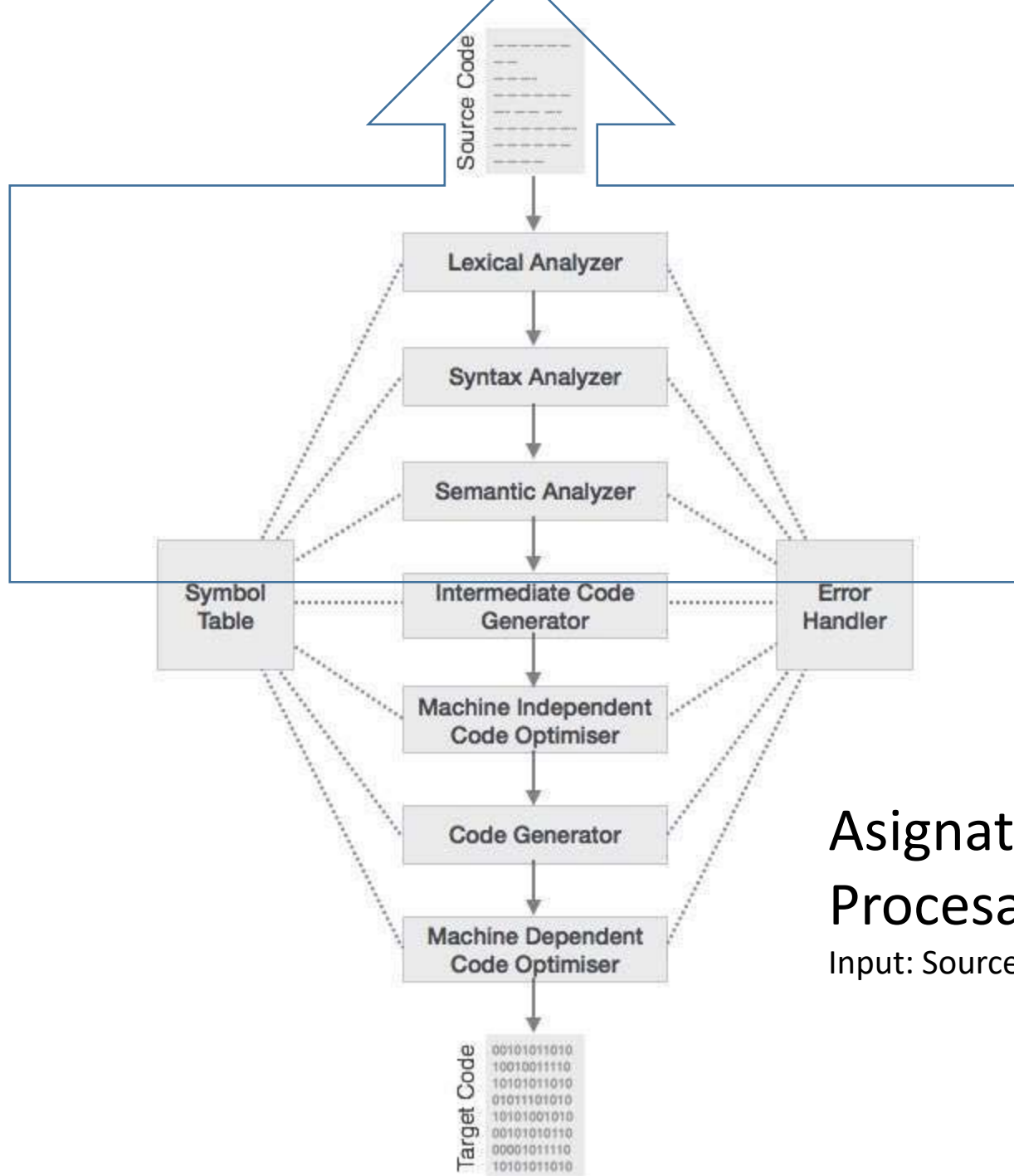




# TABLA DE SÍMBOLOS



## Front-End: Análisis

