

SondaGrafica.java

```

1 package grafico;
2
3 import java.awt.Color;
4
5
6
7
8 public class SondaGrafica {
9
10     private SimpleGUI sg;
11
12     private final static int XBASE = 500;
13
14     private final static int FRACCION_AVANCE = 3;
15
16     private static int proximoIdSonda = 1;
17
18     private double tActual = 0;
19
20     private double tmin, tmax;
21
22     private double ymin, ymax;
23
24     private double[] muestras;
25
26     int posBaseMuestras = 0;
27
28     int posMuestraActual = 0;
29
30     int margenIzquierdo = 60, margenDerecho = 6, margenInferior = 20,
31         margenSuperior = 20;
32
33     private int anchuraGrafica, alturaGrafica;
34
35
36     public SondaGrafica(String titulo, double escala) {
37
38         sg = new SimpleGUI(2 * XBASE + margenDerecho,
39             (int) (2 * XBASE / 4.0 * 3.0), false);
40         sg.setTitle("Sonda grafica " + proximoIdSonda++ + ": " + titulo);
41         ymin = -escala;
42         ymax = escala;
43         tmin = 0;
44         tmax = 2 * XBASE;
45         dibujarEjes();
46     }
47
48
49     public SondaGrafica(String titulo) {
50         this(titulo, 1);
51     }
52
53
54     public void addMuestra(double muestra) {
55         if (muestras != null) {
56
57             sg.eraseSingleDrawable("punto");
58             redimensionarEjes(muestra);
59             dibujarPunto(tActual, muestras[(posBaseMuestras + posMuestraActual)
60                 % muestras.length], 1, Color.BLACK, null);
61
62             posMuestraActual++;
63
64             if (posMuestraActual == muestras.length) {

```

SondaGrafica.java

```

65         posBaseMuestras = (posBaseMuestras + muestras.length
66             / FRACCION_AVANCE)
67             % muestras.length;
68         posMuestraActual = muestras.length - muestras.length
69             / FRACCION_AVANCE;
70         sg.eraseAllDrawables();
71         dibujarEjes();
72         double incrementoX = (tmax - tmin) / FRACCION_AVANCE;
73         tmin += incrementoX;
74         tmax += incrementoX;
75         double t1 = tmin;
76         for (int i = 0; i < muestras.length - muestras.length
77             / FRACCION_AVANCE; i++) {
78             dibujarPunto(t1, muestras[(posBaseMuestras + i)
79                 % muestras.length], 1, Color.BLACK, null);
80             t1++;
81         }
82     }
83 } else
84
85     muestras = new double[(int) tmax];
86
87     muestras[(posBaseMuestras + posMuestraActual) % muestras.length] = muestra;
88     dibujarPunto(tActual, muestra, 4, Color.RED, "punto");
89     tActual++;
90
91     try {
92         Thread.sleep(2);
93     } catch (InterruptedException e) {
94     }
95 }
96
97
98 private void dibujarPunto(double t, double y, double radio, Color color,
99     String nombre) {
100     sg.drawDot(margenIzquierdo + (t - tmin) / (tmax - tmin)
101         * anchuraGrafica, margenSuperior
102         + (1 - (y - ymin) / (ymax - ymin)) * alturaGrafica, radio,
103         color, 1, nombre);
104 }
105
106
107 private void dibujarEjes() {
108     double y, separacion;
109
110
111     anchuraGrafica = sg.getWidth() - margenIzquierdo - margenDerecho;
112     alturaGrafica = sg.getHeight() - margenInferior - margenSuperior;
113
114     sg.drawLine(margenIzquierdo, 0, margenIzquierdo, sg.getHeight());
115
116     y = ymin;
117     separacion = (ymax - ymin) / 20;
118     while (y <= ymax) {
119         sg.drawLine(margenIzquierdo - 5, (1 - (y - ymin) / (ymax - ymin))
120             * alturaGrafica + margenSuperior, margenIzquierdo,
121             (1 - (y - ymin) / (ymax - ymin)) * alturaGrafica
122             + margenSuperior);
123         String valor = String.format("%.2f", y);
124         sg.drawText(valor, 0, (1 - (y - ymin) / (ymax - ymin))
125             * alturaGrafica + margenSuperior);
126         y += separacion;

```

```

127     }
128 }
129
130
131 private void redimensionarEjes(double muestra) {
132
133     double mmin = ymin, mmax = ymax;
134     if (muestra < mmin)
135         mmin = muestra;
136     if (muestra > mmax)
137         mmax = muestra;
138
139
140     if ((mmax != ymax) || (mmin != ymin)) {
141
142         if(false) System.out.println("Redimensionando...");
143         double orden, valorSignificativo;
144         final double cercaniaOrdenRedondeo = 1;
145         orden = Math.floor(Math.log10(Math.abs(mmax - mmin)))
146             - cercaniaOrdenRedondeo;
147         valorSignificativo = Math.floor(mmin / Math.pow(10, orden));
148         ymin = valorSignificativo * Math.pow(10, orden) * 2;
149         valorSignificativo = Math.ceil(mmax / Math.pow(10, orden));
150         ymax = valorSignificativo * Math.pow(10, orden) * 2;
151
152         if(false) System.out.println("Redimensionando y:" + ymax + "-" + ymin);
153         if (Math.signum(ymax) != Math.signum(ymin)
154             && Math.abs(Math.abs(ymax / ymin) - 1) <= 0.25) {
155             double max = Math.max(Math.abs(ymax), Math.abs(ymin));
156             ymax = max;
157             ymin = -max;
158         }
159         sg.eraseAllDrawables();
160         dibujarEjes();
161
162         for (int i = 0; i < posMuestraActual; i++)
163             dibujarPunto(i, muestras[(posBaseMuestras + i)
164                 % muestras.length], 1, Color.BLACK, null);
165
166     }
167 }
168
169 }
170

```