



En el artículo sobre **JAVA** presentado en la revista N°275 propusimos un ejercicio para poner a prueba los conocimientos adquiridos sobre este lenguaje de programación. En este número publicamos la solución, incluyendo el listado completo del código fuente del programa y las correspondientes explicaciones sobre su realización.

# Programar en JAVA

El artículo sobre **Java** presentado en la revista N°275 se publicó con el objetivo de exponer las **técnicas necesarias** para tener un **primer contacto** con el lenguaje y ser capaces de **controlar dispositivos hardware**, **independientemente** del **sistema operativo** utilizado. Como dispositivo utilizamos nuestra versátil **interfaz serie-paralelo LX.1127**.

Los numerosos correos que hemos recibido confirman que nuestro objetivo ha sido cumplido.

Efectivamente, **Java** es un **lenguaje universal** de **alto nivel** muy adecuado para desarrollar aplicaciones **independientes** del **sistema operativo utilizado**.

Un programa escrito en **Java** puede ser ejecutado en **múltiples plataformas y sistemas**, tales como entornos **Windows**, **MAC** o **Linux**. De hecho hasta los **teléfonos móviles actuales** pueden ejecutar **programas Java**.

El artículo de la revista N°275 concluyó con el planteamiento de un **ejercicio práctico** que pusiera de manifiesto los conocimientos adquiridos. Se trataba de construir una **Interfaz Gráfica de Usuario (GUI)** capaz de mostrar las **posiciones (valor binario)** del **dipswitch** incluido en la **interfaz LX.1127** al pulsar un **botón en pantalla**.

En el propio artículo expusimos las **“pistas”** principales para realizar el ejercicio.

Estamos seguros de que muchos lo han realizado, no obstante hoy exponemos la **solución detallada** al ejercicio planteado.

### SOLUCIÓN al ejercicio JAVA

Partimos de la premisa de que se ha **leído**, y entendido, el **primer artículo** sobre **Java**, por lo que nos vamos a centrar en las explicaciones de las **funciones nuevas** y de las **funciones esenciales**.

Para exponer la solución de forma detallada y clara se adjunta el **listado completo** del **código fuente** del programa.

Recordamos una vez más que los **números** situados a la **izquierda** de cada línea **no** son parte del **código (no deben copiarse)**, los utilizamos como **referencia** para localizar de forma precisa las diferentes líneas en la descripción del programa.

Comenzamos la descripción por la **línea 87**, correspondiente a la función **main**. Con estas instrucciones indicamos al programa que vamos a crear una **interfaz gráfica**, de hecho en la **línea 97** se llama a la función **createAndShowGUI** (definida en la **línea 76** y siguientes).Vamos a analizarla.

En primer lugar se define un **frame** (o **ventana**) en la que posteriormente se **insertarán** los **componentes, botones y etiquetas**.

Además de realizar algunas operaciones estándar en la **línea 81** se llama al método **addComponentsToPane**, que procede a **asignar** los componentes que se **dibujarán**.

La **definición** de los **componentes asignados**, o lo que es lo mismo, la **inclusión de elementos** en **addComponentsToPane** se realiza en la **línea 41** y siguientes.

Hemos de imaginar la **ventana** como una serie de **rejillas** compuestas por **líneas** y **columnas** dentro de las que colocaremos los **componentes**.

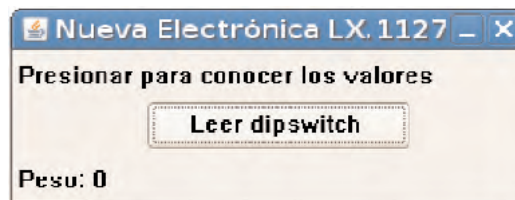
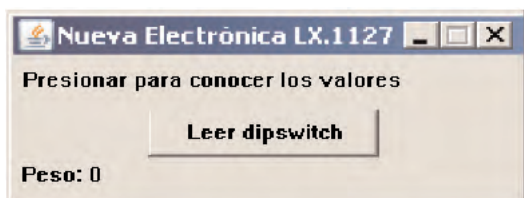
En la **primera rejilla**, que será la **principal**, se definen **3 líneas** y **0 columnas** (**línea 43**).

