UNIVERSIDAD SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1
201830313 DENILSON FLORENTÍN DE LEÓN AGUILAR
MANUAL TÉCNICO

La aplicación de graficar figuras geométricas fue desarrollado por Denilson Florentín de León Aguilar, en el IDE Android Studio, además se requirió de herramientas como java, jflex y cup, los cuales fueron la base para el desarrollo de la aplicación.

Jflex: El Jflex nos sirvió para leer la entrada de datos, analizar los tokens correctos y enviarlo al parser que crea cup. Los parámetros enviados son los siguientes:

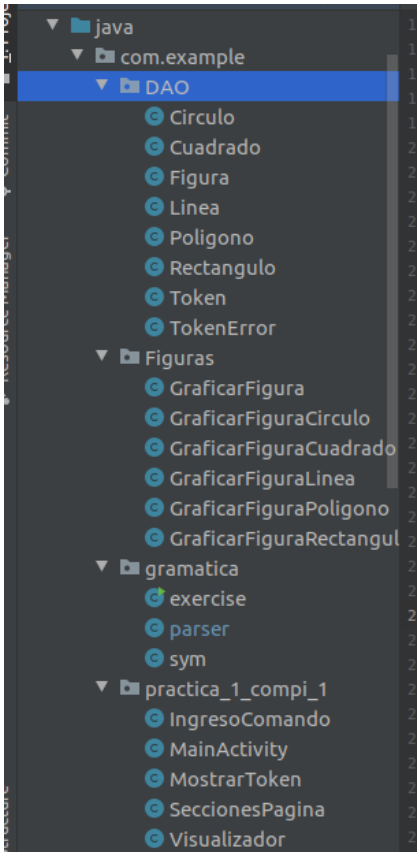
```
/*Tercera accion, expresiones*/
<YYINITIAL>{
    {DIGITO}    { return retornarSimbolo(ENTERO         , "ENTERO"         , yytext(), yyline + 1, yycolumn + 1); }
    "+"         { return retornarSimbolo(SUMA           , "SUMA"           , yytext(), yyline + 1, yycolumn + 1); }
    "-"         { return retornarSimbolo(RESTA          , "RESTA"          , yytext(), yyline + 1, yycolumn + 1); }
    "*"         { return retornarSimbolo(MULTIPLICACION , "MULTIPLICACION" , yytext(), yyline + 1, yycolumn + 1); }
    "/"         { return retornarSimbolo(DIVISION       , "DIVISION"       , yytext(), yyline + 1, yycolumn + 1); }
    "("         { return retornarSimbolo(PARENTESISA    , "PARENTESISA"    , yytext(), yyline + 1, yycolumn + 1); }
    ")"         { return retornarSimbolo(PARENTESISB    , "PARENTESISB"    , yytext(), yyline + 1, yycolumn + 1); }
    ","         { return retornarSimbolo(COMA           , "COMA"           , yytext(), yyline + 1, yycolumn + 1); }
    {CLR}       { return retornarSimbolo(COLOR          , "COLOR"          , yytext(), yyline + 1, yycolumn + 1); }
    {CURV}      { return retornarSimbolo(CURVA          , "CURVA"          , yytext(), yyline + 1, yycolumn + 1); }
    {CIRC}      { return retornarSimbolo(CIRCULO        , "CIRCULO"        , yytext(), yyline + 1, yycolumn + 1); }
    {CUAD}      { return retornarSimbolo(CUADRADO       , "CUADRADO"       , yytext(), yyline + 1, yycolumn + 1); }
    {RECTAN}    { return retornarSimbolo(RECTANGULO     , "RECTANGULO"     , yytext(), yyline + 1, yycolumn + 1); }
    {LIN}       { return retornarSimbolo(LINEA          , "LINEA"          , yytext(), yyline + 1, yycolumn + 1); }
    {POLIGON}   { return retornarSimbolo(POLIGONO       , "POLIGONO"       , yytext(), yyline + 1, yycolumn + 1); }
    {GRAF}      { return retornarSimbolo(GRAFICAR       , "GRAFICAR"       , yytext(), yyline + 1, yycolumn + 1); }
    {ANIM}      { return retornarSimbolo(ANIMAR         , "ANIMAR"         , yytext(), yyline + 1, yycolumn + 1); }
    {OBJ}       { return retornarSimbolo(OBJETO         , "OBJETO"         , yytext(), yyline + 1, yycolumn + 1); }
    {ANTER}     { return retornarSimbolo(ANTERIOR       , "ANTERIOR"       , yytext(), yyline + 1, yycolumn + 1); }
    {SEPARADORES}   {/*  */}
}

//Los simbolos que no concuerdan con las expresiones especificadas son errores lexicos
//Recibe: TipoError, Lexema, Mensaje del error/descripcion, fila y columna
    [^]         { addErrorLexico ("LEXICO", yytext(), "Token no valido",yyline + 1, yycolumn + 1);}
```

