



**Instituto Politécnico Nacional**  
**Escuela Superior de Cómputo**  
**Departamento de Ciencias e Ingeniería**  
**de la Computación**



## **Compiladores**

### **Práctica 2: Uso de Java para modo gráfico**

#### **Opción: Dibujo de círculos, líneas y rectángulos.**

**Grupo:** 5CM2

**Alumno:**

Sandoval Hernández Eduardo

**Profesor:**

Tecla Parra Roberto

**Fecha de entrega:**

22 de abril de 2022

## **Introducción**

Una interfaz gráfica de usuario, conocida también como GUI (“Graphical User Interface” por sus siglas en inglés) es un programa que actúa como interfaz de usuario, para ello utiliza un conjunto de imágenes y objetos gráficos que representan la información y acciones disponibles en la interfaz. El principal objetivo de estas interfaces gráficas es proporcionar un entorno visual sencillo para facilitar la comunicación entre el usuario y el sistema [1].

El presente trabajo tiene como finalidad reportar el desarrollo de la práctica 2 relacionada al entorno gráfico que provee el lenguaje de programación Java, la opción elegida fue el Dibujo de círculos, líneas y rectángulos. En esta práctica se debe hacer uso del código en la carpeta grafibasico para realizar el dibujo de un coche usando 2 círculos y 2 rectángulos, el dibujo de una casa usando 2 rectángulos y dos líneas inclinadas a  $45^\circ$  y  $225^\circ$ , y finalmente el dibujo de una persona con 1 círculo, 1 línea horizontal, 1 línea vertical y 2 líneas inclinadas a  $45^\circ$  y  $225^\circ$ .

## Desarrollo

Para la realización de esta práctica se utilizó el código presente en la carpeta grafibasic, en ella se encontraron todos los archivos necesarios para la ejecución de la interfaz gráfica que permite el realizar dibujos a partir de código, el archivo forma.y incluye la implementación en Yacc para la gramática, dicha gramática contiene los siguientes símbolos terminales: NUMBER, LINE, CIRCULO, RECTANGULO, COLOR, PRINT. En cuanto a las producciones se tienen las siguientes:

La primera producción permite el ingreso continuo de instrucciones, es decir, se podrán procesar varias instrucciones conforme se van ingresando:

```
/*
list => *vacio*
list => | list ';'
list => | list inst ';'
*/
list:
| list ';'
| list inst ';' {
    maq.code("print");
    maq.code("STOP");
    return 1 ;
}
;
```

La siguiente y última producción es la encargada de procesar las instrucciones para las figuras, el color de estas y sus parámetros (coordenadas y tamaño):

```

/*
inst => NUMBER
inst => RECTANGULO NUMBER NUMBER NUMBER NUMBER [X,Y,Ancho,Alto]
inst => LINE NUMBER NUMBER NUMBER NUMBER [X1,Y1,X2,Y2]
inst => CIRCULO NUMBER NUMBER NUMBER [Radio,X,Y]
inst => COLOR NUMBER [0,1,2] <= RGB: (0,Rojo) (1,Verde) (2,Azul)
*/

inst: NUMBER {
  ((Algo)$.obj).inst = maq.code("constpush");
  maq.code(((Algo)$1.obj).simb);
}
| RECTANGULO NUMBER NUMBER NUMBER NUMBER {
  //X
  maq.code("constpush");
  maq.code(((Algo)$2.obj).simb);
  //Y
  maq.code("constpush");
  maq.code(((Algo)$3.obj).simb);
  //Ancho
  maq.code("constpush");
  maq.code(((Algo)$4.obj).simb);
  //Alto
  maq.code("constpush");
  maq.code(((Algo)$5.obj).simb);
  maq.code("rectangulo");
}
| LINE NUMBER NUMBER NUMBER NUMBER {
  //X1
  maq.code("constpush");
  maq.code(((Algo)$2.obj).simb);
  //Y1
  maq.code("constpush");
  maq.code(((Algo)$3.obj).simb);
  //X2
  maq.code("constpush");
  maq.code(((Algo)$4.obj).simb);
  //Y2
  maq.code("constpush");
  maq.code(((Algo)$5.obj).simb);
  maq.code("line");
}
| CIRCULO NUMBER NUMBER NUMBER {
  //Radio
  maq.code("constpush");
  maq.code(((Algo)$2.obj).simb);
  //X
  maq.code("constpush");
  maq.code(((Algo)$3.obj).simb);
  //Y
  maq.code("constpush");
  maq.code(((Algo)$4.obj).simb);
  maq.code("circulo");
}
| COLOR NUMBER {
  //NUMBER
  maq.code("constpush");
  maq.code(((Algo)$2.obj).simb);
  maq.code("color");
}
;

