

Unidad 1

Elementos de Interfaces Gráficas

Computación Gráfica

Ejemplos de uso de primitivas de dibujo

Proyecto Símbolo

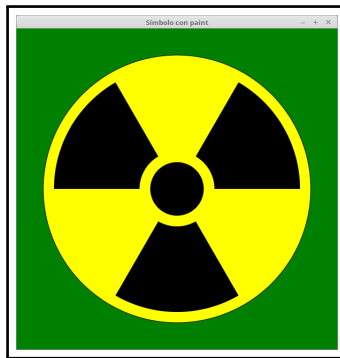


Figura 1.1: Señal de riesgo por radiación ionizante dibujado en el contexto gráfico.

Código 1.1: Aplicación para dibujar el símbolo de riesgo radiactivo.

```
public class Main extends Application {  
  
    private static final int WIDHT = 600;  
    private static final int HEIGHT = 600;  
  
    @Override  
    public void start(Stage primaryStage) {  
        Parent root = new SymbolPane(WIDHT, HEIGHT);  
        primaryStage.setTitle("Símbolo_con_paint");  
        primaryStage.setScene(new Scene(root, WIDHT, HEIGHT));  
        primaryStage.show();  
    }  
  
    public static void main(String[] args) { launch(args); }  
}
```

Método translate

Mueve el centro del sistema de coordenadas al punto especificado.

- void translate(double x, double y)

Código 1.2: Clase RadioactiveSymbol.

```
public class RadioactiveSymbol extends Pane {

    private final Canvas canvas;

    public RadioactiveSymbol(double width, double height) {
        canvas = new Canvas(width, height);
        init();
        draw();
    }

    private void init() {
        getChildren().add(canvas);
        setBackground(new Background(
            new BackgroundFill(
                Color.GREEN, CornerRadii.EMPTY, Insets.EMPTY)));
    }

    public final void draw() {
        GraphicsContext gc = canvas.getGraphicsContext2D();

        // Obtener las dimensiones del panel
        double width = canvas.getWidth();
        double height = canvas.getHeight();

        // Mover el origen al centro del panel
        gc.translate(width / 2, height / 2);

        int exteriorR = 250; //Radio del círculo externo (círculo amarillo)
        gc.setFill(Color.YELLOW);
        gc.fillOval(-exteriorR, -exteriorR, 2 * exteriorR, 2 * exteriorR);
        gc.setStroke(Color.BLACK);
        gc.strokeOval(-exteriorR, -exteriorR, 2 * exteriorR, 2 * exteriorR);

        int arcR = 230; //Radio de los arcos para las hélices.
        gc.setFill(Color.BLACK);
        gc.fillArc(-arcR, -arcR, arcR * 2, arcR * 2, 0, 60, ArcType.ROUND);
        gc.fillArc(-arcR, -arcR, arcR * 2, arcR * 2, 120, 60, ArcType.ROUND);
        gc.fillArc(-arcR, -arcR, arcR * 2, arcR * 2, 240, 60, ArcType.ROUND);

        int interiorR = 70; //Radio del círculo interior (círculo amarillo)
        gc.setFill(Color.YELLOW);
        gc.fillOval(-interiorR, -interiorR, interiorR * 2, interiorR * 2);

        int centerR = 50; //Radio del círculo interior (círculo negro)
        gc.setFill(Color.BLACK);
        gc.fillOval(-centerR, -centerR, centerR * 2, centerR * 2);
    }
}
```

