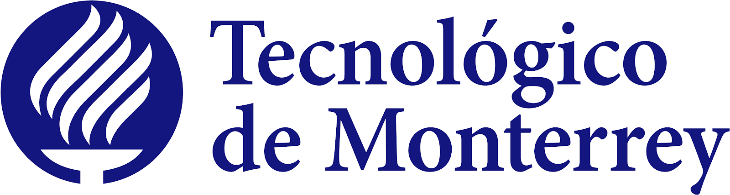
**Instituto Tecnológico y de Estudios Superiores**

**de Monterrey, Campus Monterrey**

****

**Diseño de Compiladores**

**Profesores**

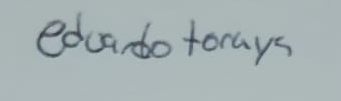
**Elda Quiroga González**

**Héctor Ceballos Cancino**

***Documentación Final***

***Forever Alone***

**Eduardo Alejandro Toraya Solís A00819785**



**Fecha de entrega: 2 de Junio del 2020.**

**Índice**

**a) DESCRIPCIÓN DEL PROYECTO**

**a.1) Propósito, Objetivos y Alcance del Proyecto.**

**El Propósito de este proyecto es desarrollar un compilador en Python utilizando como herramienta lexx and yacc para parsear de tal forma que cumpla con los requerimientos funcionales y de diseño propuestos por el documento de ForeverAlone.**

**a.2) Análisis de Requerimientos y Casos de Uso generales.**

**Requerimientos:**

* **Debe manejar tipos de dato int, float y carácter.**
* **Debe contar con una función principal y funciones adicionales opcionales.**
* **Debe poder declarar variables, obtener variables del usuario y mostrar variables al usuario.**
* **Debe manejar arreglos de una dimensión en sus operaciones.**
* **Debe contar con operaciones aritméticas lineales.**
* **Debe contar con llamadas a funciones.**
* **Debe contar con manejo de estatutos cíclicos y condicionales.**
* **Maneja expresiones como +, -, \*, /, &, |, <, <= , == , <>, >=, y > para floats e integers y para caracteres sólo las lógicas.**
* **Las funciones podrán ser de tipo void(no regresan valor) o de tipos carácter, float o integer que regresarán valor de su mismo tipo.**
* **Permite recursividad**

**Casos de uso generales:**

* **Definir/Ejecutar programa.**
* **Declarar variables individuales o de arreglo.**
* **Llamar variables individuales o de arreglo**
* **Iniciar/terminar estatutos no secuenciales (while, for, if).**
* **Llamar a funciones para su ejecución.**
* **Asignar valores.**
* **Mostar/obtener valores para usuario**
* **Escribir comentarios.**
* **Escribir strings.**
* **Iniciar o detener recursión.**

**a.3) Descripción de los principales test cases.**

**T.C.1: En el testCase del file MVTestOperaciones, se prueba la consistencia de las operaciones de suma, resta, multiplicación, operadores de comparación, asignación, lectura de valor de usuario y escritura.**

**T.C.2: MVTestCiclos, se prueba la consistencia de operaciones no lineales. Se maneja el while, el if/else, el for y estos mismos anidados. Mientras se realizan operaciones aritméticas, escritura y asignación.**

**T.C.3: MVTestFunciones, se prueba la consistencia y el correcto funcionamiento de la memoria en funciones void, con retorno y recursivas, así como llamadas recursivas y con llamada interna para confirmar el no colisionamiento de espacio de dirección. Este test case incluye la producción de un factorial y de la serie de Fibonacci hasta un valor dado por el usuario.**

**T.C.4: MVTestArrays, se prueba la consistencia de asignación de memoria de los arreglos, en el cual se prueba la asignación, lectura y operaciones aritméticas correctas de los arreglos. Se revisa el funcionamiento de arreglos anidados ejemplo: A[A[3]] y se prueba un sorteo de un arreglo definido en el mismo código combinando con características de ciclos.**

**a.4) Descripcion del PROCESO.**

**El proceso para llevar a cabo este compilador, fue el desarrollo por partes de acuerdo a los avances del mismo. Se comenzó con escoger una propuesta de proyecto de las ofrecidas por los profesores, en este caso se desarrolló el lexer, de ahí el código para operaciones lineales, seguido de operaciones no lineales, seguido de funciones y finalmente se desarrollaron los arreglos en el parser. Posteriormente, se desarrollo una máquina virtual que pudiese leer los cuadruplos generados en un texto objeto por el parser y los interpreta en Python.**

**Reflexión principal, Se tiene un mejor entendimiento del proceso interno que se puede llevar para poder traducir el código de un programador en instrucciones realizables por una computadora.**

**Reporte Entrega 13 Abril**

**Entrega del analizador lexico y sintactico**

**Issues a reparar para siguiente entrega:**

**I1.1 No indenfitica errores en específico.**

**I1.2 Faltaron pruebas extensivas.**

**I1.3 Se duda de que la gramática sea sólida en un 100%**

**Reporte Entrega 20 Abril**

**Entrega del analizador lexico y sintactico de cambio de proyecto a ForeverAlone**

**Diagramas de sintaxis: https://www.lucidchart.com/documents/edit/5059454c-35d8-4bb4-8807-630eae053432/0\_0**

**Issues a reparar para siguiente entrega:**

**I1.1 No indenfitica errores en específico.**

**I1.2 Faltaron pruebas extensivas.**

**I1.3 Atraso total entrega 2, directorio de procedimientos y tabla de variables.**

**Reporte Entrega 27**

**Entrega del cubo semántico y otros.**

**Issues a reparar para siguiente entrega:**

**I1.1 No indenfitica errores en específico.**

**I1.2 Atraso parcial entrega 3 -> falto generar cuádruplos con estatutos lineales.**

**Progreso:**

**1.- tokens, diagramas sintaxis, cubo semantico, dir funciones, tabla de variables.**

**Reporte entrega 4 mayo**

**Entrega de atraso en estatutos lineales y entrega de estatutos no lineales.**

**I1.- Posible que se requiera agregar un dato más al cuádruplo con número de cuádruplo.**

**I2.- Falta manejar memoria.**

**Reporte entrega 17 mayo:**

**Faltan arreglos y máquina virtual.**

**Se me olvido subirlo.**

**Reporte 25 Mayo:**

**Avance hasta funciones de la máquina virtual. incluye desarrollo funciones**

**Falta: desarrollar arreglos en máquina virtual, probar arreglos en parser,**

**documentación.**

**Reporte 31 mayo.**

**Compilador terminado. Con pruebas.**

<https://github.com/EduardoToraya/Compiler-in-python>

Commits on May 28, 2020

1. [update before documentation](https://github.com/EduardoToraya/Compiler-in-python/commit/f09b6739565e7df01afd8d1fecdaffa8203f78f2)

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed 1 minute ago

[**f09b673**](https://github.com/EduardoToraya/Compiler-in-python/commit/f09b6739565e7df01afd8d1fecdaffa8203f78f2)

Commits on May 27, 2020

1. [Funcionalidad completada.](https://github.com/EduardoToraya/Compiler-in-python/commit/1c875408df5acbc2722d0daad3d1f22483f8b12c)

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed 23 hours ago

[**1c87540**](https://github.com/EduardoToraya/Compiler-in-python/commit/1c875408df5acbc2722d0daad3d1f22483f8b12c)

Commits on May 25, 2020

1. [Finished parser and functions working](https://github.com/EduardoToraya/Compiler-in-python/commit/0bd6bf227850eb9355e9e47941893b804ff9591a) …

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed 3 days ago

[**0bd6bf2**](https://github.com/EduardoToraya/Compiler-in-python/commit/0bd6bf227850eb9355e9e47941893b804ff9591a)

Commits on May 19, 2020

1. [Functions added and bug repair.](https://github.com/EduardoToraya/Compiler-in-python/commit/7b6a0d653a0474f1d926f4ce35be149d6761f781) …

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed 9 days ago

[**7b6a0d6**](https://github.com/EduardoToraya/Compiler-in-python/commit/7b6a0d653a0474f1d926f4ce35be149d6761f781)

Commits on May 17, 2020

1. [Update a funciones](https://github.com/EduardoToraya/Compiler-in-python/commit/7a7f7d3a6d9f20abde86f3c2a956f1a69d0854bc)

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed 11 days ago

[**7a7f7d3**](https://github.com/EduardoToraya/Compiler-in-python/commit/7a7f7d3a6d9f20abde86f3c2a956f1a69d0854bc)

1. [changes to function](https://github.com/EduardoToraya/Compiler-in-python/commit/8337952f93e3ffee00fd3910c5c136e83718d4f3)

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed 11 days ago

[**8337952**](https://github.com/EduardoToraya/Compiler-in-python/commit/8337952f93e3ffee00fd3910c5c136e83718d4f3)

Commits on May 13, 2020

1. [Update on memory management](https://github.com/EduardoToraya/Compiler-in-python/commit/b675be28afe9411d2078d79954db04cc80dc6dbf)

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed 15 days ago

[**b675be2**](https://github.com/EduardoToraya/Compiler-in-python/commit/b675be28afe9411d2078d79954db04cc80dc6dbf)

Commits on May 3, 2020

1. [Update estatutos lineales y no lineales.](https://github.com/EduardoToraya/Compiler-in-python/commit/7392353deadb7fca3ab9ab7dee7d899ed347bff7)

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed 25 days ago

[**7392353**](https://github.com/EduardoToraya/Compiler-in-python/commit/7392353deadb7fca3ab9ab7dee7d899ed347bff7)

Commits on Apr 27, 2020

1. [3rd update on compiler](https://github.com/EduardoToraya/Compiler-in-python/commit/5a32f9701810dc82522c901756e284f86724ede7) …

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed on 27 Apr

[**5a32f97**](https://github.com/EduardoToraya/Compiler-in-python/commit/5a32f9701810dc82522c901756e284f86724ede7)

Commits on Apr 20, 2020

1. [FirstUpload](https://github.com/EduardoToraya/Compiler-in-python/commit/c069a1830a4f961599e0cda688c1852de39f9b3d)

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed on 20 Apr

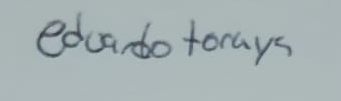
[**c069a18**](https://github.com/EduardoToraya/Compiler-in-python/commit/c069a1830a4f961599e0cda688c1852de39f9b3d)

1. [Initial commit](https://github.com/EduardoToraya/Compiler-in-python/commit/1c0ad64384d0d0e62cdee3a1ad7479355f641408)

[@EduardoToraya](https://github.com/EduardoToraya)

[**EduardoToraya**](https://github.com/EduardoToraya/Compiler-in-python/commits?author=EduardoToraya) committed on 20 Apr

[**1c0ad64**](https://github.com/EduardoToraya/Compiler-in-python/commit/1c0ad64384d0d0e62cdee3a1ad7479355f641408)

**Firma:** 

**b) DESCRIPCIÓN DEL LENGUAJE:**

**b.1) Nombre del Lenguaje: ForeverAlone**

**b.2) Descripción genérica de las principales características del lenguaje.**

**Este lenguaje utiliza palabras en español para la sus palabras reservadadas de operaciones. Comienza definiendo el nombre del programa seguido de su declaración de variables globales, posteriormente es opcional crear las funciones las cuales deberán ser declaradas con la palabra clave de la misma, seguida de su tipo y de los parámetros que tendrá la misma, similar a la instanciación de funciones en c++. Posteriormente se cuenta con un bloque Principal() de código y debe colocarse hasta el final del código para correcta ejecución, con las funciones auxiliares arriba del mismo.**