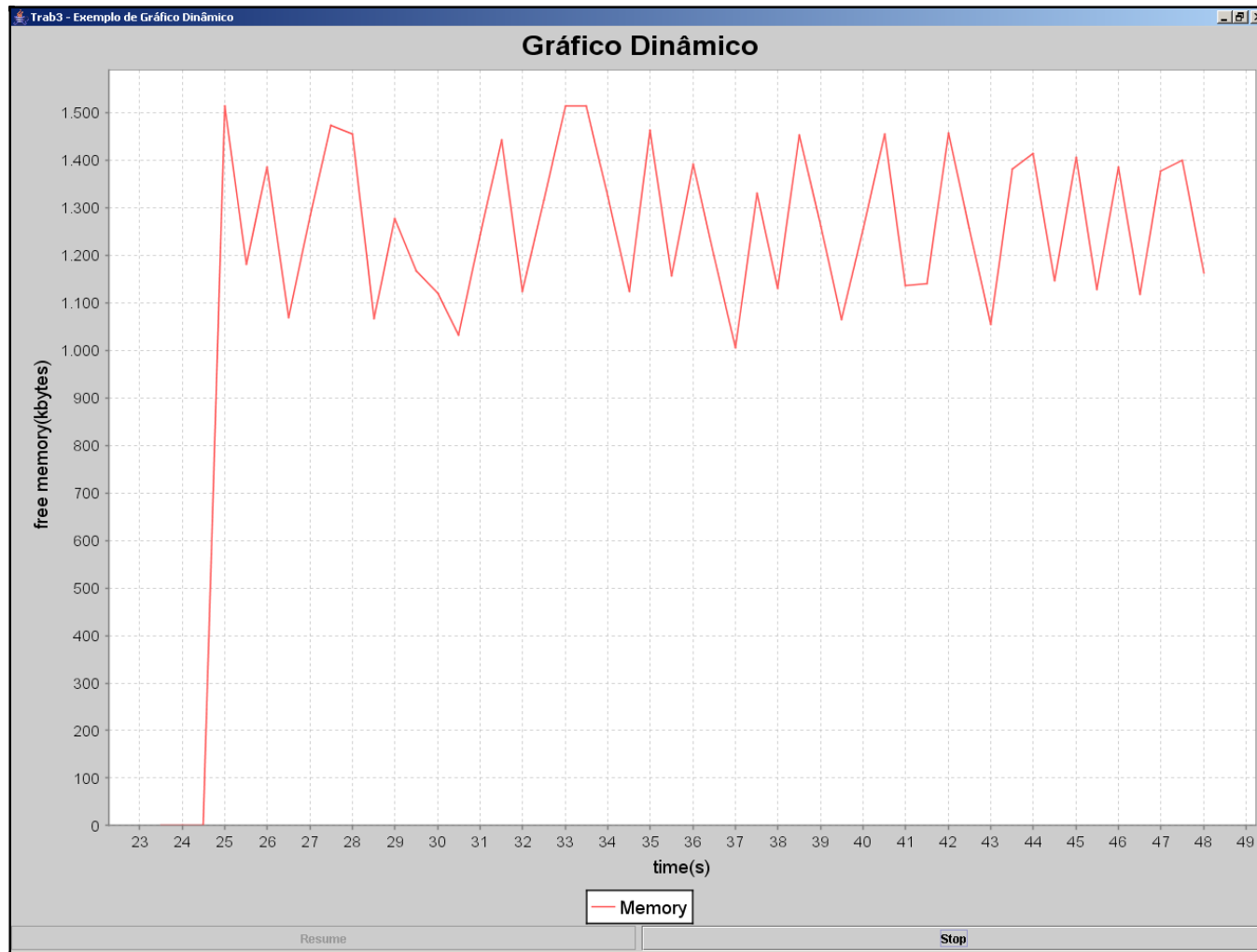
Alguns Tipos de Gráficos

XY Charts

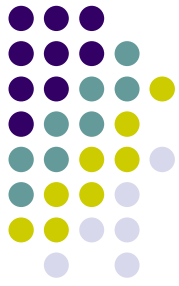


Alguns Tipos de Gráficos

Gráficos Dinâmicos



Outros recursos

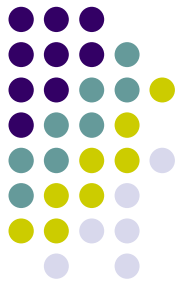
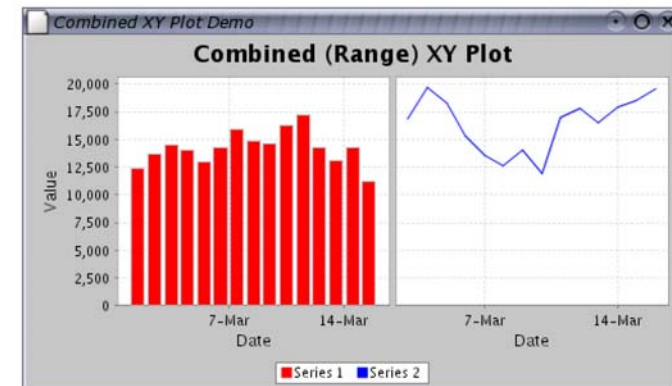
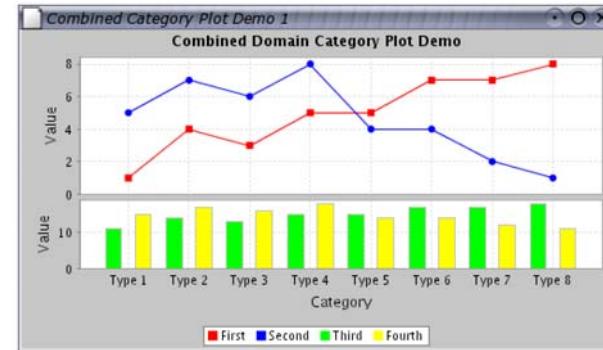


- **JDBC:**
 - API para gerar datasets através de banco de dados relacionais
 - Datasets: JDBC PieDataset,
JDBC CategoryDataset,
JDBC XYDataset

Outros recursos

- **Gráficos combinados:**

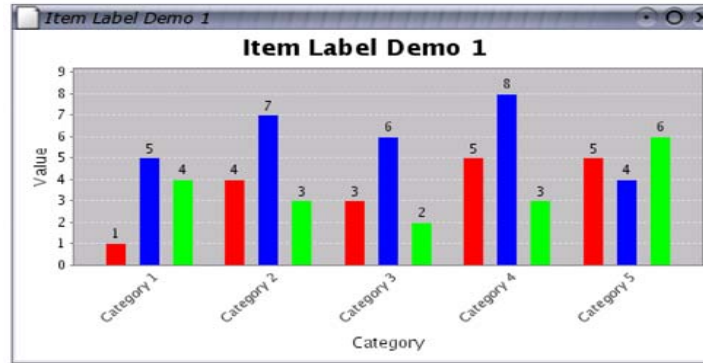
- Tipos: Domínio compartilhado
Imagem compartilhada
- Plots: CombinedDomainCategoryPlot
CombinedRangeCategoryPlot
CombinedDomainXYPlot
CombinedRangeXYPlot



```
CategoryAxis domainAxis = new CategoryAxis( "Categoria" );  
CombinedDomainCategoryPlot plot = new CombinedDomainCategoryPlot(domainAxis);  
plot.add(subplot1, 2);  
plot.add(subplot2, 1);
```

Outros recursos

- **Legenda em gráficos**



- Exibir anotações dentro de gráficos
- Problemas na versão atual: Alguns Renderers não suportam, alguns eixos não são ajustados automaticamente.

