



PATRONES PARA GRÁFICAS EN JAVA: JFreeChart

ASIGNATURA:

Programación Avanzada Orientada a Objetos

CATEDRÁTICO:

José Juan Hernández Mora

ALUMNO:

José Montoya Guzmán

FECHA DE ASIGNACIÓN:

29 de noviembre de 2016

FECHA DE ENTREGA:

4 de diciembre de 2016

JFreeChart

JFreeChart es un librería open source para el lenguaje de programación Java, el cual permite la creación de gráficos complejos de forma simple.

JFreeChart es compatible con una serie de gráficas diferentes, incluyendo cuadros combinados. Después de tipos de gráficos son compatibles:

- Gráficos XY (línea, spline y dispersión).
- Gráfico circular.
- Diagrama de Gantt.
- Gráficos de barras (horizontal y vertical, apilado e independiente).
- Histogramas.
- Single valued (termómetro, brújula, indicador de velocidad) que luego se pueden colocar sobre el mapa.
- Varias gráficas específicas (tabla de viento, gráfica polar, burbujas de diferentes tamaños, etc.).

Entre las bondades de este patrón de diseño, se encuentran que permite dibujar automáticamente las escalas de los ejes y leyendas. Con el ratón informático se puede hacer zoom en la interfaz de la gráfica automáticamente y cambiar algunos ajustes a través del menú local. Las tablas existentes pueden actualizarse fácilmente a través de los listeners que la biblioteca tiene en sus colecciones de datos.

Documentación oficial

<http://www.jfree.org/jfreechart/api/gjdoc/>

EJEMPLOS DE GRÁFICAS CON JFreeChart

Gráfico de Pastel

PiePlot3D es una clase que permite realizar graficos circulares en 3D, utilizando datos de cualquier clase que implementa la interfaz PieDataset.

Aunque esta clase extiende PiePlot, actualmente no soporta secciones explotadas.

Código

GraficaPie.java

```
package prueba;
import javax.swing.*;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PiePlot3D;
import org.jfree.data.general.DefaultPieDataset;
import org.jfree.data.general.PieDataset;
import org.jfree.util.Rotation;
/* @author jose_*/

public class GraficaPie extends JFrame{

    public GraficaPie(String appTitulo, String grafTitulo)
    {
        super(appTitulo);
        PieDataset datos = creaDatos();
        JFreeChart chart = creaGrafica(datos, grafTitulo);
        ChartPanel panel = new ChartPanel(chart);
        panel.setPreferredSize(new java.awt.Dimension(800, 500));
        setContentPane(panel);
    }
}
```

```

private PieDataset creaDatos() {
    DefaultPieDataset datos = new DefaultPieDataset();
    datos.setValue("Física", 1);
    datos.setValue("Computación", 96);
    datos.setValue("Matemáticas", 1);
    datos.setValue("Biología", 1);
    datos.setValue("Actuaría", 1);
    return datos;
}

private JFreeChart creaGrafica(PieDataset datos, String titulo) {

    JFreeChart grafica = ChartFactory.createPieChart3D(
        titulo, datos, true, false, false);
    PiePlot3D plot = (PiePlot3D) grafica.getPlot();
    plot.setStartAngle(290);
    plot.setDirection(Rotation.CLOCKWISE);
    plot.setForegroundAlpha(0.5f);
    return grafica;
}
}

```

Prueba.java

```

package prueba;

public class Prueba {

    public static void main(String[] args) {
        // TODO code application logic here
        GraficaPie demo = new GraficaPie("Aplicación de prueba de JFreeCHart",
            "Carreras en la Facultad de Ciencias");
        demo.pack();
    }
}

