Figuras geométricas en Java

Miguel Ángel Bañol Cifuentes

Politécnico Colombiano Jaime Isaza Cadavid

Medellín, Antioquia

2022

Notas del autor

Estudiante de 4to semestre de Ingeniería Informática

**Contenido**

[**INTRODUCCIÓN** 3](#_Toc96534775)

[**JUSTIFICACIÓN** 4](#_Toc96534776)

[**OBJETIVOS** 5](#_Toc96534777)

[**ALCANCE** 6](#_Toc96534778)

[**MARCO TEÓRICO** 7](#_Toc96534779)

[**CÓDIGO DE DESARROLLO** 8](#_Toc96534780)

[**CIBERGRAFÍA** 14](#_Toc96534781)

# **INTRODUCCIÓN**

En la práctica de la programación en Java, se crean diversas oportunidades para desempeñar diferentes tareas que la sociedad requiere. En este ente se dará a conocer e identificar por medio de un algoritmo desarrollado en el lenguaje Java, un pequeño esquema de figuras geométricas básicas en el que el usuario aprecia un tamaño y una dimensión de la misma, teniendo la posibilidad de elegir estas propiedades además de su color. Para este trabajo se ha implementado una serie de librerías y herramientas gráficas que Java dispone en su infinita lista de paquetes. Se ha trabajado con conceptos de POO (Programación Orientada a Objetos) para la ejecución y realización del proyecto, tales como herencia, encapsulamiento y sobrecarga de métodos, entre otros.

Este producto sirve de enseñanza para los niños que están empezando a experimentar con formas y colores que perciben de la vida real, o bien que están viendo geometría muy básica en la escuela. Aquí es donde ellos interactúan con colores, formas y tamaños, aprenden a familiarizarse con el entorno tecnológico y así poder entender que es una ayuda para el mundo.

# **JUSTIFICACIÓN**

La implementación de los conceptos POO en este trabajo lleva aún más a entender al lenguaje Java, en cuanto a su paradigma, si bien, Java va al grano, es un lenguaje orientado a objetos y en efecto esto se trata en el proyecto. Se puede asimilar con las mismas clases construidas dentro de él. Anteriormente se hablaba de figuras geométricas, pues bien, son considerados como objetos en el mundo físico, y esto es lo que hace Java, transforma de manera irreal, o más bien virtual estos “objetos” del mundo en el que vivimos; así se entiende el paradigma, orientado a objetos.

De acuerdo a lo anterior, se aborda el manejo de librerías gráficas de Java ya que son relevantes y pertinentes para el contenido de la materia Lenguajes de Programación 1 en las cuales se ven reflejadas en interfaces gráficas. Entorno a la POO es indispensable aplicar interfaces, clases, métodos, objetos, paquetes, etc. Debido a que es la “plantilla” donde se dará solución a un problema, a un algoritmo, a un cliente o a una empresa. Esta es la importancia que se le da al manejo de estas librerías dentro de este trabajo.

