

MANUAL TÉCNICO.
Fredy José Gabriel Herrera Funes - 202130478

- **Detalle de organización del proyecto:**

El proyecto se encuentra organizado por paquetes, el paquete principal es: com.example.aplicacioncompi1, que deriva de la carpeta main de la aplicación con Kotlin.

1. Luego esta la subcarpeta: “uitheme”, que dentro de ella se encuentra la carpeta “logic”, en la cual se encuentran las clases principales de lógica de la aplicación, como clase principal está la clase de Kotlin, llamada: “Analizador”, que es la que se encarga de llamar a las clases “Lexer” y “Parser” para realizar el análisis léxico y sintáctico de la entrada de texto que se tendrá.
2. La clase “Analizador” luego del análisis también se encarga de llamar a las clases Generadores(GeneradorDiagrama, GeneradorReportes, GeneradorReportesError) que tienen como tarea generar la información que necesita el usuario para la utilización de la aplicación.
3. También se tiene en cuenta que para la estructura de las clases “Lexer” y “Parser”, se utilizó las gramáticas correspondientes y se generaron con las herramientas JFlex y CUP.
4. Luego en la subcarpeta “uitheme”, se encuentra adentro la carpeta “ViewModel”, la cual contiene una clase de Kotlin llamada “DiagramView” que se encarga de generar la vista del diagrama con la información y nodos que le manda el GeneradorDiagrama.
5. Como último tenemos la clase MainActivity, que se encuentra en el paquete principal, esta clase es la encargada de generar e inicializar toda la vista inicial, y de recolectar el archivo de entrada, para su posterior análisis, como generación de diagrama y reportes.

- **Gramática analizador léxico y sintáctico:**

$$G = \{V, \Sigma, R, S\}$$

- **SÍMBOLOS NO TERMINALES (V):**

V
= {programa, bloque_codigo, lista_sentencias, sentencia, condicion, expresion, termino, factor, bloque_config, lista_config, instrucion_config, color, rgb, figura, letra}

- **SÍMBOLOS TERMINALES (Σ):**

$$\Sigma =$$

{INICIO, FIN, SI, ENTONCES, MIENTRAS, HACER, MOSTRAR, LEER, VAR, SEP
ARADOR, DEFAULT, COLOR_TEXTO_SI, COLOR_SI, FIGURA_SI, LETRA_SI,
LETRA_SIZE_SI, COLOR_TEXTO_MIENTRAS, COLOR_MIENTRAS, FIGUR

A_MIENTRAS,LETRA_MIENTRAS,LETRA_SIZE_MIENTRAS,COLOR_TEXTO_BLOQUE,COLOR_BLOQUE,FIGURA_BLOQUE,LETRA_BLOQUE,LETRA_SIZE_BLOQUE,ELIPSE,CIRCULO,PARALELOGRAMO,RECTANGULO,ROMBO,RECTANGULO_REDONDEADO,ARIAL,TIMES_NEW_ROMAN,COMIC_SANS,VERDANA,PARENTESIS_ABIERTO,PARENTESIS_CERRADO,MAS,MENOS,MULTIPLICADO,DIVIDIDO,IGUAL,IGUAL_IGUAL,DIFERENTE,MENOR,MAYOR,MENOR_IGUAL,MAYOR_IGUAL,AND,OR,NOT,COMA,PIPE,CADENA,ENTERO,DECIMAL,ID,COLOR_HEX}

- REGLAS DE PRODUCCIÓN (R):

programa -> bloque_codigo
| bloque_codigo SEPARADOR bloque_config

bloque_codigo -> INICIO lista_sentencias FIN

lista_sentencias -> lista_sentencias sentencia
| sentencia

sentencia -> VAR ID IGUAL expresion
| ID IGUAL expresion
| MOSTRAR CADENA
| MOSTRAR ID
| LEER ID
| SI PARENTESIS_ABIERTO condicion PARENTESIS_CERRADO
ENTONCES lista_sentencias FIN SI
| MIENTRAS PARENTESIS_ABIERTO condicion
PARENTESIS_CERRADO HACER lista_sentencias FIN MIENTRAS

condicion -> expresion IGUAL_IGUAL expresion
| expresion DIFERENTE expresion
| expresion MENOR expresion
| expresion MAYOR expresion
| expresion MENOR_IGUAL expresion
| expresion MAYOR_IGUAL expresion
| condicion AND condicion
| condicion OR condicion
| NOT condicion
| PARENTESIS_ABIERTO condicion PARENTESIS_CERRADO

expresion -> expresion MAS termino
| expresion MENOS termino
| termino

termino -> termino MULTIPLICADO factor
| termino DIVIDIDO factor
| factor

```

factor -> ENTERO
| DECIMAL
| ID
| PARENTESIS_ABIERTO expresion PARENTESIS_CERRADO

bloque_config -> lista_config

lista_config -> lista_config instrucion_config
| instrucion_config

instrucion_config -> DEFAULT IGUAL expresion
| COLOR_TEXTO_SI IGUAL color PIPE expresion
| COLOR_SI IGUAL color PIPE expresion
| FIGURA_SI IGUAL figura PIPE expresion
| LETRA_SI IGUAL letra PIPE expresion
| LETRA_SIZE_SI IGUAL expresion PIPE expresion
| COLOR_TEXTO_MIENTRAS IGUAL color PIPE expresion
| COLOR_MIENTRAS IGUAL color PIPE expresion
| FIGURA_MIENTRAS IGUAL figura PIPE expresion
| LETRA_MIENTRAS IGUAL letra PIPE expresion
| LETRA_SIZE_MIENTRAS IGUAL expresion PIPE expresion
| COLOR_TEXTO_BLOQUE IGUAL color PIPE expresion
| COLOR_BLOQUE IGUAL color PIPE expresion
| FIGURA_BLOQUE IGUAL figura PIPE expresion
| LETRA_BLOQUE IGUAL letra PIPE expresion
| LETRA_SIZE_BLOQUE IGUAL expresion PIPE expresion

color -> COLOR_HEX
| rgb

rgb -> expresion COMA expresion COMA expresion

figura -> ELIPSE
| CIRCULO
| PARALELOGRAMO
| RECTANGULO
| ROMBO
| RECTANGULO_REDONDEADO

letra -> ARIAL
| TIMES_NEW_ROMAN
| COMIC_SANS
| VERDANA

```

- SÍMBOLO INICIAL (S):

S= {programa}

- **Diagrama de Clases:**