```

El orden de los parámetros para las figuras fue el indicado por las instrucciones de la práctica, quedando como se muestra a continuación:

Círculo:

círculo	NUMBER	NUMBER	NUMBER
	Radio	X	Y

Rectángulo:

rectángulo	NUMBER	NUMBER	NUMBER	NUMBER
	X	Y	Ancho	Alto

Línea:

line	NUMBER	NUMBER	NUMBER	NUMBER
	X1	Y1	X2	Y2

En el código en Maquina.java se realizaron modificaciones para que pudiera realizar el dibujo de las figuras con los parámetros dados, quedando de la siguiente forma:

```

void line(){
    double X1, Y1, X2, Y2;
    //Y1
    Y1 = ((Double)pila.pop()).doubleValue();
    //X1
    X1 = ((Double)pila.pop()).doubleValue();
    //Y2
    Y2 = ((Double)pila.pop()).doubleValue();
    //X2
    X2 = ((Double)pila.pop()).doubleValue();

    if(g!=null){
        ( new Linea((int)X1, (int)Y1, (int)X2, (int)Y2) ).dibuja(
g);
    }
}

void circulo(){
    double radio, X, Y;
    //X
    X = ((Double)pila.pop()).doubleValue();
    //Radio
    radio = ((Double)pila.pop()).doubleValue();
    //Y
    Y = ((Double)pila.pop()).doubleValue();
    if(g!=null){
        ( new Circulo((int)radio, (int)X, (int)Y) ).dibuja(g);
    }
}

void rectangulo(){
    double X, Y, ancho, alto;
    //Alto
    alto = ((Double)pila.pop()).doubleValue();
    //Ancho
    ancho = ((Double)pila.pop()).doubleValue();
    //Y
    Y = ((Double)pila.pop()).doubleValue();
    //X
    X = ((Double)pila.pop()).doubleValue();
    if(g!=null){
        ( new Rectangulo((int)X, (int)Y, (int)ancho, (int)alto
) ).dibuja(g);
    }
}