Aquí vamos a crear un **único botón** que posicionamos dentro de una **rejilla** compuesta por **0 líneas** y **1 columna** (**línea 45**).

Llegado este punto **creamos** el **botón**, le asignamos una **acción** y lo **colocamos** en la **rejilla** (**líneas 47-48-49**).

# eI PUERTO SERIE II

**Fig.1** Con las instrucciones del programa GuiEjercicio, cuyo código fuente completo se presenta en este artículo, se crea una ventana que incluye un botón tradicional, esto es, cuando se deja de presionar vuelve a su estado normal.



**Fig.2** Si el sistema operativo utilizado es Windows 2000 la ejecución del programa mostrará una interfaz similar a ésta. Se puede apreciar claramente el gran parecido que tiene con el programa ejecutándose bajo Windows XP (Fig.1).

```

1 import gnu.io.CommPort;
2 import gnu.io.CommPortIdentifier;
3 import gnu.io.SerialPort;
4
5 import java.awt.*;
6 import java.awt.event.*;
7 import java.io.InputStream;
8 import java.io.OutputStream;
9
10 import javax.swing.*;
11
12 public class GuiEjercicio extends JFrame implements ActionListener {
13
14     private static final long serialVersionUID = 1L;
15     private int peso = 0;
16     private JLabel pesoLabel = null;
17     private OutputStream out = null;
18     private InputStream dipSwitch = null;
19
20     public GuiEjercicio(String name) throws Exception {
21         super(name);
22         setResizable(false);
23
24         CommPortIdentifier portIdentifier =
CommPortIdentifier.getPortIdentifier("COM1");
25
26         if (portIdentifier.isCurrentlyOwned()) {
27             System.out.println("Error: El puerto está en uso.");
28         } else {
29             CommPort commPort = portIdentifier.open("GuiEjercicio",
2000);
30             if (commPort instanceof SerialPort) {
31                 SerialPort serialPort = (SerialPort) commPort;
32                 serialPort.setSerialPortParams(2400,
SerialPort.DATABITS_8, SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
33                 out = serialPort.getOutputStream();
34                 dipSwitch = serialPort.getInputStream();
35             } else {
36                 System.out.println("Error: Solo se aceptan puertos
serie.");
37             }
38         }
39     }
40
41     public void addComponentsToPane(final Container pane) {
42         final JPanel mainPanel = new JPanel();
43         mainPanel.setLayout(new GridLayout(3, 0));
44         JPanel controls = new JPanel();
45         controls.setLayout(new GridLayout(0, 1));
46
47         JButton button = new JButton("Leer dipswitch");
48         button.addActionListener(this);
49         controls.add(button);
50

```

Para **compilar** el programa **GuiEjercicio** con un único comando teclear:

**C:\Archivos de programa\java\jdk1.6.0\_04\bin\javac.exe C:\ProgramasJava\GuiEjercicio.java**

```

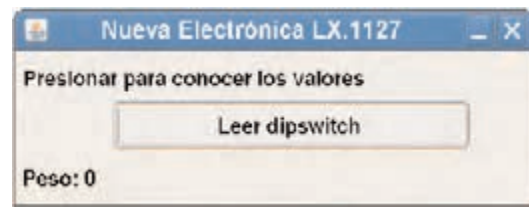
51         mainPanel.add(new JLabel(" Presionar para conocer los
valores"));
52     mainPanel.add(controls);
53     pesoLabel = new JLabel("Peso: " + peso);
54     mainPanel.add(pesoLabel);
55     pane.add(mainPanel);
56 }
57
58     public void actionPerformed(ActionEvent e) {
59         try {
60             receiveData();
61         } catch (Exception e1) {
62             e1.printStackTrace();
63         }
64         pesoLabel.setText("Peso: " + peso);
65     }
66 }
67     private void receiveData() throws Exception {
68         out.write((byte) 1);
69         out.write((byte) 0);
70         out.write((byte) 3);
71         byte[] buffer = new byte[1];
72         dipSwitch.read(buffer);
73         peso = 255 - (0xFF & ((char) buffer[0]));
74     }
75 }
76     private static void createAndShowGUI() throws Exception {
77         // Crea la ventana principal
78         GuiEjercicio frame = new GuiEjercicio("Nueva Electrónica
LX.1127");
79         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
80         // Asigna el contenido de la ventana
81         frame.addComponentsToPane(frame.getContentPane());
82         // Visualiza la ventana tra
83         frame.pack();
84         frame.setVisible(true);
85     }
86
87     public static void main(String[] args) {
88         //Usa el tema del sistema operativo
89         try {
90             UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
91         } catch (Exception ex) {
92             ex.printStackTrace();
93         }
94         javax.swing.SwingUtilities.invokeLater(new Runnable() {
95             public void run() {
96                 try {
97                     createAndShowGUI();
98                 } catch (Exception e) {
99                     e.printStackTrace();
100                 }
101             }
102         });
103     }
104 }

```

Para **ejecutar** el programa **GuiEjercicio** con un único comando teclear:

**C:\Archivos de programa\java\jdk1.6.0\_04\bin\java.exe C:\ProgramasJava\GuiEjercicio**

**Fig.3** Con un sistema operativo Linux el programa GuiEjercicio mostrará una interfaz similar a la aquí expuesta, siempre y cuando se utilicen las mismas líneas de código.



Seguramente, si se compara con el botón creado en el programa **GuiLX1127** del artículo de la **revista N°275**, se apreciará que es **diferente**. Veamos esas diferencias.

En el programa **GuiLX1127** utilizamos la función **JToggleButton** para dibujar un botón que se **quedaba pulsado** después de hacer **click**. Se realizó de esta forma para controlar adecuadamente el **diodo LED** que estaba **encendido**.

En el caso que nos ocupa ahora la utilización de **JToggleButton** sería **inútil e inoportuna**, ya que basta con utilizar un **botón común**, es decir una vez que se **deja de pulsar** vuelve a su **estado normal**.

Después de esta aclaración basta con decir que los **botones sin funciones particulares** se definen en **Java** mediante la clase **JButton**.

La última función que vamos a analizar es **actionPerformed** (línea 58). Esta función se ocupa de **interaccionar** con el **botón** actualizando, cuando se pulsa, la **etiqueta** de los **pesos**.

Cada vez que se **presiona** el **botón** se invoca esta **función**, eso sí, sólo se la invoca al hacer **click** sobre el único botón presente en nuestra **GUI**.

La primera operación a realizar es **leer** el valor del **dipswitch** mediante **receiveData**. Seguidamente el valor leído se vuelca a la **etiqueta**.

El resto del programa es **similar** al expuesto en la revista **N°275**: Las **líneas 68 a 73** reciben los **datos** procedentes del **puerto serie**, mientras que las **líneas 31 a 34** se ocupan de **programar** el puerto **COM** según las directivas descritas en el artículo anterior.

**NOTA** En sistemas **Linux** el **puerto serie 1 (COM1)** generalmente se identifica como **/dev/ttyS0**, el **puerto serie 2 (COM2)** como **/dev/ttyS1**, etc.

Poco más. Ha sido muy sencillo realizar una **GUI personalizada** que procese la **información recibida** por un **puerto serie**.

Como se vio en el artículo anterior para **compilar** y **ejecutar** el programa cambiamos al directorio **bin** del **entorno de desarrollo** y tecleamos las órdenes correspondientes:

```
CD "\\Archivos de programa\\java\\jdk1.6.0_04\\bin\"  
ENTER
```

```
Javac C:\\ProgramasJava\\GuiEjercicio.java  
ENTER
```

```
Java C:\\ProgramasJava\\GuiEjercicio  
ENTER
```

## PARA CONCLUIR

Han sido varias las consultas recibidas en las que nos habéis preguntado dónde es posible hallar **manuales** para aprender a **programar en Java**.

Aquí exponemos algunas **direcciones interesantes** de la empresa desarrolladora de **Java (Sun Microsystems)**. El único requisito para su utilización es ... saber **inglés**.

**Tutorial Java** con un buen conjunto de **ejemplos prácticos** para **utilizar inmediatamente**:

<http://java.sun.com/docs/books/tutorial/>

**Libros y manuales** para programar en **Java**:

[http://www.sun.com/books/java\\_series.html](http://www.sun.com/books/java_series.html)