



**TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA**

TÓPICOS AVANZADOS DE PROGRAMACIÓN

**PRACTICA 2. CREACIÓN DE UNA APLICACIÓN “GUI”
USANDO COMPONENTES DISPONIBLES POR EL
LENGUAJE.**

**PRESENTA ESTUDIANTE DE LA CARRERA INGENIERÍA EN SISTEMAS
COMPUTACIONALES:
ZARATE CARREÑO JOSÉ VALENTÍN**

DOCENTE: HERNANDEZ ABREGO ANAYANSI

GRUPO: 4SA

HORA: 9:00- 10:00

Oaxaca de Juárez, Oax, 09 de marzo de 2020.

COMPETENCIA A DESARROLLAR

Diseña e implementa componentes y librerías para lograr la reutilización de código.

INTRODUCCIÓN

Las aplicaciones tipo GUI usan invariablemente componentes de este tipo que facilitan enormemente su desarrollo. En las prácticas anteriores se han utilizado componentes GUI de los paquetes de java: swing y awt. Ahora corresponde ejercitar el proceso de su desarrollo, partiendo desde su diseño, implementación y la manera de generar con ellos bibliotecas para cumplir la propiedad de encapsulamiento. Aquí se propone crear un tipo de componente para un ambiente visual que además utiliza un poco manejo de gráficos con la intención de mostrar que un componente puede ser para cualquier ambiente.

CORRELACIÓN CON LOS TEMAS Y APLICACIÓN EN EL CONTEXTO.

Desde la primera práctica se han utilizados componentes de tipo GUI, así que los dos primeros temas de esta segunda competencia (unidad) se llegan a reforzar en esta práctica: 'definición conceptual de componentes, paquetes / bibliotecas' y 'uso de bibliotecas proporcionadas por el lenguaje'. Además, estos temas se llegan a integrar con el último tema que se trata de lleno en esta práctica: Creación de componentes (visuales y no visuales) definidos por el usuario. Los componentes utilizados y a desarrollar están dentro del contexto de la propuesta de Oracle –Java, utilizando el estándar ahí propuesto. El acceso a su funcionalidad y elementos internos de cada componente es mediante sus métodos get y set a nivel de programación.

MATERIAL Y EQUIPO NECESARIO

- Equipo de cómputo: Laptop o PC Software: Aquí se emplea el “IDE” NetBeans (el cual cuenta con el asistente para crear un *.jar), pero puedes utilizar cualquier otro que maneje java con un entorno de diseño de GUI- Java, versión del jdk 1.8.0

- . Asegurarse que dentro de la carpeta bin de java se encuentre la herramienta: jar.exe

METODOLOGÍA

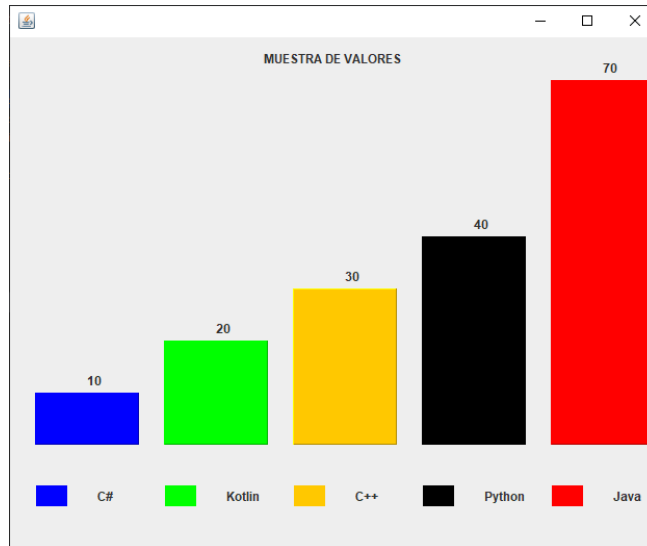
Se parte de la descripción de lo que se pretende realizar, presentado los pasos a seguir para el desarrollo de componentes y una descripción sobre el caso a diseñar e implementar, se expone un caso de un componente visual-gráfico.

