



Manual de Tabla de Símbolos

Zeuddy Segovia

Endelkys Matos

Facultad de Ingeniería, Universidad Valle del Momboy

Lenguaje y Compiladores

Ing. Katituska Morillo

4 de Abril de 2025

Introducción:

La Tabla de Símbolos es una estructura de datos fundamental en el proceso de compilación de Anny a JavaScript. Su función principal es almacenar y gestionar información sobre los identificadores (variables, funciones, etc.) declarados en el código fuente, permitiendo validaciones semánticas y generación de código eficiente.

2. Estructura de la Tabla

Cada entrada en la tabla contiene la siguiente información:

Campo	Tipo	Descripción
nombre	String	Nombre del identificador (ej: edad, sumar)
tipo	String	Tipo de dato (ej: entero, cadena, funcion)
ámbito	String	Contexto donde fue declarado (ej: global, funcion_principal)
línea	Integer	Línea de declaración en el código fuente

3. Funcionalidades Clave

3.1. Agregar Símbolos

```
symbol_table.add_symbol(nombre, tipo, ámbito, línea)
```

- **Parámetros:**

- **nombre:** Nombre del identificador.
- **tipo:** Tipo de dato (entero, decimal, cadena, booleano, funcion).
- **ámbito:** Contexto de declaración (por defecto: global).
- **línea:** Número de línea en el código fuente.

- **Comportamiento:**

- Si el símbolo ya existe en el ámbito actual:
 - Genera error semántico (Error 100: Variable ya declarada).
- Si es válido: lo almacena en la tabla.

3.2. Buscar Símbolos

```
symbol = symbol_table.get_symbol(nombre, ámbito)
```

- **Parámetros:**

- nombre: Nombre a buscar.
- ámbito: Ámbito donde buscar (opcional).

- **Retorna:**

- Diccionario con los datos del símbolo si existe.
- None si no se encuentra.

3.3. Verificar Existencia

```
existe = symbol_table.exists(nombre, ámbito)
```

- **Retorna:**

- True si el símbolo existe en el ámbito especificado.
- False en caso contrario.

4. Gestión de Errores

La tabla de símbolos trabaja en conjunto con la Tabla de Errores para reportar:

Código	Mensaje	Causa
100	Variable ya declarada	Redeclaración en el mismo ámbito
200	Variable no declarada	Uso de identificador no definido
300	Tipo incompatible	Asignación con tipos no compatibles

5. Ejemplos de Uso

5.1. Declaración Válida

```
entero edad = 25;
```

```
symbol_table.add_symbol("edad", "entero", "global", 1)
```

Tabla Resultante:

nombre	tipo	ámbito	línea
edad	entero	global	1

5.2. Error de Redeclaración


```
decimal precio = 99.99;  
decimal precio = 50.0; // Error 100
```

Salida de Errores:

Código	Mensaje	Línea	Columna
100	Variable ya declarada	2	8

5.3. Uso de Variable no Declarada

```
imprimir(total); // Error 200
```

Salida de Errores:

Código	Mensaje	Línea	Columna
200	Variable no declarada	1	10

6. Integración con el Sistema

Fase de Análisis:

- Durante el parsing, se invoca `add_symbol` en cada declaración.
- Se verifica el ámbito actual (ej: dentro de una función).

Validación Semántica:

- Antes de usar un identificador, se verifica con `exists()`.

Generación de Código:

El listener de JavaScript accede a la tabla para resolver nombres.