```

```

demo.setVisible(true);
}
}

```

Ejecución

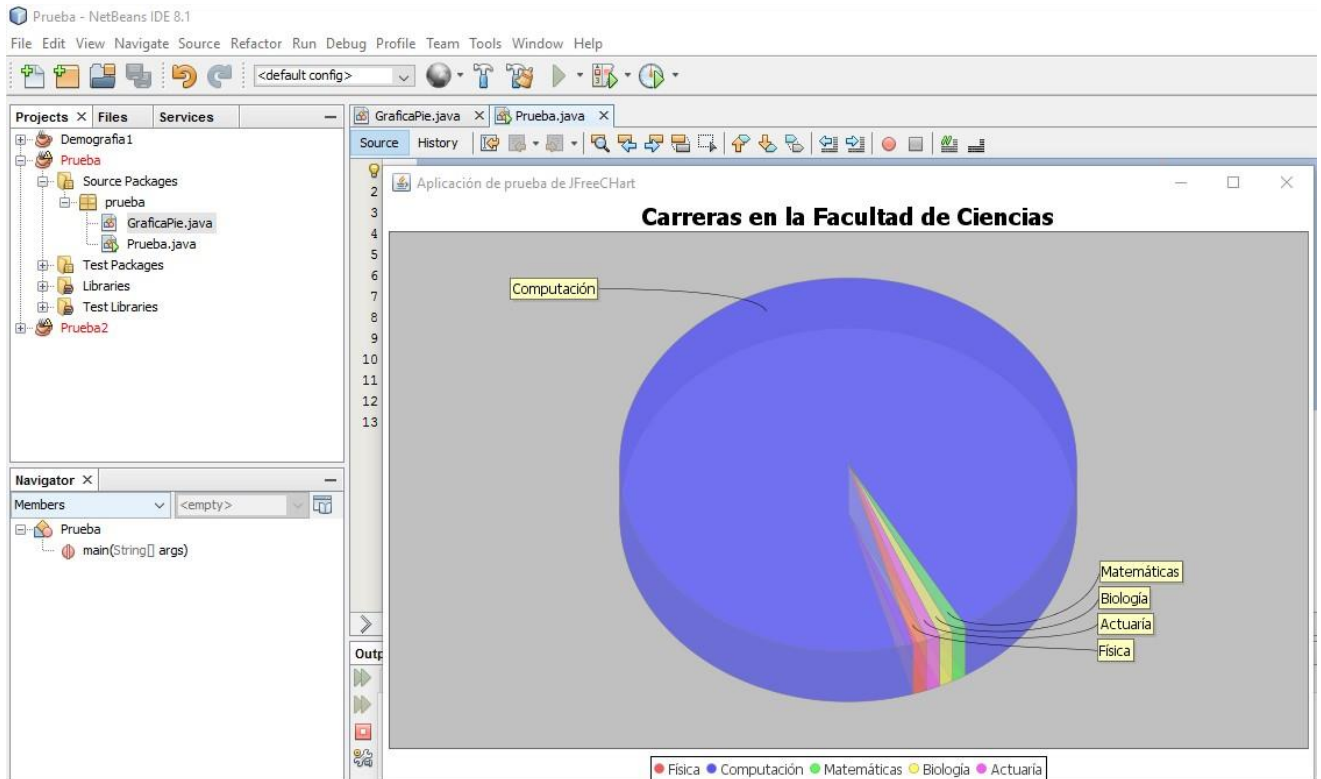


Gráfico de una función

Una interfaz a través de la cual se puede acceder a datos en forma de elementos (x, y). Es ideal para graficar funciones matemáticas.

Código

FuncionSenoX.java

```
package prueba2;
```

```
import javax.swing.*;
```

```
import java.awt.Dimension;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.xy.XYDataset;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;
/* @author jose_*/
```

```
public class FuncionSenoX extends JFrame{
    public FuncionSenoX(String appTitulo) {
        super(appTitulo);
        XYDataset datos = creaDatos();
        JFreeChart diagrama = creaGrafica(datos);
        ChartPanel panel = new ChartPanel(diagrama);
        panel.setPreferredSize(new Dimension(500, 400));
        setContentPane(panel);
    }
```

```
    private XYDataset creaDatos() {
        XYSeries datos = new XYSeries("Datos");
        for (double x = 254-20; x <= 254+40; x += 0.01) {
            double y = Math.sin((x*2*Math.PI)/33);
            datos.add(x, y);
        }
        XYSeriesCollection conjuntoDatos = new XYSeriesCollection();
        conjuntoDatos.addSeries(datos);

        return conjuntoDatos;
    }
```

```

private JFreeChart creaGrafica(XYDataset datos) {
    JFreeChart diag = ChartFactory.createXYLineChart(
        "Gráfica", "Eje X", "Eje Y", datos,
        PlotOrientation.VERTICAL, false, false,
        false);
    return diag;
}
}

```

Prueba2.java

```

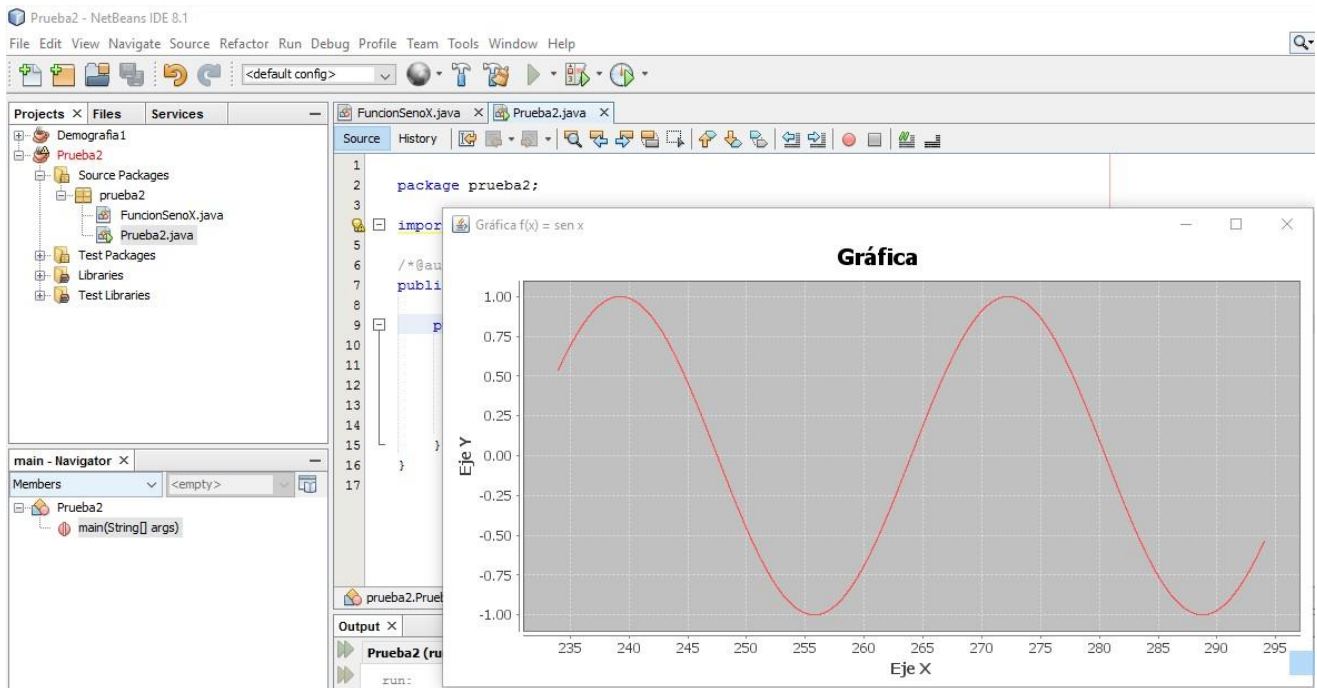
package prueba2;
import org.jfree.ui.RefineryUtilities;

/*@author jose_*/
public class Prueba2 {

    public static void main(String[] args) {
        // TODO code application logic here
        FuncionSenoX grafica = new FuncionSenoX("Gráfica  $f(x) = \sin x$ ");
        grafica.pack();
        //RefineryUtilities.centerFrameOnScreen(grafica);
        grafica.setVisible(true);
    }
}

```

Ejecución



Gráfica de líneas

La clase ChartFactory una colección de métodos de utilidad para crear algunos gráficos estándar con JFreeChart.

Código

Demografia1.java

```
package demografia1;
```

```
/* @author jose_ */
```

```
import org.jfree.chart.ChartFactory;
```

```
import org.jfree.chart.ChartFrame;
```

```
import org.jfree.chart.JFreeChart;
```



```

import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;

public class Demografia1 {

    public static void main(String[] args) {

        DefaultCategoryDataset dataset = new DefaultCategoryDataset();

        double poblacion = 25;
        int esperanza_de_vida = 5;
        double defunciones;
        double poblacion_neta;

        double tc = 0.2; // tasa de crecimiento 20%
        double tm = 0.4; // tasa de mortalidad 40%

        for (int tiempo = 0; tiempo < esperanza_de_vida; tiempo++) {

            poblacion = poblacion * (1 + tc);
            dataset.addValue(poblacion, "Crecimiento", "" + tiempo);

            defunciones = poblacion * tm;
            dataset.addValue(defunciones, "Mortalidad", "" + tiempo);

            poblacion_neta = poblacion - defunciones;
            dataset.addValue(poblacion_neta, "Crecimiento neto", "" + tiempo);

        }

        JFreeChart chart = ChartFactory.createLineChart(

```

```

        "Calculo demografico",
        "Tiempo",
        "Población",
        dataset,
        PlotOrientation.VERTICAL,
        true,
        false,
        false
    );
    ChartFrame fr = new ChartFrame("Calculo Demografico I", chart);
    fr.pack();
    fr.setVisible(true);
}
}

```

Ejecución