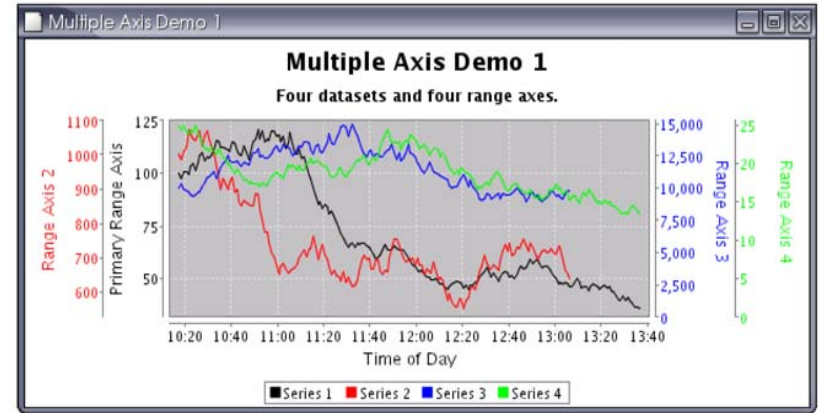
```
CategoryItemRenderer renderer = plot.getRenderer( );  
renderer.setLabelGenerator( new StandardCategoryItemLabelGenerator( ));  
renderer.setItemLabelsVisible( true );
```

Outros recursos



- **Gráficos Múltiplos**

- Múltiplos datasets, renderers, eixos de imagem e eixos de domínio



Métodos:

```
plot.setSecondaryRangeAxis(0, axis);
```

```
plot.setSecondaryDataset(0, dataset);
```

```
plot.setSecondaryRenderer(0, renderer);
```

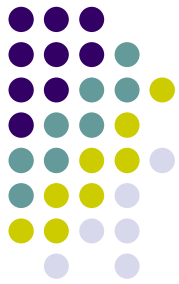
Outros recursos



- **ToolTips**

- JFreeChart inclui mecanismos de exibir, gerar e coletar ToolTips
- **Gerar:** PieToolTipGenerator, CategoryToolTipGenerator, etc.
- **Coletar:** Classe ChartRenderingInfo
- **Exibir:** Classe ChartPanel: setDisplayToolTips(boolean flag);

Outros Tipos de Gráficos



Séries Temporais

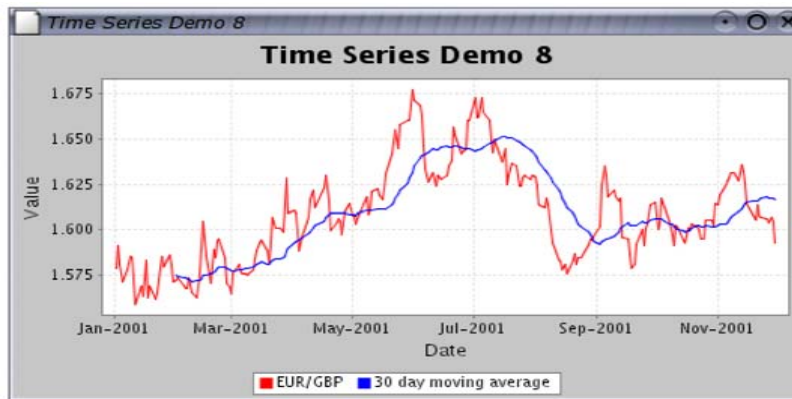


Gráfico de Gantt

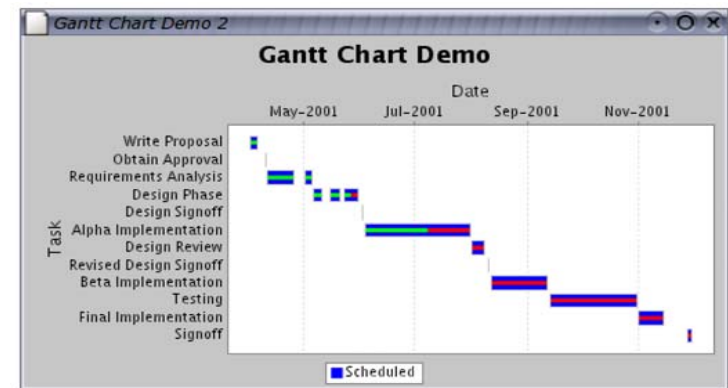
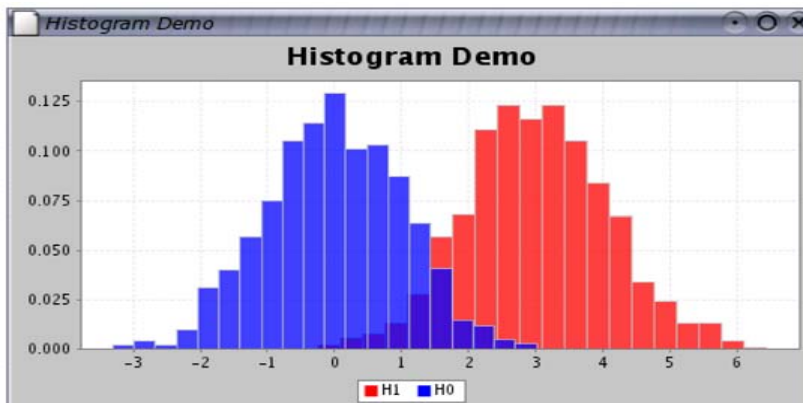


