

Manual Técnico

1. Organización del Proyecto.

La aplicación móvil desarrollada en este proyecto esta conformada por 3 capas, cada una manifiesta un comportamiento y una funcionalidad indicada para llegar al objetivo final de la misma, el poder graficar y animar formas geométricas por medio de un lenguaje sencillo, las capas del proyecto son:

I. Analizadores

Los analizadores son los encargados de tomar el código fuente y convertirlo en objetos, primero se comienza por el analizador léxico, este toma la entrada (Código fuente) y la divide en caracteres, luego toma cada carácter y verifica que estos formen los lexemas indicados para el lenguaje, cuando forma una cadena, ya sea aceptada o no, este la clasifica y envía cada lexema al analizador sintáctico, este analizador se encarga de verificar que cada lexema venga en el orden correcto según sus reglas de producción, cuando le termina de servir un lexema le pide el siguiente al analizador léxico hasta que la entrada de texto finalice, cuando el analizador sintáctico se encuentra con una entrada de una forma este crea un objeto del tipo indicado, luego lo envía a una segunda clase que se encarga de recolectar todas las figuras a graficar, cuando una de las figuras tiene una animación se le agrega como un atributo a la misma, de lo contrario este atributo esta vacío, en el mismo proceso el analizador sintáctico se encarga de ejecutar las operaciones matemáticas con la precedencia indicada, así mismo contiene listas que se encargan de recolectar la información debida para los reportes.

II. Objetos

Los objetos se encargan de representar lo que el usuario pide a través del código, cada vez que se pide una figura se llama a la clase correspondiente y se instancia un objeto, así mismo también hay objetos que nos ayudan a administrar los demás objetos, en este caso la clase Graficador se encarga de guardar todas las figuras con su respectiva animación (Si la hay) a graficar, luego esta es obtenida la clase FigurasActivity que la contiene en una lista esta lista es enviada a la clase Dibujador que se encarga de graficar las figuras solicitadas, luego de graficar, se obtienen los reportes, estos los obtenemos por medio de nuestros analizadores anteriormente mencionados, en estos analizadores tenemos los objetos correspondientes a cada dato, estos objetos se agregan a una lista y esta lista es la que obtenemos como habíamos mencionado con anterioridad, luego estas listas son enviadas al presionar el botón a nuestra clase ReportesActivity, esta se encarga de recibirlas y mostrarlas al usuario.

III. Interfaz Grafica

La interfaz es sencilla, tenemos una actividad principal donde se ingresara el código fuente correspondiente, al presionar el botón esta enviara los datos a la siguiente interfaz que se encargara de mostrar las figuras de una manera grafica y al presionar el botón de reportes se mostraran las listas anteriormente mencionadas.

2. Analizador Léxico

Como se menciono antes, el analizador léxico esta encargado de dividir el código fuente en lexemas y clasificarlo para luego enviarlos al analizador sintáctico.

Para realizar esta tarea primero se basa en la siguiente expresión regular que expresa las cadenas aceptadas por el lenguaje:

*((|) | [0-9]+ | azul | rojo | verde | amarillo | naranja | morado | café | negro | + | - | * | / | circulo | cuadrado | rectangulo | linea | poligono | graficar | , | animar | objeto | anterior | curva)*

Este toma carácter por carácter para moverse entre los estados del AFD que genera, cuando este entra en un estado de aceptación se le enviá el lexema al analizador sintáctico, cada lexema formado es una instancia de un Token, en la siguiente tabla se podrá ver la clasificación de estos tokens:

Token	Lexemas
PARENTESIS_A	“ (“
PARENTESIS_C	“) ”
CUADRADO	“cuadrado”
CIRCULO	“circulo”
RECTANGULO	“rectangulo”
LINEA	“linea”
POLIGONO	“poligono”
GRAFICAR	“graficar”

COMA	“ , ”
ENTERO	Numero enteros del 0 al infinito
SUMA	“ + ”
RESTA	“ - ”
MULTI	“ * ”
DIVISION	“ / ”
COLOR	("azul" "rojo" "verde" "amarillo" "naranja" "morado" "cafe" "negro")
ANIMAR	“animar”
OBJETO	“objeto”
ANTERIOR	“anterior”
CURVA	“curva”

3. Analizador Sintáctico

Este analizador se encarga de recibir cada lexema enviado por el léxico, este verifica que el orden en el que vienen los lexemas sea el correcto, en caso contrario maneja el error para poder seguir analizando el código fuente.

Este también usa un autómata para el análisis, en este caso un autómata de pila que va siguiendo las instrucciones dadas por las siguiente gramática:

Simbolo inicial = {inicio}

T =

{GRAFICAR,CUADRADO,CIRCULO,RECTANGULO,LINEA,POLIGONO,COMA,COLOR,PARENTESIS_A,PARENTESIS_C,SUMA,RESTA,MULTI,DIVISION,ANIMAR,OBJETO,ANTERIOR,CURVA,error}

N = {inicio,a,b,c,d,e,f,g}

P=

inicio → a

| error

a → GRAFICAR b a

| lamda

**b → CUADRADO PARENTESIS_A c COMA c COMA c COMA COLOR
PARENTESIS_C e**

**| CIRCULO PARENTESIS_A c COMA c COMA c COMA COLOR
PARENTESIS_C e**

| RECTANGULO PARENTESIS_A c COMA c COMA c COMA c COMA
COLOR PARENTESIS_C e

| LINEA PARENTESIS_A c COMA c COMA c COMA c COMA COLOR
PARENTESIS_C e

| POLIGONO PARENTESIS_A c COMA c COMA c COMA c COMA c
COMA COLOR PARENTESIS_C e

| error

c → d

| error

d → d SUMA d

| d RESTA d

| d MULTI d

| d DIVISION d

| ENTERO

| PARENTESIS_A d PARENTESIS_C

| error

e → ANIMAR g

| error

g → OBJETO ANTERIOR PARENTESIS_A c COMA c COMA f PARENTESIS_C

| error

f → CURVA

| LINEA

4. Diagrama de Clases

