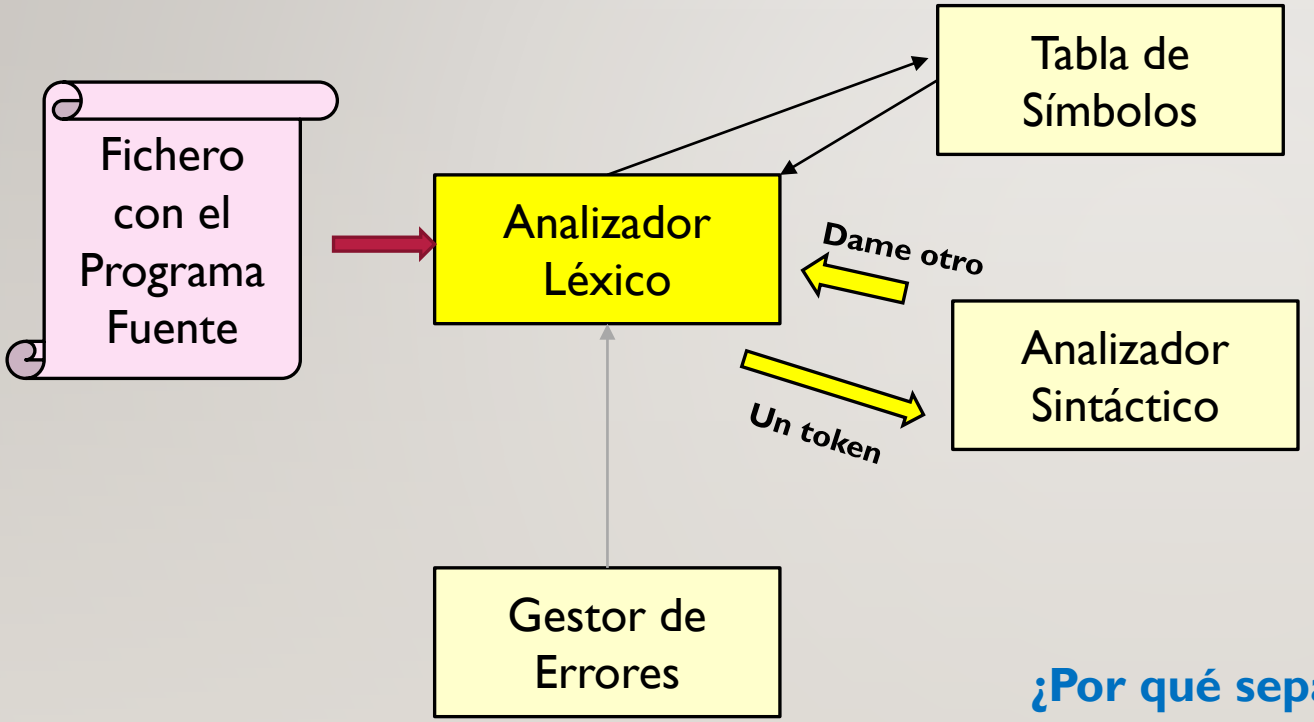
2. ANÁLISIS LÉXICO

JUAN PEDRO CARAÇA-VALENTE HERNÁNDEZ

SEPTIEMBRE 2020

2.1. FUNCIÓN DEL ANALIZADOR LÉXICO



- Manejar el fichero del Texto Fuente
- Leer 1 a 1 los caracteres de entrada
- Generar Componentes Léxicos - Tokens
- Elimina información no relevante (espacios, tabuladores, ¡comentarios!)
- Relacionar errores con su posición en el Texto Fuente
- Preprocesado del Texto Fuente
- Rellenar parte de la información en la TS

¿Por qué separar el An.Léxico y el An.Sintáctico?

Divide y Vencerás: diseño más sencillo, portable, eficiente

APLICACIONES DEL AN. LÉXICO

- Interprete de Comandos de un Sta. Operativo
- Analizar especificaciones de cualquier tipo de sistema
- Pasar información de ficheros de texto a Bases de Datos
- Front-end de un Compilador
- Formateadores de texto
- Análisis de búsquedas en browser de internet

2.2 DEFINICIONES

- Componente Léxico o Token

  Cada uno de los elementos del lenguaje con significado propio *(serán los símbolos terminales de la gramática sintáctica del lenguaje fuente)*

- Patrón (del token)

  Regla que describe el conjunto de cadenas de entrada que produce como salida el mismo tipo de token

- Lexema

  Secuencia de caracteres concreta que aparece en el texto fuente, que concuerda con el patrón de un token.

2.2 TOKEN

- **Se definen como una tupla de dos elementos:**

<Tipo_Token, Atributo >

Grosso modo, **tipo_token** relevante para el An. Sintáctico, **atributo** relevante para el An. Semántico y para el Traductor

- **Tipo_Token: que tipo de palabra ha leído el An.Léxico**
- **Atributo: el An.Léxico informa al An. Sintáctico/Semántico de información adicional sobre la palabra leída (cuando hay más de un lexema que concuerda con el patrón)**

6 EJEMPLOS

Tipo-Token	Patrón	Lexema (texto fuente)
if	{i,I} ⊕ {f,F}	if, IF
then	{t,T} ⊕ {h,H} ⊕ {e,E} ⊕ {n,N}	Then, THEN
op-rel	>, <, >=, …	>, <, >=, …
identificador	Una letra seguida de cualquier n° de letras y dígitos	Var2, …
Constante-entera	Secuencia de digitos	592
Literal o cadena	Lista de caracteres entre “ ”	“Hola a todos!”
Separador	( ) [; , : } etc.	( ) [; …

<palabra_reservada, 7 >

<op_relacional, 2 >

<id, punteroTS >

<Cte_entera, valor >

ERRORES

Hay un error léxico cuando no se logra equiparar una secuencia de caracteres con alguno de los patrones del lenguaje

→ no se corresponde con ningún tipo de token válido en el lenguaje

Errores típicos:

- Carácter no válido en el lenguaje: €
- Nombre de usuario (identificador, variable) no válido: #pasaporte
- Constante real no válida: 12,34,45

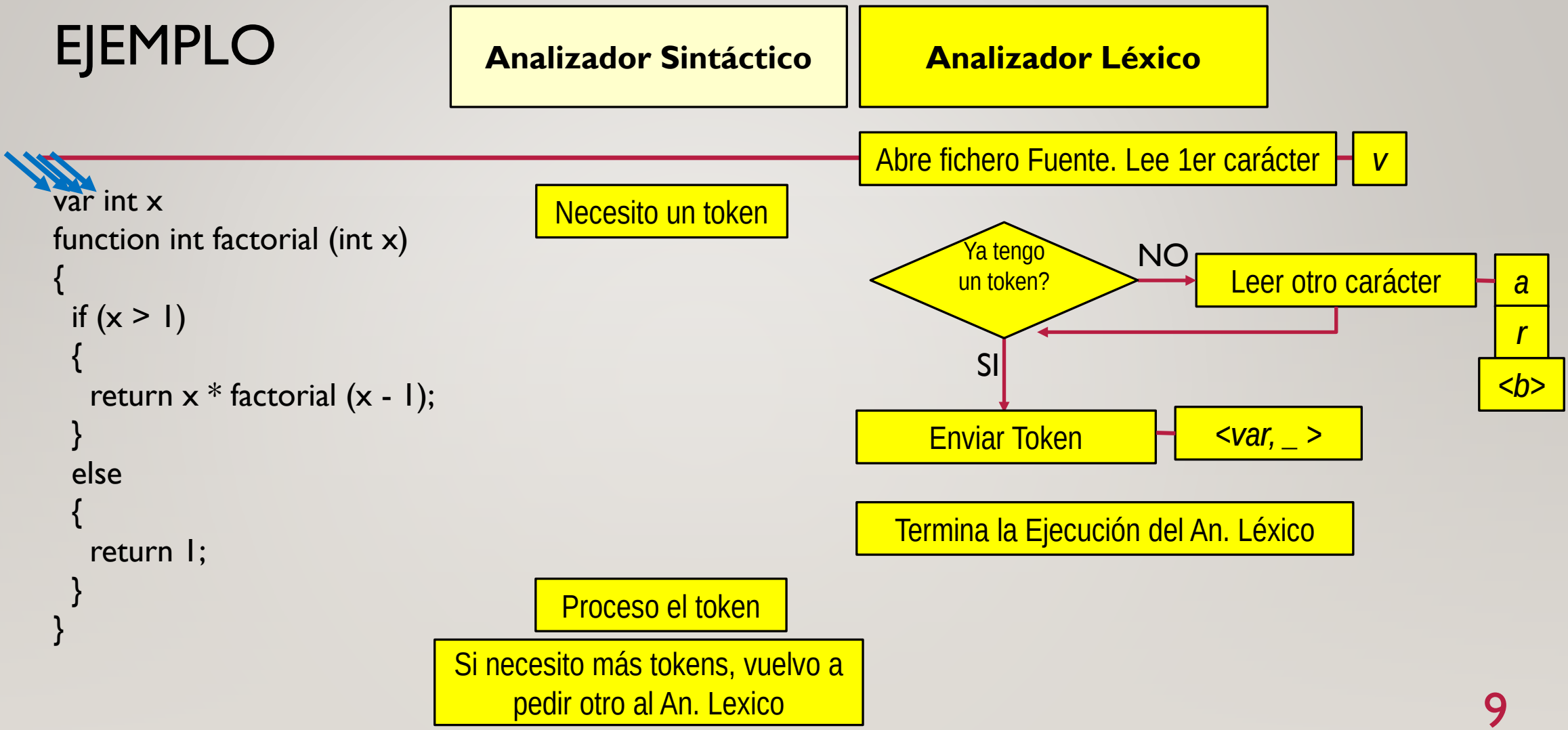
2.3 DISEÑO DEL ANALIZADOR LÉXICO

- 0. Conocer el lenguaje que vamos a analizar
- 1. Identificar los tokens
- 2. Definir la Gramática Regular *(genera todos los lexemas válidos)*
- 3. Construir el Autómata Finito Determinista *(reconoce todos los lexemas válidos)*
- 4. Incorporar Acciones Semánticas al Autómata
- 5. Incluir mensajes de error

(Implementar autómata)

EJEMPLO

```
var int x
function int factorial (int x)
{
  if (x > 1)
  {
    return x * factorial (x - 1);
  }
  else
  {
    return 1;
  }
}
```



EJEMPLO

```
var int x
function int factorial (int x)
{
  if (x > 1)
  {
    return x * factorial (x - 1);
  }
  else
  {
    return 1;
  }
}
```

Analizador Sintáctico | **Analizador Léxico**

Abre fichero Fuente. Lee 1er carácter — *v*

Necesito un token

<var, _ >

Necesito un token

<int, _ >

Necesito un token

<id, PosTS(1) >

…

Necesito un token

<Abre_Paréntesis, - >