Proyecto Dibujo

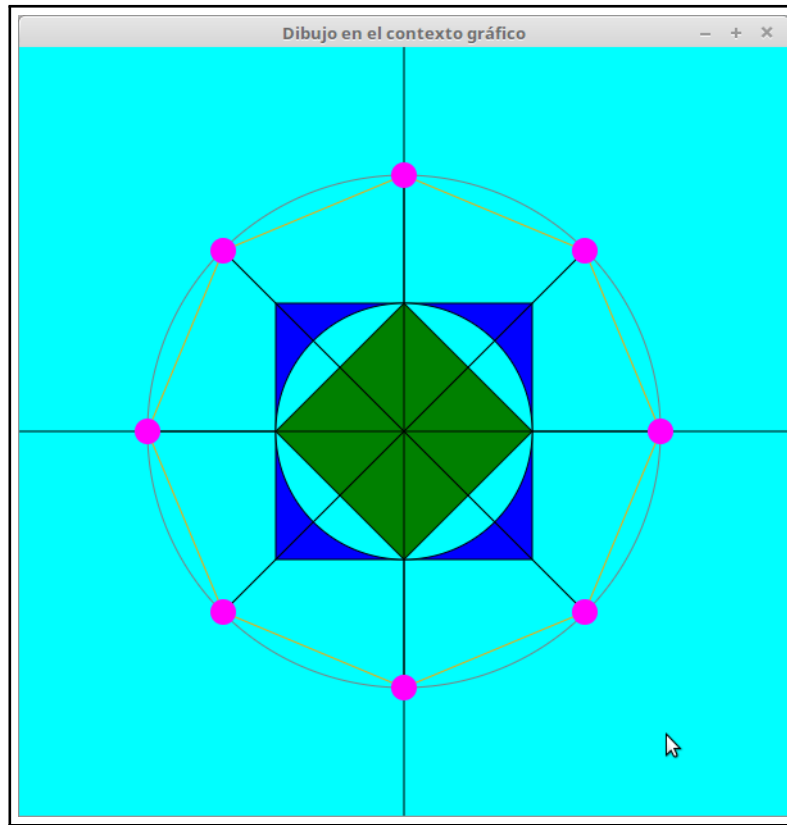


Figura 1.2: Dibujo de figuras en el contexto gráfico.

Código 1.3: Aplicación para dibujar figuras en el contexto gráfico.

```
public class Main extends Application {  
  
    private static final int WIDHT = 600;  
    private static final int HEIGHT = 600;  
  
    @Override  
    public void start(Stage primaryStage) {  
        Parent root = new DrawPane(WIDHT, HEIGHT);  
        primaryStage.setTitle("Dibujo_en_el_contexto_gráfico");  
        primaryStage.setScene(new Scene(root, WIDHT, HEIGHT));  
        primaryStage.show();  
    }  
  
    public static void main(String[] args) { launch(args); }  
}
```

Código 1.4: Clase DrawPane (Parte 1).

```
public class DrawPane extends Pane {

    private final Canvas canvas;

    public DrawPane(double width, double height) {
        canvas = new Canvas(width, height);
        init();
        draw();
    }

    private void init() {
        getChildren().add(canvas);
        setBackground(new Background(
            new BackgroundFill(
                Color.CYAN, CornerRadii.EMPTY, Insets.EMPTY)));
    }

    private void draw() {
        GraphicsContext gc = canvas.getGraphicsContext2D();

        // Obtener las dimensiones del panel
        double width = canvas.getWidth();
        double height = canvas.getHeight();

        // Mover el origen al centro
        gc.translate(width / 2, height / 2);

        // Dibujar los ejes de coordendas
        gc.strokeLine(-width / 2, 0, width / 2, 0);
        gc.strokeLine(0, -height / 2, 0, height / 2);

        // Cuadrado con centro en el origen
        int sidelenght = 200;
        gc.setFill(Color.BLUE);
        gc.fillRect(-sidelenght / 2, -sidelenght / 2, sidelenght, sidelenght);
        gc.setStroke(Color.BLACK);
        gc.strokeRect(-sidelenght / 2, -sidelenght / 2, sidelenght, sidelenght);

        // Círculo con centro en el origen
        gc.setFill(Color.CYAN);
        gc.fillOval(-sidelenght / 2, -sidelenght / 2, sidelenght, sidelenght);
        gc.strokeOval(-sidelenght / 2, -sidelenght / 2, sidelenght, sidelenght);

        // Rombo inscrito en el cuadrado
        double[] pX = {0, -sidelenght / 2, 0, sidelenght / 2};
        double[] pY = {-sidelenght / 2, 0, sidelenght / 2, 0};
        gc.setFill(Color.GREEN);
        gc.fillPolygon(pX, pY, pX.length);
        gc.strokePolygon(pX, pY, pX.length);

        // Círculo con centro en el origen
        int radius = 200;
        gc.setStroke(Color.GRAY);
        gc.strokeOval(-radius, -radius, 2 * radius, 2 * radius);

        // Dibujar rayos del centro a ocho puntos en el círculo
        gc.setStroke(Color.BLACK);
        for (int i = 0; i < 8; i++) {
            double rad = Math.toRadians(45 * i);
            int x = (int) (radius * Math.cos(rad));
            int y = (int) (radius * Math.sin(rad));
            gc.strokeLine(0, 0, x, y);
        }
        ...
    }
}
```

Código 1.5: Clase DrawPane (Parte 2).

```
...

// Dibujar líneas que unan los rayos
gc.setStroke(Color.ORANGE);
int ax = radius;
int ay = 0;
for (int i = 0; i <= 8; i++) {
    double rad = Math.toRadians(45 * i);
    int x = (int) (radius * Math.cos(rad));
    int y = (int) (radius * Math.sin(rad));
    gc.strokeLine(ax, ay, x, y);
    ax = x;
    ay = y;
}

// Dibujar 8 puntos al final de cada rayo
gc.setFill(Color.MAGENTA);
for (int i = 0; i < 8; i++) {
    double rad = Math.toRadians(45 * i);
    int x = (int) (radius * Math.cos(rad));
    int y = (int) (radius * Math.sin(rad));
    gc.fillOval(x - 10, y - 10, 20, 20);
}
}
```