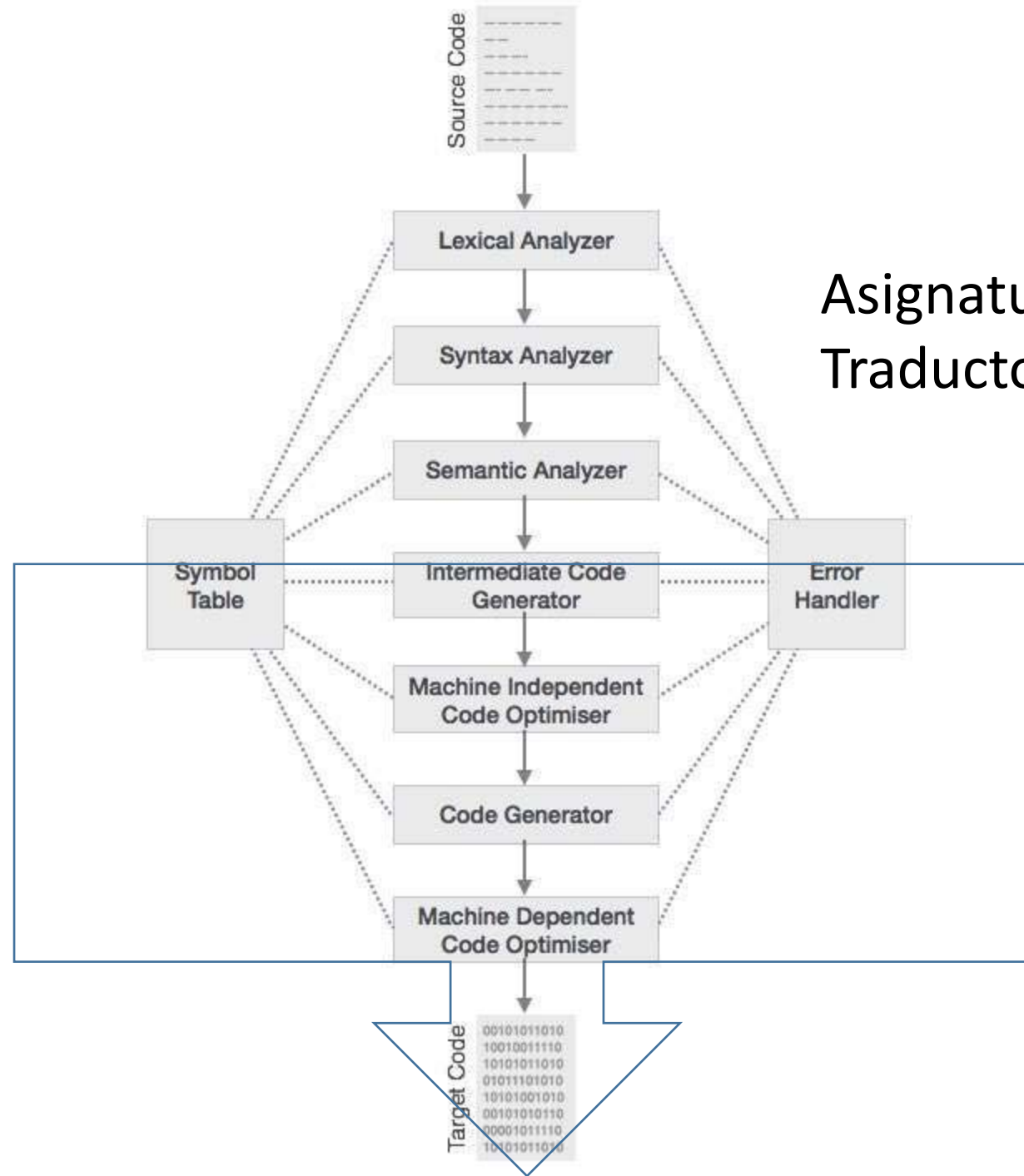


## Asignatura Procesadores de Lenguajes

Input: Source file (code or other)



