

# **Especificación de Requerimientos del Software**

**para**

## **Analizador contextual de lenguaje Triángulo**

**Integrantes del grupo:**

**Christian Leon Guevara - 2013371982**

**Gabriel Ramírez Ramírez - 201020244**

**2018/12/24** ☐

## Contenido

<b>Introducción.....</b>	<b>3</b>
Propósito.....	3
Alcance del proyecto .....	3
<b>Descripción General.....</b>	<b>3</b>
Funciones del producto .....	3
Clases de usuario y características .....	3
Ambiente de operación.....	3
Limitaciones de diseño e implementación .....	3
<b>Requerimientos de Interfaz Externa .....</b>	<b>4</b>
Interfaces de Software .....	4
<b>Características del sistema.....</b>	<b>4</b>
Revisión del Árbol Sintáctico.....	4
Poner Referencias en el árbol sintáctico .....	4
<b>Otros Requerimientos No Funcionales.....</b>	<b>4</b>
Requerimientos de rendimiento.....	4
Atributos de Calidad del Software .....	5

# **Introducción**

## **Propósito**

El producto será realizar un analizador contextual del lenguaje triangulo, este deberá ser capaz de revisar el código, de poner ciertas referencias en el árbol sintáctico y en caso de que encuentre errores de tipo o declaración este deberá reportarlos.

## **Alcance del proyecto**

Este proyecto se limitara a implementar las funciones de un compilador léxico de Triangulo (revisar errores y poner ciertas referencias en el árbol sintáctico) usando las interfaces del árbol sintáctico(Ast.ml y Ast.mli) y la de Reporte de Errores (Carpeta de Misc).

# **Descripción General**

## **Funciones del producto**

El producto solo tendrá 1 función, recibir un árbol sintáctico y devolver un árbol sintáctico decorado para el siguiente proceso que sería generación de código; en caso de que existan errores de declaración o de tipo

## **Clases de usuario y características**

El usuario principal de este módulo será el profesor Ignacio Trejos, lo usara para enseñar el curso de Compiladores E Interpretes del TEC.

## **Ambiente de operación**

El compilador en general se ejecutara en una computadora que se espera que tenga como mínimo:

- 512 MB de memoria RAM
- 500 MB de espacio de disco duro

## **Limitaciones de diseño e implementación**

El proyecto deberá de ser implementado haciendo uso de las interfaces de:

- Ast.ml
- Ast.mli

- ErrorReporter.ml
- ErrorReporter.mli

## **Requerimientos de Interfaz Externa**

### **Interfaces de Software**

- El código debe correrse en ocaml 4.07.0

## **Características del sistema**

### **Revisión del Árbol Sintáctico**

Descripción y prioridad

Deberá validar todas las declaraciones de variables y que las expresiones tengan tipos coherentes con el ámbito en el que fueron usadas. En caso de que se encuentre una incoherencia se deberá reportar el error. La importancia es muy Alta.

Requerimientos Funcionales

- REQ-1: Revisar el Árbol sintáctico
- REQ-2: Reportar errores en caso de que se encuentren

### **Poner Referencias en el árbol sintáctico**

Descripción y prioridad

Se deberán hacer referencias de las instancias de las variables a la declaración en la variable.

Requerimientos Funcionales

- REQ-1: Hacer referencias de las instancias de las variables a la declaración de la variable

## **Otros Requerimientos No Funcionales**

### **Requerimientos de rendimiento**

El compilador en general como será para fines educativos no tendrá que compilar archivos muy extensos ni complejos, por ello:

- El analizador contextual deberá poder ejecutar un árbol de 100 nodos en 20 segundos o menos.

## **Atributos de Calidad del Software**

El código deberá ser escrito siguiendo el estilo de programación CS 3110 de OCaml.