

Desarrollo de aplicaciones avanzadas de ciencias computacionales (Gpo 502)



**Tecnológico  
de Monterrey**

**Proyecto 3 Analizador Semántico**

---

Fecha de entrega: 14/05/2025

---

**Alumnos**

Francisco Martinez Gallardo Lascurain | A01782250

### Reglas lógicas utilizadas:

1. Declaración de variables:
  - a. Una variable o parámetro no puede ser declarado con tipo void.
  - b. No se permite declarar una variable con el mismo nombre dos veces dentro del mismo scope.
2. Declaración de funciones:
  - a. No se permite declarar múltiples veces una misma función en el mismo scope.
  - b. El tipo de retorno debe de ser del mismo tipo, con el que se creó la función.
3. Llamadas a variables:
  - a. Si una variable fue llamada y no ha sido declarada en el scope se lanza error.
4. Llamadas a variables
  - a. Se verifica que una función no regrese tipo void cuando se usa en expresiones.
  - b. Se checa que el número de argumentos y el tipo coincidan.
  - c. No se permite que una lista se le asigne un valor, sin poner el índice. Como no se permite que se le asigne un valor a un índice de `int` o `void`.
5. Condicionales:
  - a. La condición debe ser una expresión que no contenga void.
6. `return`
  - a. El valor del retorno debe tener el mismo valor que la función.

### Tabla de símbolos:

1. La tabla de símbolos está dada por un bucket list que contiene las siguientes llaves:
  - a. `location`: Línea de posición.
  - b. `linenos`: Líneas donde se hace referencia
  - c. `type`: El tipo que fue declarado.
  - d. `Is_array`: checa si es un arreglo.

- e. metadata: Información adicional (lo use principalmente para los parámetros).

2. Scopes:

- a. Los scopes se definieron como una pila de Bucket lists y luego una lista de nombres se usó para guardar los nombres de los scopes. Los nombres son clave ya que se utilizan para entrar a los scopes cuando se hace alguna operación. Solo se crea un nuevo scope por función y el scope global.

3. Se definieron la siguientes funciones para hacer operaciones en la tabla:

- a. st\_insert: Inserta un identificador.
- b. st\_lookup: Busca un identificador en todos los scopes
- c. .st\_lookup\_current\_scope: Busca solo en el scope actual
- d. .st\_get\_type, st\_get\_metadata: Recuperan tipo o metadatos.
- e. st\_get\_is\_array: Checa si la variable es un arreglo.