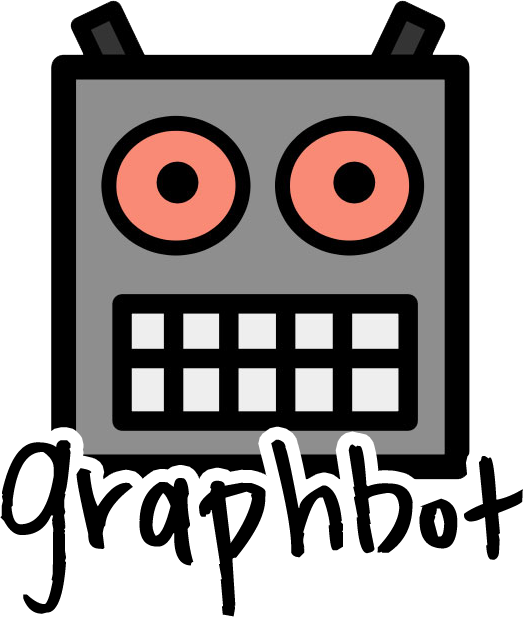
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY CAMPUS MONTERREY



Fernando Mendoza Ramírez A00515576

Claudia Graciela Hernández Rivera A01033904

Campus Monterrey

Maestras Titulares: Ing. Elda Quiroga

Curso: Compiladores

Monterrey, Nuevo León, a 7 de mayo de 2014

**a) Descripción del Proyecto**

* **Visión, Objetivos y Alcance del Proyecto**

**Visión:**

Diseñar un lenguaje gráfico que pueda ser utilizado como una herramienta de aprendizaje para jóvenes sin nociones de programación. El lenguaje permitirá que los usuarios aprendan los fundamentos de la lógica de programación de una manera divertida al utilizar una línea de comandos que irá dibujando líneas, figuras y formas en un canvas conforme el usuario vaya ejecutando los comandos.

**Objetivos:**

El lenguaje Graphbot tiene como objetivos:

* Ser un lenguaje de fácil aprendizaje para su trabajo con niños y jóvenes
* Enseñar los principales conceptos de la lógica de programación mediante el soporte para manejo de listas, ciclos, variables, entre otros
* Desarrollar habilidades metacognitivas en los niños o jóvenes mediante la presentación de retos intelectuales que puedan ser resueltos con el lenguaje

**Alcance del Proyecto:**

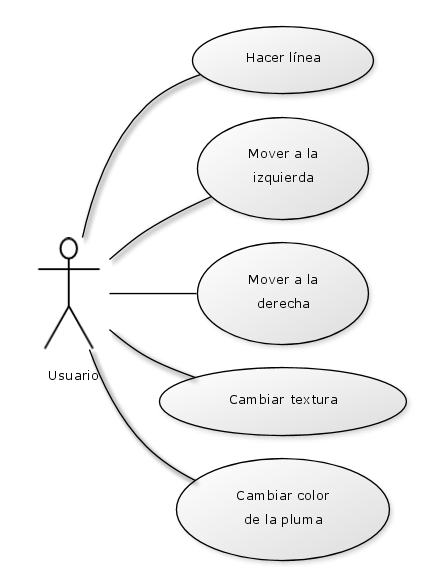
El lenguaje tiene como objetivo ser implementado en escuelas primarias para alentar a los jóvenes en el área de programación y fomentar el gusto por las ciencias computacionales. Se busca que sea un lenguaje atractivo que capture el interés de los niños para que quieran continuar programando. El mostrarles directamente en pantalla el resultado de los comandos que ellos mismos han programado, los incentivará a querer continuar utilizando el lenguaje.

* **Análisis de Requerimientos y Casos de Uso Generales**

**Requerimientos Principales**

* Manejo de Ciclos (*for, while)*
* Implementación de condicionales (*if)*
* Manejo de variables tipo lista
* Uso de módulos y funciones
* Salida gráfica
* Implementación de línea de comandos y ejecución de programas dentro de una aplicación

**Casos de Uso Generales**



**1. Hacer línea**

El usuario hace una línea mediante el comando *move* y una distancia especificada.

**2. Mover a la izquierda**

El usuario mueve al robot a la izquierda al utilizar el comando *leftturn* seguido de una cantidad en grados.

**3. Mover a la derecha**

El usuario mueve al robot a la izquierda al utilizar el comando *rightturn* seguido de una cantidad en grados.

**4. Cambiar textura**

El usuario cambia el fondo de la aplicación utilizando el comando *setbackgroundtxt* seguido del número de textura que quiera utilizar.