Descripción

El proceso que aquí se sigue para el desarrollo de un componente java es:

1. Descripción de la funcionalidad, donde se especifica lo que se espera que realice el componente y se hace un esbozo de su diseño dependiendo del tipo que es (GUI o solo funcional)
2. Diseño de los elementos que lo conforman. Aquí se especifican sus interfaces de entrada y salida (métodos get y set)
3. Implementación. Es la creación de la clases o clases necesarias siguiendo las convenciones propuestas por Oracle.
4. Pruebas. Esto es un punto vital para garantizar que se cumple con la funcionalidad especificada
5. Generación del componente java *.jar e integración de este en alguna biblioteca para su utilización.

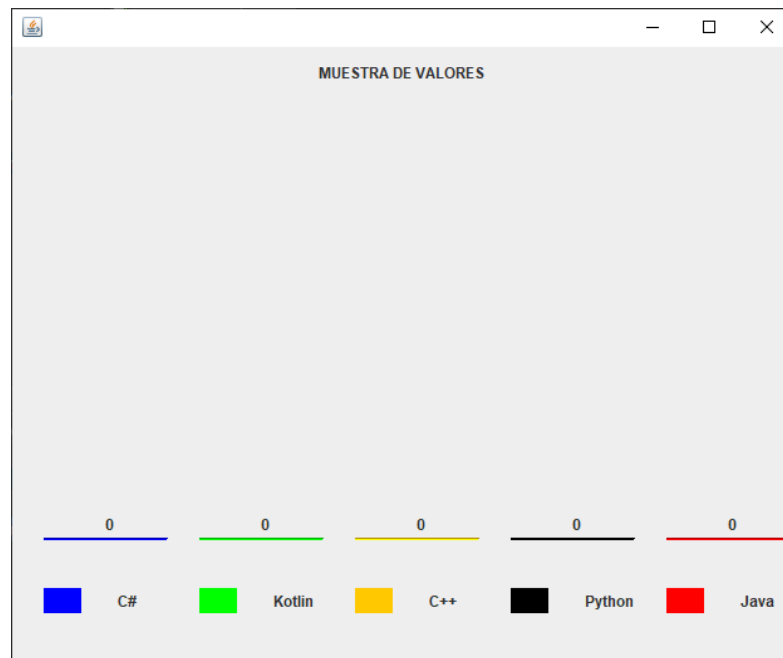
A continuación se describe el componente propuesto: Descripción de la funcionalidad del componente tipo GUI denominado "Grafico". Funcionalidad: Dibujar un gráfico de barras de colores que indiquen en su dimensión los valores que se le proporcionen y las leyendas que las identifiquen. El número de barras será de acuerdo a las leyendas de identificación que se proporcionen y cada vez que se cambie algunos de los valores estas deben reflejar en su dimensión el valor correspondiente. Abajo se muestra un ejemplo de cómo debe ser.