```

Para la realización de los dibujos solicitados se utilizaron los siguientes parámetros:

```
//Carro
color 0 ;
rectangulo 200 200 200 50 ;
color 1 ;
rectangulo 250 150 100 50 ;
color 2 ;
circulo 50 220 225 ;
circulo 50 330 225 ;

//Casa
color 2 ;
line 500 200 600 100 ;
line 600 100 700 200 ;
color 1 ;
rectangulo 500 200 200 200 ;
color 0 ;
rectangulo 575 300 50 100 ;

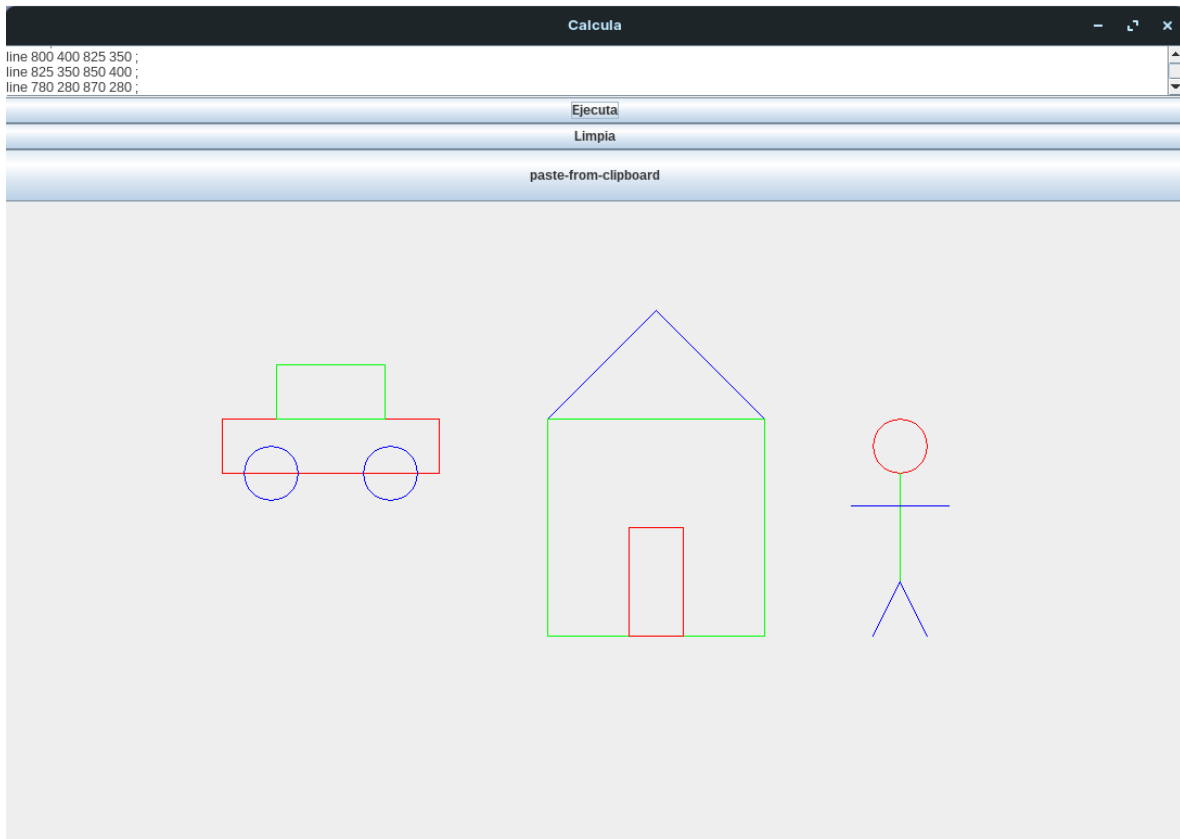
//Persona
color 0 ;
circulo 50 800 200 ;
color 1 ;
line 825 250 825 350 ;
color 2 ;
line 800 400 825 350 ;
line 825 350 850 400 ;
line 780 280 870 280 ;
```

Para la compilación del código fueron necesarios los siguientes comandos:

```
//Compilacion
byaccj -J forma.y
javac -d . *.java
java Parser
```

Cabe recalcar que si resulta un error al compilar Parser.java, es necesario eliminar la línea de código donde se indica el error y posteriormente volver a compilar Parser.java

Finalmente, al ejecutar el programa y darle los parámetros anteriores se obtiene como resultado las figuras mostradas a continuación:



## Conclusión

El comienzo de esta práctica me resultó particularmente difícil ya que no entendí muy bien a la primera lo que debíamos de realizar, además de que no sabía como compilar los archivos de la carpeta grafibasi por lo que tuve que investigar, posteriormente me encontré con el error mencionado anteriormente al compilar Parser.java, este error me desconcertó bastante ya que fue generado por el mismo Yacc a lo cual la única solución que encontré fue borrar la línea de código donde se encontraba el error y para fortuna mía funcionó y pude compilar los archivos, posteriormente solo fue realizar el código necesario para que se realizaran los dibujos y el realizarlos a punta de prueba y error hasta que resultaron dibujos decentes. Considero que esta práctica me ha sido de utilidad ya que no soy muy bueno con las interfaces gráficas y porque no llevo mucho tiempo programando en Java.



## **Referencias**

- [1] Wikipedia (s.f.). "Interfaz gráfica de usuario". [Internet]. Disponible en [https://es.wikipedia.org/wiki/Interfaz\\_gr%C3%A1fica\\_de\\_usuario](https://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario)