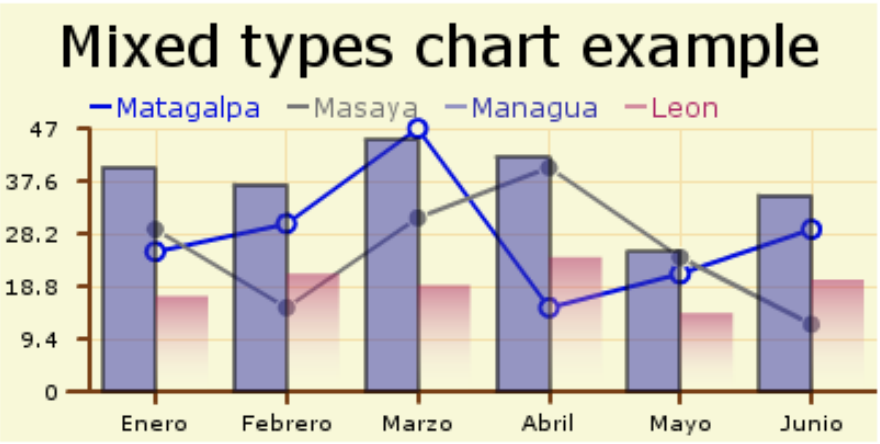
Curso de gráficos con Java y OpenFlashChart para aplicaciones web

Manual del alumno



Ing. Cedric Simon – Tel: 2268 0974 – Cel: 8888 2387 – Email: cedric@solucionjava.com – Web: www.solucionjava.com

SolucionJava.com

Índice de contenido

1. Introducción al curso

1.1. Objetivo de este curso

En este curso vamos a aprender a crear gráficos para aplicaciones web JSP.

1.2. Manual del alumno

Este manual del alumno es una ayuda para el alumno, para tenga un recuerdo del curso. Este manual contiene un resumen de las materias que se van a estudiar durante el curso, pero el alumno debería de tomar notas personales para completas este manual.

1.3. Ejercicios prácticos

Para captar mejor la teoría, se harán muchos ejercicios con los alumnos, para probar la teoría y verificar la integración de la materia.

También, el alumno podrá copiar sus códigos en una memoria flash al fin del curso para llevarse, con fin de seguir la práctica en su hogar.

1.4. Requisitos para atender a este curso

El conocimiento de los lenguaje HTML, Java, y Javascript es requerido para poder atender a este curso.

Si el alumno tiene dificultades en un u otro capitulo, el debe sentirse libre de pedir explicaciones adicionales al profesor.

Pero si aparece que el alumno no posee los requisitos mínimos para este curso, por respeto a los otros alumnos que ya poseen esta materia, el alumno podría ser traslado para otro curso en el futuro, cuando el cumplirá con los requisitos.

1.5. Soporte después del curso

Si tienes preguntas sobre la materia del curso en tus ejercicios prácticos, puedes escribir tus preguntas a cedric@solucionjava.com .

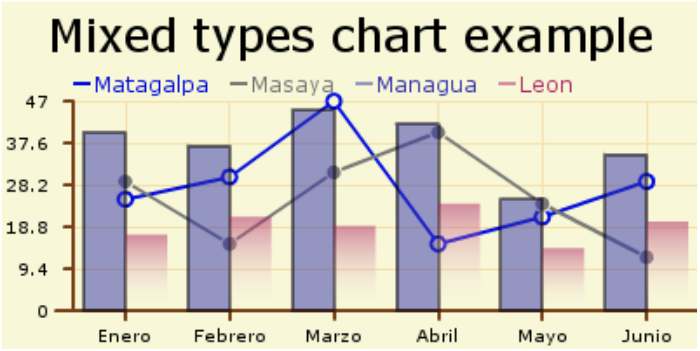
Para informaciones sobre otros cursos, visita el sitio web www.solucionjava.com.

2. Introducción

2.1. Gráficos con OpenFlashChart

OpenFlashChart es una librería Flash que permite crear gráficos en Flash usando Javascript para manejar el gráficos, y entonces sin escribir nada en Flash.