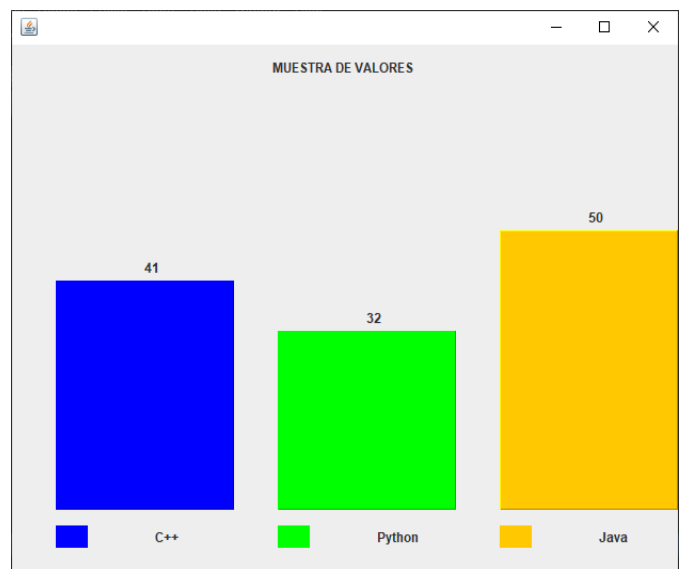
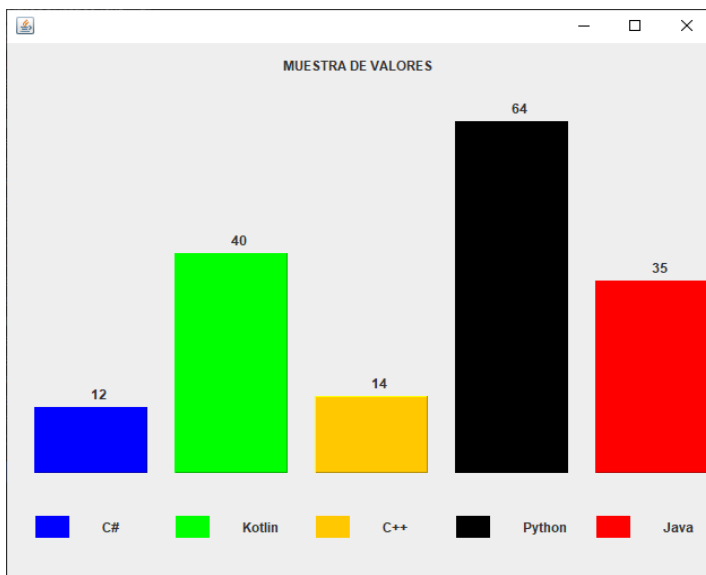
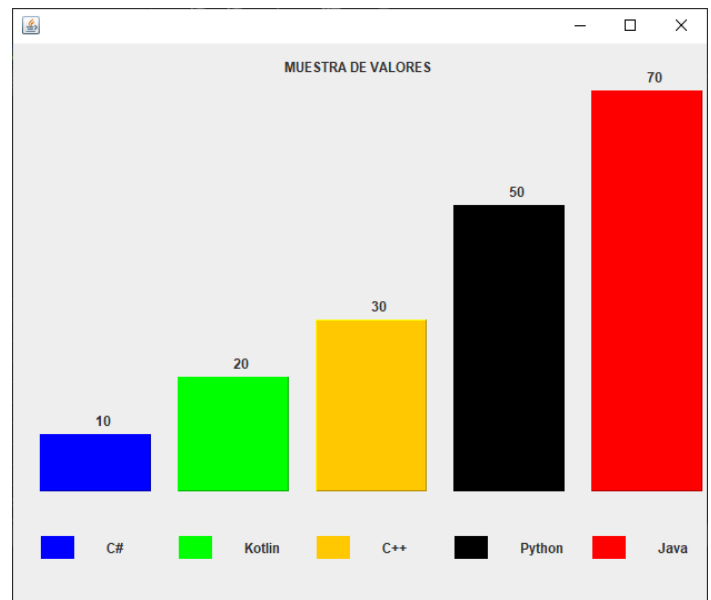
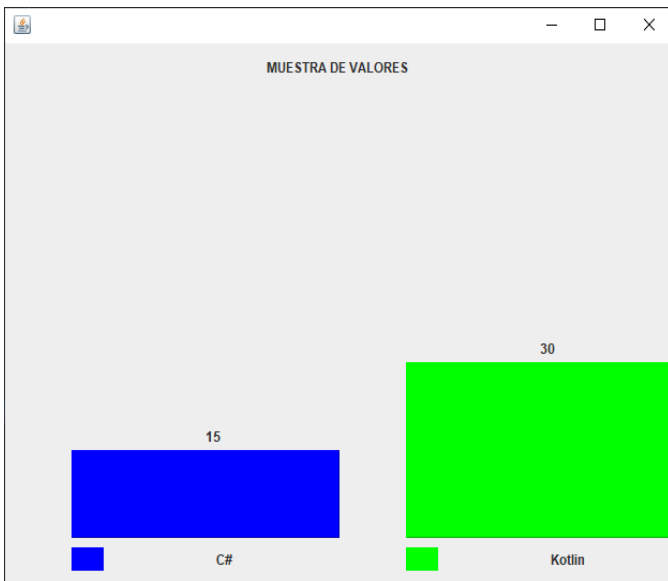
DESARROLLO

