

## Construcción de Compilador YAPL

### Generación de Código Intermedio

La generación de código intermedio es la siguiente fase del diseño de nuestro compilador. Luego de haber realizado el análisis semántico (análisis de tipos estático), utilizarán sus estructuras de datos (árboles sintácticos, tablas de símbolos) para generar una representación intermedia del código de alto nivel. Esta representación intermedia les será de utilidad al momento de la generación de código assembler (u objeto). Esta fase contempla:

- Agregar acciones semánticas necesarias sobre el árbol sintáctico, con el objetivo de generar código intermedio. La sintaxis del código intermedio a utilizar es a discreción del diseñador (pueden utilizar la sintaxis de alguna bibliografía conocida).
- Complementar la tabla de símbolos con información necesaria para la generación de código assembler (direcciones de memoria, etiquetas, temporales, etc.)
- Implementar un algoritmo para asignación y reciclaje de variables temporales.

El diseño del lenguaje intermedio tiene que estar hecho de tal forma que los ayude a generar el código assembler (u objeto) en la última fase de la compilación. Es por esto que en este punto del proyecto deben de considerar la sintaxis y estructura del lenguaje assembler (u objeto) que utilizarán en la fase final.

### ¿Qué entregar?

- Desarrollo Ágil
  - Tablero Kanban elaborado para esta fase [Periodicidad Semanal]
  - Retrospectiva de Sprint [Cambios y/o Sugerencias]
- Interfaz de Usuario
  - Que permita la escritura de programas en YAPL y despliegue del código intermedio generado.

### Sugerencias

- Presentar avances semanales.
- Estudien la sintaxis y el uso del código assembler (u objeto) que generarán en la última fase.