# **OBJETIVOS**

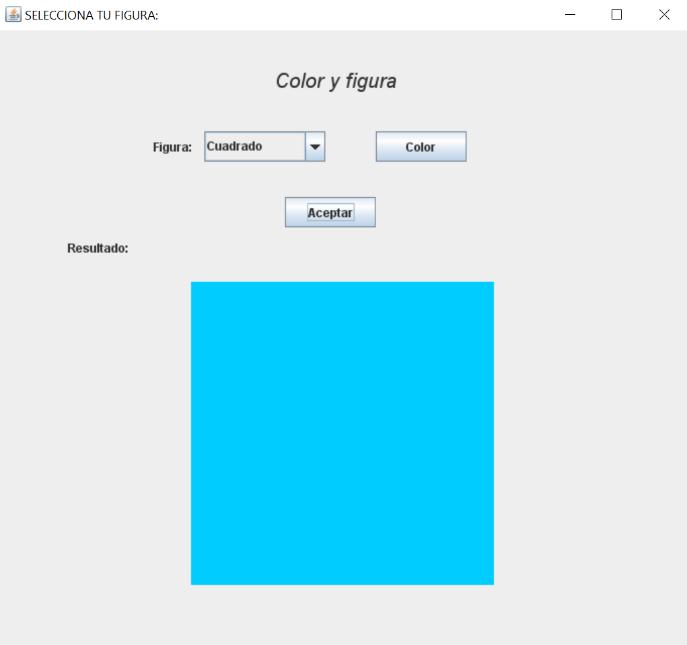
* Reconocer las librerías gráficas para la realización de figuras geométricas.
* Identificar algunas clases las librerías gráficas que tiene Java, como la clase Graphics y sus respectivos métodos Paint() y Repaint().
* Manipular los conceptos de POO (herencia, polimorfismo, encapsulamiento, sobrecarga de métodos, etc.).
* Manejar errores y excepciones de clases y objetos.
* Implementar interfaces gráficas como awt y swing para la presentación de ventanas de la navegabilidad.

# **ALCANCE**

Se busca exponer de manera detallada las figuras geométricas básicas (Cuadrado, Círculo, Triángulo, Rectángulo) con ayuda de las librerías gráficas de Java, implementando sus métodos respectivos para rellenar este tipo de figuras, de manera que el usuario elija la forma el color y el tamaño de estas. De este mismo modo es posible afianzar conocimientos sobre interfaces gráficas, interfaz de usuario y navegabilidad para una próxima tarea, o bien, para dar solución a un problema, a una empresa o cliente en específico como lo mencionamos anteriormente.

# **MARCO TEÓRICO**

Si bien para la elaboración de la obra, se requieren de muchos recursos de Java que dan base para la objetividad que tiene el trabajo. Se presentan muchos individuos y temas que fueron elaborados con su respectiva técnica para el buen desarrollo. Puede apreciarse la ejecución de elementos como la paleta de colores, gracias a ella se puede elegir un color muy específico, con propiedades de valores hexadecimales, saturación, opacidad, matiz, entre otras. En la aplicación de esta paleta se investigó la creación de ella, el cómo adecuarla a la necesidad del problema, ya que se necesitaba poner el color en las figuras y arrojará este resultado:



Como se puede apreciar está la gráfica de la figura deseada con un color ya seleccionado, pero detrás de esto, implica la inserción de los métodos que grafican el cuadrado, allí se trabaja una especie de lienzo para la elaboración de la figura que se está viendo en pantalla.

La acción de escucha de los botones como por ejemplo “color” tiene el selector de la paleta de colores, el cual fue minuciosamente elaborado para no cometer errores futuros, los demás botones generalmente llevan a otra ventana lo que implica que hayan interfaces gráficas de las librerías swing y awt.

# **CÓDIGO DE DESARROLLO**

Clase Inicio (método main)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | **package** proyecto;  **public** **class** **Inicio** {    **public** **static** **void** **main**(String[] args) {    Ventana\_bienvenida nueva = **new** Ventana\_bienvenida();  nueva.setVisible(**true**);    }  } |

Clase Ventana Bienvenida (interfaz gráfica)

**package** proyecto;

**import** **java.awt.Color**;

**import** **java.awt.Font**;

**import** **java.awt.event.ActionEvent**;

**import** **java.awt.event.ActionListener**;

**import** **javax.swing.JButton**;

**import** **javax.swing.JFrame**;

**import** **javax.swing.JLabel**;

**import** **javax.swing.JOptionPane**;

**import** **javax.swing.JPanel**;

**import** **javax.swing.SwingConstants**;

**public** **class** **Ventana\_bienvenida** **extends** JFrame {

//Defino atributos gráficos que contendrá la ventana

**private** JLabel lblbienvenido;

**private** JButton btncontinuar, btnsalir;

**private** JPanel pnlbienvenida;

**public** **Ventana\_bienvenida**() {

//Propiedades de la ventana

**this**.setSize(**600**, **500**);

**this**.setTitle("BIENVENIDA");

**this**.setLocationRelativeTo(**null**);

**this**.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

//Propiedades del label "Bienvenido"