OpenFlashCart permite crear gráficos más atractivos que JFreeChart, pero tiene menos tipos de gráficos disponibles, y no es un objeto Java, si no que se ha creado una acción personalizada para poder crear fácilmente los gráficos desde una página JSP. OpenFlashChart solo está disponible para clientes web.



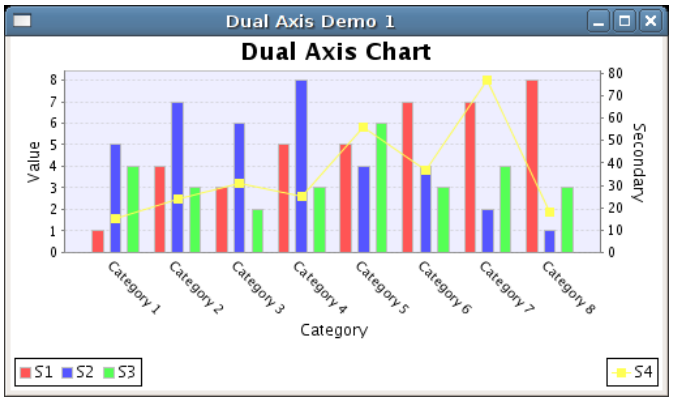
2.2. Gráficos con Java

Existen varios paquetes que permiten crear gráficos con Java. La más conocida, y probablemente la más completa es el paquete JFreeChart, que se puede usar solo o empaquetado en otros paquetes como JasperReport.

JFreeChart es un software libre, bajo licencia GPL, y se puede descargar desde http://www.jfree.org/jfreechart/ .

JFreeChart permite crear todos tipos de gráficos, ya sea área, barra, ejes,lineales, linea de tiempo, etc...

Las clases de JFreeChart crean objetos Graphic2D que se pueden en aplicaciones de escritorio, o exportar a imágenes o PDF para incluirlos en aplicaciones web, sin necesidad de plugin especial.



3. OpenFlashChart

3.1. Instalación

Para poder usar JFreeChart necesitamos :

1. El archivo SjFramework.jar
2. Una subcarpeta llamada 'chart' debajo de 'WebContent' (o raiz del sitio) con los archivos open-flash-chart.swf y swfobject.js, y opcionalmente el archivo flash_detect.js
3. Un navegador con el plugin de Macromedia Flash instalado para poder ver los gráficos
4. La base de datos disponible como fuente de datos.

OpenFlashChart no necesita Java para funcionar. Pero vamos a ver en este curso como usar la librería de acción personalizada TagLib diseñada por SolucionJava.com y liberada bajo licencia GPL, que nos permite crear gráficos basados en una consulta a una base de datos.

3.2. Primer gráfico

Para crear gráficos en OpenFlashChart con la librería de SolucionJava.com, es muy fácil: usa el taglib, menciona los parámetros que quieres para personalizar el gráfico, y pone el SQL para traer los datos.

Se guardo el nombre de variable original de OpenFlashChart (que no respeta las reglas de nombramiento de Java) para poder fácilmente ligar el Java con el Javascript.

Para info sobre los parámetros disponibles, ver la documentación del API de OpenFlashChartTag, y para su significado, ver la documentación en el el sitio web oficial de OpenFlashChart (http://teethgrinder.co.uk/open-flash-chart/).

Ejemplo:

```
<%@ taglib uri="/WEB-INF/lib/SjFramework.jar" prefix="sj" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Primer gr&aacute;fico</title>
<script type="text/javascript" src="chart/swfobject.js"></script>
</head>
<body>
<script type="text/javascript" src="chart/flash_detect.js"></script>
<h1>Primer gr&aacute;fico</h1>
<sj:OpenFlashChart
width="400"
height="200"
title="Simple Line chart example"
title_style="font-size:26px"
occurence="1"
line_hollow_colour="#bdf"
line_hollow_dot_size="5"
line_hollow_text="Matagalpa"
line_hollow_font_size="12"
line_hollow_width="2"
dbObjectName="bd.LeerDB"
>
select   tm.value_es,
         total_ventas ,
         date_format(fecha,'%m')
```

```
        from
        ventas tt
        join months tm on (date_format(fecha,'%m')=tm.orden)
        where tt.fecha  between '2007-01-01' and '2007-06-30'
        and no_site= 4
        order by 3
</sj:OpenFlashChart>
</body>
</html>
```

3.3. Otros gráficos

Para obtener otros gráfico, solo modifica los parametros de la etiqueta <sj:OpenFlashChart>.

