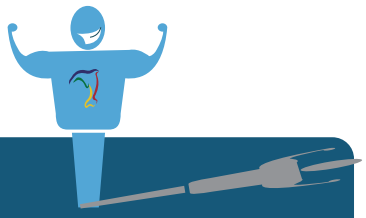
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

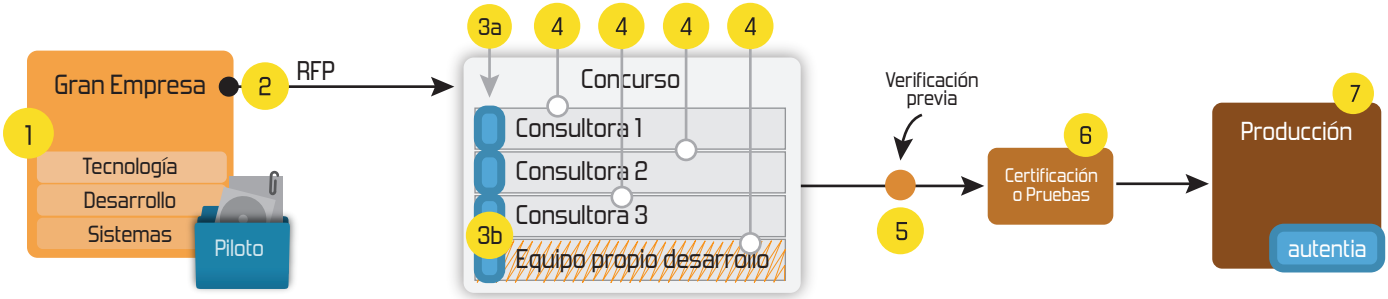
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Control de autenticación y acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

- JFreeChart 1.0.13.

3. Diseñando la gráfica.

Como hemos dicho anteriormente, nos vamos a basar en datos de pilotos que dan vueltas de entrenamiento a un circuito para generar este ejemplo. Las características que tendrá nuestra gráfica serán las siguientes:

- Cada serie (de tiempos por vuelta de un piloto) se representará de forma lineal.
- El eje Y de la gráfica mostrará los tiempos en segundos en un rango de 120 a 135. El número de segundos del eje se visualizará de dos en dos (120, 122, 124...).
- El eje X de la gráfica mostrará el número de vueltas, de la vuelta 1 a la vuelta 5.
- Cada punto de la gráfica debe de mostrarse con un círculo (evidentemente los círculos serán unidos por la línea que representará la serie).
- Los colores de las series deben ser distintos, así como las líneas de la gráfica que hacen de guía (líneas discontinuas que salen en el fondo de la gráfica).
- En el pie de la gráfica debe figurar una leyenda con los nombres de los pilotos.

4. Construyendo la gráfica.

Pues bien, aquí tenemos la clase que nos generará la gráfica:

```
01 import java.awt.Color;
02 import java.io.File;
03 import org.jfree.chart.ChartFactory;
04 import org.jfree.chart.ChartUtilities;
05 import org.jfree.chart.JFreeChart;
06 import org.jfree.chart.axis.NumberAxis;
07 import org.jfree.chart.axis.NumberTickUnit;
08 import org.jfree.chart.plot.PlotOrientation;
09 import org.jfree.chart.plot.XYPlot;
10 import org.jfree.chart.renderer.xy.XYLineAndShapeRenderer;
11 import org.jfree.data.xy.XYSeries;
12 import org.jfree.data.xy.XYSeriesCollection;
13
14 public class PruebaJFreeChart {
15
16     private static Color COLOR_SERIE_1 = new Color(255, 128, 64);
17
18     private static Color COLOR_SERIE_2 = new Color(28, 84, 140);
19
20     private static Color COLOR_RECUADROS_GRAFICA = new Color(31, 87, 4);
21
22     private static Color COLOR_FONDO_GRAFICA = Color.white;
23
24     public JFreeChart crearGrafica(XYSeriesCollection dataset) {
25
26         final JFreeChart chart = ChartFactory.createXYLineChart("Tiempos de
entrenamientos", "Vuelta", "Tiempo (segundos)",
27                 dataset,
28                 PlotOrientation.VERTICAL,
29                 true, // uso de leyenda
30                 false, // uso de tooltips
31                 false // uso de urls
32                 );
33         // color de fondo de la gráfica
34         chart.setBackgroundPaint(COLOR_FONDO_GRAFICA);
35
36         final XYPlot plot = (XYPlot) chart.getPlot();
37         configurarPlot(plot);
38
39         final NumberAxis domainAxis = (NumberAxis)plot.getDomainAxis();
40         configurarDomainAxis(domainAxis);
41
42         final NumberAxis rangeAxis = (NumberAxis)plot.getRangeAxis();
43         configurarRangeAxis(rangeAxis);
44
45         final XYLineAndShapeRenderer renderer =
(XYLineAndShapeRenderer)plot.getRenderer();
46         configurarRendered(renderer);
47
48         return chart;
49     }
50
51     // configuramos el contenido del gráfico (damos un color a las líneas
que sirven de guía)
52     private void configurarPlot (XYPlot plot) {
53         plot.setDomainGridlinePaint(COLOR_RECUADROS_GRAFICA);
54         plot.setRangeGridlinePaint(COLOR_RECUADROS_GRAFICA);
55     }
56
57     // configuramos el eje X de la gráfica (se muestran números enteros y de
uno en uno)
58     private void configurarDomainAxis (NumberAxis domainAxis) {
59         domainAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());
60         domainAxis.setTickUnit(new NumberTickUnit(1));
61     }
62
```

```
    // configuramos el eje y de la gráfica (números enteros de dos en dos y rango entre 120 y 135)
    private void configurarRangeAxis (NumberAxis rangeAxis) {
        rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());
        rangeAxis.setTickUnit(new NumberTickUnit(2));
        rangeAxis.setRange(120, 135);
    }

    // configuramos las líneas de las series (añadimos un círculo en los puntos y asignamos el color de cada serie)
    private void configurarRendered (XYLineAndShapeRenderer renderer) {
        renderer.setSeriesShapesVisible(0, true);
        renderer.setSeriesShapesVisible(1, true);
        renderer.setSeriesPaint(0, COLOR_SERIE_1);
        renderer.setSeriesPaint(1, COLOR_SERIE_2);
    }

}
```

