



TECNOLÓGICO
NACIONAL DE MÉXICO



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA

ASIGNATURA: TOPICOS AVANZADOS DE PROGRAMACIÓN

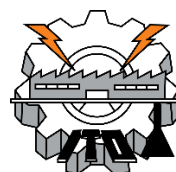
CATEDRÁTICO: HERNANDEZ ABREGO ANAYANSI CRISTINA

ALUMNO: GARCÍA GARCÍA JOSÉ ÁNGEL

UNIDAD: 2

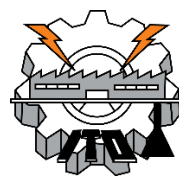
**PRACTICA 1- Creación de una aplicación “GUI” usando componentes
disponibles por el lenguaje**

OAXACA DE JÚAREZ, OAX, 08/MARZO/2020



Índice

BITACORA	3
CÓDIGO FUENTE DEL COMPONENTE	5
CÓDIGO FUENTE DE LA APLICACIÓN PRUEBA.....	9
CÓDIGO FUENTE DE LA APLICACIÓN IMPORTADO DESDE BIBLIOTECA	15



BITACORA

La interface `Serializable` sirve para poder representar a los objetos de la clase por una secuencia de bytes que incluye los datos del objeto, el tipo de objeto y los tipos de datos que contiene el objeto. Es utilizado en nuestro para poder crear la biblioteca, ya que podemos guardar el estado de un objeto en fichero, y permite posteriormente leer esos bytes para reconstruir el objeto y su estado en la memoria de nuevo. Además, puede ser serializado en una máquina y deserializado en otra máquina distinta. La mayoría de las clases del API de Java implementan esta interface.

Es necesario el constructor sin parámetros para poder inicializar los atributos de la clase y no tener errores en ejecución, además de que siempre se debe tener un constructor simple como buena práctica, por si al cliente se le olvida agregar parámetros al constructor.

La parte que completa el código del método `iniciarElementos()` es:

```
for (int i = 0; i < nfiguras; i++) {  
    colorB[i] = new JLabel(); // Se crea la etiqueta a utilizar  
    colorB[i].setOpaque(true); // Permite establecer un color de fondo a la etiqueta  
    colorB[i].setBackground(color[i]); // Se establece el color de fondo a la etiqueta  
    add(colorB[i]); // Se agrega la etiqueta al componente  
}
```

El método `setValores()`:

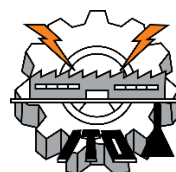
`if(valores.length == nfiguras) // Si el numero de valores no es el mismo que el de barras, las construye en 0`

```
for (int i = 0; i < nfiguras; i++) {  
    vals[i] = valores[i]; // Establece los nuevos valores a cada elemento del arreglo de valores  
    etis[i].setText("" + vals[i]); // Establece el valor en tipo texto a cada etiqueta  
}
```

La parte que tambien nos solicitan agregar al método `paintComponen()` es:

`colorB[nfiguras-1-i].setBounds(separa,getHeight()-(16*j)-(3*i),15,15); // Se posiciona el cuadro de color para cada leyenda en el componente`









`leyendaBarra[nfiguras-1-i].setBounds(separa+25,getHeight()-(18*j)-(2*i),getWidth(),20); // Se posiciona la leyenda en el componente`



Se hizo uso de GitHub para guardar cada cambio realizado durante la práctica. Aquí se muestra los cambios realizados de cada código. El primero es el del componente, el segundo corresponde a las pruebas del mismo y el tercero es la aplicación que hace uso del componente.





Historial de [tópicos](#) / [tap_practica_04](#) / [src](#) / [com_grafico](#)

Se compromete el 7 de marzo de 2020

ProyectoAplicacion ChepeAicrag12 comprometido hace 17 horas	 7fbd80c 
PruebasDeGrafico ... ChepeAicrag12 comprometido hace 23 horas Se tienen en clases diferentes, las pruebas solicitadas.	 80c3ff1 
ClaseProbada0 ... ChepeAicrag12 comprometido ayer ClaseGrafico probada con valores en 0	 c6b8759 
CreacionClaseGrafico ChepeAicrag12 comprometido ayer	 b5579a1 

Historia de [Temas](#) / [ProyectoPrueba](#) / [src](#) / [proyectorprueba](#)

Se compromete el 7 de marzo de 2020

ProyectoAplicacion ChepeAicrag12 comprometido hace 17 horas	 7fbd80c 
CreacionDeJAR ... ChepeAicrag12 comprometido hace 23 horas Se ha creado el .jar de proyecto donde está gráfico.	 1429fa4 

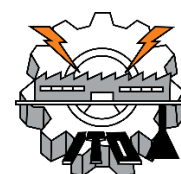
Historia de [Temas](#) / [ProyectoAplicado](#) / [src](#) / [proyectoaplicado](#)

Se compromete el 8 de marzo de 2020

TerminadoProyectoConGraficoAplicado ... ChepeAicrag12 comprometido hace 2 minutos Se culmina la aplicación que hace uso del componente Gráfico.	 a175ec2 
--	---

