Tecnológico de Monterrey

Campus Monterrey

Compiladores

TC 3040

**MARFLE**

Autores

Rubén Alejandro Valdez Ruiz Ana Eline Guajardo Mozo  
A00803015 A01033641  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8 de mayo de 2013

[IMAGEN]

**Índice**

1. Descripción del proyecto…………………………....................................................................
   1. Visión, Objetivos y Alcance del proyecto……………………………………………………………
   2. Análisis de Requerimientos y Casos de Uso Generales………………………………………
   3. Descripción de los principales Test Cases………………………………………………………….
   4. Descripción del Proceso…………………………………………………………………………………….
2. Descripción del Lenguaje…………………………………………………………………………………………..
   1. Nombre del lenguaje………………………………………………………………………………………….
   2. Descripción genérica…………………………………………………………………………………………..
   3. Descripción de errores………………………………………………………………………………………..
3. Descripción del Compilador………………………………………………………………………………………..
4. Descripción de la Máquina Virtual………………………………………………………………………………
   1. General………………………………………………………….……………………………………………………
   2. Arquitectura………………………………………………………….……………………………………………
5. Pruebas del Funcionamiento……………………………………………………………………………………..
6. Listados del Proyecto………………………………………………………….………………………………………

Documentación

1. Descripción del proyecto
   1. Visión, Objetivos y Alcance del Proyecto
      1. **Visión**

Para el año 2020, se busca que el lenguaje esté al nivel de Visual Basic .NET™ y Python™ como el lenguaje ideal para un principiante, es decir, que sea la primera opción para aprender a programar.

* + 1. **Objetivo del lenguaje**

Se busca que el lenguaje Marfle sea un buen comienzo para aquellos que están empezando a programar en lenguajes orientado a objetos, como es el caso con Python, para que cuando el usuario se decida por aprender lenguajes más avanzados como C++ ya tenga un conocimiento básico de cómo es programar, y que no lo sienta tan abrumador.

* + 1. **Alcance**
  1. Análisis de Requerimientos y Casos de Uso Generales
     1. **Componentes del léxico**

Las palabras reservadas en el lenguaje son:

|  |  |  |  |
| --- | --- | --- | --- |
| Program | If | Else | For |
| While | Do | Int | Float |
| String | Char | Print | Object |
| Read | Main | Bool | void |

Operadores dentro del lenguaje.

|  |  |
| --- | --- |
| +, -,\*, / | Aritméticos |
| <, >, >=, <=, !=, == | Comparación |
| &&, ||, ! | Lógicos |
| = | Asignación |
| - | Negativo |

Expresiones regulares para instancia múltiple.

