Índice

- **Tabla de Símbolos**
- **Analizador Morfológico**
- **Ejercicios en clase**

Información de la TABLA DE SÍMBOLOS para ALFA

Administra todos los nombres que aparecen en los programas:

- ✓ Las **variables** (globales del programa y locales de las funciones)
- ✓ Los **parámetros** de las funciones
- ✓ Las **funciones**

- **Clave de acceso**: Identificador (lexema) de la variable, parámetro o función.
- **Categoría del elemento** (variable, parámetro de función y función): define la categoría del identificador que está almacenado en la tabla de símbolos. A tener en cuenta:

```
#define VARIABLE 1
#define PARAMETRO 2
#define FUNCION 3
```

- **Tipo básico de dato** (boolean, int): para variables, parámetros de funciones, vectores y tipo de datos de retorno de función. A tener en cuenta:

```
#define BOOLEAN 1
#define INT 2
```

- **Clase** (escalar, vector): identifica la estructura de la información asociada al identificador de una variable o un parámetro. A tener en cuenta:

```
#define ESCALAR 1
#define VECTOR 2
```

- **Tamaño** (número de filas, valores 1-64): sólo para identificadores de vectores.
- **Número de parámetros** de función.
- **Posición del parámetro** dentro de la lista de parámetros (0 a n-1).
- **Número de variables locales** dentro de una función (0 a n-1).
- **Posición de la variable local** dentro de una función (1 a n).

Ámbitos de la TABLA DE SÍMBOLOS para ALFA

Código fuente ALFA

```
main {

  //Variables globales
  int g1, g2, …;

  //Definición de funciones
  function int f1(int p1, int p2, …) //Parámetros para f1
  {
    // Variables locales f1
    int l1, l2, …;
    ...
  }
  function int f2(int q1, int q2, …) //Parámetros para f2
  {
    // Variables locales f2
    int m1, m2, …;
    ...
  }

  // Sentencias del programa
  ...
}
```

Ámbito	Declara	Variables
Global	g1, g2, f1, f2	g1, g2, f1, f2
Función f1	p1, p2, l1, l2	g1, g2, f1, p1, p2, l1, l2
Función f2	q1, q2, m1, m2	g1, g2, f1, f2, q1, q2, m1, m2

Programa de prueba para la TABLA DE SÍMBOLOS para ALFA

./pruebaTS <fichero_entrada> <fichero_salida>

Fichero de entrada

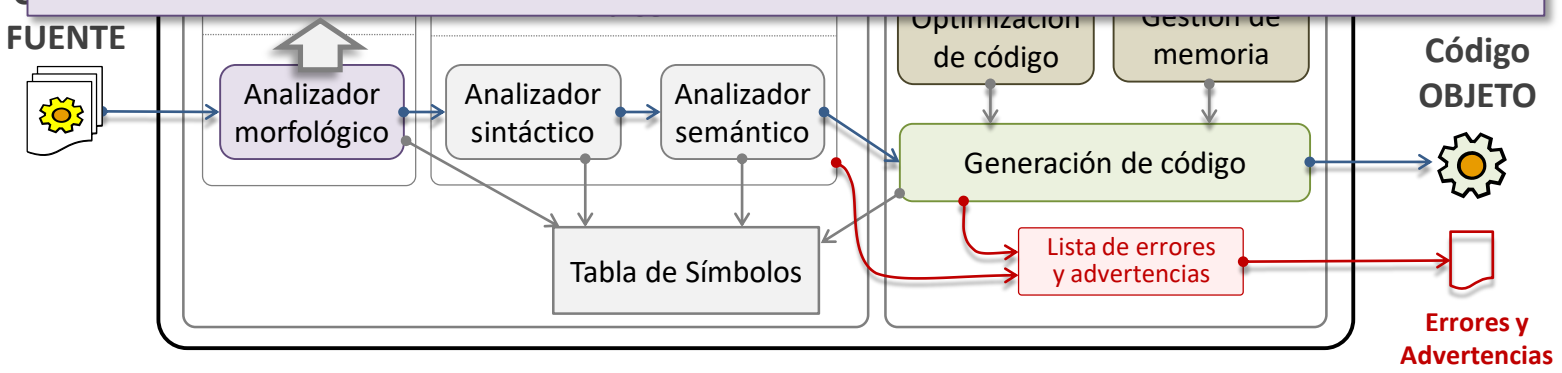
- **Inserción de un elemento** [N]≥0
  Definición> [ID] [N]
  Ejemplo> global 10
- **Búsqueda de un elemento**
  Definición> [ID]
  Ejemplo> global
- **Apertura de un ámbito** [N]≤-1
  Definición> [ID] [N]
  Ejemplo> función -20
- **Cierre de ámbito activo**
  Definición> cierre -999
  Ejemplo> cierre -999

Fichero de salida

- **Inserción/Apertura con con éxito**
  Definición> [ID]
  Ejemplo> global
- **Inserción/Apertura sin éxito**
  Definición> -1 [ID]
  Ejemplo> -1 nombre
- **Búsqueda con éxito**
  Definición> [ID] [N]
  Ejemplo> global 10
- **Búsqueda sin éxito**
  Definición> [ID] -1
  Ejemplo> nombre -1
- **Cierre de ámbito local**
  Definición> cierre
  Ejemplo> cierre

[ID]: *Identificador*
[N]: *Número entero*

Analizador morfológico: transforma el código fuente del programa de una secuencia de caracteres, a una secuencia de unidades sintácticas (tokens).



Código FUENTE

```
begin
  int A;
  A := 100;
  A := A+A;
  output A
end
```

1 Análisis morfológico

```
(<PC>,begin)
  (<PC>,int) (<ID>,A) (<SS>,;)
  (<ID>,A) (<SS>,:=) (<CN>,100) (<SS>,;)
  (<ID>,A) (<SS>,:=) (<ID>,A) (<SS>,+) (<ID>,A) (<SS>,;)
  (<PC>,output) (<ID>,A)
(<PC>,end)
```

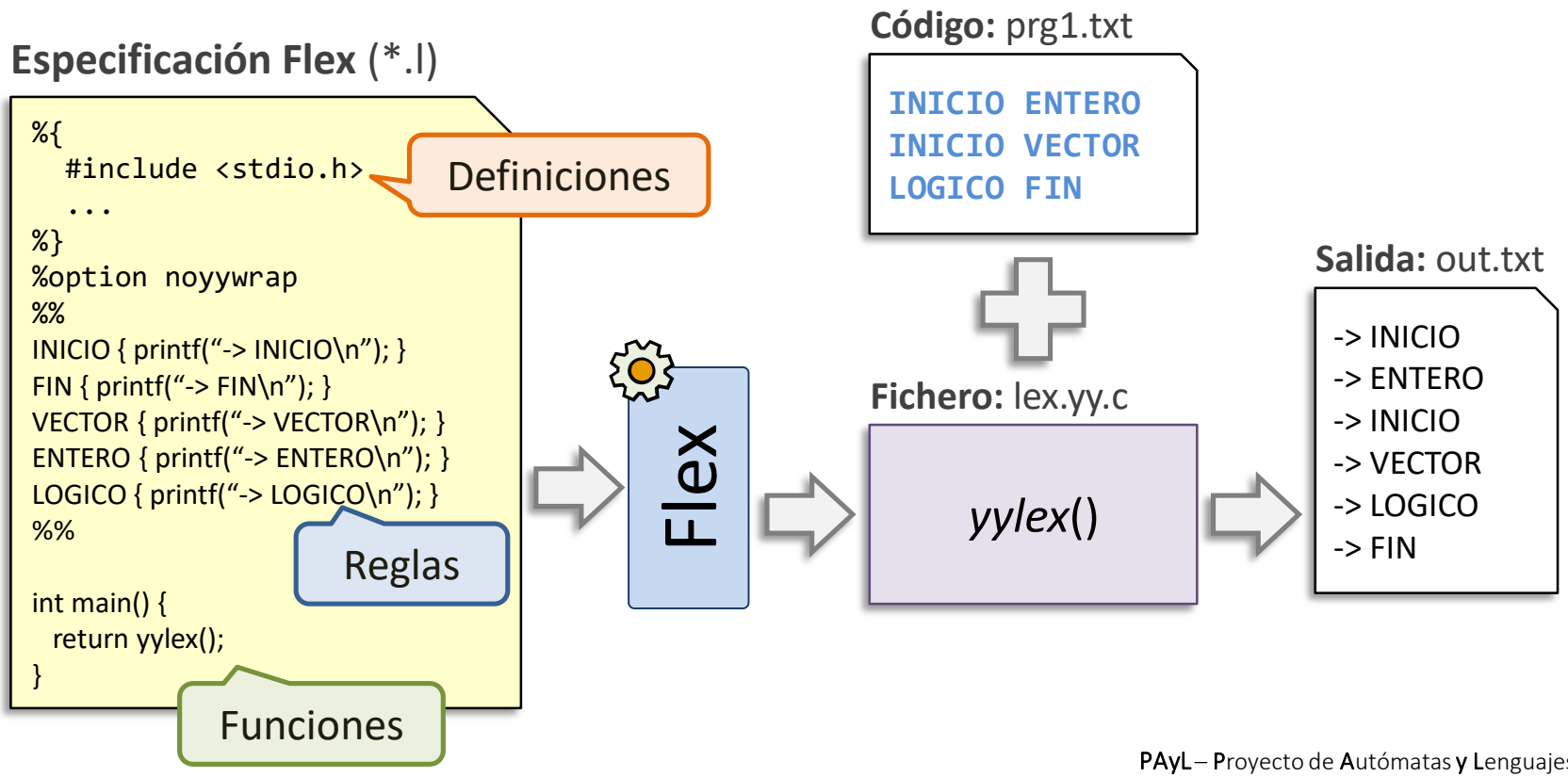
Token (<..>,valor)

Tokens

<ID> Identificadores, **<PC>** Palabras reservadas, **<CN>** Constantes numéricas; **<CS>** Constantes literales; **<SS>** Símbolos simples; **<SM>** Símbolos múltiples

1 Flex: herramienta para la creación del Análisis Morfológico

- **Flex** es una herramienta que permite **generar automáticamente autómatas finitos que reconocen lenguajes regulares expresados mediante patrones Flex**.
- Los **patrones** de Flex **son extensiones de las expresiones regulares**.
- Flex recibe como entrada **un lenguaje regular expresado como un conjunto de patrones**, y genera la función C ***yylex***() que es un autómata finito que **reconoce cadenas** pertenecientes a dicho lenguaje de entrada y opcionalmente **ejecuta alguna acción** cada vez que se reconoce una de las cadenas del lenguaje.
- A la entrada a la herramienta Flex se le llama **especificación Flex**.



A partir de la especificación <file>.l se genera el ejecutable de la siguiente manera:

0 Instalación Flex

```
$ sudo apt-get update
$ sudo apt-get install flex
```

1 Compilar la Especificación Flex

```
$ flex <file>.l
> lex.yy.c  //Genera el fichero
```

2 Generación de ejecutable

```
$ gcc –Wall –o execute lex.yy.c
```

3 Ejecución

```
$ ./execute
```