Gráfico Cascata



Outros Tipos de Gráficos

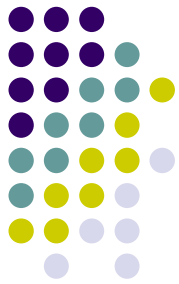


Gráfico Diferença

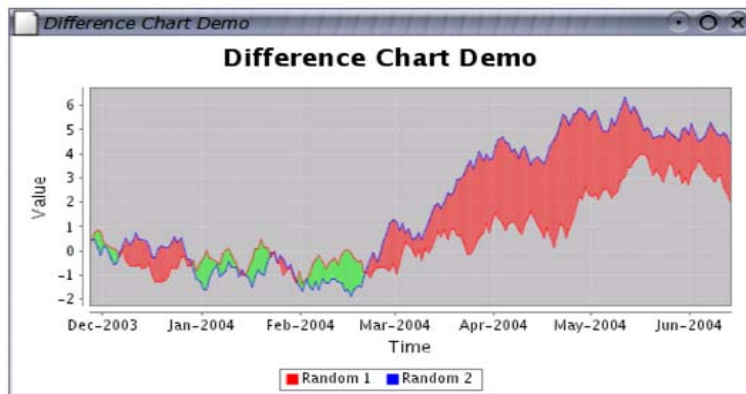


Gráfico de Área

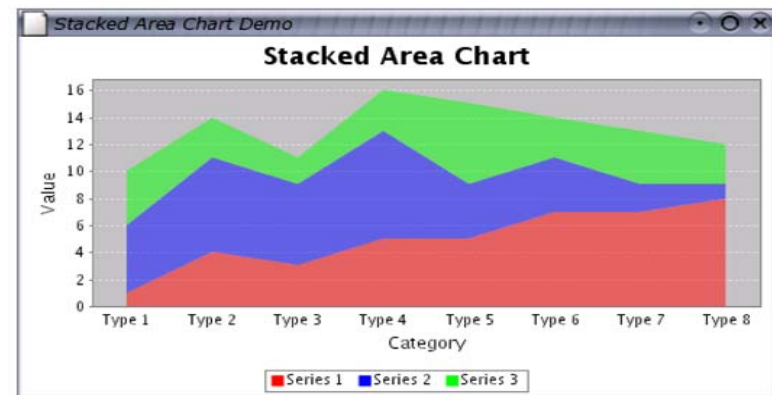


Gráfico Escada

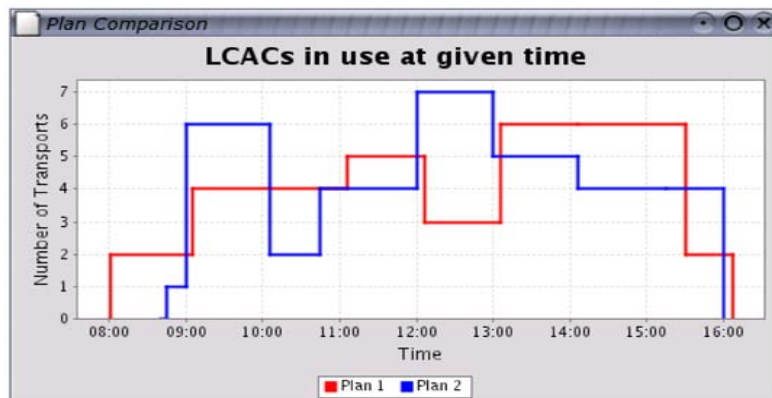
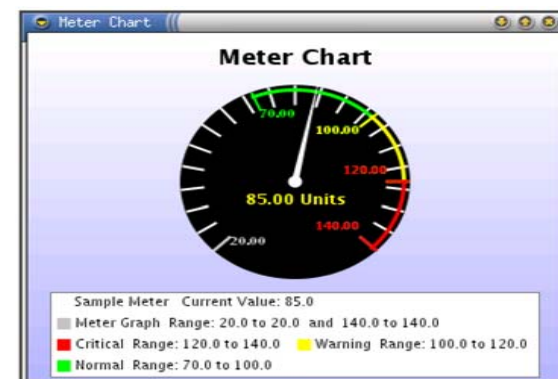


Gráfico Agulha



Outros Tipos de Gráficos



Gráfico de ventos

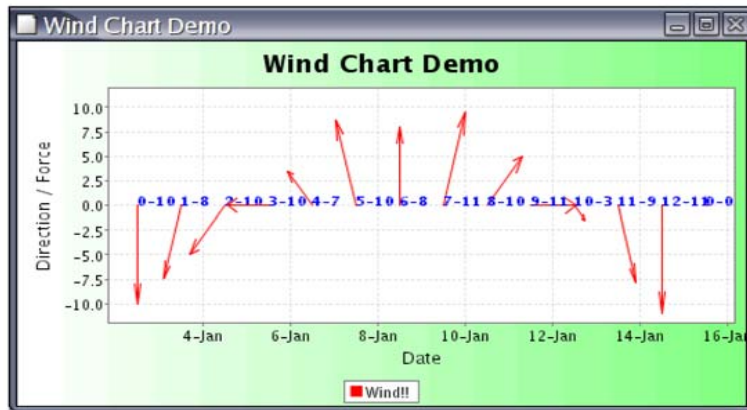
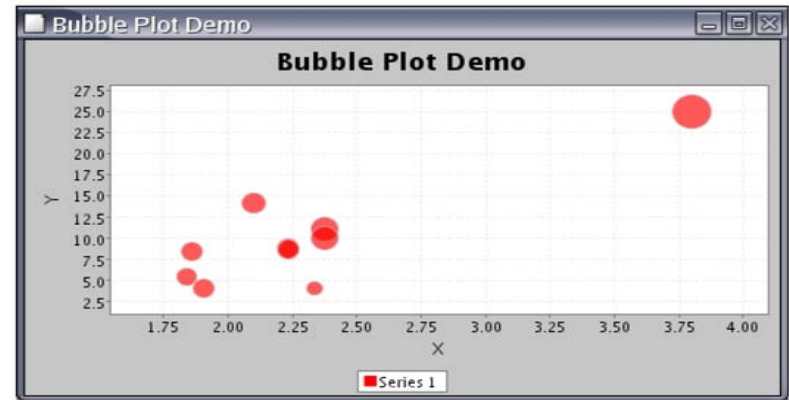
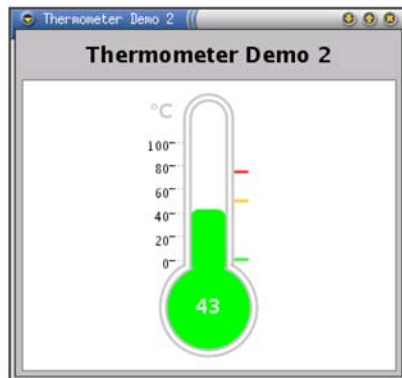