**Identificadores:** [a-z][A-Za-z0-9\_]\*

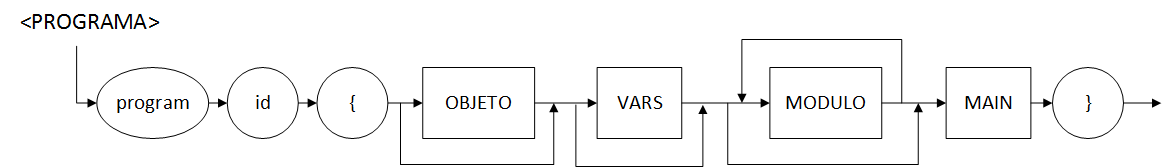
**Constantes enteras:** [0-9]+

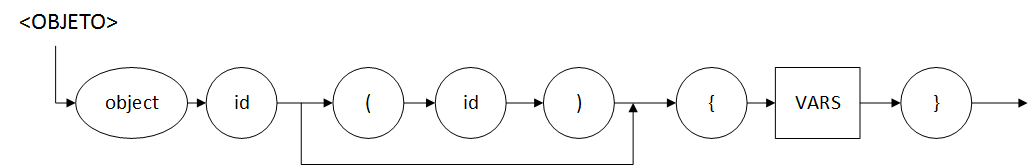
**Constantes flotantes:** [0-9]+.[0-9]+

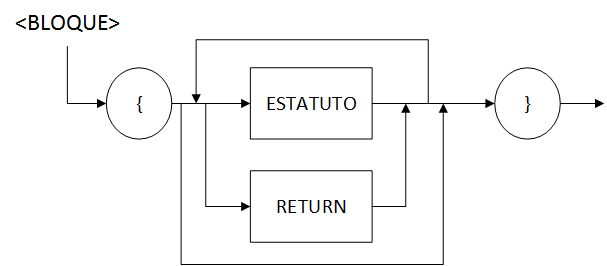
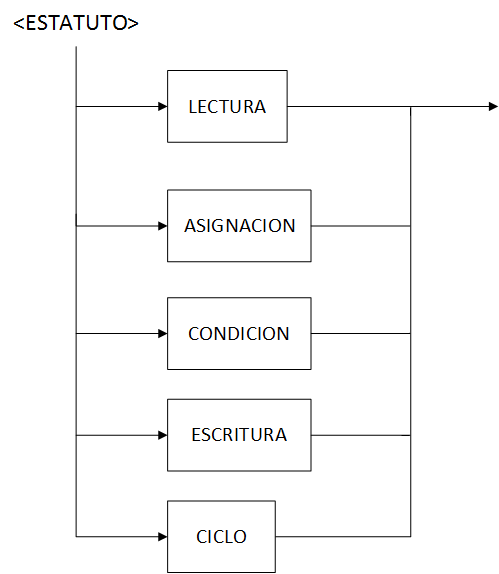
**Constante string:** \”(Identificadores | Const. Entera | Const. Flotante)\*\”

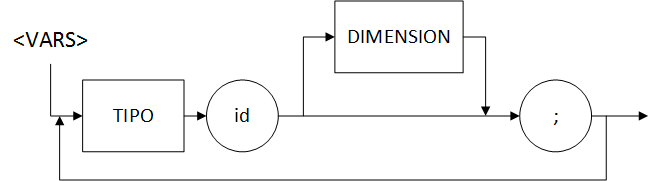
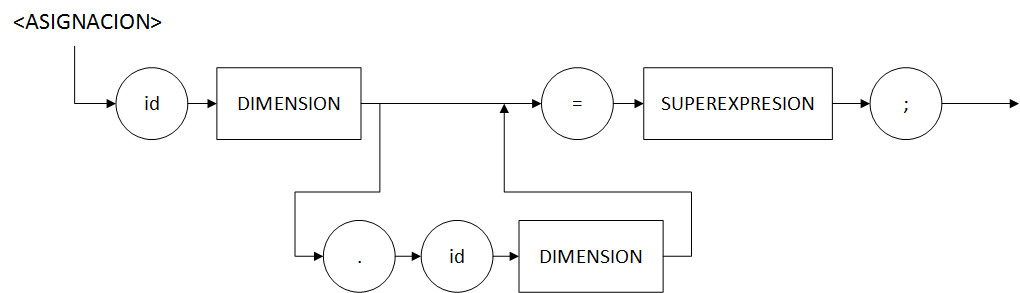
**Constante booleana:** (true | false)