# Asignatura Traductores de Lenguajes



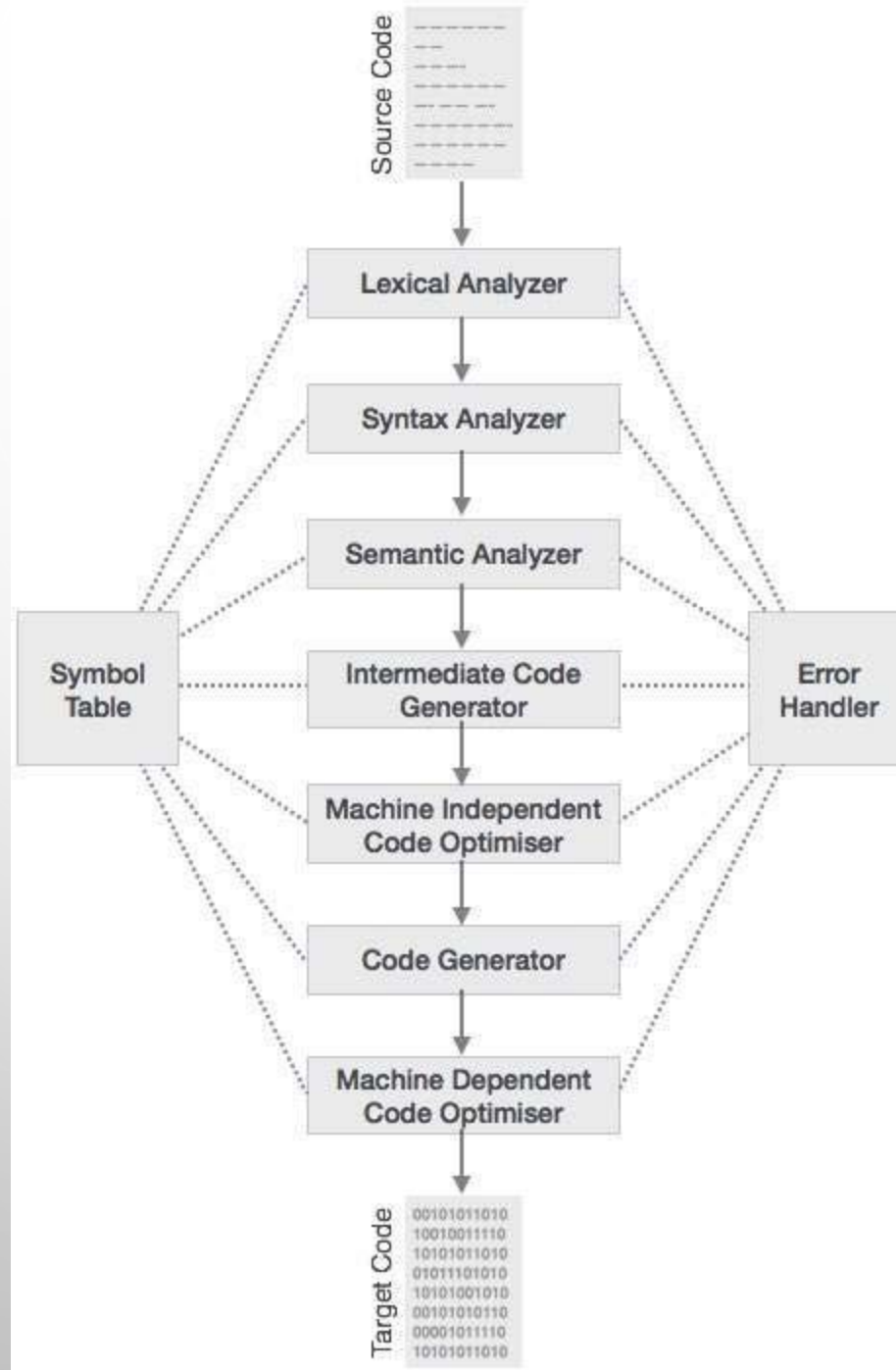
Back-End:  
Síntesis



## ¿QUÉ ES LA TABLA DE SÍMBOLOS?

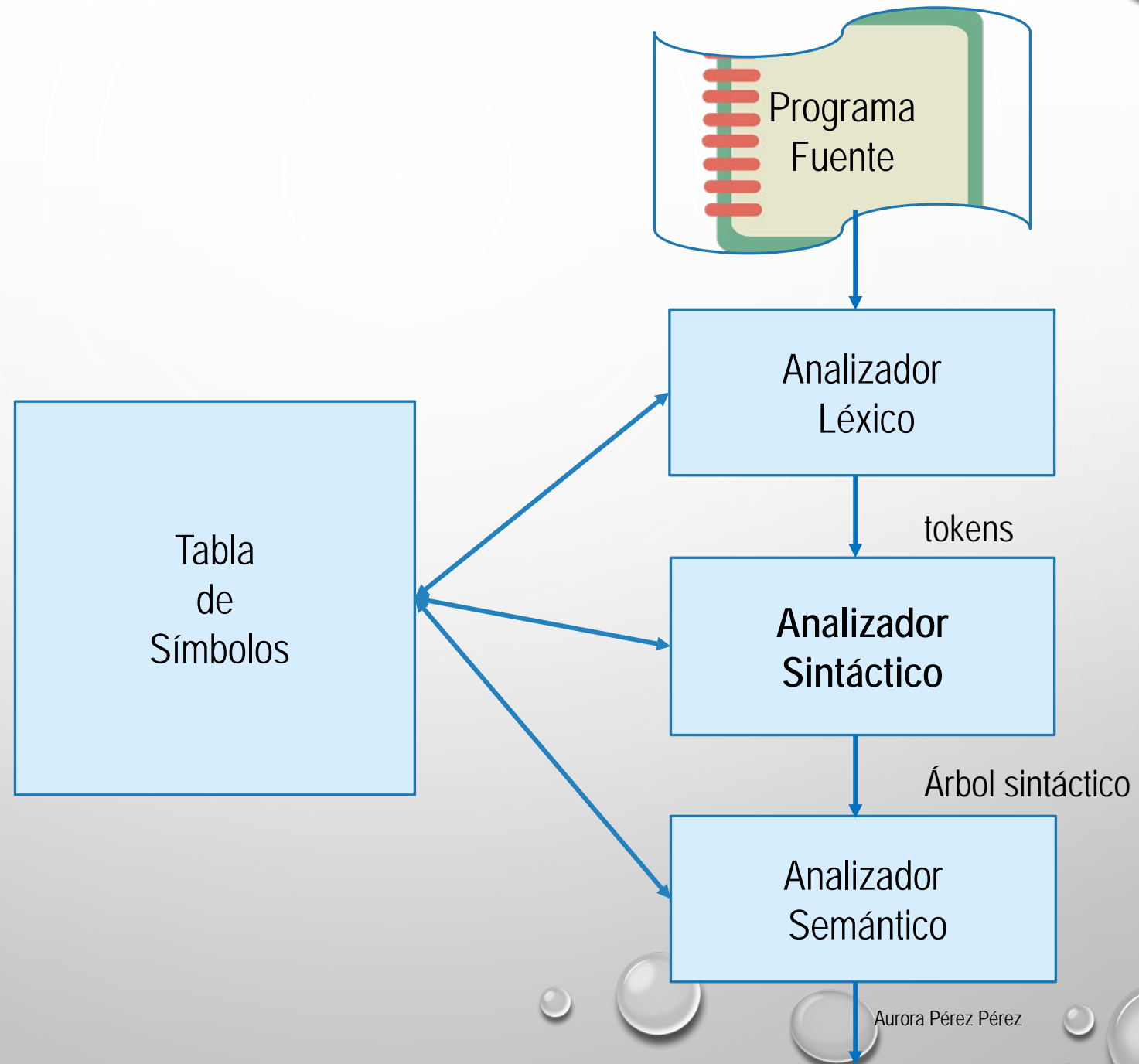
Estructura de datos donde se almacena toda la información relevantes sobre los identificadores del programa

Todos los módulos tienen acceso a ella (para almacenar o para consultar información)





Estructura de datos  
donde se almacena  
toda la información  
relevantes sobre los  
identificadores del  
programa

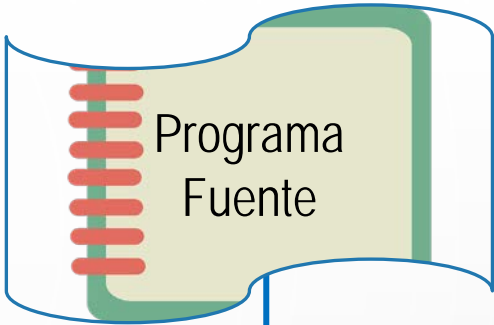




Estructura de datos donde se almacena toda la información relevantes sobre los identificadores del programa

Lexema	Tipo	...	...	...

Inserta los **lexemas** de los identificadores



Analizador Léxico

tokens

Analizador Sintáctico

Árbol sintáctico

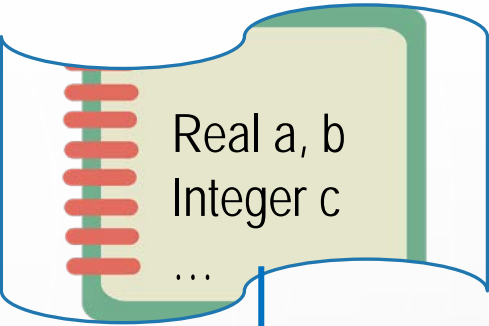
Analizador Semántico



Estructura de datos donde se almacena toda la información relevantes sobre los identificadores del programa

Lexema	Tipo	...	...	...
a				
b				
c				
...				

Inserta los **lexemas** de los identificadores



Analizador  
Léxico

<PR\_real, > **<id,1>** <coma, >  
<id,2> <PR\_int, > <id, 3> ...