**5. Cambiar color de la pluma**

El usuario cambia el color de la pluma (comienza en negro) utilizando el comando *setcolor* seguido de los tres valores R,G,B del color al que quiere cambiar.

* **Descripción de los principales Test Cases**

espiral.gbot

Programa que dibuja una espiral. El programa comienza asignando la primera textura a la aplicación y después declara una variable i con valor inicial 10.0 Después utilizando esa variable entra a un ciclo for que aumenta la variable i en una cantidad de 10.0 cada vez que completa una iteración y lo repite hasta llegar a la cantidad 110.0. En cada iteración del for, se dibuja una línea con distancia i y se hace una vuelta a la derecha en 90º.

figuras.gbot

Programa que dibuja un asterisco y 4 octagonos alrededor. El programa comienza declarando la función *octagon* y declara una variable i que después es utilizada dentro de un ciclo for que se repite 8 veces y realiza líneas con una distancia de 50.0 para después girar a la derecha en 45º. Se declara otra función llamada *asterisk* que salva una variable j y la utiliza en un ciclo for que repite 20 veces y realiza una línea con distancia de 80.0, luego se dirige hacia abajo y vuelve a moverse en 80.0 para terminar dando una vuelta a la derecha en 18º. Finalmente llega el programa *main* que asigna la segunda textura al canvas, luego cambia el color de la pluma a blanco, llama la función *asterisk*, luego guarda otra variable i y repite la función *octagon* cuatro veces sobre el canvas.

* **Descripción del proceso general seguido para el desarrollo del proyecto**

**Semana 2 (Marzo 10 – 14)**

* Resolución de conflictos
  + - 75 reduce/reduce
    - 5 shift/Reduce
* Directorio de procedimientos
* Inserción de procedimientos por declaración
* Tablas de Variables
* Inserción a tablas de variables por declaración y asignación
* Verificación de uso de variables inexistentes
* Creación de Makefile

**Semana 3 (Marzo 17 – 21)**

* Generación de Cuádruplos
  + - Expresiones Aritméticas
    - Asignación, Lectura (Comandos)
    - Comparadores
* Impresión a archivo de texto
* Revaluación ciclos y condiciones

**Semana 4 (Marzo 25 – 29)**

* Generación de Cuádruplos
  + - If
    - While
    - For

**Semana 5 (Abril 31 – 4)**

* Codificar comandos
* Funciones/Módulos
* Manejo de Variables Tipo Lista (tabla de variables, memoria virtual)
* Corregir estructura de datos utilizada para la tabla de variables y directorio de procedimientos
* Corregir cuando se mandan llamar dos funciones definidas por el usuario seguidas

**Semana 6 (Abril 7 – 11)**

* Implementar diseño de memoria
* Codificar comandos
* Convertir constantes numéricas en direcciones virtuales
* Codificar comparadores y operadores aritméticos
* Convertir saltos en direcciones virtuales
* Convertir variables en direcciones virtuales
* Convertir temporales en direcciones virtuales
* Cálculo de tamaño de funciones

**Semana Santa (Abril 15 – 19)**

* Cargar memoria en ejecución
* Cargar constantes
* Ejecución de funciones aritméticas

**Semana 7 (Abril 21 – 25)**

* Manejo de Memorias
* Ejecución de Estatutos Condicionales
* Instrucciones *move*, *rightturn* y *leftturn* listas

**Semana 8 (Abril 28 – 2)**

* Implementación de nuevos comandos
* Show
* Hide
* PenUp
* PenDown
* RightTurn
* LeftTurn
* Integración con WX Widgets (GUI)
* Implementación de comentarios
* Primera Versión de Documentación

**Principales aprendizajes logrados:**

**Fernando Mendoza:**

“*Durante el prcoa adasjdjasbksjdkajsdnkjasndkajsds*”

**Claudia Hernández:**

“*Durante el prcoa adasjdjasbksjdkajsdnkjasndkajsds*”

**b) Descripción del Lenguaje**

* **Nombre del Lenguaje**

Graphbot

* **Descripción genérica de las principales características del lenguaje**

Graphbot es un lenguaje que permite crear programas y funciones que tienen salida gráfica. Mediante sus diferentes tipos de comandos, Graphbot permite la creación de líneas, figuras y dibujos en pantalla. La línea de comandos permite evaluar de manera automática en pantalla la salida de los comandos, mientras que el *text area* permite la creación de programas que se ejecutarán al ser llamados desde línea de comandos. Graphbot permite la creación de infinidad de dibujos gracias a sus ciclos, condicionales y demás opciones que brindan variedad a la forma de dibujar sobre el canvas.

