

**Diseño de Compiladores**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Martha Lisett Benavides Martínez José González Ayerdi

A01280115 A01136121

6 Marzo, 2017

Contenido

[1. Visión y Propósito 3](#_Toc476569429)

[2. Objetivo Principal del Lenguaje 3](#_Toc476569430)

[3. Requerimientos del Lenguaje 3](#_Toc476569431)

[3.1. Elementos Básicos de Léxico 3](#_Toc476569432)

[3.2. Diagramas de Sintaxis para todas las estructuras del Lenguaje 4](#_Toc476569433)

[3.2.1 Gramáticas 10](#_Toc476569434)

[3.3. Principales Características de Semántica 11](#_Toc476569435)

[3.4. Funciones Especiales del Lenguaje (Output) 12](#_Toc476569436)

[3.5. Tipos de Datos 12](#_Toc476569437)

[4. Lenguaje y Plataforma 12](#_Toc476569438)

[5. Bibliografía 12](#_Toc476569439)

# ***1. Visión y Propósito***

Hoy en día el uso de las Tecnologías de Información se encuentra en auge en distintas áreas de la vida diaria, tal como lo son los casos de grandes empresas de distintos giros, escuelas, aplicaciones web y móviles, así como videojuegos en 2D y 3D. Dado lo anterior, el propósito del presente proyecto es el de fomentar en jóvenes y adultos el aprendizaje de conceptos básicos de programación de manera sencilla y didáctica, en donde puedan tener la oportunidad de ver la perspectiva de desarrollo de los sistemas/programas que utilizan diariamente, de modo que obtengan una herramienta que les sirva como base para desarrollar sus propias ideas, que incluso, quizá en un futuro este aprendizaje sea suficiente para buscar estudiar una carrera universitaria o técnica relacionada con las Tecnologías de Información, en donde la creatividad de cada persona es el límite.

# ***2. Objetivo Principal del Lenguaje***

El principal objetivo de APODORAX es el producir, como ya se mencionó con anterioridad, un lenguaje sencillo de utilizar, en donde las personas puedan aprender de forma simplificada y didáctica la lógica de la programación que se encuentra detrás de todas las aplicaciones que usan, dando paso a que con un poco de esfuerzo ellos mismos puedan generar instrucciones que les permitan observar de manera gráfica el resultado de las mismas (output gráfico).

De igual manera, el lenguaje tendrá como característica principal que estará basado en español, de modo que sea más sencillo para las personas que no tienen conocimiento del idioma inglés el declarar variables, ciclos, condiciones y funciones, ya que como es conocido, la mayoría de los lenguajes de programación usados a nivel mundial se encuentran escritos en inglés, logrando intimidar a aquellas personas con un bajo entendimiento del idioma.

# ***3. Requerimientos del Lenguaje***

Debido al hecho de que se mencionó que APODORAX estará desarrollado en español, la estructura del mismo, así como sus palabras clave, será representada en el mismo idioma.

## ***3.1. Elementos Básicos de Léxico***

Dentro de esta sección se establecen los Tokens a utilizar dentro del lenguaje, tal y como lo son las palabras reservadas, identificadores, operadores aritméticos y delimitadores.

|  |  |  |  |
| --- | --- | --- | --- |
| **Palabras Reservadas** | | | |
| inicio | fin | funcion | var |
| && | || | entero | flotante |
| cadena | bool | caracter | programa |
| mientras | verdadero | falso | vacio |
| desplegar | regresar | negro | gris |
| azul | amarillo | verde | rojo |
| insertaTexto | insertaRectangulo | insertaCirculo | insertaOvalo |
| insertaTriangulo | insertaPunto | insertaLinea | insertaCurva |
| no | entrada | c\_entero | c\_flotante |
| c\_cadena | c\_bool | c\_caracter | entonces |
| si | sino |  |  |

|  |  |
| --- | --- |
| **Operadores** | == , != , +, -, /, \*, > ,<, <=, >= |
| **Delimitadores** | ( , ) , { , } , [ , ] , ; , : , , , “” |
| **Identificadores** | Comienzan con una letra mayúscula, seguido de más letras minúsculas o números. |

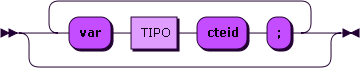
Asimismo, se utilizarán arreglos de una dimensión como un elemento estructurado.

## ***3.2. Diagramas de Sintaxis para todas las estructuras del Lenguaje***

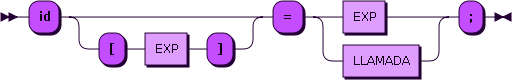
**<PROGRAMA>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCacheContent.Word\PROGRAMA.PNG

**<DECLARACION>**



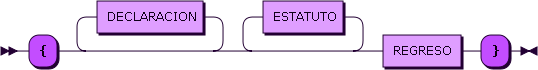
**<ASIGNACION>**



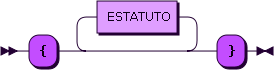
**<FUNCION>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\FUNCTION.PNG

**<BLOQUEFUN>**



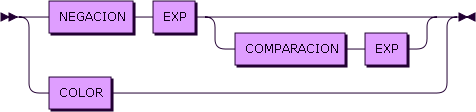
**<BLOQUE>**



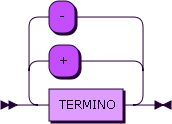
**<ESTATUTO>**



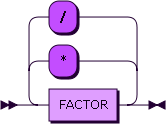
**<EXPRESION>**



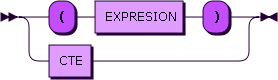
**<EXP>**



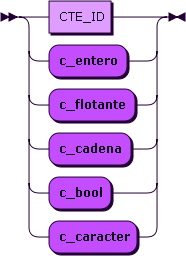
**<TERMINO>**



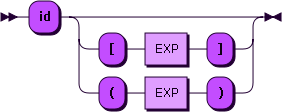
**<FACTOR>**



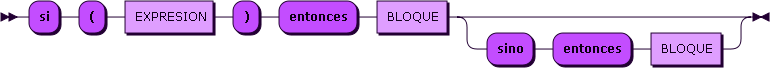
**<CTE>**



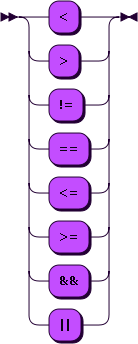
**<CTE\_ID>**



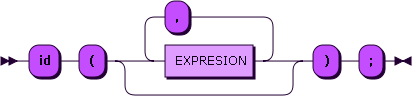
**<CONDICION>**



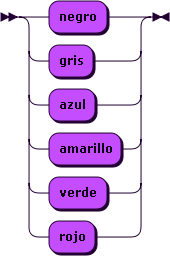
**<COMPARACION>**



**<LLAMADA>**



**<COLOR>**



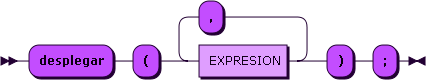
**<CICLO>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\CICLO.PNG

**<NEGACION>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\NEGACION.PNG

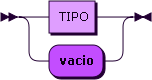
**<ESCRITURA>**



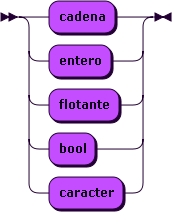
**<REGRESO>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\REGRESO.PNG

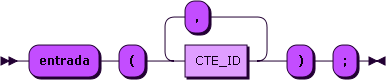
**<TIPO\_REGRESO>**



**<TIPO>**



**<INGRESO>**



**<TEXTO>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\TEXTO.PNG

**<RECTANGULO>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\RECTANGULO.PNG

**<TRIANGULO>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\TRIANGULO.PNG

**<CIRCULO>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\CIRCULO.PNG

**<OVALO>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\OVALO.PNG

**<PUNTO>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\PUNTO.PNG

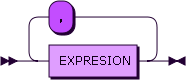
**<LINEA>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\LINEA.PNG

**<CURVA>**

C:\Users\César Benavides\AppData\Local\Microsoft\Windows\INetCache\Content.Word\CURVA.PNG

**<ARGS>**



### **3.2.1 Gramáticas**

PROGRAMA ::= 'programa' 'id' ':' DECLARACION FUNCION 'inicio' BLOQUE 'fin'

DECLARACION ::= ('var' TIPO 'cteid' ';')\*

ASIGNACION ::= ('id' ('[' EXP ']' |))'=' (EXP | LLAMADA) ';'

FUNCTION ::= 'funcion' TIPO\_REGRESO 'id' '(' FUNCTIONPAM ')' BLOQUEFUN

FUNCTIONPAM ::= ('var' TIPO 'id' (',' FUNCTIONPAM)\* |)

BLOQUEFUN ::= '{' DECLARACION\* ESTATUTO\* REGRESO '}'

BLOQUE ::= '{' ESTATUTO\* '}'

ESTATUTO ::= INGRESO | ESCRITURA | TEXTO | RECTANGULO | CIRCULO | OVALO | TRIANGULO | PUNTO | LINEA | CURVA | LLAMADA ';' | ASIGNACION | CONDICION | CICLO

NEGACION ::= ('no')?

EXPRESION ::= NEGACION EXP | NEGACION EXP COMPARACION EXP | COLOR

EXP ::= TERMINO | TERMINO '+' EXP | TERMINO '-' EXP

TERMINO ::= FACTOR | FACTOR '\*' TERMINO | FACTOR '/' TERMINO

FACTOR ::= '(' EXPRESION ')' | CTE

CTE ::= CTE\_ID | 'c\_entero' | 'c\_flotante' | 'c\_cadena' | 'c\_bool' | 'c\_caracter'

CTE\_ID ::= 'id' | 'id' '[' EXP ']'| 'id' '(' EXP ')'

CONDICION ::= 'si' '(' EXPRESION ')' 'entonces' BLOQUE ( 'sino' 'entonces' BLOQUE | )

COMPARACION ::= '<' | '>' | '!=' | '==' | '<=' | '>=' | '&&' | '||'

PARAMETROS ::= 'var' TIPO 'id' | 'var' TIPO 'id' ',' PARAMETROS

LLAMADA ::= 'id' '(' (LLAMADA2 |) ')' ';'

LLAMADA2 ::= EXPRESION | EXPRESION ',' LLAMADA2

COLOR ::= 'negro' | 'gris' | 'azul' | 'amarillo' | 'verde' | 'rojo'

CICLO ::= 'mientras' '('EXPRESION ')' BLOQUE

ESCRITURA ::= 'desplegar' '(' ESCRITURA2 ')' ';'

ESCRITURA2::= EXPRESION | EXPRESION ',' ESCRITURA2

REGRESO ::= ('regresar' EXP ';' | )

TIPO\_REGRESO ::= TIPO | 'vacio'

TIPO ::= 'cadena' | 'entero' | 'flotante' | 'bool' | 'caracter'

INGRESO ::= 'entrada' '(' INGRESO2 ')' ';'

INGRESO2 ::= CTE\_ID | CTE\_ID ',' INGRESO2

ARGS ::= EXPRESION | EXPRESION ',' ARGS

TEXTO ::= 'insertaTexto' '(' ARGS ')' ';'

RECTANGULO ::= 'insertaRectangulo' '(' ARGS ')' ';'

CIRCULO ::= 'insertaCirculo' '(' ARGS ')' ';'

OVALO ::= 'insertaOvalo' '(' ARGS ')' ';'

TRIANGULO ::= 'insertaTriangulo' '(' ARGS ')' ';'

PUNTO ::= 'insertaPunto' '(' ARGS ')' ';'

LINEA ::= 'insertaLinea' '('ARGS ')' ';'

CURVA ::= 'insertaCurva' '(' ARGS ')' ';'

## ***3.3. Principales Características de Semántica***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operadores Aritméticos (+,-,\*,/)** | | | | |
|  | *cadena* | *flotante* | *entero* | *bool* |
| *cadena* | X | X | X | X |
| *flotante* | X | flotante | flotante | X |
| *entero* | X | flotante | entero | X |
| *bool* | X | X | X | X |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operadores Relacionales (==,!=,<,>)** | | | | |
|  | *cadena* | *flotante* | *entero* | *bool* |
| *cadena* | bool | x | x | x |
| *flotante* | x | bool | bool | x |
| *entero* | x | bool | bool | x |
| *bool* | x | x | x | bool |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operadores Booleanos (y, o, no)** | | | | |
|  | *cadena* | *flotante* | *entero* | *bool* |
| *cadena* | x | x | x | x |
| *flotante* | x | x | x | x |
| *entero* | x | x | x | x |
| *bool* | x | x | x | bool |

## ***3.4. Funciones Especiales del Lenguaje (Output)***

Con ayuda de la librería de graphics.py, se podrán implementar las siguientes funciones para realizar el despliegue gráfico de las siguientes figuras:

* insertaLinea(coordXinicial, coordYinicial, coordXfinal, coordYfinal, color, grosor)
* insertaTexto(coordX, coordY, palabras, color)
* insertaPunto(coordX, coordY, color)
* insertaTriangulo(coordX, coordY, coordXm, coordYm, coordXf, coordYf, color, grosor)
* insertaCirculo(coordX, coordY, radio, color, grosor)
* insertaOvalo(coordXinicial, coordYinicial, coordXfinal, coordYfinal, color, grosor)
* insertaRectangulo(coordXinicial, coordYinicial, coordXfinal, coordYfinal, color, grosor)
* insertaCurva(coordXinicial, coordYinicial, coordXintermedia, coordYintermedia, coordXfinal, coordYfinal, grosor): Las coordenadas iniciales y las intermedias formarán una línea, mientras que las coordenadas intermedias y las finales formarán otra, logrando tener dos líneas pegadas.

Cabe destacar que, de acuerdo a la figura o método, será la cantidad de coordenadas y parámetros necesarios para hacer el despliegue de los mismos.

## ***3.5. Tipos de Datos***

Como se especificó en las secciones de Léxico y de Sintaxis, los tipos de datos a utilizar en el lenguaje serán cadena, flotante IEEE 754 a precisión simple, entero (–2.147.483.648 a 2.147.483.647) y bool (verdadero / falso).

# ***4. Lenguaje y Plataforma***

El lenguaje a utilizar para el desarrollo del lenguaje será Python 2.7, mientras que los Sistemas Operativos de las computadoras a usar para la implementación del mismo serán Windows 10 y Ubuntu Linux.

Por otro lado, se utilizará Github para el control de versiones y la herramienta PLY como analizador de léxico y de sintaxis.

De igual manera, se tomará la librería graphics.py como base para el desarrollo del output gráfico, con la cual se pueden mostrar líneas, círculos, puntos, entre otras figuras, de manera gráfica.

# ***5. Bibliografía***

Zelle, J. (2016). Librería graphics.py. Wartburg. Recuperado de: <http://mcsp.wartburg.edu/zelle/python/graphics.py>

Zelle, J. (2016). Librería graphics.py. Wartburg. Recuperado de: <http://mcsp.wartburg.edu/zelle/python/graphics/graphics/node2.html>

Microsoft. (2016). Intervalos de Tipos de Datos. Microsoft. Recuperado de: <https://msdn.microsoft.com/es-es/library/s3f49ktz.aspx>

Rademacher, G. (2017). Railroad Diagram Generator. Bottle Caps. Recuperado de: <http://bottlecaps.de/rr/ui>