Bien, como vemos el método ChartFactory.createXYLineChart nos creará una gráfica de series representadas por líneas.

El método configurarPlot se ocupa de establecer el color de las líneas de fondo de la gráfica que hacen de guía. Son las líneas discontinuas verticales y horizontales que aparecen en el fondo de la gráfica.

El método configurarDomainAxis actua sobre el eje X, el eje del número de vueltas, y hace que se muestren números enteros y de uno en uno.

El método configurarRangeAxis actua sobre el eje Y, el de los tiempos, y hace que se muestren los tiempos comprendidos en el rango de 120 a 135 pero mostrando un número si, un número no.

El método configurarRndered hace que en cada punto (relación tiempo - vuelta de una serie) se muestre un círculo y que cada serie sea de un color.

5. Ejecutando el ejemplo.

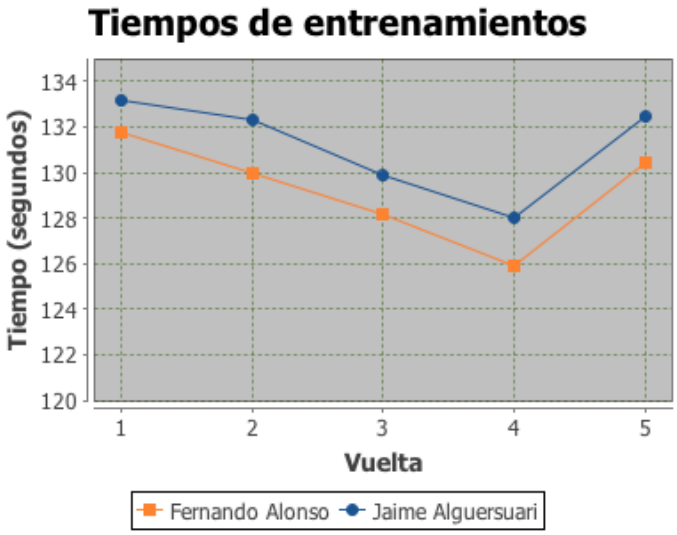
Por último, nos queda cargar los datos de los pilotos (usaremos los del otro tutorial) y generar la gráfica en un fichero.

```
public static final int ANCHO_GRAFICA = 400;

public static final int ALTO_GRAFICA = 300;

public static void main(String args[]) {

    final XYSeries serie1 = new XYSeries("Fernando Alonso");
    serie1.add(1, 131.78);
    serie1.add(2, 129.95);
    serie1.add(3, 128.16);
    serie1.add(4, 125.91);
    serie1.add(5, 130.44);

    final XYSeries serie2 = new XYSeries("Jaime Alguersuari");
    serie2.add(1, 133.16);
    serie2.add(2, 132.32);
    serie2.add(3, 129.86);
    serie2.add(4, 128.02);
    serie2.add(5, 132.45);

    final XYSeriesCollection collection = new XYSeriesCollection();
    collection.addSeries(serie1);
    collection.addSeries(serie2);

    try {
        final PruebaJFreeChart prueba = new PruebaJFreeChart();
        final JFreeChart grafica = prueba.crearGrafica(collection);
        ChartUtilities.saveChartAsPNG(new File("tiempos-entrenamientos.png"), grafica, ANCHO_GRAFICA, ALTO_GRAFICA);
    } catch (Exception e) {
        e.printStackTrace();
    }

}
```

Observemos que añadimos los tiempos de cada piloto en una serie llamada XYSeries. A esta serie únicamente le indicamos el nombre, que se corresponderá con el nombre del piloto, y el tiempo por vuelta. Ej: en la serie 1 hemos indicado que en la vuelta 1 el tiempo fue de 131.78 segundos, en la vuelta 2 fue de 129.95, etc...

Añadimos las gráficas a un dataset de tipo XYSeriesCollection y se lo pasamos al método crearGrafica.

Por último, con el método saveChartAsPNG de la clase ChartUtilities generamos una imagen en formato png, con el alto y ancho que queramos y que que dará como resultado una imagen llamada tiempos-entrenamiento.png. La gráfica resultante es la siguiente:



6. Referencias.

- JFreeChart Demo

7. Conclusiones.

En este tutorial hemos visto lo sencillo que es generar gráficas de series lineales con ayuda de la librería JFreeChart.

Nótese que esta librería es capaz de generar una gran cantidad de tipos de gráficas que pueden verse en la demo de su página web.

Espero que este tutorial os haya sido de ayuda. Un saludo.

Miguel Arlandy

marlandy@autentia.com

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

COMENTARIOS

sin clicks + + + + + + + +

powered by karmacracy