1. Dentro del IDE NetBeans, crea un proyecto Java tipo biblioteca(Library). Después dar clic en el botón Next y nombralo como tap_practica_04, en él crea un paquete fuente con el nombre 'com_grafico' y dentro de este último crea una clase, llámala Grafico.
 - a) Hereda la clase Grafico de JComponent, esto es para cumplir con la función principal sobre producir un gráfico de barras.
2. Declaración de los atributos, crea los atributos de la clase.
3. Creación de constructores
 - a) Crea un constructor sin parámetros con atributos con valores por omisión.
 - b) Crea un constructor que pase como valores el título del gráfico y leyendas, y tenga valores por omisión.
 - c) Se crea el método iniciarElementos.

- d) Crea el código indicado, que sirva para asignar color a las etiquetas que acompañan a las leyendas de las barras, cuyo identificador es "colorB" y se agregan al componente.
4. Creación de los métodos para asignar título, leyendas y valores
- a) Se crea un método para asignar título.
 - b) Se crea un método para asignar leyendas.
 - c) Se crea un método para ingresar valores a las gráficas.
5. Creación del método para crear el gráfico:
- a) Se crea el código para fijar la posición de los elementos que muestran el color de identificación y texto de las leyendas correspondientes a cada barra(lo cual está indicado como comentarios en el código de arriba).
6. Pruebas de la clase
- a) Crea una clase prueba que cuente con un JFrame que agregue a su contenedor un objeto tipo 'Grafico' y se le comuniquen valores del título y etiquetas sin valores. La gráfica debe mostrarse como la siguiente:



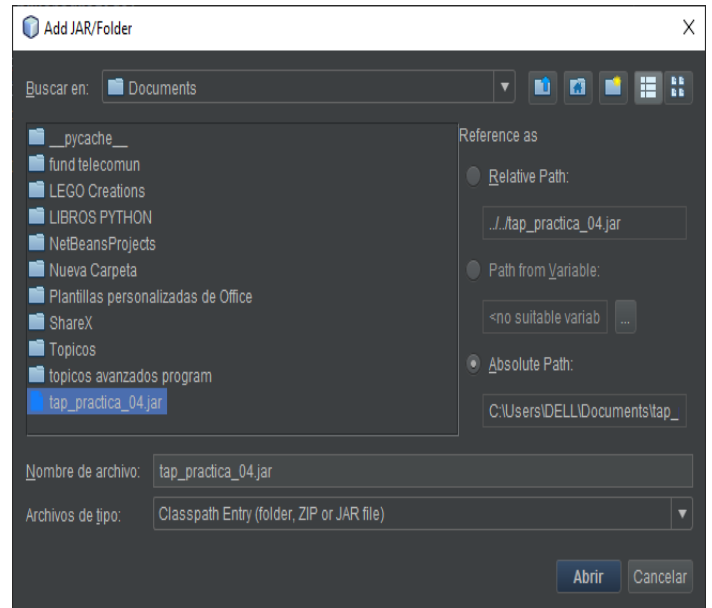
- b) Cambio de valores. Usa el método setValores() para asignarle valores a las barras, usa los valores mostrados en la descripción del componente para que al ejecutar de nuevo el programa se vea como en dicha muestra.
- c) Comprobar que se distribuyen en forma equidistante y sus leyendas debajo del gráfico. Varía el ancho de la ventana de prueba para comprobar que se distribuyen en forma equidistante
- d) Prueba de numero de barras, crea un gráfico con dos etiquetas con sus valores de barras y otro de cinco, deben verse como:



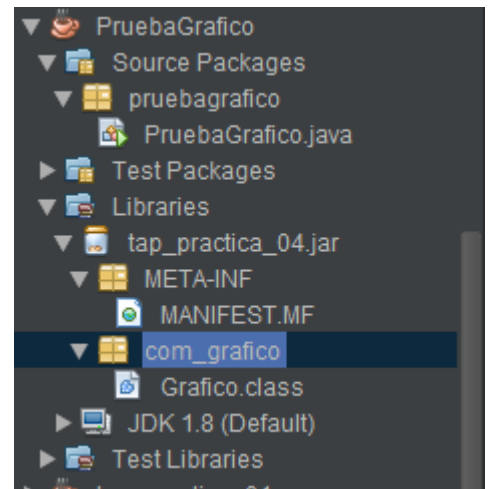
7. Creación como una biblioteca compartida dentro del 'IDE' de NetBeans