Analizador  
Sintáctico

Árbol sintáctico

Analizador  
Semántico



## ¿QUÉ INFORMACIÓN CONTIENE LA TABLA DE SÍMBOLOS?

- Una entrada para cada identificador, con una serie de campos para atributos:

→ **Lexema** (nombre del identificador)

→ **Tipo** (entero, real, cadena, lógico, array, función, etiqueta,...)

→ Dependiendo del Tipo, se tendrán unos atributos u otros (no es una tabla homogénea):

- entero, real, cadena, lógico, array...: **Desplazamiento** (dirección relativa)
- array: núm de **dimensiones**, límite **inf** y límite **sup** de cada dimensión,...
- función: **número y tipo de parámetros**, **método de paso** de parámetros, **tipo devuelto** por la función, **etiqueta** de comienzo del código de la función, ...
- ...

Una TS  
para cada  
ámbito

Las entradas no  
son homogéneas.  
Pero todas tienen  
Lexema y Tipo

- Alcance (ámbito de visibilidad de los identificadores)





## • GESTIÓN DE LA TS

→ GESTIÓN DEL ÁMBITO O ALCANCE (al entrar en un nuevo ámbito o salir de uno)

→ AÑADIR INFORMACIÓN (cuando se “declara” un nuevo identificador en el programa fuente )

→ CONSULTAR INFORMACIÓN (cuando se usa un identificador que ya se había “declarado”)

## FUNCIONALIDADES

- Crear una nueva TS
- Destruir una TS ← liberar el espacio que ocupa
- Crear nueva entrada ← **Añadir un nuevo lexema**
- Completar información de los atributos de una entrada existente
- **Buscar un lexema** ← Comprobar si ya se creó una entrada para ese identificador
- Consultar atributos de una entrada

La búsqueda de un lexema tiene que ser EFICIENTE

## • MÉTODOS DE ORGANIZACIÓN DE LA TS

→ Lineal

→ Ordenada

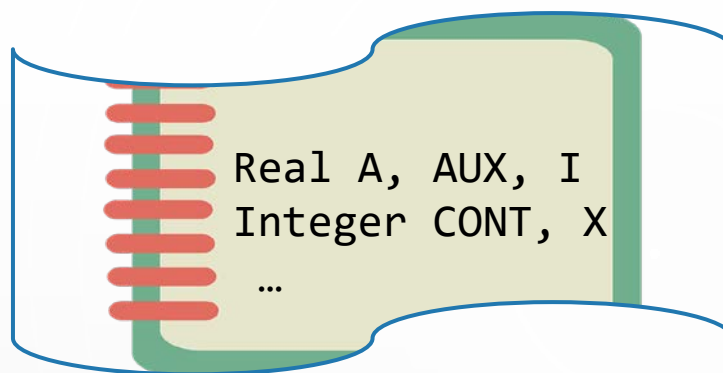
→ Hash: hash abierta, hash con desbordamiento (overflow), hash con encadenamiento

• **TAMAÑO:** Fijo / Dinámico

Analizador  
Léxico



## Ejemplo



### 1. TS Lineal

Lexema	Tipo	...	...	...
A				
AUX				
I				
CONT				
X				

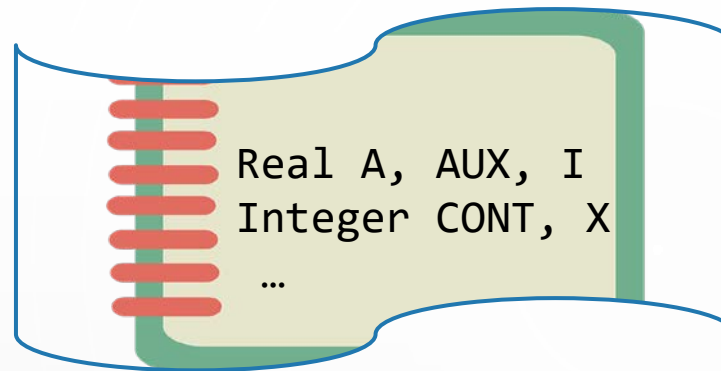
### 2. TS ordenada (alfabéticamente)

Lexema	Tipo	...	...	...
A				
AUX				
I				

Lexema	Tipo	...	...	...
A				
AUX				
CONT				
I				
X				



## Ejemplo



### 3. TS hash con encadenamiento

Hash(A) = 27  
Hash(CONT) = 27  
Hash(X) = 27

Hash(AUX) = 42  
Hash(I) = 42

