

*Universidad de San Carlos de Guatemala USAC.  
División de Ciencias de la Ingeniería.  
Centro Universitario de Occidente CUNOC.  
Organización de lenguajes y compiladores 1  
Ing. Moises*

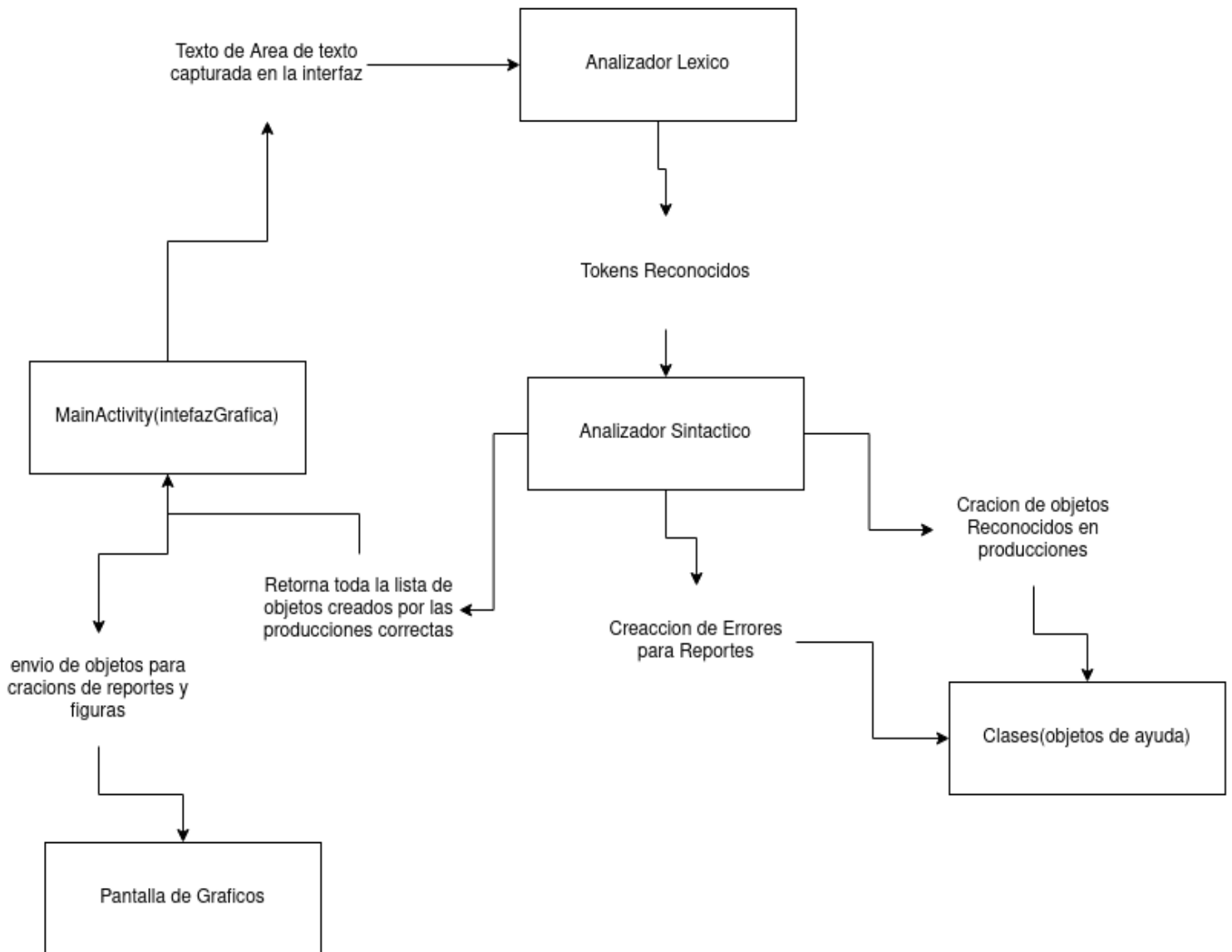


*Estudiante  
Elvis Lizandro Aguilar Tax*

*Carnet  
201930304*

*Practica No. 1 – Manual Tecnico  
Generador de gráficos a base de un texto de entrada estructurada de órdenes y atributos  
para el Gráfico(Barras, Pie)*

## Organización del proyecto



*Se inicia con el activityMain clase principal donde arranca el proyecto este misma instancia un objeto del tipo parser(analizador sintáctico) en el que el mismo instancia un analizador Léxico que le solicita los tokens.*