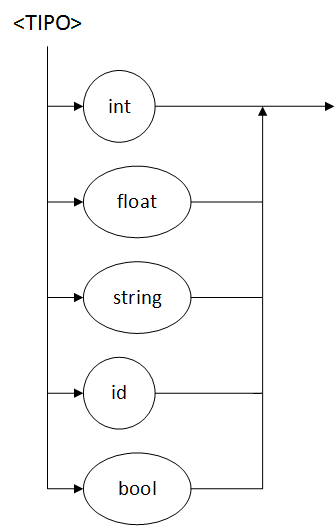
* + 1. **Diagramas de sintaxis**

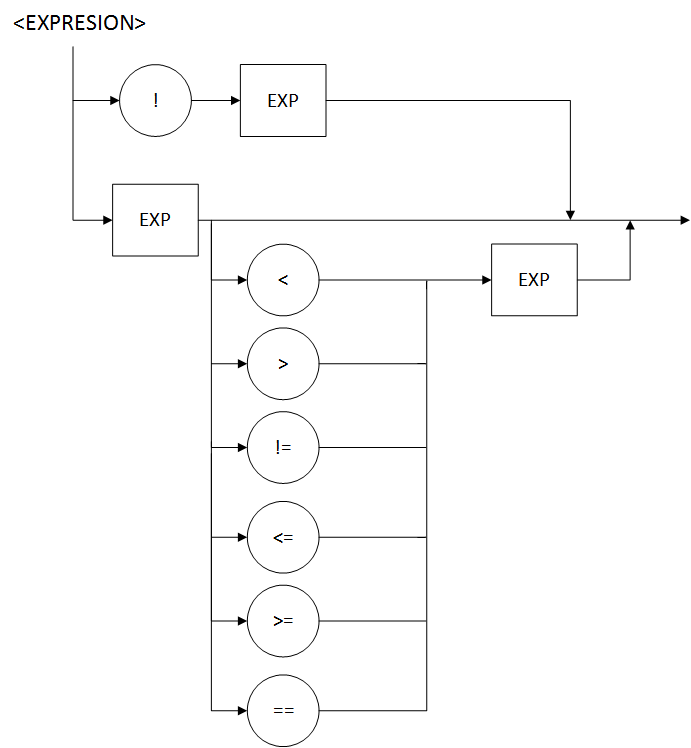
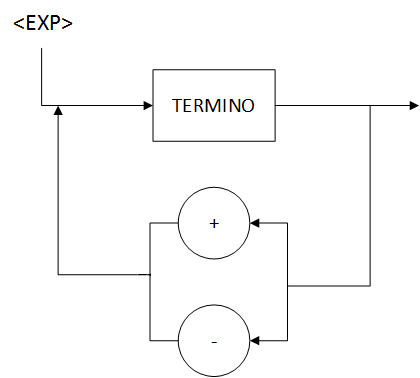


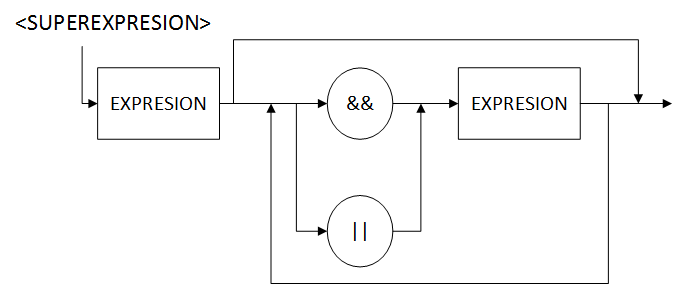


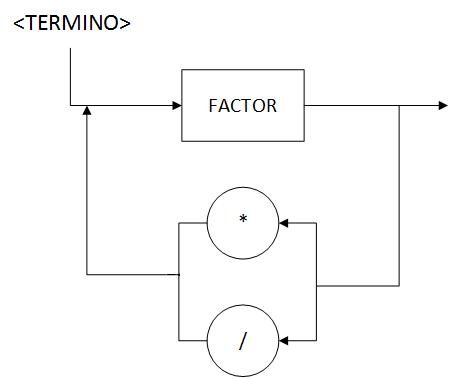
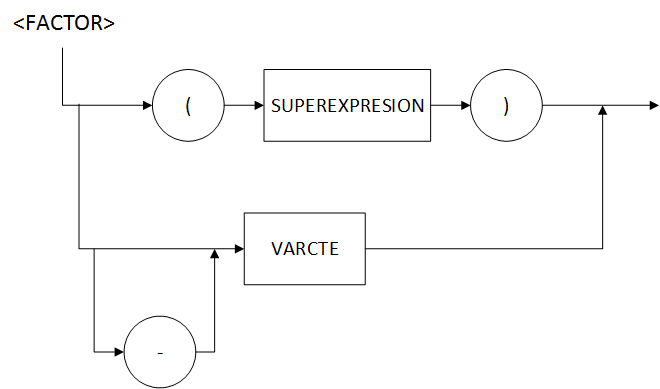