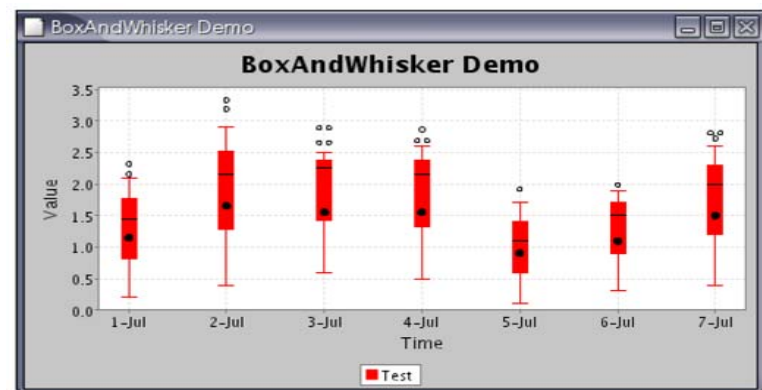
Gráfico bolha



Termômetro



Outros

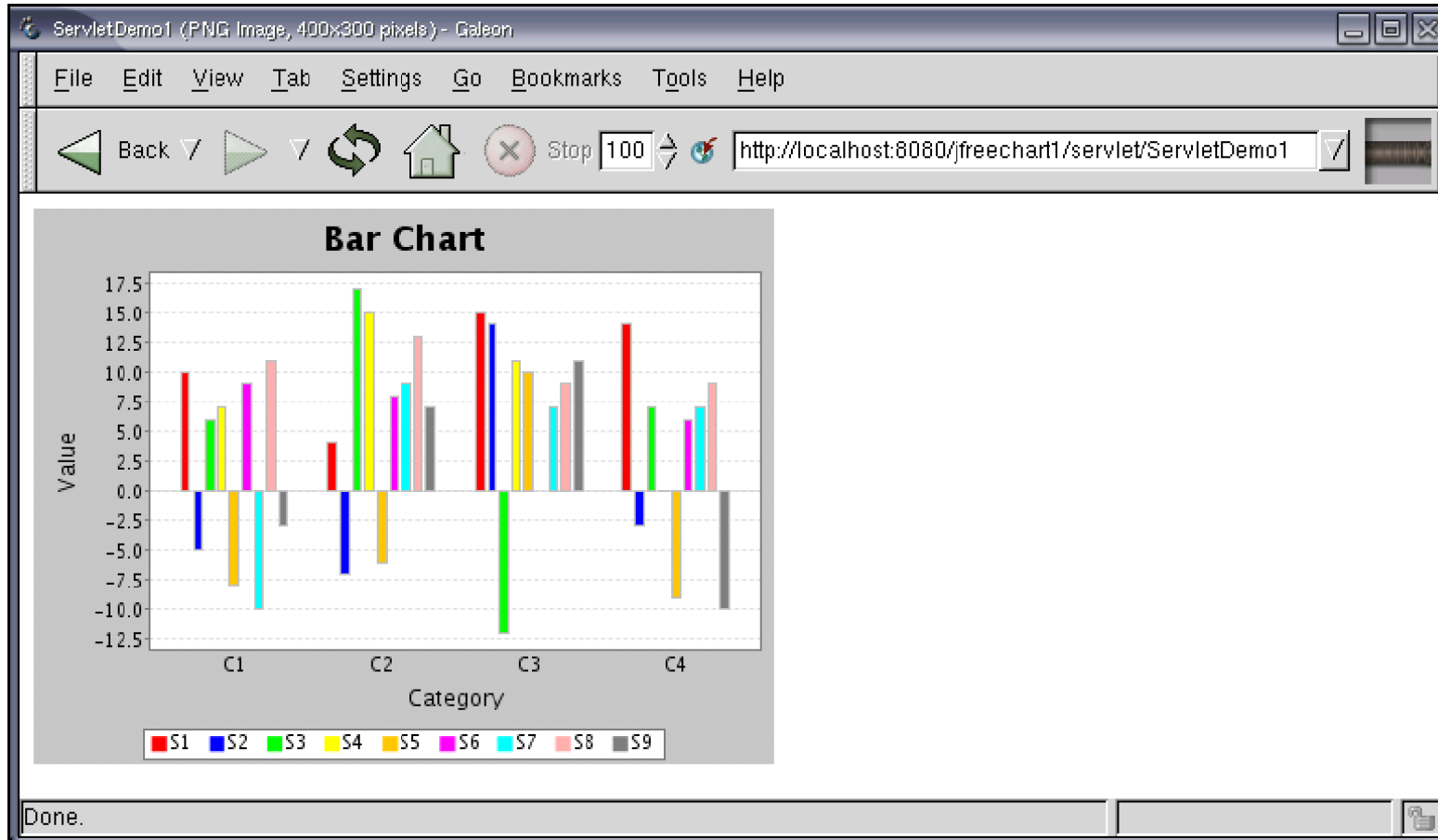


4. Tipos de Apresentações



- Aplicação Swing
- Servlets
- Applets
- JSP

ServLets



ServLets



```
public class Servlet1 extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException
    {
        OutputStream out = response.getOutputStream();
        try
        {
            DefaultCategoryDataset dataset = new DefaultCategoryDataset();
            dataset.addValue(10.0, "S1", "C1"); dataset.addValue(4.0, "S1", "C2");
            dataset.addValue(15.0, "S1", "C3"); dataset.addValue(14.0, "S1", "C4");
            dataset.addValue(-5.0, "S2", "C1"); dataset.addValue(-7.0, "S2", "C2");
            dataset.addValue(14.0, "S2", "C3"); dataset.addValue(-3.0, "S2", "C4");
            dataset.addValue(6.0, "S3", "C1"); dataset.addValue(17.0, "S3", "C2");
            dataset.addValue(-12.0, "S3", "C3"); dataset.addValue(7.0, "S3", "C4");
            dataset.addValue(7.0, "S4", "C1"); dataset.addValue(15.0, "S4", "C2");
            dataset.addValue(11.0, "S4", "C3"); dataset.addValue(0.0, "S4", "C4");
            dataset.addValue(-8.0, "S5", "C1"); dataset.addValue(-6.0, "S5", "C2");
            dataset.addValue(10.0, "S5", "C3"); dataset.addValue(-9.0, "S5", "C4");
            dataset.addValue(9.0, "S6", "C1"); dataset.addValue(8.0, "S6", "C2");
            dataset.addValue(null, "S6", "C3"); dataset.addValue(6.0, "S6", "C4");
            dataset.addValue(-10.0, "S7", "C1"); dataset.addValue(9.0, "S7", "C2");
```

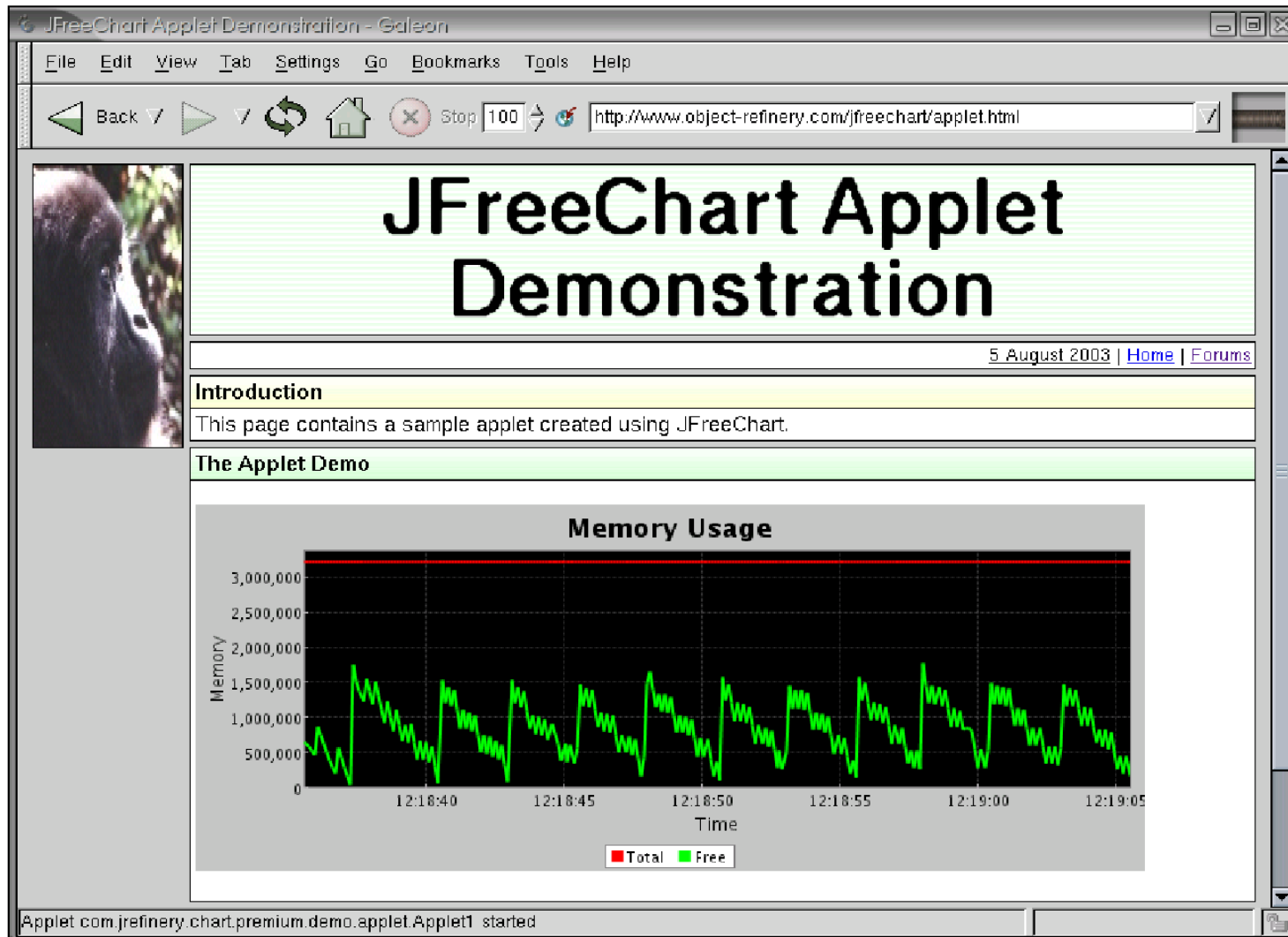
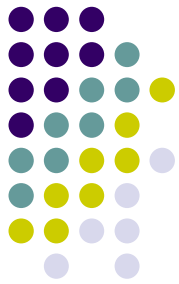
ServLets



```
dataset.addValue(7.0, "S7", "C3"); dataset.addValue(7.0, "S7", "C4");
dataset.addValue(11.0, "S8", "C1"); dataset.addValue(13.0, "S8", "C2");
dataset.addValue(9.0, "S8", "C3"); dataset.addValue(9.0, "S8", "C4");
dataset.addValue(-3.0, "S9", "C1"); dataset.addValue(7.0, "S9", "C2");
dataset.addValue(11.0, "S9", "C3"); dataset.addValue(-10.0, "S9", "C4");
```

```
JFreeChart chart;
chart = ChartFactory.createBarChart( "Bar Chart", "Category", "Value", dataset,
                                     PlotOrientation.VERTICAL, true, true, false );
response.setContentType( "image/png" );
ChartUtilities.writeChartAsPNG(out, chart, 400, 300);
}
catch (Exception e) {
    System.err.println(e.toString());
}
finally {
    out.close();
}
}
}
```