*al reconocer una estructura correcta de la gramática se crean o instancian las gráficas para poder ser usadas mas adelante así mismo se registran errores si es que existen retorna las gráficas al activityMain y luego este le envía las gráficas a la pantalla Graficos para poder ser graficada.*

## ***Análisis de gramática para analizador léxico***

*En analizador léxico es el encargado de reconocimiento de expresiones regulares, que definen tokens o símbolos que estos serán usados en un análisis sintáctico, en base a un texto de entrada, esta misma la recibirá de la aplicación android mediante a un multiline Text.*

*Análisis: en base a qué se necesita palabras reservadas e identificadores así como también operadores algebraicos e incluso bloques de comentario se crean 4 tipo de reconocimientos.*

### ***—Palabras reservadas***

***Barras, Def o def, Pie, título, ejex, ejey, etiquetas, valores, Cantidad, Porcentaje, unir, tipo, total, extra, Ejecutar.***

### ***—Identificador***

*Para los identificadores es más preciso reconocer una expresión regular compuesta de signo (") seguida de letras o número o espacios varias veces, terminando cuando encuentre el signo (").* — (") (letras|números| espacios)\*(")

### ***—OperadoreAlgebraicos***

*signos conocidos como operadores de forma* — (+|-|\*|/)

### ***—Signos de Puntuación***

*signos conocidos como puntuación de forma* — (;|,|:|.)

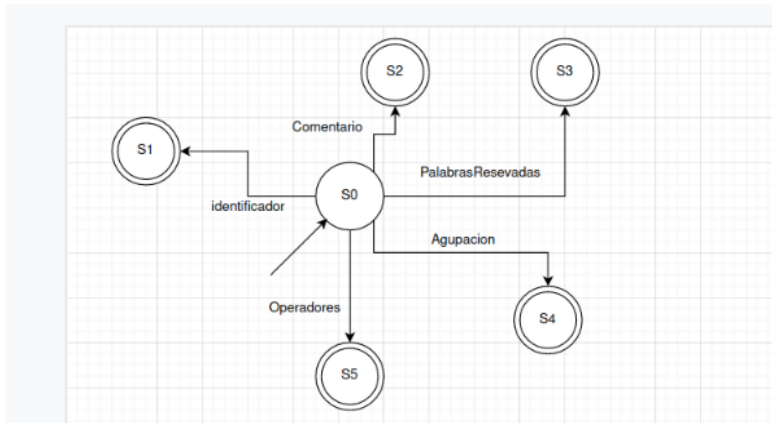
### ***—Signo de agrupación***

*signos conocidos como agrupación de forma* — ([|]|{|})

### ***—Comentario Bloque***

*signos conocidos como operadores de forma* — (# seguida de cualquier expresión)

## ***Autómata Determinista General***



***LETRA = [a-zA-Z]***

***WhiteSpace = [\r|\n|\r\n] | [\t\f]***

***SIGNO\_MENOS = "-"***

***SIGNO\_MAS = "+"***

***SIGNO\_POR = "\*"***

***SIGNO\_DIVISION = "/"***

***SIGNO\_PARENTESISISA = "("***

***SIGNO\_PARENTESISISC = ")"***

***SIGNO\_LLAVEA = "{"***

***SIGNO\_LLAVEC = "}"***

***SIGNO\_CORCHETEA = "["***

***SIGNO\_CORCHETEC = "]"***

***SIGNO\_COMILLAS = "\""***

***SIGNO\_PUNTO = "."***

***SIGNO\_COMA = ","***

***SIGNO\_PUNTOCOMA = ";"***

***SIGNO\_DOPUNTO = ":"***

***NUMERO = [0-9]+***

***DECIMAL = (({NUMERO})({SIGNO\_PUNTO})({NUMERO}))***

***INICIO\_COMENTARIO\_BLOQUE = "#"***

***FIN\_COMENTARIO\_BLOQUE = [\n]***

***DEF= ("Def" | "def")***

***BARRAS = "Barras"***

***PIE = "Pie"***

***TITULO = "titulo"***

***EJEX = "ejex"***

***EJEY = "ejey"***

***ETIQUETAS = "etiquetas"***

***VALORES = "valores"***

***CANTIDAD = "Cantidad"***

***PORCENTAJE = "Porcentaje"***

**UNIR** = "unir"  
**TIPO** = "tipo"  
**TOTAL** = "total"  
**EXTRA** = "extra"  
**EJECUTAR** = "Ejecutar"  
**IDD** = (\"({LETRA}{NUMERO}|s)\*\)

## **Gramática para analizador sintáctico**

El analizador sintáctico, conocido como parser, o en el lenguaje Java la Herramienta de Parseo(CUP), se encarga de plantear la validez sintáctica por medio de revisión de gramáticas que representan una secuencia de tokens enviadas por el analizador léxico, revisando el orden de las mismas y que la expresión sea reconocida de acuerdo con las reglas gramaticales que se estén implementando.

## **Gramática (G)**

$G(N,T,P,S)$

- ❖ **N = No terminales**
- ❖ **T = Terminales**
- ❖ **P = Reglas de Producción**
- ❖ **S = Símbolo Inicial**

### **N — Símbolos no Terminales**

- *q\_inicial, comodin\_a, comodin\_b, a\_barra, b\_pie, title, y\_eje, unir, x\_eje, val, etiquet, p\_tipo, f\_ejecutar, n\_consecutivos, i\_tems, l\_total, r\_extra, h\_tupla;*
- *Integer operation;*
- *Double o\_decimal*

### **T — Símbolos Terminales**

- *MENOS, MAS, POR, DIVISION, PARENTESIS, PARENTESIS, LLAVE, LLAVE, CORCHETE, CORCHETE ;*
- *COMA, PUNTO, COMA, DOPUNTO, NUMERO, DECIMAL, DEF, BARRAS, PIE, TITULO, EJEX, EJEX, ETIQUETAS, VALORES, CANTIDAD;*
- *PORCENTAJE, UNIR, TIPO, TOTAL, EXTRA, EJECUTAR, IDD;*

### **P — Reglas de Producción**

**q\_inicial** — Producción Inicial

*q\_inicial* ::= DEF BARRAS LLAVEA comodin\_a LLAVEC *q\_inicial*  
 | DEF PIE LLAVEA comodin\_b LLAVEC *q\_inicial*  
 | DEF PIE LLAVEA comodin\_b LLAVEC *f\_ejecutar*  
 | DEF BARRAS LLAVEA comodin\_a LLAVEC *f\_ejecutar*

*comodin\_a* → *a\_barra*

*a\_barra* → title *a\_barra*  
 | *y\_eje a\_barra*  
 | *unir a\_barra*  
 | *x\_eje a\_barra*  
 |  $\lambda$

*comodin\_b* → *b\_pie*

*b\_pie* → title *b\_pie*  
 | *p\_tipo b\_pie*  
 | *etiquet b\_pie*  
 | *val b\_pie*  
 | *l\_total b\_pie*  
 | *unir b\_pie*  
 | *r\_extra b\_pie*  
 |  $\lambda$

*etiquet* → ETIQUETAS DOPUNTO CORCHETEA *i\_tems* CORCHETEC PUNTOCOMA

*val* → VALORES DOPUNTO CORCHETEA *n\_consecutivos* CORCHETEC PUNTOCOMA

*h\_tupla* → LLAVEA operation COMA operation LLAVEC COMA *h\_tupla*  
 | LLAVEA operation COMA operation LLAVEC

*l\_total* → TOTAL DOPUNTO *o\_decimal* PUNTOCOMA

*title* → TITULO DOPUNTO *IDD* PUNTOCOMA

*y\_eje* → EJEY DOPUNTO CORCHETEA *n\_consecutivos* CORCHETEC PUNTOCOMA

*x\_eje* → EJEX DOPUNTO CORCHETEA *i\_tems* CORCHETEC PUNTOCOMA

*r\_extra* → EXTRA DOPUNTO *IDD* PUNTOCOMA

*p\_tipo* → TIPO DOPUNTO PORCENTAJE PUNTOCOMA  
 | TIPO DOPUNTO CANTIDAD PUNTOCOMA

$n\_consecutivos \rightarrow o\_decimal \text{ COMA } n\_consecutivos$   
|  $o\_decimal$

$i\_tems \rightarrow IDD \text{ COMA } i\_tems$   
|  $IDD$

$f\_ejecutar \rightarrow EJECUTAR \text{ PARENTESISISA } IDD \text{ PARENTESISISC } PUNTOCOMA f\_ejecutar$   
|  $\lambda$

$operation \rightarrow operation \text{ MAS } operation$   
|  $operation \text{ MENOS } operation$   
|  $operation \text{ POR:operador } operation$   
|  $operation \text{ DIVISION } operation$   
|  $NUMERO$

$o\_decimal ::= o\_decimal \text{ MAS } o\_decimal$   
|  $o\_decimal \text{ MENOS:operador } o\_decimal$   
|  $o\_decimal \text{ POR } o\_decimal$   
|  $o\_decimal \text{ DIVISION } o\_decimal$   
|  $NUMERO$   
|  $DECIMAL$   
|  $PARENTESISISA o\_decimal \text{ PARENTESISISC}$

# Diagrama de Clases