lblbienvenido = **new** JLabel("Bienvenido a figuras geométricas");

lblbienvenido.setBounds(**50**, **30**, **500**, **40**);

lblbienvenido.setHorizontalAlignment(SwingConstants.CENTER);

lblbienvenido.setHorizontalTextPosition(SwingConstants.CENTER);

lblbienvenido.setFont(**new** Font("arial", **2**, **30**));

lblbienvenido.setOpaque(**false**);

//Propiedades de los botones

btncontinuar = **new** JButton("Continuar");

btncontinuar.setBounds(**120**, **300**, **90**, **30**);

ActionListener oyente\_continuar = (ActionEvent ae) -> {

**this**.setVisible(**false**);

Ventana\_seleccionar\_figura nueva = **new** Ventana\_seleccionar\_figura();

nueva.setVisible(**true**);

};

btncontinuar.addActionListener(oyente\_continuar);

//Creamos el botón salir

btnsalir = **new** JButton("Salir");

btnsalir.setBounds(**340**, **300**, **90**, **30**);

ActionListener oyente\_salir = (ActionEvent ae) -> {

**int** respuesta;

respuesta = JOptionPane.showConfirmDialog(**this**, "¿Esta seguro de que quiere salir?", "Confirmacion",

JOptionPane.YES\_NO\_OPTION);

**if** (respuesta == JOptionPane.YES\_OPTION) {

System.exit(**0**);

}

};

btnsalir.addActionListener(oyente\_salir);

pnlbienvenida = **new** JPanel();

pnlbienvenida.setLayout(**null**);

pnlbienvenida.add(lblbienvenido);

pnlbienvenida.add(btncontinuar);

pnlbienvenida.add(btnsalir);

**this**.getContentPane().add(pnlbienvenida);

}

}

Clase Seleccionar Figura (gráficos)

**package** proyecto;

**import** **java.awt.Font**;

**import** **java.awt.event.ActionEvent**;

**import** **java.awt.event.ActionListener**;

**import** **java.awt.Color**;

**import** **java.awt.Graphics**;

**import** **javax.swing.JButton**;

**import** **javax.swing.JColorChooser**;

**import** **javax.swing.JComboBox**;

**import** **javax.swing.JFrame**;

**import** **javax.swing.JLabel**;

**import** **javax.swing.JOptionPane**;

**import** **javax.swing.JPanel**;

**import** **javax.swing.SwingConstants**;

**public** **class** **Ventana\_seleccionar\_figura** **extends** JFrame {

**private** JLabel lblfigura, lblseleccion, lblresultado;

**private** JButton btncolor, btnaceptar;

**private** JComboBox combofigura;

**private** JPanel pnlseleccion;

**private** Color color;

**int** x, y;

**public** **Ventana\_seleccionar\_figura**() {

//Pripiedades de la ventana

**this**.setSize(**700**, **800**);

**this**.setTitle("SELECCIONA TU FIGURA: ");

**this**.setLocationRelativeTo(**null**);

**this**.setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

//Propiedades de los labels de ventana.

lblseleccion = **new** JLabel("Color y figura");

lblseleccion.setBounds(**135**, **30**, **400**, **40**);

lblseleccion.setHorizontalAlignment(SwingConstants.CENTER);

lblseleccion.setHorizontalTextPosition(SwingConstants.CENTER);

lblseleccion.setFont(**new** Font("arial", **2**, **20**));

lblseleccion.setOpaque(**false**);

//Propiedades del label Figura

lblfigura = **new** JLabel("Figura: ");

lblfigura.setBounds(**155**, **100**, **70**, **30**);

//Propiedades del label Resultado

lblresultado = **new** JLabel("Resultado: ");

lblresultado.setBounds(**70**, **200**, **70**, **30**);

//Realizamos el combobox de las figuras...