Aunque una biblioteca (Library) tiene como propósito ser una colección de componentes, sobre todo a nivel de un IDE, en esta práctica se hará solo con un componente pero posteriormente puedes agregarle más elementos.

- a) Dentro del menú principal del IDE de NetBeans, elige la opción Herramientas(Tools), dentro de ahí activa Biblioteca (Library) y da clic en el botón Nueva biblioteca (New Library), puedes llamarle MiBiblioteca como se muestra a continuación:
- b) Agregarle componentes a la biblioteca creada. En la ventana mostrada, da clic en el botón AddJar/Folder. Ubica la ruta donde se encuentra el componente como se muestra enseguida



- c) Da clic en el botón AddJar/Folder y después darle OK y listo ya se cuenta con una biblioteca.
- d) Ahora cada vez que requieras usarla en cualquier proyecto de aplicación la tienes disponible a nivel del IDE de NetBeans, puedes hacerlo en el proyecto prueba o en cualquiera otro.



CÓDIGO DE LA APLICACIÓN

```
1  package com_grafico;
2
3  import java.awt.Color;
4  import java.awt.Graphics;
5  import java.io.Serializable;
6  import javax.swing.JComponent;
7  import javax.swing.JLabel;
8
9  public class Grafico extends JComponent implements Serializable {
10
11     private int nfiguras; // numero de barras
12     private JLabel[] etis; // etiquetas para el valor a mostrar
13     private int[] vals = {10, 0, 0}; // valores de cada barra, se dan valores por omision
14     // Se consideran diez colores para diez valores
15     private static Color color[] = {Color.BLUE, Color.GREEN, Color.ORANGE, Color.black,
16                                     Color.RED, Color.YELLOW, Color.WHITE, Color.PINK, Color.MAGENTA};
17     private static JLabel leyendaBarra[]; // muestra el titulo de la leyenda
18     private static JLabel colorB[]; // identificacion del color de la barras
19     private JLabel titulo; // titulo
20     private String tTitulo; // Texto del titulo
21     private String tLeyenda[]; // titulos de las leyendas que idefican a cada barra
22
23     public Grafico() {
24         // Valores por omision
25         titulo = new JLabel("");
26         etis = new JLabel[0];
27         leyendaBarra = new JLabel[0];
28         tLeyenda = new String[0];
29     }
30 }
```

```
32     public Grafico(String encabezado, String tLeyenda[]) {
33         nfiguras = tLeyenda.length;
34         this.tLeyenda = tLeyenda; // texto de cada leyenda
35         leyendaBarra = new JLabel[tLeyenda.length];
36         tTitulo = encabezado;
37         setLeyendas(this.tLeyenda);
38         iniciarElementos(); // metodo para iniciar los valores
39     }
40
41     public void iniciarElementos() { // Creacion e inicio de los valores de cada barra
42         vals = new int[nfiguras];
43         for (int nv = 0; nv < nfiguras; nv++) { // inicia los valores
44             vals[nv] = 0;
45         }
46         // crea y agrega la etiqueta del titulo
47         titulo = new JLabel(tTitulo);
48         add(titulo); // // crea e inicia los demas arreglos
49         etis = new JLabel[nfiguras]; //
50         colorB = new JLabel[nfiguras];
51         for (int i = 0; i < nfiguras; i++) {
52             etis[i] = new JLabel("" + 0);
53             add(etis[i]);
54         } // identificacion de color de cada barra colorB[]
55         for (int x = 0; x < nfiguras; x++) {
56             colorB[x] = new JLabel();
57             colorB[x].setBackground(color[x]);
58             colorB[x].setForeground(color[x]);
59             colorB[x].setOpaque(true);
60             add(colorB[x]);
61         }
62     }
63 }
```



```

65     public void setTitulo(String encabezado) {
66         tTitulo = encabezado;
67     }
68
69     public void setLeyendas(String tLeyenda[]) {
70         nfiguras = tLeyenda.length;
71         this.tLeyenda = tLeyenda;
72         leyendaBarra = new JLabel[tLeyenda.length];
73         for (int e = 0; e < tLeyenda.length; e++) {
74             leyendaBarra[e] = new JLabel(tLeyenda[e]);
75         }
76         iniciarElementos();
77     }
78     //metodo para asignar la etiquetas y cambiar valores
79
80     public void setValores(int valores[]) {
81         vals = valores;
82         etis = new JLabel[vals.length];
83         for (int v = 0; v < vals.length; v++) {
84             etis[v] = new JLabel("" + valores[v]);
85             add(etis[v]);
86         }
87     }
88