Applet



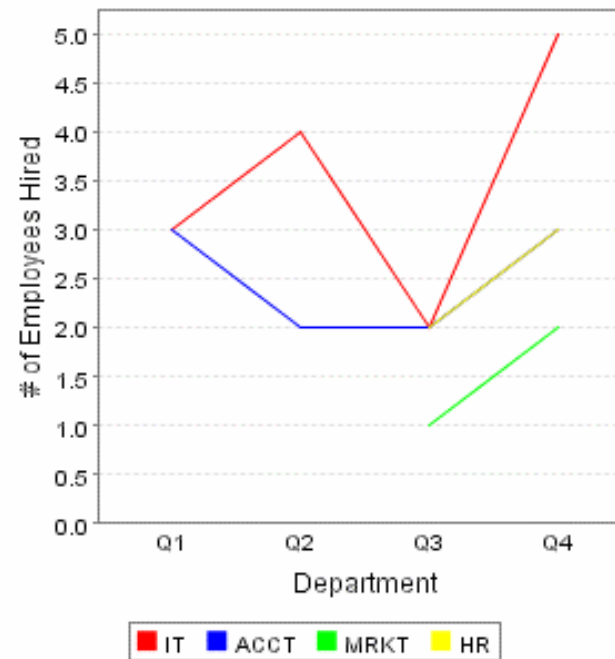
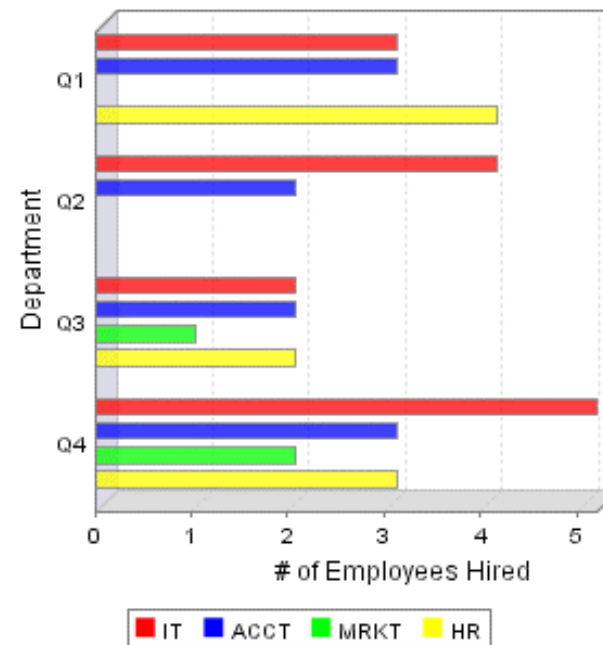
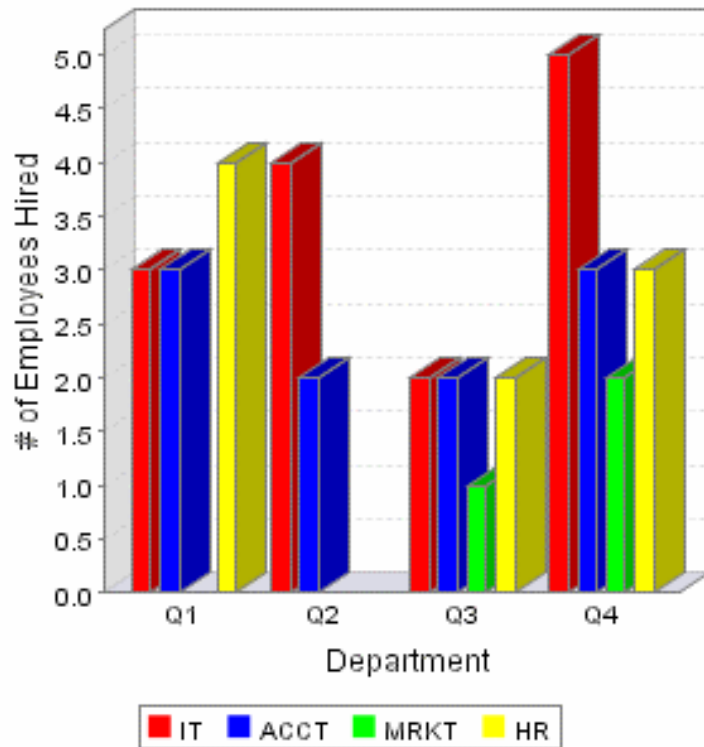
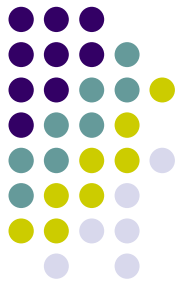
Applet



Problemas:

- Suporte a diferentes browsers
- Segurança
- Tamanho do código fonte

JSP



JSP



```
<%@page contentType= "text/html" %>
<%@page import= "org.jfree.data.*" %>
<%@page import= "java.io.Serializable" %>
```

```
<%@taglib uri= ' /WEB-INF/cewolf.tld ' prefix='cewolf' %>
```

```
<jsp:useBean id="barChartView" class="examples.EmployeesByDeptAndQtr"/>
```

```
<% String yAxis = "# of Employees Hired" ; String xAxis = "Department" ; %>
```

```
<html>
```

```
<body>
```

```
<table align= "center" border= "0" width= "60%" >
```

```
<tr><td colspan="3">
```

```
<H2 ALIGN= "CENTER" >Employees by Department and Quarter</H2></td></tr>
```

```
<tr><td>
```

```
<cewolf: chart id="barExample3DH" type="horizontalBar3D"xaxislabel=" <%=xAxis%> "
yaxislabel="<%=yAxis%> ">
```

```
<cewolf: data>
```

```
<cewolf:producer id= "barChartView" />
```

```
</cewolf:data>
```

```
<cewolf: colorpaint color= "#FFFFFF" />
```

```
</cewolf:chart>
```

```
<cewolf: img chartid ="barExample3DH" renderer="cewolf" border="0" width="300"
height="300" />
```

```
</td>
```

JSP



```
<td>
  <cewolf: chart id ="barExample3DV" type="verticalBar3D" xaxislabel=" <%=xAxis%> "
yaxislabel=" <%=yAxis%> " >
    <cewolf:data>
      <cewolf: producer id ="barChartView" />
    </cewolf:data>
    <cewolf: colorpaint color ="#FFFFFF" />
  </cewolf:chart>
  <cewolf: img chartid ="barExample3DV" renderer= "cewolf" border="0" width="300"
height="300" />
</td>

<td>
  <cewolf:chart id= "lineExample" type="line"
  xaxislabel="<%=xAxis%>" yaxislabel="<%=yAxis%>">
    <cewolf:data>
      <cewolf:producer id="barChartView" />
    </cewolf:data>
    <cewolf:colorpaint color= "#FFFFFF" />
  </cewolf:chart>
  <cewolf: img chartid ="lineExample" renderer= "cewolf" border="0" width="300"
height="300" /> </td> </tr>

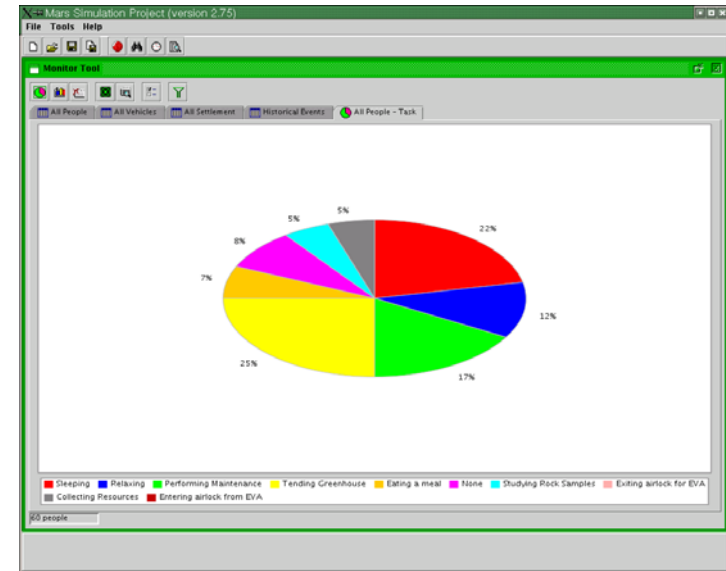
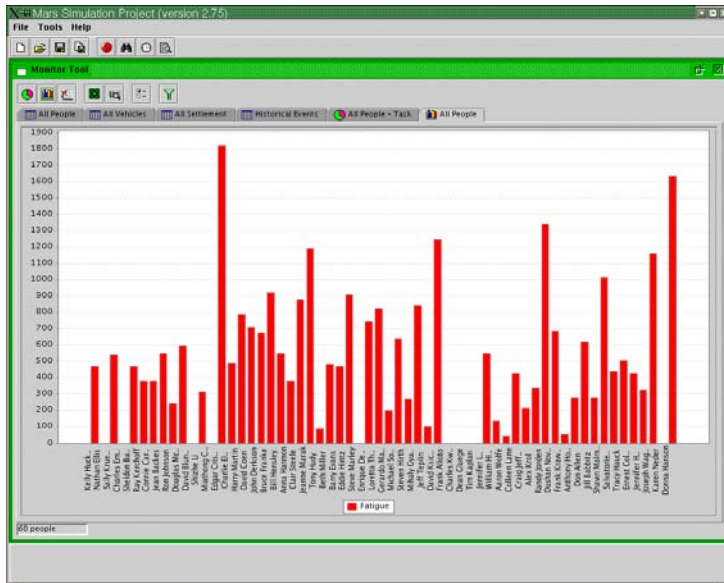
</body>
</html>
```

Softwares que utilizam JFreeChart



- The Mars Simulation Project

<http://mars-sim.sourceforge.net/>



Concorrentes



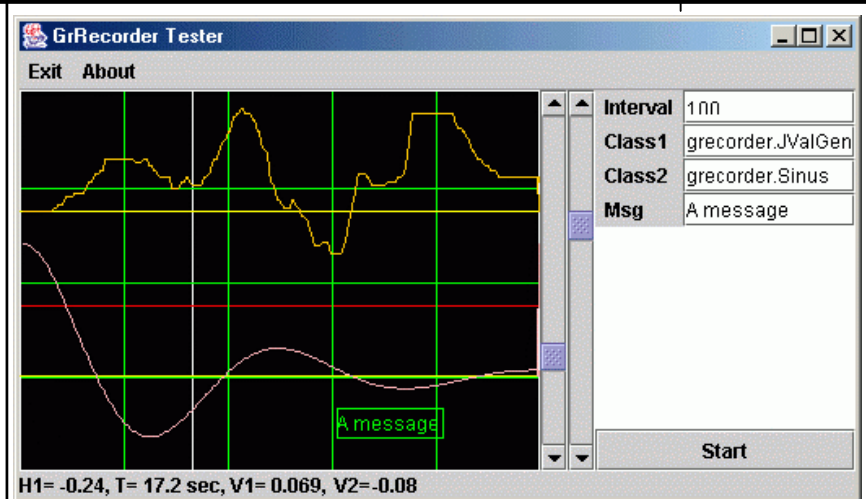
GrRecorder

http://pages.infinit.net/bigfeet/grrecorder_.html

Licença: GNU Lesser GPL

Tipo: Aplicativo

Produtor: Marcel St-Amant



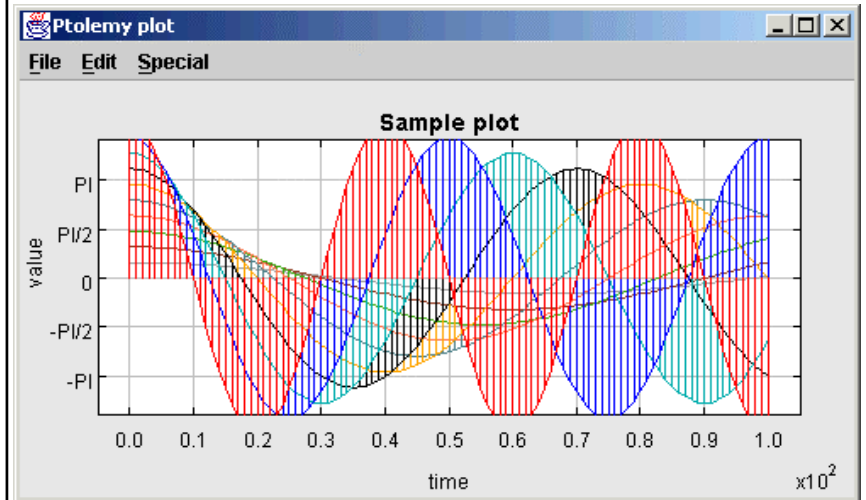
PtPlot

<http://ptolemy.eecs.berkeley.edu/java/ptplot/>

Licença : GNU Lesser GPL

Tipo: Component

Produtor: Ptplot



Concorrentes



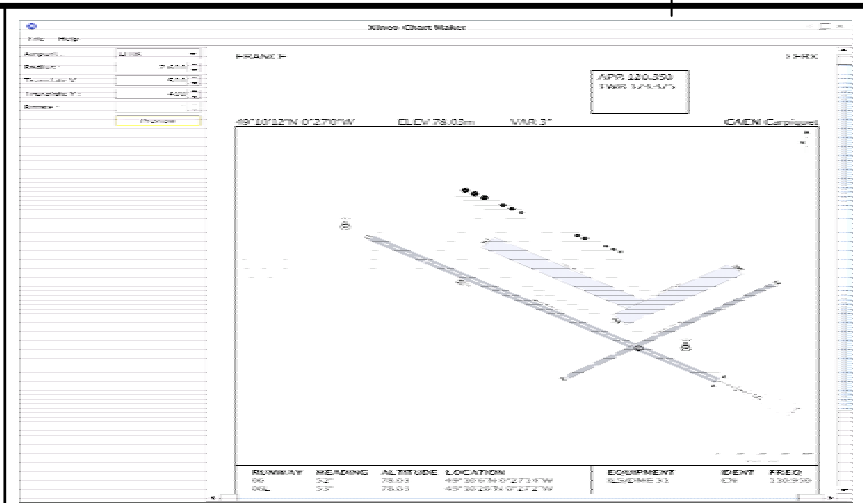
Chart Maker

<http://www.xineo.net/chartmaker.jspx>

Licença: Xineo Freeware License

Tipo : Componente

Produtor: Xineo.net



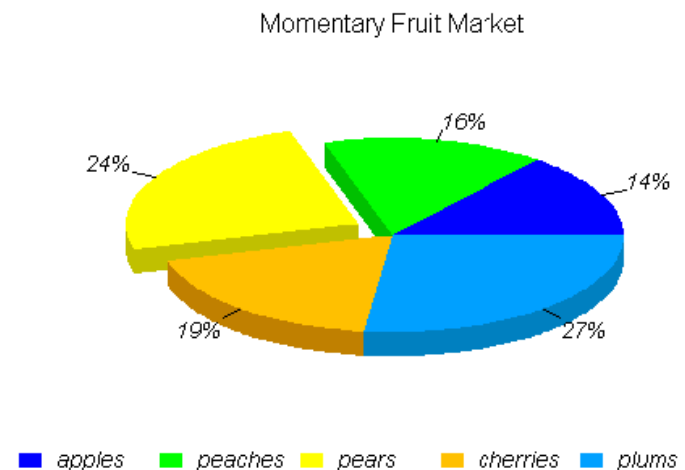
KavaChart

<http://www.ve.com/kavachart/index.html>

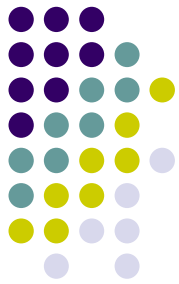
Licença: Commercial

Tipo: Componente

Produtor: Visual Engineering



Concorrentes



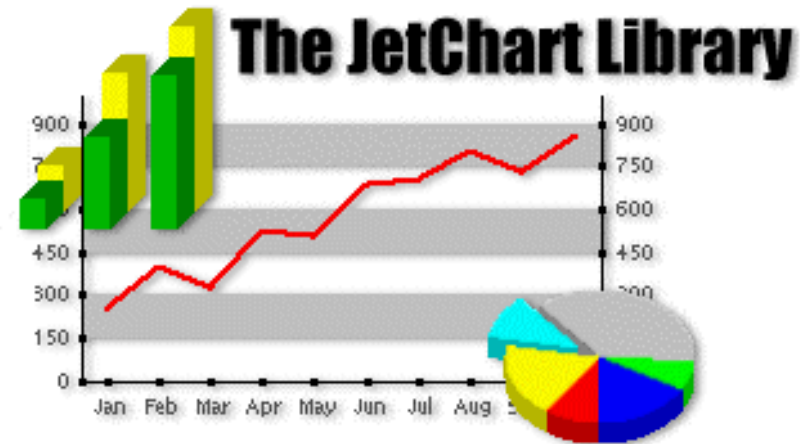
JetChart

<http://www.jinsight.com/jetchart/index.html>

Licença: Comercial

Tipo: Componente

Produtor: Jinsight Informatica



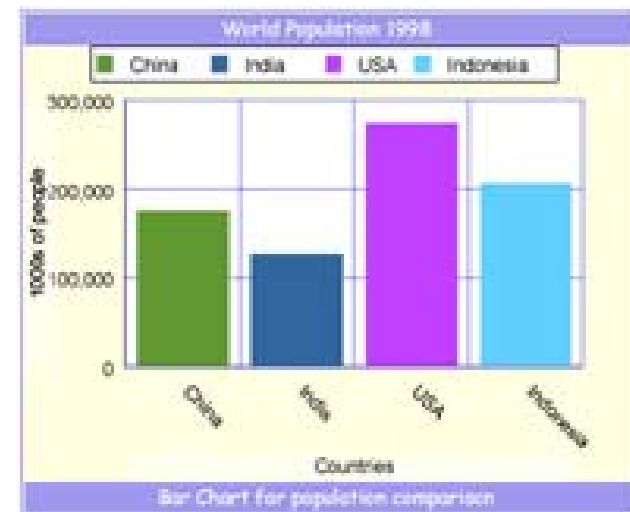
AgileBlox Chart

<http://www.elansoft.com/web/home.html>

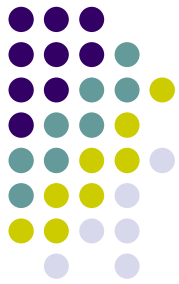
Licença: Comercial

Tipo: Componente

Produtor: Elansoft



Concorrentes



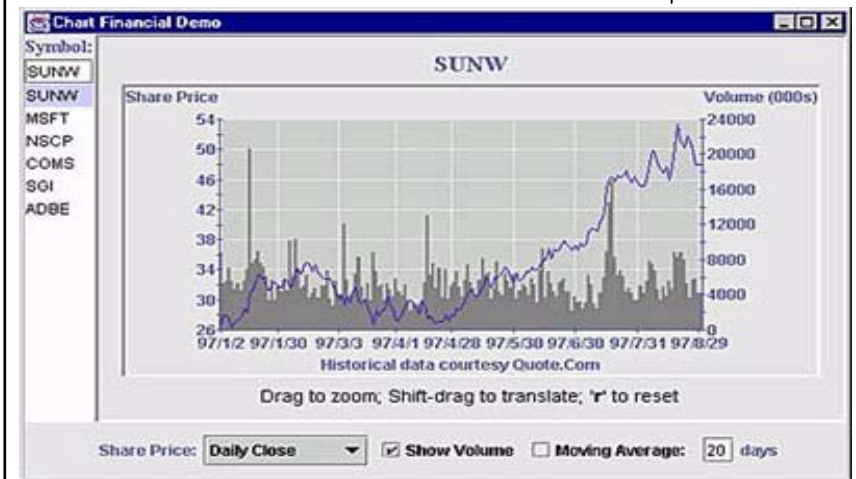
JClass Chart

http://www.quest.com/jclass_desktopviews/

Licença: Comercial

Tipo : Componente

Produtor: Quest Software, Inc



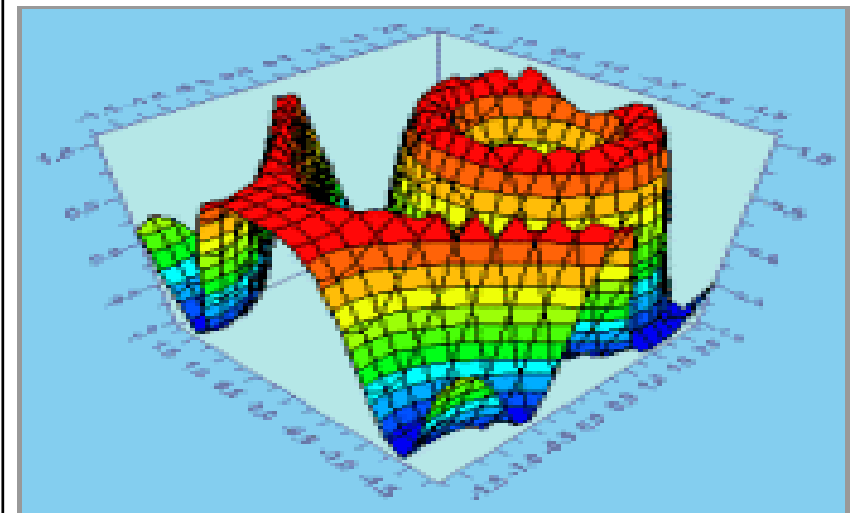
JClass Chart 3D

http://www.quest.com/jclass_desktopviews/

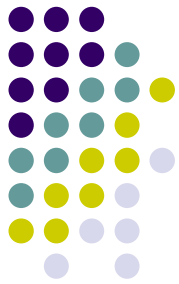
Licença: Comercial

Tipo: Componente

Produtor: Quest Software, Inc



Concorrentes



EasyCharts

<http://www.objectplanet.com/EasyCharts/>

Licença: Comercial

Tipo : Componente

Produtor: ObjectPlanet, Inc

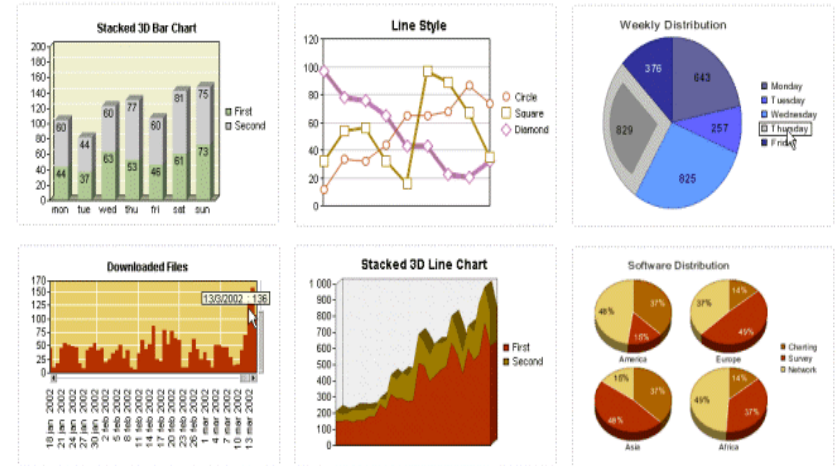


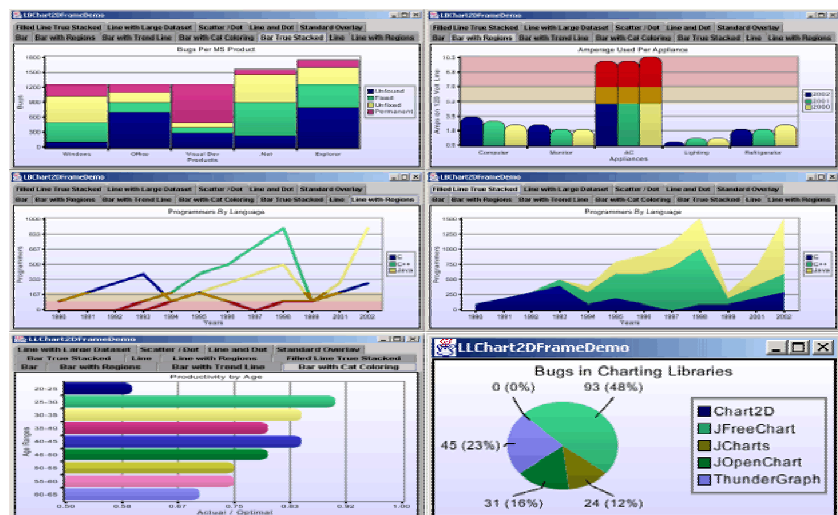
Chart2D

<http://chart2d.sourceforge.net/>

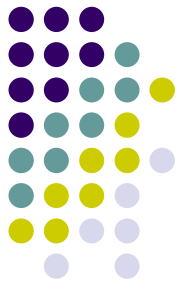
Licença: OpenSource

Tipo: Componente

Produtor: Chart2D



Concorrentes



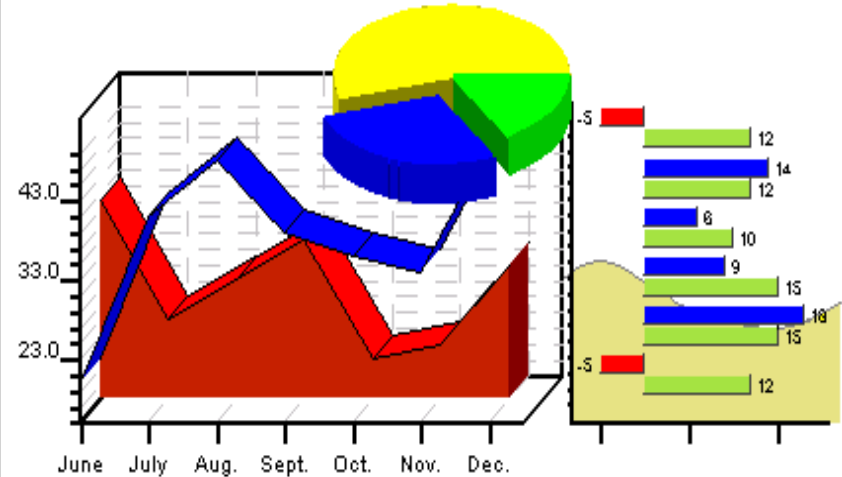
RChart

http://www.java4less.com/charts_e.htm

Licença: Comercial

Tipo : Componente

Produtor: J4L



ChartCat

<http://www.netcat.li/java-graph-and-chart-engine/>

Licença: Comercial

Tipo: Componente

Produtor: NetCat Inc