|  |  |  |  |
| --- | --- | --- | --- |
| Gráficas | Queries | Estructuras de Control | Misceláneas |
| move rightturn leftturn setpos set x set y setcolor setpensize setbackground setbackgroundtxt show hide clean home penup pendown | getcolorR  getcolorG  getcolorB  getpensize  getx  gety | program  function  save  for  if  while  end  true  false | playmusic  stopmusic  main  float  bool  about |

* **Descripción de los errores que pueden ocurrir, tanto en compilación como ejecución**

**c) Descripción del Compilador**

* **Equipo de Cómputo, lenguaje y utilerías especiales usadas en el desarrollo del proyecto**

El lenguaje se desarrolló en computadoras portátiles con OS X Versión 10.9.1. La herramienta Flex y Bison fue utilizada para el desarrollo del compilador. Se utilizó además tanto para el compilador como para la máquina virtual la versión 4.2.1 de g++.

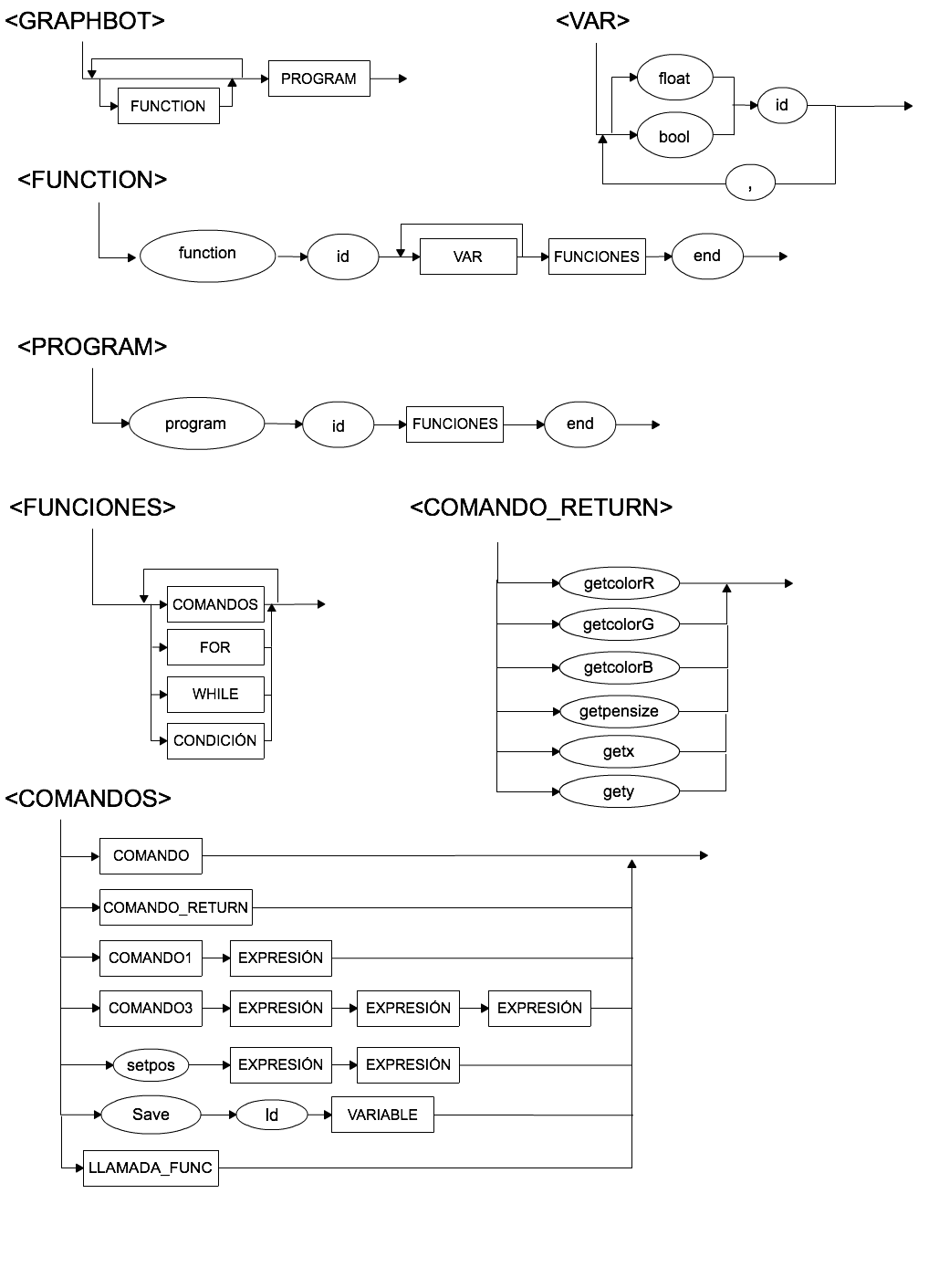
* **Descripción del Análisis de Léxico**
  + **Patrones de Construcción**
  + **Enumeración de Tokens**
* **Descripción del Código Intermedio y Análisis Semántico**
  + **Código de operación y direcciones virtuales asociadas a los elementos del código**

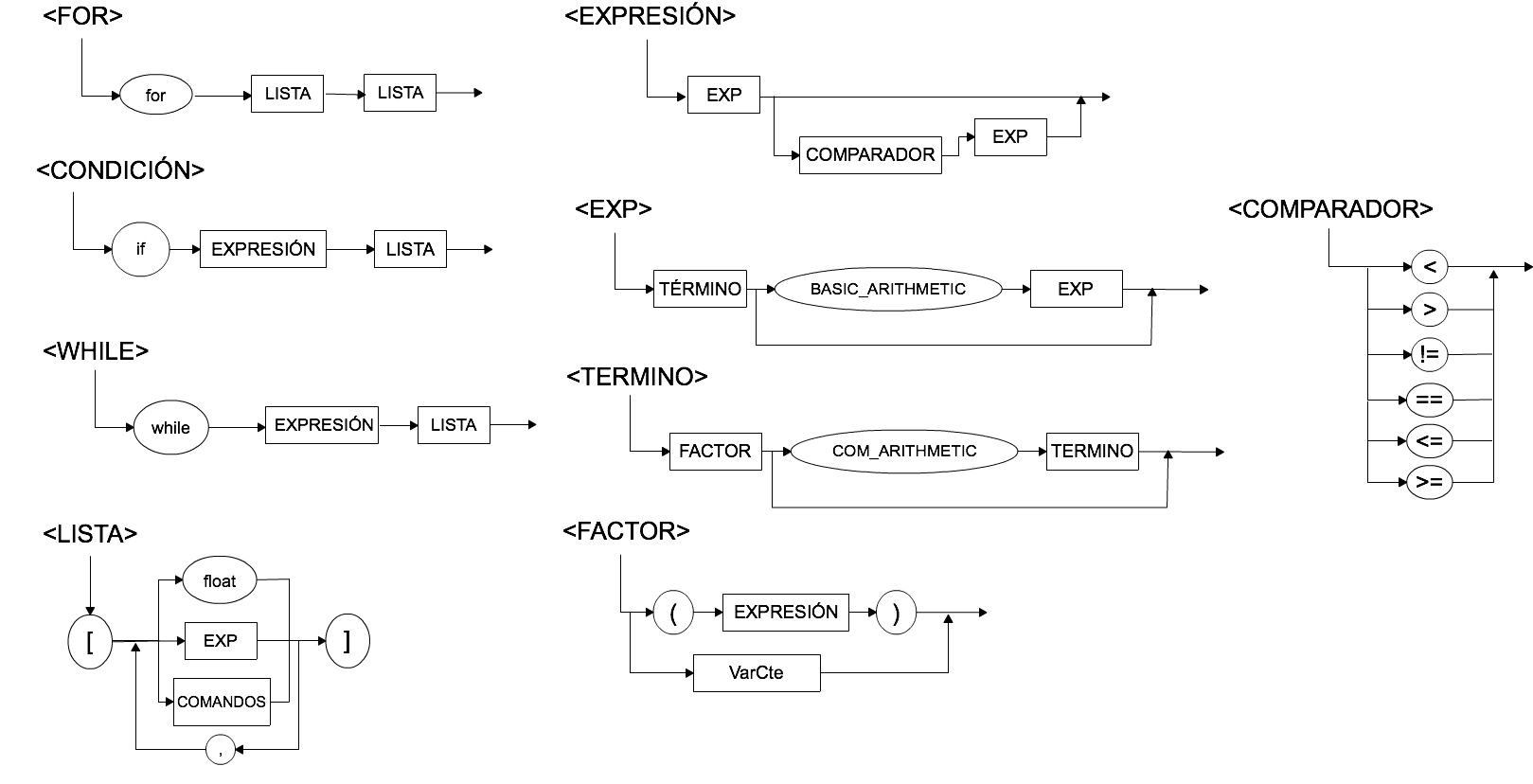
0001 - 0999 : constantes saltos  
1000 - 1999 : constantes numéricas  
2000 - 2999 : temporales  
3000 - 3999 : variables flotantes  
4000 - 4999 : variables enteras  
5000 - 5042 : comandos

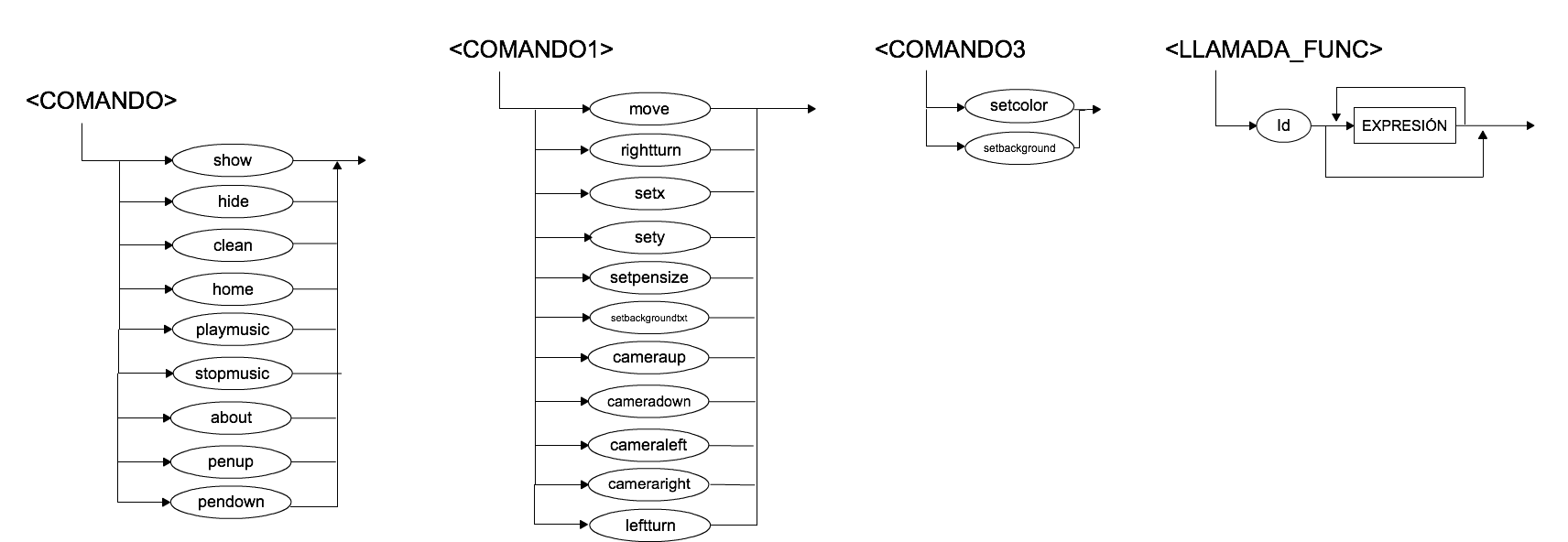
**Codificación de comandos**

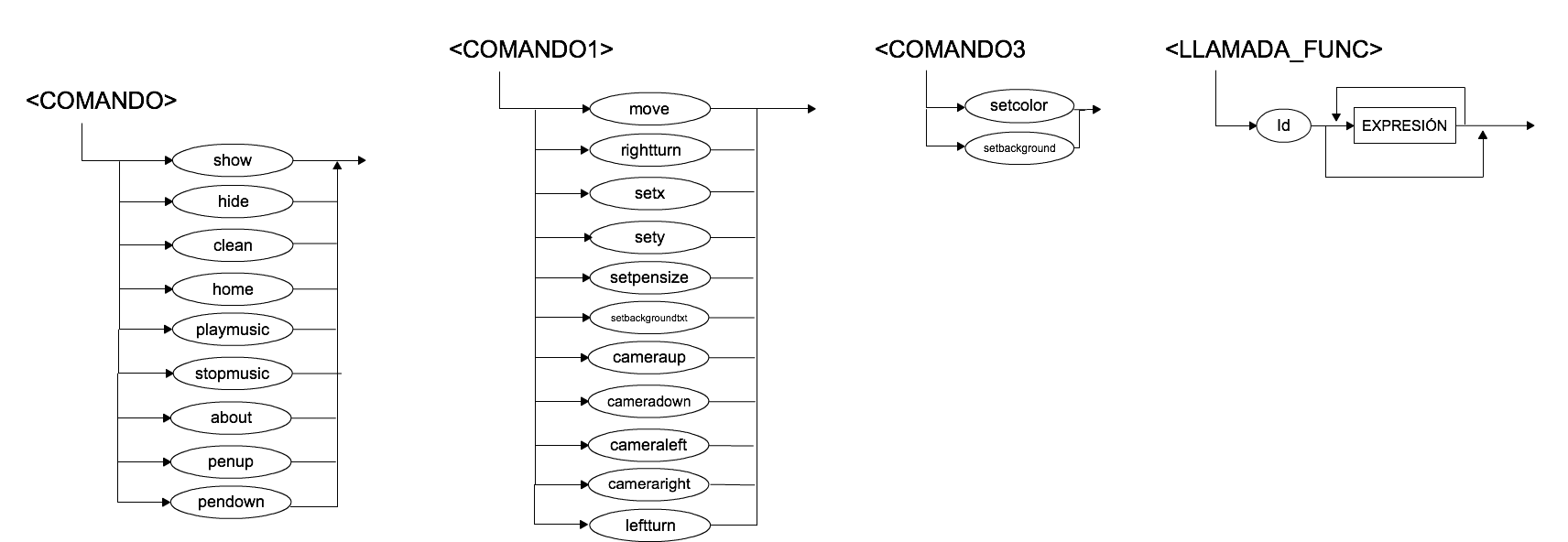
5000 : show  
5001 : hide  
5002 : clean  
5003 : home   
5004 : getcolorr  
5005 : getcolorg  
5006 : getcolorb  
5007 : getpensize  
5008 : getx  
5009 : gety  
5010 : stopmusic  
5011 : playmusic  
5012 : move  
5013 : about  
5014 : setx  
5015 : sety  
5016 : setpensize  
5017 : penup  
5018 : pendown  
5019 : rightturn  
5020 : leftturn  
5021 : setbackgroundtxt  
5022 : save  
5023 : setpos  
5024 : setcolor  
5025 : setbackground  
5026 : =  
5027 : >  
5028 : <  
5029 : !=  
5030 : >=  
5031 : <=  
5032 : +  
5033 : -  
5034 : \*  
5035 : /  
5036 : goto  
5037 : gotoF  
5038 : param  
5039 : era  
5040 : retorno función  
5041 : retorno lista  
5042 : gosub

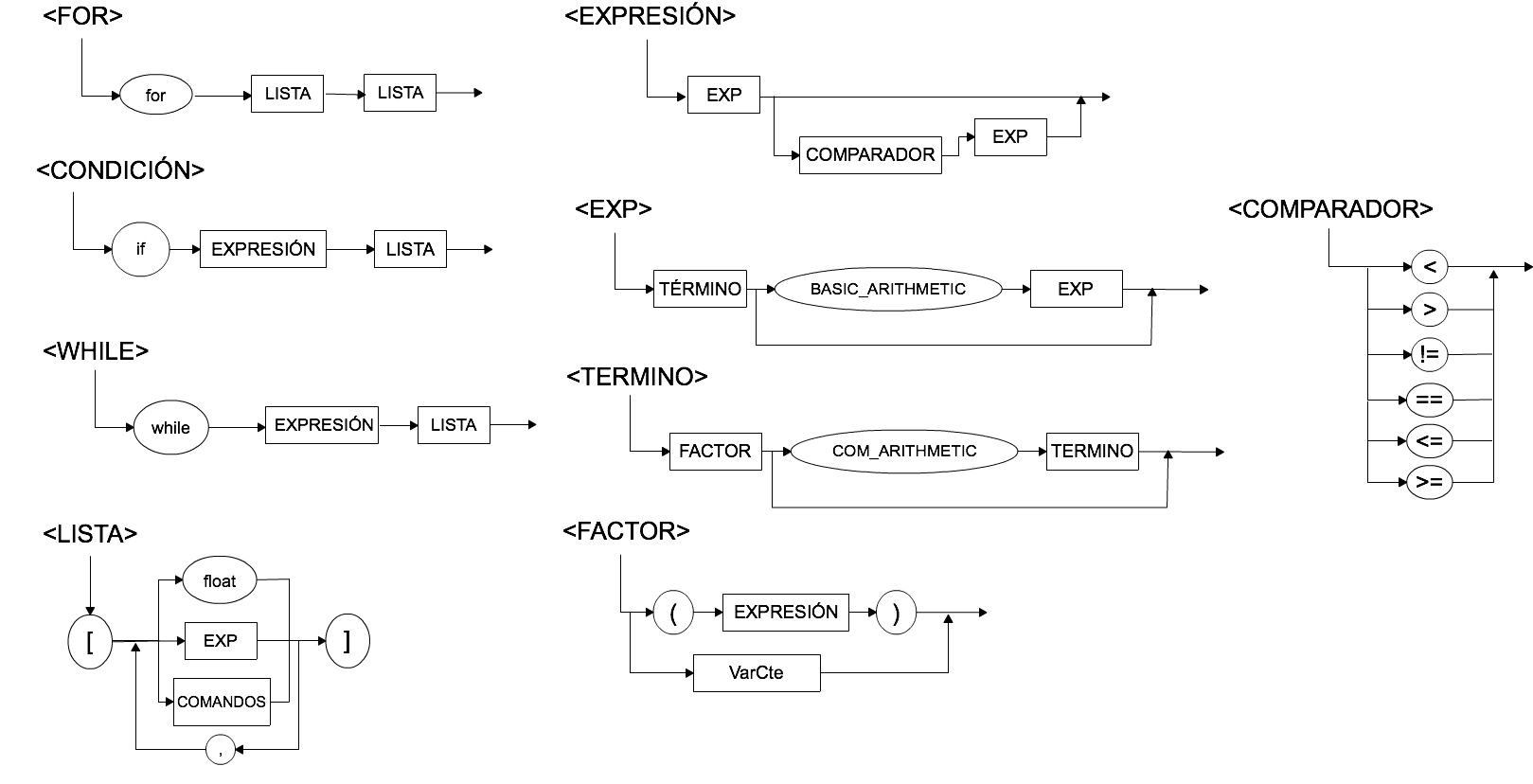
* + **Diagramas de sintaxis**

****

****

****

****

****

* + **Breve Descripción de cada una de las acciones semánticas y de código**
  + **Tabla de consideraciones semánticas**
* **Descripción detallada del proceso de Administración de Memoria usado en la compilación**

**d) Descripción de la Máquina Virtual**

* **Equipo de Computo**

La máquina virtual se desarrolló en computadoras portátiles con OS X Versión 10.9.1. Se utilizó además tanto para el compilador como para la máquina virtual la versión 4.2.1 de g++ además de la librería de gráficos OpenGL, la librería SOIL para la carga de texturas, la librería OpenAL para los sonidos y la librería WX Widgets para la implementación de la línea de comandos y la ejecución de programas.

* **Descripción detallada del proceso de Administración de Memoria en ejecución**

**e) Pruebas del Funcionamiento del Lenguaje**

* **Codificación de la prueba**

espiral.gbot

// Programa que dibuja una espiral en pantalla

program main

setbackgroundtxt 1.0 // Cambia a la primera textura

save i 10.0

for [i, 110.0, 10.0][

move i, // Mueve el robot una distancia i

rightturn 90.0 // Vuelta a la derecha

]

end

* **Resultados por el código intermedio y por la ejecución**

**CodigoInt.txt**

11

1. 5039 1 0 2

2. 5036 3

3. 5021 1000

4. 5022 1001 3000

5. 5028 3000 1002 2000

6. 5037 2000 12

7. 5012 3000

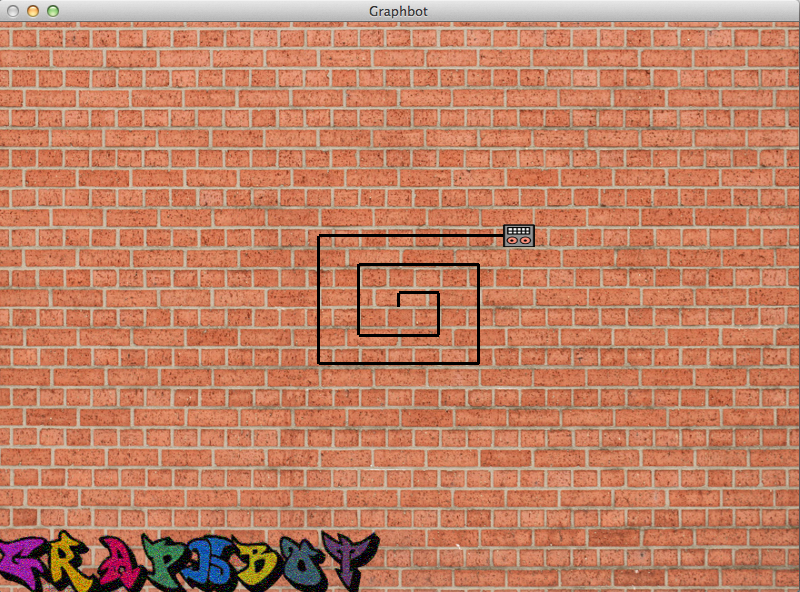
8. 5019 1003

9. 5032 1001 3000 2001

10. 5022 2001 3000

11. 5036 5

**Ejecución**

****

**f) Listados Documentados del Proyecto**

**Gráficas**

*move*

move distancia

*donde distancia es un float, una expresión aritmética o una variable tipo flotante*