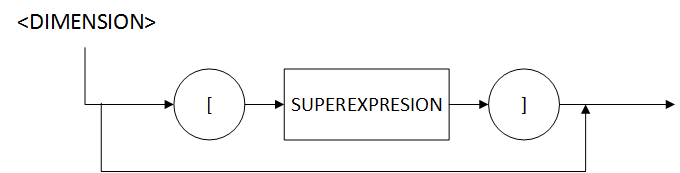


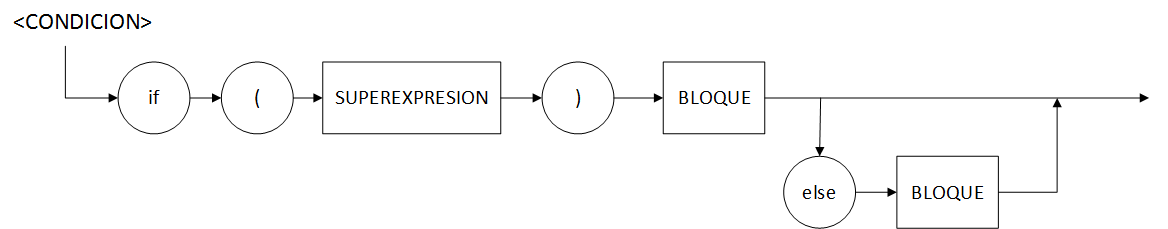


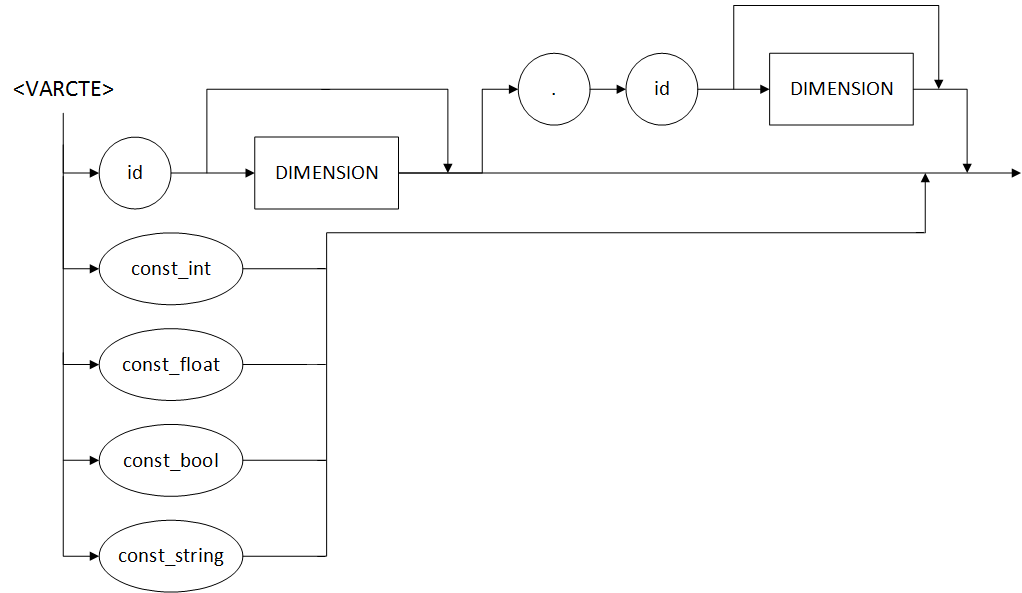


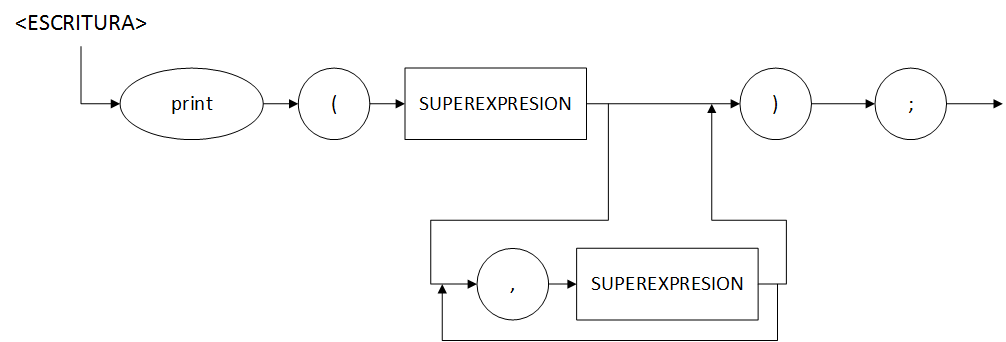


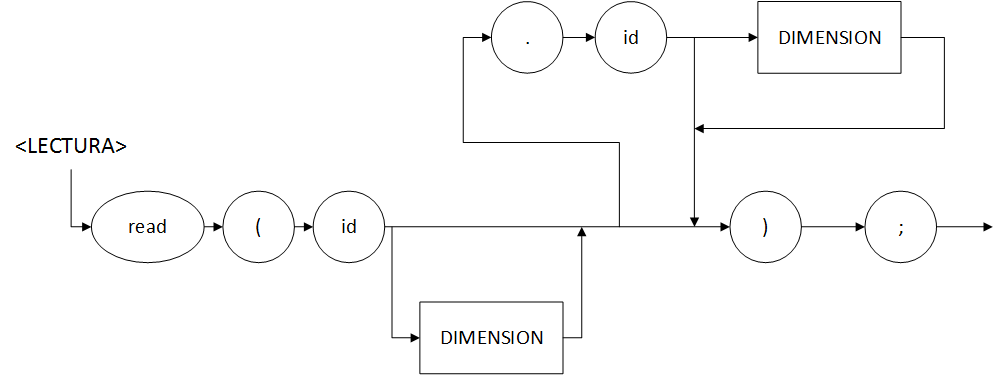


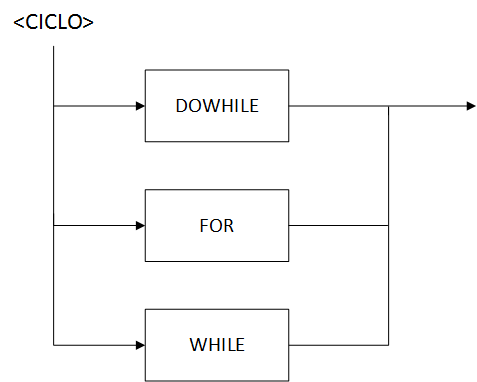


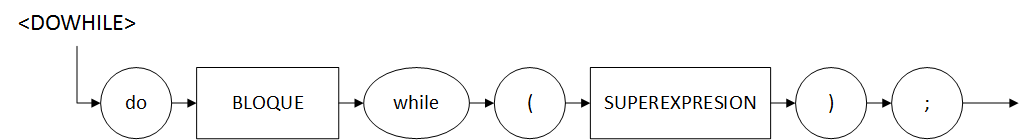


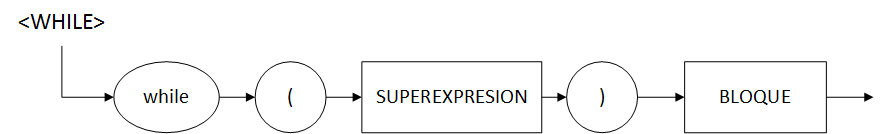


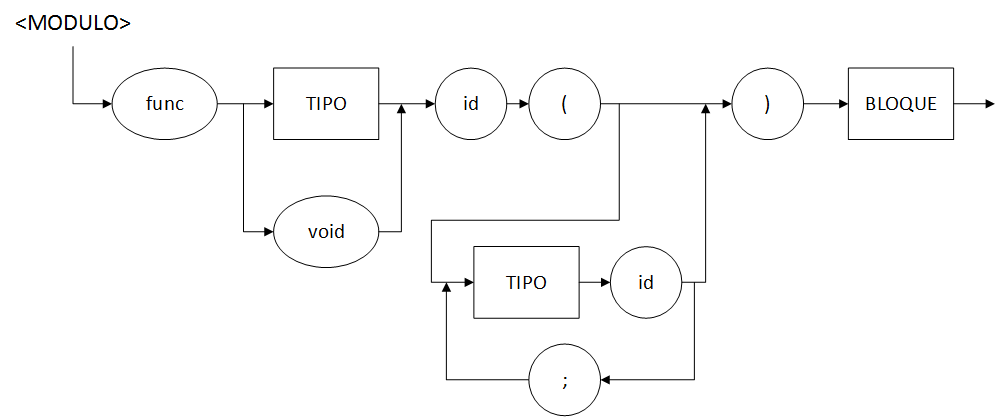


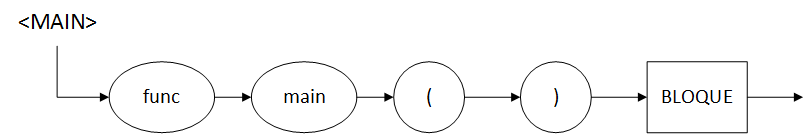


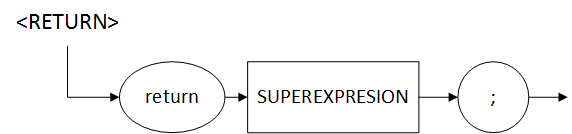


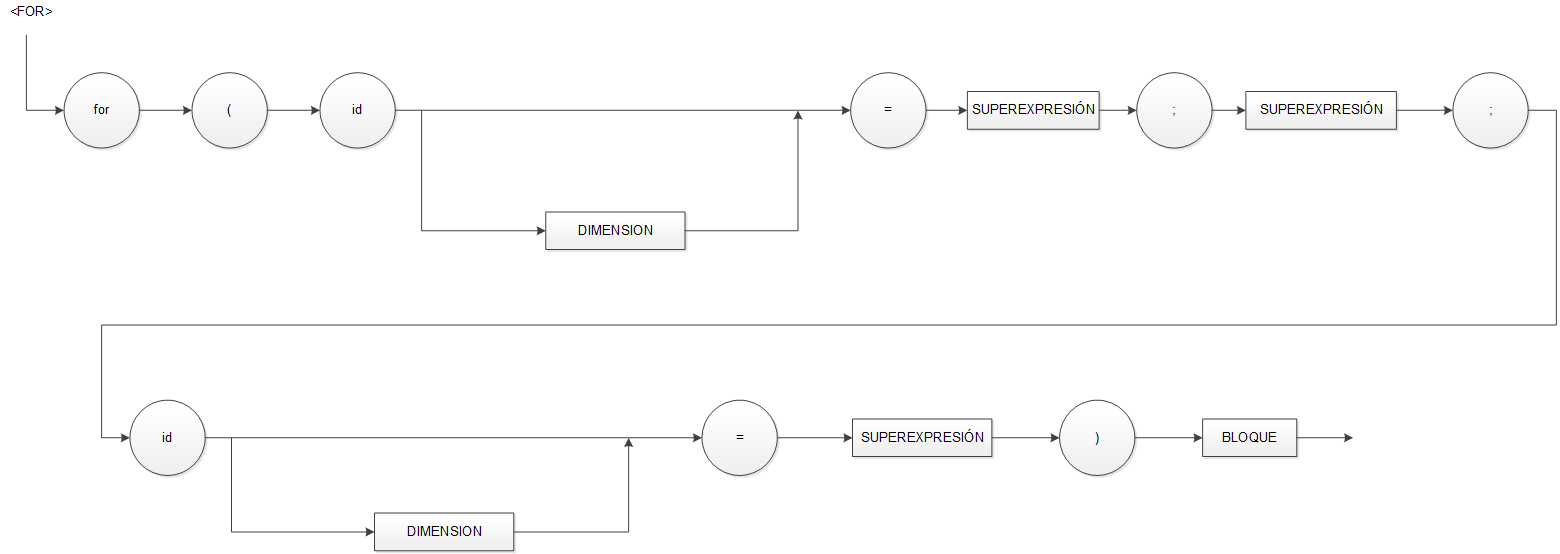












* 1. Descripción de los principales *Test Cases*
  2. Descripción del proceso

Se utilizó la herramienta GitHub para el manejo de versiones del compilador. Asimismo, se mantuvo una bitácora que se actualizaba al menos una vez por semana, en la que mencionaba en cada nueva entrada una breve descripción de cambios o avances significativos que se hayan hecho en el código.

[Reflexiones]

1. Descripción del Lenguaje
   1. Nombre del lenguaje

Marfle

* 1. Descripción genérica

Marfle es un Lenguaje Orientado a Objetos que cuenta con los requerimientos básicos de cualquier lenguaje OO; se pueden hacer operaciones aritméticas, manejar objetos, crear métodos, etc.

* 1. Descripción de errores

1. Descripción del Compilador

El compilador fue escrito en Python, utilizando PLY, una implementación de lex y yacc para Python.

1. Descripción de la Máquina Virtual
   1. General

En su mayoría, la máquina virtual fue escrita en Linux, utilizando Python.

* 1. Arquitectura

1. Pruebas del Funcionamiento del Lenguaje
2. Listados del Proyecto