```

```

89     @Override
90     public void paintComponent(Graphics f) {
91         int i;
92         int j = 1;
93         int ancho = getWidth();
94         int alto = getHeight();
95
96         titulo.setBounds((int) (getWidth() - f.getFontMetrics().stringWidth(titulo.getText()) / 2, 10,
97             titulo.getText().length() * 8, 20);
98         int separa = (int) (getWidth() / (nfiguras) * 0.20);
99         int anchoB = (int) (getWidth() / (nfiguras) - separa); //Ubica cada elemento en su posicion
100
101         //ubica cada uno de los elementos en su posicion
102         for (i = 0; i < nfiguras; i++) {
103
104             f.setColor(color[i]); //Establece el color de cada barra
105             etis[i].setBounds(separa + i * (anchoB + separa) + (anchoB / 2),
106                 (alto - 20 * colorB.length) - vals[i] * 5 - etis[i].getHeight() - 2, 30, 20);
107             f.fillRect(separa + i * (anchoB + separa), ((alto - 20 * colorB.length) - vals[i] * 5),
108                 anchoB, vals[i] * 5, true);
109
110             colorB[i].setBounds(i * (anchoB + separa) + (anchoB / 4), 430, 30, 20);
111             leyendaBarra[i].setBounds(separa + 10 + i * (anchoB + separa) + (anchoB / 2), 430, 80, 20);
112
113             add(leyendaBarra[i]);
114             j++;
115         }
116     }
117 }
118
119 }

```

Clase prueba:

```
2 package com_grafico;
3 import javax.swing.JFrame;
4
5 public class Prueba {
6     Run | Debug
7     public static void main(String[] args) {
8
9         JFrame f = new JFrame();
10        String title = "MUESTRA DE VALORES";
11        String Leyenda[] = {"C#", "Kotlin", "C++", "Python", "Java"};
12        Grafico graf = new Grafico(title, Leyenda);
13        int altura[]={10, 20, 30, 50,70};
14        graf.setValores(altura);
15        f.setVisible(true);
16        f.add(graf);
17
18
19        f.setSize(640, 530);
20        f.setResizable(true);
21        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22
23    }
24 }
25
```

Proyecto

importando la librería creada:

```
1 package pruebagrafico;
2
3 import java.util.Random;
4 import com_grafico.Grafico;
5 import javax.swing.JFrame;
6
7 public class PruebaGrafico {
8
9     public static void main(String[] args) {
10
11        JFrame f = new JFrame();
12
13        String title = "MUESTRA DE VALORES";
14        String Leyenda[] = {"C#", "Kotlin", "C++", "Python", "Java"};
15        Grafico graf = new Grafico(title, Leyenda);
16        Random rnd = new Random();
17
18        int altura[] = new int[Leyenda.length];
19
20        for (int i = 0; i < Leyenda.length; i++) {
21            int valorDado = rnd.nextInt(70) + 1;
22            altura[i] = valorDado;
23            graf.setValores(altura);
24            f.setVisible(true);
25            f.add(graf);
26        }
27        f.setSize(640, 530);
28        f.setResizable(true);
29        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30    }
31 }
```

Pruebas haciendo uso de la librería creada:

