



**Tecnológico
de Monterrey**

Diseño de compiladores

Documentación Final
MeMyself

Héctor Noel León Quiroz
A01251806

25 de noviembre 2020

Descripción del proyecto

Propósito

Este proyecto fue hecho como trabajo final para la materia de Desarrollo de Compiladores con el objetivo de demostrar lo aprendido en el transcurso de clase al desarrollar desde cero un lenguaje de programación de alto nivel tales como C++, Python, Java etc.

Objetivo

El objetivo deseado es construir un lenguaje de programación que sea capaz de tener un output gráfico, iniciando con especificar la gramática y la sintaxis hasta llegar al desarrollo de la máquina virtual la cual se encarga de ejecutar el código.

Alcance del proyecto

El lenguaje debe de ser capaz de manejar:

- Estatutos de asignación, condiciones, ciclos, lectura y escritura.
- Expresiones Matemáticas: Aritmética, Lógica y Relacional.
- Módulos, incluyendo parámetros y variables locales y globales.
- Elementos estructurados: Arreglos (una y dos dimensiones), o, listas, etc.

Análisis de requerimientos

Requerimientos Funcionales

- RF01: El lenguaje permitirá declarar variables globales y locales en módulos (funciones), especificando que tipo de variable es al momento que se crean (int, float o char)
- RF02: El lenguaje permitirá declarar variables de arreglos de 1 o 2 dimensiones, estas pueden ser globales o locales en módulos (funciones), especificando que tipo de variable es al momento que se crean (int, float o char)
- RF03: Los arreglos deben ser inicializados al momento de ser declarados con un solo valor int el cual determinará su tamaño.

- RF04: El lenguaje permitirá declarar módulos (funciones), los cuales se tiene que especificar el tipo de retorno ya sea int, float, char o void.
- RF05: El lenguaje permitirá hacer llamadas a módulos (funciones), previamente declarados.
- RF06: Las llamadas a módulos (funciones) deberán tener el número exacto de parámetros con los que fueron declarados o de lo contrario se mostrará un error.
- RF07: Las llamadas a módulos (funciones) de tipo void no podrán ser asignadas a variables, de lo contrario se mostrará un error.
- RF08: Se deberá utilizar una expresión de tipo int para poder acceder los valores de un arreglo.
- RF0: Para acceder a un arreglo se necesita un número mayor igual a 0 y menor al tamaño del arreglo, de lo contrario se mostrará un mensaje de error correspondiente.
- RF09: El lenguaje permitirá mostrar información en la consola por medio del estatuto write.
- RF10: El lenguaje permitirá obtener información mediante la consola por medio del estatuto read.
- RF11: El lenguaje permitirá utilizar llevar a cabo condiciones por medio de los estatutos if y else.
- RF12: El lenguaje permitirá hacer ciclos por medio de los estatutos while y from-to.
- RF13: El lenguaje permitirá regresar un canvas con elementos gráficos con los estatutos Line, Point, Circle, Arc, Penup, Pendown, Color, Size y Clear.

Requerimientos No Funcionales

- RNF01: El lenguaje deberá ser fácil y sencillo de utilizar para nuevos usuarios.
- RNF02: El tiempo de compilación del lenguaje deberá ser aceptable.
- RNF03: El lenguaje deberá mostrar mensajes de error adecuados para los casos donde se hayan cometidos fallas en el desarrollo de código.

Caso de Uso	Hello world
ID	UC01
Actores	User
Descripción	Se tecleara un programa el cual es capaz de imprimir "Hello world"

Precondiciones	Tener los archivos necesarios para correr el lenguaje MeMyself
Flujo de eventos	<ol style="list-style-type: none"> 1. Crear un nuevo archivo 2. Teclear algo parecido al siguiente código: Program UC1; <pre>main(){ write("Hello world"); }</pre> 3. Ejecutar tu archivo de la siguiente forma py MeMyself.py {AQUI EL NOMBRE DEL ARCHIVO}
Postcondiciones	Se mostrará en la consola el mensaje Hello world

Caso de Uso	Al cuadrado
ID	UC01
Actores	User
Descripción	Se tecleara un programa el cual es capaz de calcular e imprimir en pantalla la potencia por 2 de 1 al número deseado
Precondiciones	Tener los archivos necesarios para correr el lenguaje MeMyself
Flujo de eventos	<ol style="list-style-type: none"> 4. Crear un nuevo archivo 5. Teclear algo parecido al siguiente código: Program UC2; var int: x, i; <pre>main(){ write("Ingresa un numero"); read(x); from i=1 to x+1 do{ write("El cuadrado de ", i, " es ", i*i); } }</pre> 6. Ejecutar tu archivo de la siguiente forma py MeMyself.py {AQUI EL NOMBRE DEL ARCHIVO}
Postcondiciones	Se mostrará en la consola el todos los cuadrados del 1 al número

	deseado
--	---------

Caso de Uso	¿Quieres continuar?
ID	UC01
Actores	User
Descripción	Se tecleara un programa el cual es capaz de calcular e imprimir en pantalla la potencia por 2 de 1 al número deseado
Precondiciones	Tener los archivos necesarios para correr el lenguaje MeMyself
Flujo de eventos	<p>7. Crear un nuevo archivo</p> <p>8. Teclear algo parecido al siguiente código:</p> <pre> Program UC3; var int: x, i; char: a; module void continuar(char c); { if(c == 'Y') then{ write("Continua!"); } else { write("Adios!"); } } main(){ write("Quires proceder? Y/N"); read(a); continuar(a); } </pre> <p>9. Ejectuar tu archivo de la siguiente forma py MeMyself.py {AQUI EL NOMBRE DEL ARCHIVO}</p>
Postcondiciones	Se mostrará en la consola lo correspondiente a lo tecleado