Se compromete el 7 de marzo de 2020

ProyectoAplicacion ChepeAicrag12 comprometido hace 17 horas	 7fbd80c 
---	---



CÓDIGO FUENTE DEL COMPONENTE

CÓDIGO

```
package com_grafico;

import java.awt.Color;
import java.awt.Graphics;
import java.io.Serializable;
import javax.swing.JComponent;
import javax.swing.JLabel;

/**
 * @author García García José Ángel
 */
public class Grafico extends JComponent implements Serializable{
    private int nfiguras; // Numero de barras
    private JLabel [] etis; // etiquetas para el valor a mostrar
    private int [] vals ={0,0,0}; // valores de cada barra, se dan valores por omision
    // se consideran 10 colores para 10 valores
    private static Color color[] ={Color.BLUE,Color.GREEN,Color.CYAN
        ,Color.YELLOW,Color.PINK,Color.WHITE,Color.RED,Color.ORANGE,Color.MAGENTA};
    private static JLabel leyendaBarra[]; //Muestra el titulo de la leyenda
    private static JLabel colorB[]; // identificacion del color de las barras
    private JLabel titulo; // titulo
    private String tTitulo; // texto del titulo
    private String tLeyenda[]; // titulos de las leyendas que identifican a cada barra

    public Grafico(){ // Constructor sin parametros para crear instancias con valores; bitacora
        titulo = new JLabel(""); // Crea la etiqueta que tiene el titulo
        etis = new JLabel[0]; // Crea el arreglo de etiquetas que tiene los valores de las barras
        leyendaBarra = new JLabel[0]; // Crea el arreglo de etiquetas que tiene las leyendas de las
        barras
    }
```



```
colorB = new JLabel[0]; // Crea el arreglo de etiquetas que tiene los colores de las barras para
colocar juanto a las leyendas
```

```
tLeyenda = new String[0]; // Crea el arreglo de los titulos de las leyendas para las barras
// Se tiene el cosntructor simple, para ello se debe
// inicializar los atributos de la clase, en especial arreglos para que
// puedan se usados posteriormente.
}
```

```
// Constructor con parametros para el titulo y las leyendas de las barras
```

```
public Grafico(String encabezado, String tLeyenda[]){
    nfiguras = tLeyenda.length; // Establece el numero de barras
    this.tLeyenda = tLeyenda;
    leyendaBarra = new JLabel[tLeyenda.length];
    tTitulo = encabezado;
    setLeyendas(tLeyenda);
    iniciarElementos(); // metodo para iniciar los valores
}
```

```
private void iniciarElementos() {
    vals = new int[nfiguras]; // Crea el arreglo que guarda los valores
    for (int i = 0; i < nfiguras; i++) {
        vals[i] = 0; // Establece 0 a todos los valores, por defecto
    }
    titulo = new JLabel(tTitulo); // crea la etiqueta del titulo
    add(titulo); // Agregamos el titulo al componente
    etis = new JLabel[nfiguras]; // Creamos el arreglo de etiquetas
    colorB = new JLabel[nfiguras]; // Creamos el arreglo de colores
    for (int i = 0; i < nfiguras; i++) {
        etis[i] = new JLabel("" + 0); // Asignamos el valor de barra a la etiqueta correspondiente
        add(etis[i]); // Agregamos la etiqueta al componente
    }
}
```



```

    }
    // identificacion del color de cada barra colorB[]
    for (int i = 0; i < nfiguras; i++) {
        colorB[i] = new JLabel(); // Se crea la etiqueta a utilizar
        colorB[i].setOpaque(true); // Permite establecer un color de fondo a la etiqueta
        colorB[i].setBackground(color[i]); // Se establece el color de fondo a la etiqueta
        add(colorB[i]); // Se agrega la etiqueta al componente
    }
}

public void setTitulo(String encabezado){
    this.tTitulo = encabezado; // Establece el titulo al componente de acuerdo con el del parametro
}

public void setLeyendas(String tLeyenda[]){
    nfiguras = tLeyenda.length; // Establece la cantidad de figuras
    this.tLeyenda = tLeyenda;
    leyendaBarra = new JLabel[nfiguras]; // Crea el arreglo de las leyendas de barra
    for (int i = 0; i < nfiguras; i++) {
        leyendaBarra[i] = new JLabel(tLeyenda[i]); // establece el texto a cada etiqueta leyenda
        add(leyendaBarra[i]); // Agrega las leyendas de cada barra al componente
    }
    iniciarElementos();
}

public void setValores(int valores[]){
    if(valores.length == nfiguras) // Si el numero de valores no es el mismo que el de barras, las
    construye en 0
    for (int i = 0; i < nfiguras; i++) {
        vals[i] = valores[i]; // Establece los nuevos valores a cada elemento del arreglo de valores
        etis[i].setText("" + vals[i]); // Establece el valor en tipo texto a cada etiqueta
    }
}

```



```

    }
}

@Override
public void paintComponent(Graphics f){
    int i;
    int j = 1;
    int ancho = getWidth(); // Se almacena el ancho del componente
    int alto = getHeight(); // Se almacena el alto del componente
    titulo.setBounds((int)((ancho-f.getFontMetrics().stringWidth(titulo.getText()))/2), 10,
    titulo.getText().length()*8, 20); // Se posiciona el titulo
    int separa = (int)(ancho / (nfiguras)* 0.10); // Se guarda un valor separador entre barras
    int anchoB = (int)(ancho / (nfiguras) - separa); // Se guarda el ancho entre barras

    for (i = 0; i < nfiguras; i++) {
        f.setColor(color[i]); // Establece el color de la barra
        etis[i].setBounds(separa+i*(anchoB+separa)+(anchoB/2), (alto-20*colorB.length)-vals[i]*5-
        etis[i].getHeight()-2,30,20); // Posiciona el valor de cada barra
        f.fillRect(separa+i*(anchoB+separa),((alto-20*colorB.length)-vals[i]*5), anchoB, vals[i]*5,
        true); // Crea la barra de color en posicion especificada
        colorB[nfiguras-1-i].setBounds(separa,getHeight()-(16*j)-(3*i),15,15); // Se posiciona el cuadro
        de color para cada leyenda en el componente
        leyendaBarra[nfiguras-1-i].setBounds(separa+25,getHeight()-(18*j)-(2*i),getWidth(),20); // Se
        posiciona la leyenda en el componente
        j++;
    }
}
}
}

```



CÓDIGO FUENTE DE LA APLICACIÓN PRUEBA

CÓDIGO

```
package com_grafico;

import javax.swing.JFrame;

/**
 * @author Garcia Garcia Jose Angel
 * Prueba 2
 */
public class prueba extends JFrame{

    public prueba(){

        setSize(600, 400);

        setVisible(true);

        setLocationRelativeTo(null);

        String[] tLeyenda = {"Leyenda de barra 1 ", "Leyenda de barra 2 ",
                             "Leyenda de barra 3 ", "Leyenda de barra 4 "};

        Grafico g = new Grafico("Muestra con 2 valores", tLeyenda);

        add(g);

        setDefaultCloseOperation(EXIT_ON_CLOSE);

    }

    public static void main(String[] args) {

        prueba b = new prueba();

    }

}
```



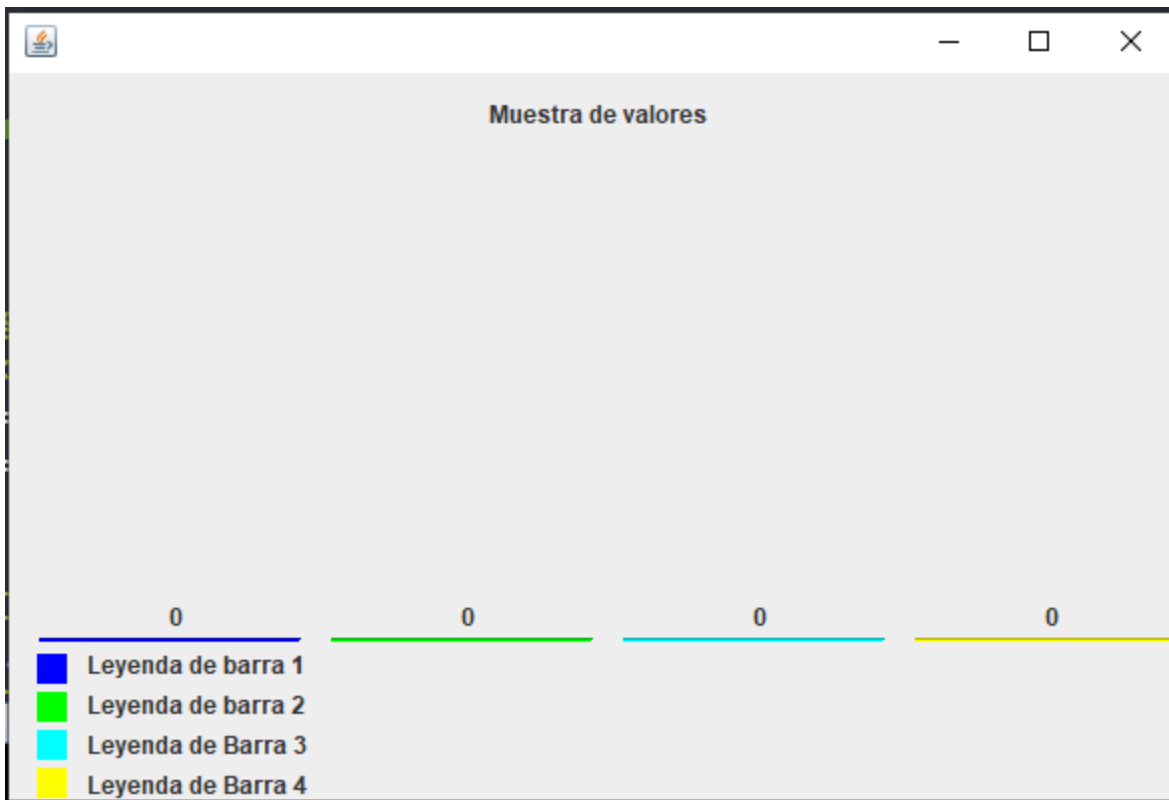


Ilustración 1- Código 1

```
package com_grafico;

import javax.swing.JFrame;

/**
 * @author Garcia Garcia Jose Angel
 * Prueba 1
 */
public class ProyectoPrueba extends JFrame{

    public ProyectoPrueba(){
        setSize(600, 400);
        setVisible(true);
        setLocationRelativeTo(null);
        String[] tLeyenda = {"Leyenda de barra 1 ", "Leyenda de barra 2 ",
                             "Leyenda de barra 3 ", "Leyenda de barra 4 ", "Leyenda de barra 5 "};
        Grafico g = new Grafico("Muestra con 5 valores", tLeyenda);
    }
}
```



```

int[] a = {10,30,20,15,18};
g.setValores(a);
add(g);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}

public static void main(String[] args) {
    new ProyectoPrueba();
}
}

```

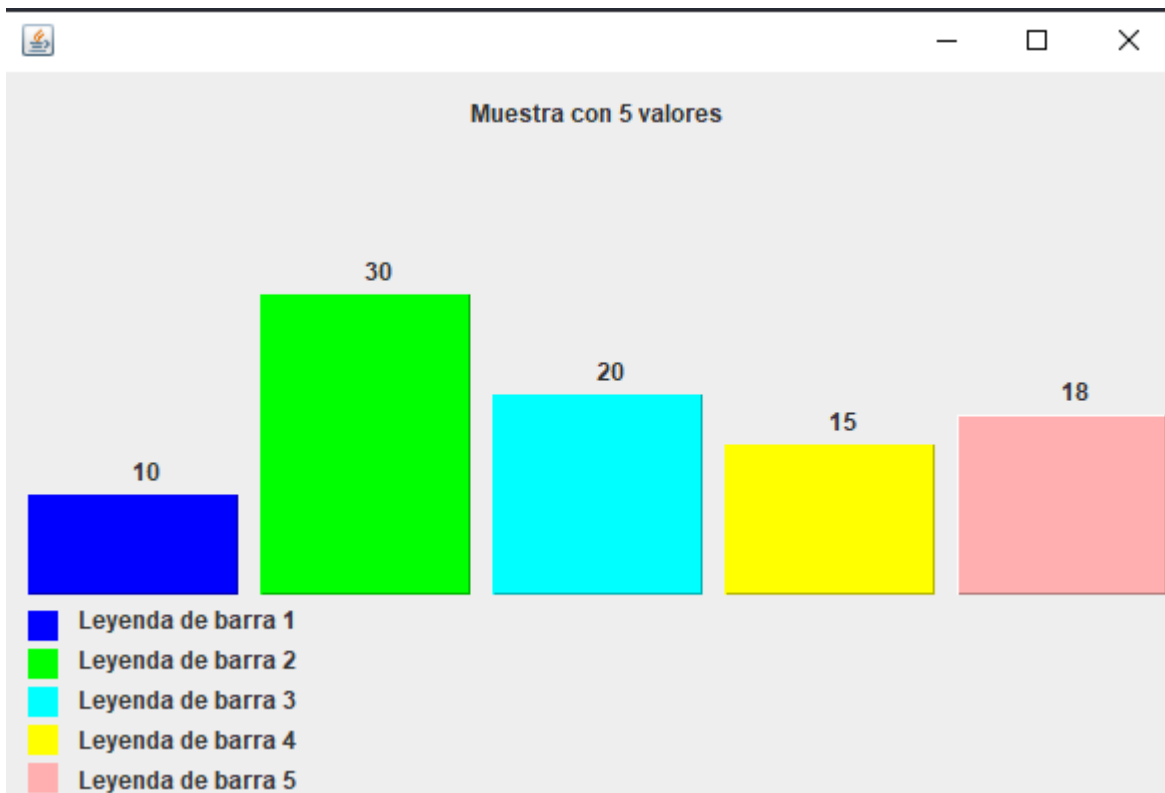


Ilustración 2- Código 2



```

package com_grafico;

import javax.swing.JFrame;

/**
 *
 * @author Garcia Garcia Jose Angel
 */
public class pruebaB extends JFrame{

    public pruebaB(){

        setSize(600, 400);

        setVisible(true);

        setLocationRelativeTo(null);

        String[] tLeyenda = {"Leyenda de barra 1 ", "Leyenda de barra 2 ",
                               "Leyenda de barra 3 ", "Leyenda de barra 4 "};

        Grafico g = new Grafico("Muestra con 4 valores", tLeyenda);

        int[] a = {10,30,20,10};

        g.setValores(a);

        add(g);

        setDefaultCloseOperation(EXIT_ON_CLOSE);

    }

    public static void main(String[] args) {

        pruebaB b = new pruebaB();

    }

}