Cuando hay varios grupos de valores, se menciona el número de grupos de datos con el parámetro “numberOfDataSets”, y el valor del identificador de cada grupo (cuarto campo de la consulta) con “setXValue” donde X representa el numero del grupo. Luego define el color y el texto de cada grupo.

Ejemplo:

```
<%@ taglib uri="/WEB-INF/lib/SjFramework.jar" prefix="sj" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Gr&aacute;fico de varios grupos de datos</title>
<script type="text/javascript" src="chart/swfobject.js"></script>
</head>
<body>
<script type="text/javascript" src="chart/flash_detect.js"></script>
<h1>Gr&aacute;fico de varios grupos de datos</h1>
<sj:OpenFlashChart
width="400"
height="200"
title="Multiple lines chart example"
title_style="font-size:26px"
numberOfDataSets="4"
set1Value="1"  set2Value="2"  set3Value="3"  set4Value="4"

occurence="3"  line_hollow_colour="#3333ad"  line_hollow_dot_size="5"     line_hollow_text="Managua"
line_hollow_font_size="12"   line_hollow_width="2"

line_circles="15"     line_size="12" line_width="2" line2_colour="#af2566"       line2_text="Leon"

line_dot_dot_size="5" line_dot_font_size="12"      line_dot_width="2"    line3_dot_colour="#777777"
line3_dot_text="Masaya"

line4_hollow_colour="#bdf"   line4_hollow_text="Matagalpa"

dbObjectName="bd.LeerDB"
>
select   tm.value_es,
         total_ventas ,
         date_format(fecha,'%m'),
         no_site
         from  ventas tt
         join months tm on (date_format(fecha,'%m')=tm.orden)
         where tt.fecha  between '2007-01-01' and '2007-06-30'
         order by 3 ,4
</sj:OpenFlashChart>
</body>
</html>
```

4. JFreeChart

4.1. Instalación

Para poder usar JFreeChart necesitamos las librerías jfreechart-1.0.13.jar, jcommon-1.0.16.jar, y iText-2.1.5.jar incluidas en el classpath (les podemos copiar debajo WEB-INF/lib).

4.2. Primer gráfico

Vamos a crear un primer gráfico en un servlet, y una página HTML para mostrarlo.

Primero tenemos que crear el arreglo de valores (dataset), luego creamos el gráfico, y lo enviamos por la salida.

Ejemplo:

```
package demo;

import java.io.IOException;
import java.io.OutputStream;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;

public class ServletDemo1 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");
            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 = 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();
        }
    }
}
```

jfdemo1.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Primer grafico JFreeChart</title>
</head>
<body>
<h1>Primer grafico JFreeChart</h1>
<hr>
<img alt="Primer grafico JFreeChart" src="ServletDemo1">
</body>
</html>
```

4.3. Tipos de gráficos y tipos de arreglos de datos

A cada tipo de gráfico corresponde un tipo de arreglo de datos.

Los principales tipos de gráficos son: Pie, Bar, Line, y TimeSerie.

a) Pie chart

Crear un arreglo de datos de tipo PieDataset y luego crear el gráfico.

Servlet PieDemo1:

```
package demo;
```

```
import java.io.IOException;
import java.io.OutputStream;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.data.general.DefaultPieDataset;
public class PieDemo1 extends HttpServlet {
       private static final long serialVersionUID = 1L;
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
               doPost(request, response);
       }
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        OutputStream out = response.getOutputStream();
        try {
            JFreeChart chart = null;
            chart = createPieChart();
            if (chart != null) {
                response.setContentType("image/png");
                ChartUtilities.writeChartAsPNG(out, chart, 400, 300);
            }
        }
        catch (Exception e) {
            System.err.println(e.toString());
        }
        finally {out.close();
        }
       }
    private JFreeChart createPieChart() {
        DefaultPieDataset data = new DefaultPieDataset();
        data.setValue("Uno", new Double(43.2));
        data.setValue("Dos", new Double(10.0));
        data.setValue("Tres", new Double(27.5));
        data.setValue("Cuatro", new Double(17.5));
        data.setValue("Cinco", new Double(11.0));
        data.setValue("Seis", new Double(19.4));
        JFreeChart chart = ChartFactory.createPieChart(
                "Pie Chart Demo 1",  // chart title
                data,             // data
                true,                // include legend
                true,                           // tooltips
                false                           // urls
        );
        return chart;
    }
}
```

b) ***Bar chart***

Crear un arreglo de datos de tipo DefaultCategoryDataset y luego crear el gráfico.

Servlet BarDemo1:

```
package demo;

import java.io.IOException;
import java.io.OutputStream;
```

```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;

public class BarDemo1 extends HttpServlet {
       private static final long serialVersionUID = 1L;
          protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                  doPost(request, response);
             }
             protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
              OutputStream out = response.getOutputStream();
              try {
                  JFreeChart chart = null;
                  chart = createBarChart();
                  if (chart != null) {
                      response.setContentType("image/png");
                      ChartUtilities.writeChartAsPNG(out, chart, 400, 300);
                  }
              }
              catch (Exception e) {
                  System.err.println(e.toString());
              }
              finally {out.close();
              }
             }
          private JFreeChart createBarChart() {
              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");
              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 = ChartFactory.createBarChart3D(
                    "Bar Chart Demo 1",       // chart title
                    "Category",               // domain axis label
                    "Value",                  // range axis label
                    dataset,                  // data
                    PlotOrientation.VERTICAL, // orientation
                    true,                     // include legend
                    true,                     // tooltips?
                    false                     // URLs?
                );
                return chart;
            }
}
```

c) ***Line chart***

Crear un arreglo de datos de tipo DefaultCategoryDataset y luego crear el gráfico.

Servlet LineDemo1:

```
package demo;

import java.io.IOException;
import java.io.OutputStream;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;
public class LineDemo1 extends HttpServlet {
       private static final long serialVersionUID = 1L;
          protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                    doPost(request, response);
              }
             protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
              OutputStream out = response.getOutputStream();
              try {
                  JFreeChart chart = null;
                  chart = createLineChart();
                  if (chart != null) {
                      response.setContentType("image/png");
                      ChartUtilities.writeChartAsPNG(out, chart, 600, 300);
                  }
              }
              catch (Exception e) {
                  System.err.println(e.toString());
              }
              finally {out.close();
              }
             }
 private JFreeChart createLineChart() {
        DefaultCategoryDataset dataset = new DefaultCategoryDataset();
     dataset.addValue(212, "Classes", "JDK 1.0");
```

```
     dataset.addValue(504, "Classes", "JDK 1.1");
     dataset.addValue(1520, "Classes", "JDK 1.2");
     dataset.addValue(1842, "Classes", "JDK 1.3");
     dataset.addValue(2991, "Classes", "JDK 1.4");
     dataset.addValue(3500, "Classes", "JDK 1.5");
     JFreeChart chart = ChartFactory.createLineChart(
             "Java Standard Class Library",   // chart title
             null,                         // domain axis label
             "Class Count",                   // range axis label
             dataset,                         // data
             PlotOrientation.VERTICAL,        // orientation
             false,                           // include legend
             true,                            // tooltips
             false                            // urls
         );
     return chart;
 }
}
```

d) ***TimeSerie chart***

Crear un arreglo de datos de tipo XYDataset y luego crear el gráfico.

Servlet TimeSerieDemo1:

```
package demo;
import java.io.IOException;
import java.io.OutputStream;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.data.time.Day;
import org.jfree.data.time.TimeSeries;
import org.jfree.data.time.TimeSeriesCollection;
import org.jfree.data.xy.XYDataset;
import org.jfree.date.MonthConstants;

public class TimeSerieDemo1 extends HttpServlet {
       private static final long serialVersionUID = 1L;
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
                    doPost(request, response);
              }
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

              OutputStream out = response.getOutputStream();
              try {
                  JFreeChart chart = null;
                  chart = createTimeSeriesChart();
                  if (chart != null) {
                      response.setContentType("image/png");
                      ChartUtilities.writeChartAsPNG(out, chart, 400, 300);
                  }
              }
              catch (Exception e) {
                  System.err.println(e.toString());
              }
              finally {out.close();
              }
             }
```

```
private JFreeChart createTimeSeriesChart() {

        // here we just populate a series with random data...
        TimeSeries series = new TimeSeries("Random Data");
        Day current = new Day(1, MonthConstants.JANUARY, 2009);
        for (int i = 0; i < 100; i++) {
            series.add(current, Math.random() * 100);
            current = (Day) current.next();
        }
        XYDataset data = new TimeSeriesCollection(series);
        JFreeChart chart = ChartFactory.createTimeSeriesChart(
                "Time Series Chart", // title
                "Date",              // x-axis label
                "Rate",                  // y-axis label
                data,            // data
                true,               // create legend?
                true,               // generate tooltips?
                false               // generate URLs?
        );
        return chart;
    }
}
```

e) Otros gráficos

Como pudieron ver, es siempre casi el mismo código:

1) crear un arreglo de datos
2) crear el gráfico

Para los otros tipos de gráficos, ver la documentación API de JfreeChart.

4.4. Conexión con la base de datos

Para traer los datos de la base de datos, hay dos posibilidades: llenar el arreglo de datos con una bucle sobre el arreglo de resultados, o usar un arreglo de datos JDBC.

Cada arreglo de datos tiene su pendiente JDBC.

PieDataset → JDBCPieDataset
XYDataset → JDBCXYDataset
DefaultDataset → JDBCDefaultDataset
etc...

El uso del JDBC dataset es muy sencillo: crear una conexión, pasarla al dataset, ejecutar la consulta, y cerrar la conexión.

```
con = DriverManager.getConnection(url, "jfreechart", "password");
data = new JDBCPieDataset(con);
String sql = "SELECT * FROM PIEDATA1;";
data.executeQuery(sql);
con.close();
```

Si usamos el marco de tarbajo de SolJava:

```
LeerDB leerDB = new LeerDB();
data = new JDBCPieDataset(leerDB.getCon());
```

```
            String sql = "SELECT * FROM PIEDATA1;";
            data.executeQuery(sql);
            leerDB.cleanup();
```

Servlet PieDemo2:

```
package demo;

import java.io.IOException;
import java.io.OutputStream;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.data.jdbc.JDBCPieDataset;
import bd.LeerDB;

public class PieDemo2 extends HttpServlet {
       private static final long serialVersionUID = 1L;

       protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
              doPost(request, response);
       }
       protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
              OutputStream out = response.getOutputStream();
              try {
                     JFreeChart chart = null;
                     chart = createPieChart();
                     if (chart != null) {
                            response.setContentType("image/png");
                            ChartUtilities.writeChartAsPNG(out, chart, 400, 300);
                     }
              }
              catch (Exception e) {
                     System.err.println(e.toString());
              }
              finally {out.close();
              }
       }
       private JFreeChart createPieChart() {
              JDBCPieDataset data=null;
              try {
                     LeerDB leerDB = new LeerDB();
                     data = new JDBCPieDataset(leerDB.getCon());
                     String sql = "select"+
                     " tm.value_es,"+
                     " total_ventas"+
                     " from "+
                     " ventas tt  "+
                     " join months tm on (date_format(fecha,'%m')=tm.orden)"+
                     " where tt.fecha  between '2007-01-01' and '2007-06-30'"+
                     " and no_site= 4";
                     data.executeQuery(sql);
                     leerDB.cleanup();
              } catch (SQLException e) {
                     e.printStackTrace();
              } catch (Exception e) {
                     e.printStackTrace();
              }
              JFreeChart chart = ChartFactory.createPieChart(
```