Cup: Recibe los parámetros de Jflex y los trabaja de la siguiente forma:
Reglas de graficación:

```
s ::= graficar
    | graficar s
    ;
graficar ::= GRAFICAR CIRCULO PARENTESISA producto:posX COMA producto:posY COMA producto:radio COMA COLOR:color PARENTESISB
    {:
        Figura aux = new Circulo(color.getLexema(),
                    Double.parseDouble(posX),
                    Double.parseDouble(posY),
                    Double.parseDouble(radio));
        listadoFiguras.add(aux);
    :}
    | GRAFICAR CUADRADO PARENTESISA producto:posX COMA producto:posY COMA producto:longitudLado COMA COLOR:color PARENTESISB
    {:
        Figura aux = new Cuadrado(color.getLexema(),
                    Double.parseDouble(posX),
                    Double.parseDouble(posY),
                    Double.parseDouble(longitudLado));
        listadoFiguras.add(aux);
    :}
    | GRAFICAR RECTANGULO PARENTESISA producto:posX COMA producto:posY COMA producto:alto COMA producto:ancho COMA COLOR:color PARENTESISB
    {:
        Figura aux = new Rectangulo(color.getLexema(),
                    Double.parseDouble(posX),
                    Double.parseDouble(posY),
                    Double.parseDouble(alto),
                    Double.parseDouble(ancho));
        listadoFiguras.add(aux);
    :}
    | GRAFICAR LINEA PARENTESISA producto:posX COMA producto:posY COMA producto:posX1 COMA producto:posY1 COMA COLOR:color PARENTESISB
    {:
        Figura aux = new Linea(color.getLexema(),
                    Double.parseDouble(posX),
                    Double.parseDouble(posY),
                    Double.parseDouble(posX1),
                    Double.parseDouble(posY1));
        listadoFiguras.add(aux);
    :}
    | GRAFICAR POLIGONO PARENTESISA producto:posX COMA producto:posY COMA producto:alto COMA producto:ancho COMA producto:cantidadLados COMA COLOR:color PARENTESISB
    {:
        Figura aux = new Poligono(color.getLexema(),
                    Double.parseDouble(posX),
                    Double.parseDouble(posY),
                    Double.parseDouble(alto),
                    Double.parseDouble(ancho),
                    Double.parseDouble(cantidadLados));
        listadoFiguras.add(aux);
    :}
```

Operaciones:

```
producto ::= producto:n1 SUMA producto:n2
    {:
        //System.out.print(" + ");
        Double num1 = Double.parseDouble(n1);
            Double num2 = Double.parseDouble(n2);
            Double aux = num1 + num2;
            RESULT = String.valueOf(aux);
    :}
    | producto:n1 RESTA producto:n2
    {:
        //System.out.print(" - ");
        Double num1 = Double.parseDouble(n1);
            Double num2 = Double.parseDouble(n2);
            Double aux = num1 - num2;
            RESULT = String.valueOf(aux);
    :}
    | producto:n1 MULTIPLICACION producto:n2
    {:
        //System.out.print(" * ");
        Double num1 = Double.parseDouble(n1);
            Double num2 = Double.parseDouble(n2);
            Double aux = num1 * num2;
            RESULT = String.valueOf(aux);
    :}
    | producto:n1 DIVISION producto:n2
    {:
        //System.out.print(" / ");
        Double num1 = Double.parseDouble(n1);
            Double num2 = Double.parseDouble(n2);
            Double aux = num1 / num2;
            RESULT = String.valueOf(aux);
    :}
    | PARENTESISA producto:p
    {:
        //System.out.print(" ( )");
            RESULT = p;
    :}
    PARENTESISB
    | ENTERO:e
    {:
        //System.out.print(" Entero");
            RESULT = e.getLexema();
    :}
    | RESTA producto:n1
    {:
        //System.out.print(" Entero negativo ");
```



Paquetes: Para el desarrollo de la aplicación, se crearon varios paquetes, que sirven para tener el código ordenado.

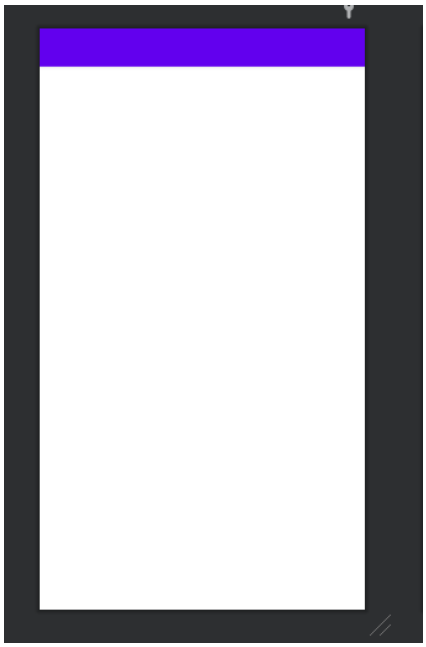
- DAO: Este paquete contiene todas las clases que solo poseen atributos, getters y setters.
- Figuras: Posee la graficación de todas las figuras, donde la clase padre es “GraficarFigura” y el resto hereda de ella.
- Gramatica: Posee las clases generadas por jflex y cup.
- practica_1_compi_1: Posee las clases principales, donde se ejecuta todo el programa.

Las clases más importantes son las del último paquete, con lo cual, se explicará cada una.

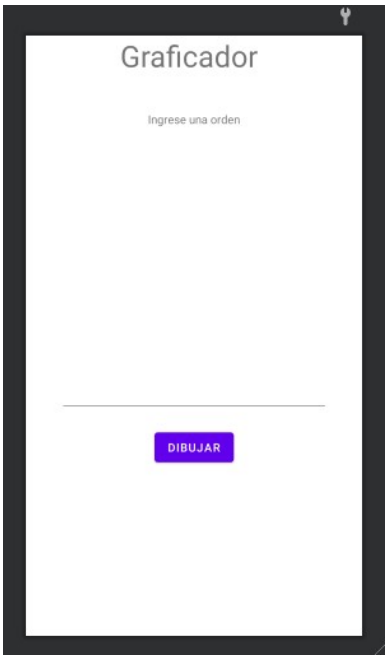
- MainActivity: Es la página principal y contiene todos los demás componentes enlazados a las clases de IngresoComando, Visualizador y MostrarToken.
- SeccionesPagina: Corresponde a un tab en el menú principal y se encarga de enlazar los componentes de los paneles a la ventana principal.
- IngresoComando: Esta clase está enlazado a la interfaz de usuario del panel donde se ingresan los comandos, después de recibir la entrada de datos, envía la información a la clase Visualizador.
- Visualizador: Ejecuta el lexer y el parser, además de graficar las figuras recibidas y enviar los datos de los reportes a la clase MostrarToken.
- MostrarToken: Muestra la tabla con todos los reportes.

Ventanas:

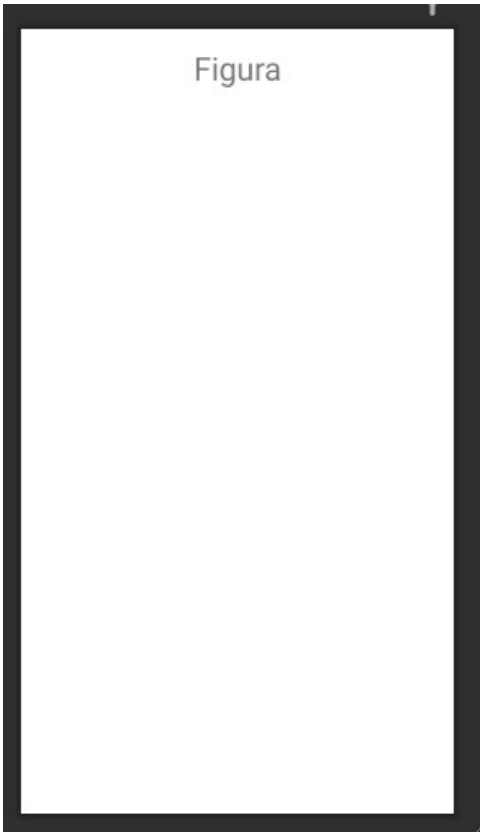
Ventana principal:



Fragmento de ingreso de datos:



Fragmento de graficos:



Fragmento de reportes:

En el espacio azul podemos notar un cuadro vacío, ese espacio corresponde a la tabla.