```



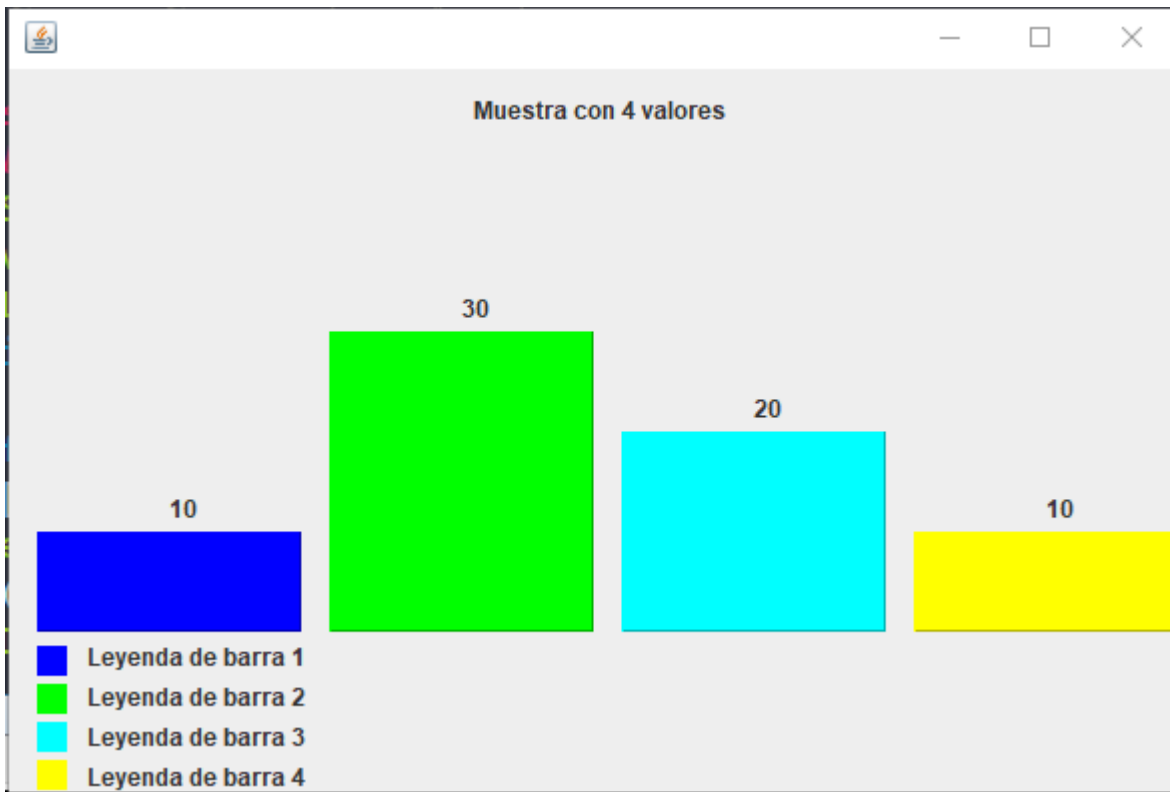


Ilustración 3- Código 3

```
package com_grafico;

import javax.swing.JFrame;

/**
 * @author Garcia Garcia Jose Angel
 * Prueba 3
 */
public class pruebaC extends JFrame{

    public pruebaC(){

        setSize(800, 400);

        setVisible(true);

        setLocationRelativeTo(null);

        String[] tLeyenda = {"Leyenda de barra 1 ", "Leyenda de barra 2 "};

        Grafico g = new Grafico("Muestra con 2 valores", tLeyenda);

        int[] a = {10,30};

        g.setValores(a);
    }
}
```



```
add(g);  
setDefaultCloseOperation(EXIT_ON_CLOSE);  
}  
public static void main(String[] args) {  
    pruebaC b = new pruebaC();  
}  
}
```

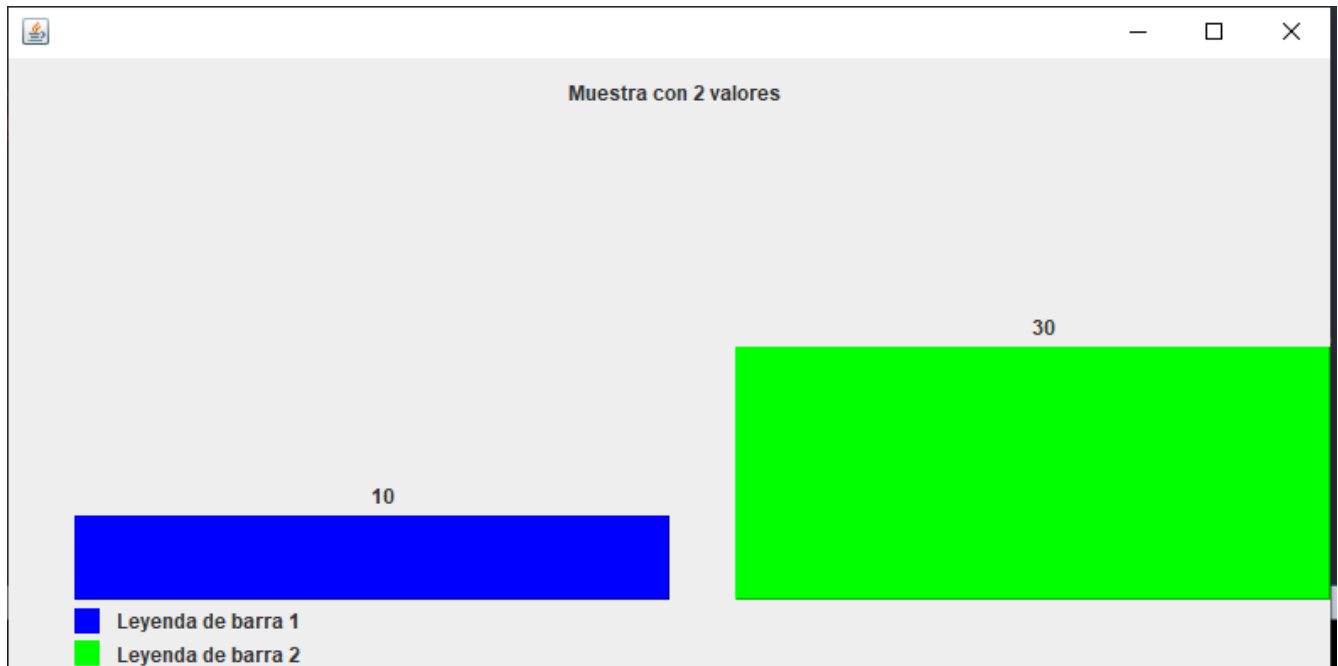


Ilustración 4- Prueba 4



CÓDIGO FUENTE DE LA APLICACIÓN IMPORTADO DESDE BIBLIOTECA

CLASE MVC

```
/*
 * Proyecto creado con la intención de usar el componente recién creado (Grafico)
 * Se aplica el diseño MVC.
 */
package proyectoaplicado;
import javax.swing.JFrame;
/**
 *
 * @author Garcia Garcia Jose Angel
 */
public class MVC {
    public static void main(String[] args) {
        Vista v = new Vista();
        Modelo m = new Modelo();
        Controlador c = new Controlador(v, m);
        v.conectarControlador(c);
        v.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

CLASE CONTROLADOR

```
package proyectoaplicado;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import javax.swing.JOptionPane;
```



```

/**
 *
 * @author Garcia Garcia Jose Angel
 */
public class Controlador implements KeyListener, ActionListener {

    private Vista v;
    private Modelo m;

    public Controlador(Vista v, Modelo m) {
        this.v = v;
        this.m = m;
    }

    @Override
    public void keyTyped(KeyEvent ke) {

    }

    @Override
    public void keyPressed(KeyEvent ke) {

    }

    @Override
    public void keyReleased(KeyEvent ke) {
        try {
            m.agregarCuadros(v);
        } catch (Exception e) {
        }
    }

```




```
}
```

```
@Override
```

```
public void actionPerformed(ActionEvent ae) {
```

```
    JOptionPane.showMessageDialog(v, "Mostrando grafico en otro Frame");
```

```
    m.mostrar(v);
```

```
}
```

```
}
```

CLASE MODELO

```
package proyectoaplicado;
```

```
/**
```

```
*
```

```
* @author Garcia Garcia Jose Angel
```

```
*/
```

```
public class Modelo {
```

```
    public Modelo() {
```

```
}
```

```
    public void mostrar(Vista v) {
```

```
        v.printf();
```

```
}
```

```
    public void agregarCuadros(Vista v) {
```

```
        v.agregarOpciones();
```

```
        v.bloquearTxt(false);
```

```
}
```



```
}
```

CLASE VISTA

```
package proyectoaplicado;
```

```
import com_grafico.Grafico;
```

```
import java.awt.Container;
```

```
import java.awt.event.ActionListener;
```

```
import java.awt.event.KeyListener;
```

```
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.JTextField;
```

```
import javax.swing.SpringLayout;
```

```
/**
```

```
*
```

```
* @author Garcia Garcia Jose Angel
```

```
*/
```

```
public class Vista extends JFrame {
```

```
    private JTextField txt1;
```

```
    private JTextField txt2[];
```

```
    private JTextField txtLeyendas[];
```

```
    private JLabel etqG[];
```

```
    private JLabel etq1;
```

```
    private JLabel etq2;
```

```
    private JPanel p;
```

```
    private Grafico g;
```



```

private JButton graficar;
private JTextField titulo;
private JLabel etqTitulo;
private JLabel valor;
private JLabel leyen;

SpringLayout s;

public Vista() {
    super("Aplicación con el complemento \"Grafico\"");
    setResizable(false);
    setVisible(true);
    s = new SpringLayout();
    g = new Grafico();
    add(componente());
    setLocationRelativeTo(null);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    p.updateUI();
}

public int getTxt1() {
    if (txt1.getText().isEmpty()) {
        return 0;
    }
    int a = Integer.parseInt(txt1.getText());
    if (a < 11 && a > 0) {
        return a;
    }
    return 0;
}

```



```

public String getTitulo() {
    if (titulo.getText().isEmpty()) {
        return "Sin titulo";
    }
    return titulo.getText();
}

```

```

public Container componente() {
    setSize(400, 100);
    p = new JPanel();
    p.setLayout(s);
    graficar = new JButton("Graficar");
    etqTitulo = new JLabel("Introduce el titulo");
    titulo = new JTextField(10);
    etq1 = new JLabel("Introduce la cantidad de barras");
    txt1 = new JTextField(10);
    p.add(etqTitulo);
    s.putConstraint(SpringLayout.WEST, etqTitulo, 12, SpringLayout.WEST, p);
    s.putConstraint(SpringLayout.NORTH, etqTitulo, 12, SpringLayout.NORTH, p);
    p.add(titulo);
    s.putConstraint(SpringLayout.WEST, titulo, 90, SpringLayout.EAST, etqTitulo);
    s.putConstraint(SpringLayout.NORTH, titulo, 12, SpringLayout.NORTH, p);
    p.add(etq1);
    s.putConstraint(SpringLayout.WEST, etq1, 12, SpringLayout.WEST, p);
    s.putConstraint(SpringLayout.NORTH, etq1, 12, SpringLayout.SOUTH, etqTitulo);
    p.add(txt1);
    s.putConstraint(SpringLayout.NORTH, txt1, 12, SpringLayout.SOUTH, titulo);
    s.putConstraint(SpringLayout.WEST, txt1, 12, SpringLayout.EAST, etq1);
    etq2 = new JLabel("Introduce los datos de las barras");
}

```



```

p.add(etq2);
s.putConstraint(SpringLayout.NORTH, etq2, 25, SpringLayout.SOUTH, etq1);
s.putConstraint(SpringLayout.WEST, etq2, 140, SpringLayout.WEST, p);
valor = new JLabel("Valor");
p.add(valor);
s.putConstraint(SpringLayout.NORTH, valor, 12, SpringLayout.SOUTH, etq2);
s.putConstraint(SpringLayout.WEST, valor, 145, SpringLayout.WEST, p);
leyen = new JLabel("Leyenda");
p.add(leyen);
s.putConstraint(SpringLayout.NORTH, leyen, 12, SpringLayout.SOUTH, etq2);
s.putConstraint(SpringLayout.WEST, leyen, 90, SpringLayout.EAST, valor);
return p;
}

```

```

public void agregarOpciones() {
    setSize(430, 470);
    int nBarras = getTxt1();
    txt2 = new JTextField[nBarras];
    etqG = new JLabel[nBarras];
    txtLeyendas = new JTextField[nBarras];
    for (int i = 0; i < nBarras; i++) {
        etqG[i] = new JLabel("Datos Barra " + (i + 1));
        txt2[i] = new JTextField(10);
        txtLeyendas[i] = new JTextField(15);
        p.add(etqG[i]);
        p.add(txt2[i]);
        p.add(txtLeyendas[i]);
        s.putConstraint(SpringLayout.WEST, etqG[i], 12, SpringLayout.WEST, p);
        s.putConstraint(SpringLayout.WEST, txt2[i], 15, SpringLayout.EAST, etqG[i]);
        s.putConstraint(SpringLayout.WEST, txtLeyendas[i], 15, SpringLayout.EAST, txt2[i]);
    }
}

```



```

    if (i == 0) {
        s.putConstraint(SpringLayout.NORTH, txt2[i], 12, SpringLayout.SOUTH, leyen);
        s.putConstraint(SpringLayout.NORTH, etqG[i], 15, SpringLayout.SOUTH, leyen);
        s.putConstraint(SpringLayout.NORTH, txtLeyendas[i], 12, SpringLayout.SOUTH, leyen);
    } else {
        s.putConstraint(SpringLayout.NORTH, txt2[i], 10, SpringLayout.SOUTH, txt2[i - 1]);
        s.putConstraint(SpringLayout.NORTH, etqG[i], 14, SpringLayout.SOUTH, etqG[i - 1]);
        s.putConstraint(SpringLayout.NORTH, txtLeyendas[i], 10, SpringLayout.SOUTH,
txtLeyendas[i - 1]);
    }
    if (getTxt1() == 0) {
        graficar.setEnabled(false);
    }
    p.add(graficar);
    s.putConstraint(SpringLayout.WEST, graficar, 15, SpringLayout.EAST, txt1);
    s.putConstraint(SpringLayout.NORTH, graficar, 12, SpringLayout.NORTH, p);
    p.updateUI();
}

}

```

```

private String[] txtLeyendas() {
    String[] l = new String[getTxt1()];
    for (int i = 0; i < l.length; i++) {
        try {
            if (txtLeyendas[i].getText().isEmpty()) {
                throw new Exception();
            }
            l[i] = txtLeyendas[i].getText();
        } catch (Exception e) {
            l[i] = "Sin titulo";
        }
    }
}

```



```

    }
}
return l;
}

```

```

public void printf() {
    g = new Grafico(getTitulo(), txtLeyendas());
    g.setValores(valoresDeText());
    JFrame ex = new JFrame();
    ex.setTitle("Grafico solicitado");
    ex.setSize(800, 600);
    ex.setVisible(true);
    ex.setResizable(false);
    ex.setLocationRelativeTo(null);
    ex.setDefaultCloseOperation(HIDE_ON_CLOSE);
    ex.add(g);
    p.updateUI();
}

```

```

private int[] valoresDeText() {
    int nBarras = getTxt1();
    int valores[] = new int[nBarras];
    for (int i = 0; i < nBarras; i++) {
        int val;
        try {
            val = Integer.parseInt(txt2[i].getText());
        } catch (Exception e) {
            val = 0;
        }
        valores[i] = val;
    }
}

```



```

    }

    return valores;
}

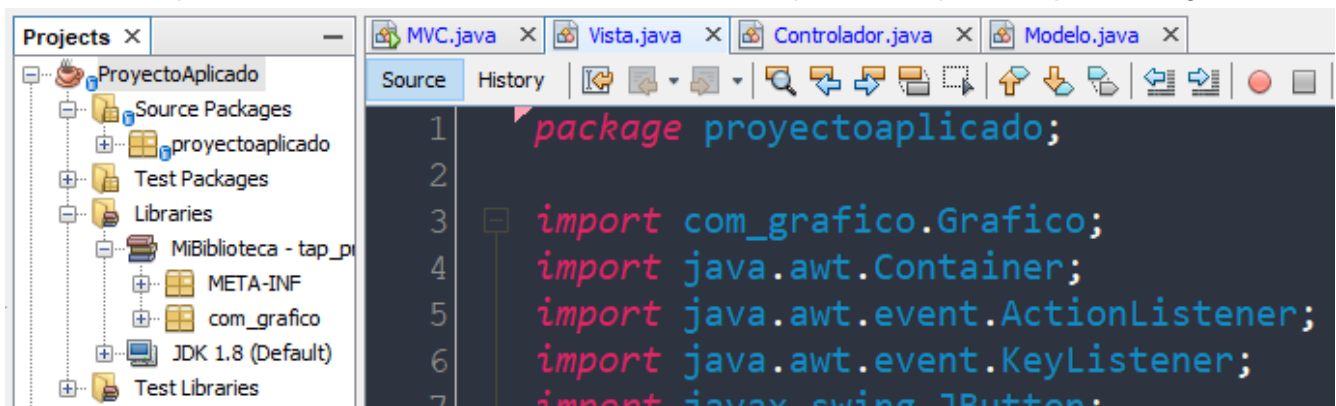
public void bloquearTxt(boolean b) {
    txt1.setEditable(b);
    titulo.setEditable(b);
}

public void conectarControlador(Controlador c) {
    txt1.addKeyListener((KeyListener) c);
    graficar.addActionListener((ActionListener) c);
}
}

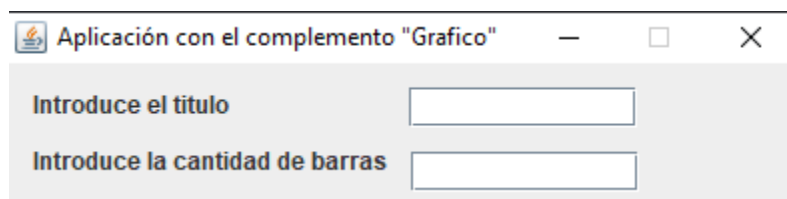
```

PROYECTO - PRUEBAS

El proyecto que utiliza la aplicación en su clase Vista, hace uso de la librería *MiBiblioteca*, creada durante esta práctica, entonces se hace lo solicitado en el reporte. Comprobado por la imagen.



Al ejecutar el programa, la primera vista que tenemos es la siguiente. Donde se solicita el título general que tendrá el gráfico y la cantidad de barras a dibujar; el programa está validado, pero hacer máximo 10 barras.



Aplicación con el complemento "Grafico"

Introduce el título

Introduce la cantidad de barras

Introduce los datos de las barras

	Valor	Leyenda
Datos Barra 1	<input type="text"/>	<input type="text"/>
Datos Barra 2	<input type="text"/>	<input type="text"/>
Datos Barra 3	<input type="text"/>	<input type="text"/>
Datos Barra 4	<input type="text"/>	<input type="text"/>
Datos Barra 5	<input type="text"/>	<input type="text"/>
Datos Barra 6	<input type="text"/>	<input type="text"/>

Introduciendo valores para hacer la prueba, se vería algo así. Observando que el tamaño de la ventana se ha incrementado.

Luego de rellenar los 2 primeros campos mostrados anteriormente, aparecen nuevos campos que solicitan el valor y leyenda de cada barra.

También aparece un botón "Graficar" que sirve para ejecutar el programa y hace la entrada en funcionamiento del componente "Grafico".

Introduce el título

Introduce la cantidad de barras

Introduce los datos de las barras

	Valor	Leyenda
--	-------	---------

Aquí se muestra un ejemplo de que pasaría si introducimos una cantidad de barras mayor al máximo que podemos graficar.

El programa no mostraría los demás campos.



Aplicación con el complemento "Grafico"

Introduce el título

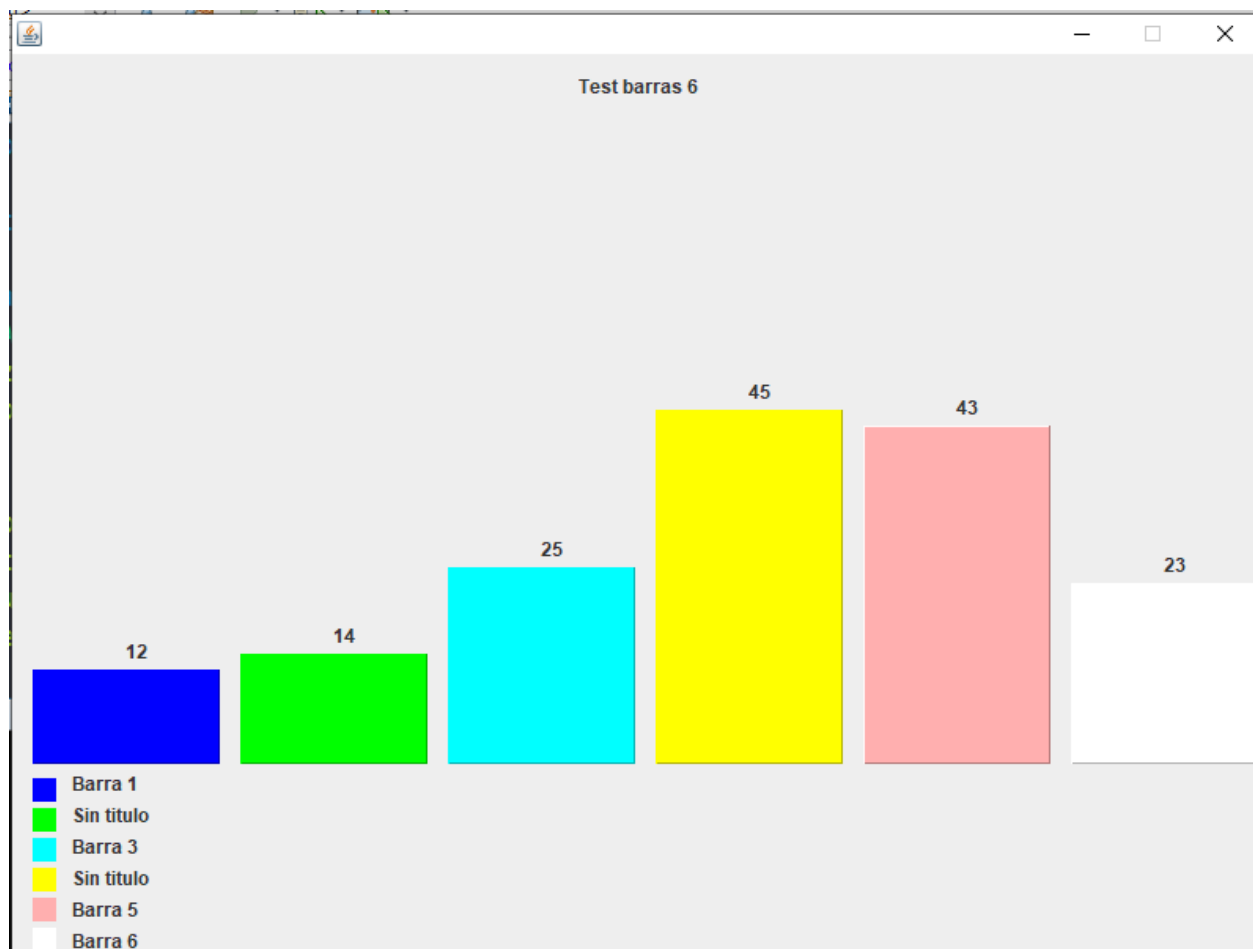
Introduce la cantidad de barras

Introduce los datos de las barras

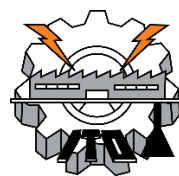
	Valor	Leyenda
Datos Barra 1	<input type="text" value="12"/>	<input type="text" value="Barra 1"/>
Datos Barra 2	<input type="text" value="14"/>	<input type="text"/>
Datos Barra 3	<input type="text" value="25"/>	<input type="text" value="Barra 3"/>
Datos Barra 4	<input type="text" value="45"/>	<input type="text"/>
Datos Barra 5	<input type="text" value="43"/>	<input type="text" value="Barra 5"/>
Datos Barra 6	<input type="text" value="23"/>	<input type="text" value="Barra 6"/>

Rellenamos los campos de forma aleatoria para muestra de esta aplicación.

Dejamos a propósito 2 campos para ver como funciona el programa sin rellenar el campo de leyenda de la barra.



Este es el resultado, se muestra otra ventana con el grafico solicitado.



Teniendo así 2 ventanas, la que mantiene los datos y la que muestra el grafico.

