

## **Reimplementaciones de Triángulo ( $\Delta$ ) en ML(s) – Analizador Contextual**

**Contexto:** Es un sistema técnico que desarrolla componentes que coexisten y colaboran entre ellos. Se utilizará el lenguaje ML (Standard ML, OCaml o F#).

### **¿Qué existe?**

Una implementación de un compilador para el lenguaje de programación Triangulo utilizado por el profesor Ignacio Trejos Zelaya en el curso ‘Compiladores e intérpretes’, de la carrera de Ingeniería en Computación en el TEC. El compilador es de tres pasadas y se encuentra escrito en Java.

### **Análisis Contextual – Tareas técnicas:**

1. Aprendizaje del lenguaje ML y algunas herramientas pertinentes.
2. Investigar si hay reglas de estilos para OCaml (indentación, nomenclatura).
3. Investigar soporte del lenguaje ML a la modularización de los componentes y la definición de interfaces (técnicas) entre componentes.
4. Aprendizaje de lenguaje fuente ( $\Delta$ ) y su sistema de tipos.
5. Definir algoritmos y estructuras de datos para el manejo de tablas de identificación, con consideración a posibles extensiones de  $\Delta$  en materia de declaraciones compuestas (paralelas o colaterales, privadas o locales, mutuamente recursivas).
6. Diseño, construcción y prueba de la estructura de datos y sus algoritmos.
7. Análisis contextual de expresiones.
8. Análisis contextual de comandos.
9. Análisis contextual de declaraciones.
10. Manejo de recursión.
11. Comprobación de tipos para funciones y procedimientos con parámetros.
12. Manejo de errores.
13. Comprobación de variables, simples y compuestas.
14. Manejo de denotadores de tipos.
15. Ambiente Estándar (Integer, Boolean, Char, ...)
16. Pruebas de Integración (QA).
17. Pruebas de validación y verificación (QA).
18. Documentación técnica de los componentes que integran el Analizador contextual.
19. Integración del Analizador contextual con el Analizador sintáctico<sup>1</sup>

---

<sup>1</sup> El Analizador sintáctico también comprende al Analizador léxico y esconde la definición de ‘tokens’; tiene como interfaz los Árboles de sintaxis abstracta (ASTs). El Analizador sintáctico lo está desarrollando otro equipo de estudiantes del curso ‘Proyecto de Ingeniería del Software’.