mueve al robot hacia adelante una distancia especificada

*rightturn*

rightturn grados

*donde grados es un float, una expresión aritmética o una variable tipo flotante*

mueve al robot a la derecha de acuerdo al ángulo especificado medido en grados

*leftturn*

leftturn grados

*donde grados es un float, una expresión aritmética o una variable tipo flotante*

mueve al robot a la izquierda de acuerdo al ángulo especificado medido en grados

*setpos*

setpos X Y

*donde X y Y son un float, una expresión aritmética o una variable tipo flotante*

mueve al robot a una posición absoluta en el canvas, el comando recibe como entrada las coordenadas X y Y

*setx*

*setx coorx*

*donde coorx es un float, una expresión aritmética o una variable tipo flotante*

mueve al robot horizontalmente de su posición antigua a una nueva coordenada X

*sety*

*sety coory*

*donde coory es un float, una expresión aritmética o una variable tipo flotante*

mueve al robot verticalmente de su posición antigua a una nueva coordenada Y

*setcolor*

*setcolor r g b*

*donde r, g, y b son un float, una expresión aritmética o una variable tipo flotante*

establece un color a la pluma de acuerdo al color generado por los tres valores flotantes que le siguen, estos siendo valores entre 0.0 y 1.0

*setpensize*

*setpensize tamaño*

*donde tamaño es un float, una expresión aritmética o una variable tipo flotante*

establece el espesor de la pluma, la entrada es un número entero

*setbackground*

*setbackground r g b*

*donde r, g, y b son un float, una expresión aritmética o una variable tipo flotante*

pinta el fondo del canvas de acuerdo al color generado por los tres valores flotantes que le siguen, estos siendo valores entre 0.0 y 1.0

*setbackgroundtxt*

*setbackgroundtxt textura*

*donde textura es un float [1.0, 2.0, 3.0, 4.0]*

pinta el fondo del canvas de acuerdo a la textura elegida

*show*

vuelve al robot visible

*hide*

vuelve al robot invisible

*clean*

borra todo lo dibujado en el canvas

*home*

regresa al robot al centro del canvas

**Queries**

*getcolorr*

regresa en la línea de comandos el valor de R

*getcolorg*

regresa en la línea de comandos el valor de G

*getcolorb*

regresa en la línea de comandos el valor de B

*getpensize*

regresa en la línea de comandos el tamaño actual del espesor de la pluma

*getx*

regresa en la línea de comandos la coordenada x en la que se encuentra el robot

*gety*

regresa en la línea de comandos la coordenada y en la que se encuentra el robot

**Estructuras de Control**

*function*

*function nombrefunción parametros*

*comandos*

*end*

*donde parametros es una lista de ids, comandos es una lista de comandos o una variable del tipo lista de comandos*

genera una función local al programa

*save*

*save nombrevariable valor*

*donde valor es un float, una expresión aritmética, o una lista*

guarda una variable nueva con su respectivo nombre y valor

*for*

*for [var final incremento] instrucciones*

*donde instrucciones es una lista de comandos o una variable del tipo lista de comandos*

repite una serie de instrucciones de acuerdo al valor inicial establecido, el incremento y el valor límite

*if*

*if condición instrucción*

*donde instrucción es una lista de comandos o una variable del tipo lista de comandos*

ejecuta una instrucción si y solo si la condición establecida es verdadera

*while*

*while condición instrucciones*

*donde instrucciones es una lista de comandos o una variable del tipo lista de comandos*

ejecuta una instrucción continuamente siempre y cuando la condición establecida sea verdadera

**Misceláneas**

*playmusic*

*playmusic*

reproduce música durante la ejecución de la aplicación

*stopmusic*

*stopmusic*

detiene la música

*penup*

*penup*

levanta la pluma del canvas y permite no dibujar en el mismo mientras se ejecutan otras instrucciones

*pendown*

*pendown valor*

apunta la pluma al canvas y permite dibujar en el mismo mientras se ejecutan otras instrucciones

*about*

*about*

muestra información de los creadores de Graphbot