```
                            "Pie Chart Demo 1",  // chart title
                            data,             // data
                            true,                // include legend
                            true,                             // tooltips
                            false                             // urls
                );
                return chart;
        }
}
```

4.5. Formateo fino del gráfico

Los gráficos de JfreeChart se pueden formatear de manera muy fina.

Casi todos los elementos del gráfico se pueden personalizar, ya sea el título, los ejes, el fondo, las etiquetas, etc...

Existen hasta la posibilidad de crear gráficos interactivos.

Como las opciones de formateo fino son inmensas, referirse a la documentación en línea para descubrir todas las posibilidades.

Servlet BarDemo2:

```
package demo;
import java.awt.BasicStroke;
import java.awt.Color;
import java.awt.GradientPaint;
import java.io.IOException;
import java.io.OutputStream;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.CategoryPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.title.TextTitle;
import org.jfree.data.jdbc.JDBCCategoryDataset;
import org.jfree.ui.RectangleInsets;
import bd.LeerDB;
public class BarDemo2 extends HttpServlet {
       private static final long serialVersionUID = 1L;
          protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                    doPost(request, response);          }
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
               OutputStream out = response.getOutputStream();
               try {
                   JFreeChart chart = null;
                   chart = createBarChart();
                   if (chart != null) {
                       response.setContentType("image/png");
                       ChartUtilities.writeChartAsPNG(out, chart, 400, 300);
                   }
               }
               catch (Exception e) {
```

```
                System.err.println(e.toString());
            }
            finally {out.close();
            }
           }
        private JFreeChart createBarChart() {
                  JDBCCategoryDataset data=null;
                  try {
                         LeerDB leerDB = new LeerDB();
                         data = new JDBCCategoryDataset(leerDB.getCon());
                         String sql = "select"+" tm.value_es,"+" total_ventas"+
                         " from "+" ventas tt  "+
                         " join months tm on (date_format(fecha,'%m')=tm.orden)"+
                         " where tt.fecha  between '2007-01-01' and '2007-06-30'"+
                         " and no_site= 4";
                         data.executeQuery(sql);
                         leerDB.cleanup();
                  } catch (SQLException e) {e.printStackTrace();
                  } catch (Exception e) {e.printStackTrace();}
            JFreeChart chart = ChartFactory.createBarChart(
                    "Bar Chart Demo 2",       // chart title
                    null,               // domain axis label
                    "Valor",                  // range axis label
                    data,                  // data
                    PlotOrientation.VERTICAL, // orientation
                    true,                     // include legend
                    true,                     // tooltips?
                    false                     // URLs?
            );
            TextTitle title = chart.getTitle();
            title.setBorder(0, 0, 1, 0);
            title.setBackgroundPaint(new GradientPaint(0f, 0f, Color.red, 350f,
                    0f, Color.white, true));
            title.setExpandToFitSpace(true);

            chart.setBackgroundPaint(new GradientPaint(0f, 0f, Color.yellow, 350f,
                    0f, Color.white, true));
            // get a reference to the plot for further customisation...
            CategoryPlot plot = (CategoryPlot) chart.getPlot();
            plot.setNoDataMessage("NO DATA!");
            plot.setBackgroundPaint(null);
            plot.setInsets(new RectangleInsets(10, 5, 5, 5));
            plot.setOutlinePaint(Color.black);
            plot.setRangeGridlinePaint(Color.gray);
            plot.setRangeGridlineStroke(new BasicStroke(1.0f));
            // change the margin at the top of the range axis...
            NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis();
            rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());
            rangeAxis.setRange(0.0, 30.0);
            rangeAxis.setTickMarkPaint(Color.black);

            return chart;
        }
}
```