String[] figuras = {"Circulo", "Cuadrado", "Rectangulo", "Triangulo"};

combofigura = **new** JComboBox(figuras);

combofigura.setBounds(**205**, **100**, **120**, **30**);

//Propiedades del boton color

btncolor = **new** JButton("Color");

btncolor.setBounds(**375**, **100**, **90**, **30**);

ActionListener oyente\_color = (ActionEvent ae) -> {

//Paleta de colores

color = JColorChooser.showDialog(**null**, "Selecciona uno", color);

**if** (color == **null**) {

color = (java.awt.Color.BLACK);

}

};

btncolor.addActionListener(oyente\_color);

//Propiedades del botón aceptar

btnaceptar = **new** JButton("Aceptar");

btnaceptar.setBounds(**285**, **165**, **90**, **30**);

ActionListener oyente\_aceptar = (ActionEvent ae) -> {

// Aqui ira el resultado obtenido de la figura y el color

// System.out.println(combofigura.getSelectedItem().toString() + color);

// Aqui validamos que figura seleccionó el usuario para empezar a graficar.

//Graphics g = null;

x = Integer.parseInt(JOptionPane.showInputDialog("Ingrese el ancho x:"));

y = Integer.parseInt(JOptionPane.showInputDialog("Ingrese el alto y:"));

repaint();

};

btnaceptar.addActionListener(oyente\_aceptar);

//Instanciamos el panel para colocar todos estos elementos dentro de el

pnlseleccion = **new** JPanel();

pnlseleccion.setLayout(**null**);

pnlseleccion.add(lblfigura);

pnlseleccion.add(lblseleccion);

pnlseleccion.add(lblresultado);

pnlseleccion.add(combofigura);

pnlseleccion.add(btncolor);

pnlseleccion.add(btnaceptar);

**this**.getContentPane().add(pnlseleccion);

} // constructor

**@Override**

**public** **void** **paint**(Graphics g) {

**super**.paint(g);

**if** (combofigura.getSelectedItem().toString().equals("Circulo")) {

pintarComponenteCirculo(g, color, x, y);

}

**if** (combofigura.getSelectedItem().toString().equals("Cuadrado")) {

pintarComponenteRectangulo(g, color, x, y);

}

**if** (combofigura.getSelectedItem().toString().equals("Rectangulo")) {

pintarComponenteRectangulo(g, color, x, y);

}

**if** (combofigura.getSelectedItem().toString().equals("Triangulo")) {

pintarComponenteTriangulo(g, color);

}

// Aqui ira el if del triangulo...

}

**public** **void** **pintarComponenteRectangulo**(Graphics g, Color c, **int** x, **int** y) {

//super.paintComponents(g);

g.setColor(c);

g.fillRect(**200**, **280**, x, y);

}

**public** **void** **pintarComponenteCirculo**(Graphics g, Color c, **int** x, **int** y) {

//super.paintComponents(g);

g.setColor(c);

g.fillOval(**200**, **280**, x, y);

}

**public** **void** **pintarComponenteTriangulo**(Graphics g, Color c) {

**int** vecx[] = {**320**, **180**, **250**};

**int** vecy[] = {**300**, **300**, **200**};

g.setColor(c);

g.fillPolygon(vecx, vecy, **3**);

}

}

# **CIBERGRAFÍA**

khaterine Rodriguez. (05 de Octubre de 2012). Swing y Awt en Java [video]. Youtube. Obtenido de https://www.youtube.com/watch?v=0BKgDuuMBGs

Oracle. (03 de 09 de 2000). Class JColorChooser. Web. Obtenido de https://docs.oracle.com/javase/7/docs/api/