


# Compiladores e Intérpretes

## Análisis Léxico – Esquema General

Universidad Nacional del Sur  
Departamento de Ciencias e Ingeniería de la Computación  
2024



# Esquema General del Análisis Léxico

- El **programa fuente** es una cadena de caracteres
- Tenemos que reconocer si esa cadena **pertenece al lenguaje** utilizando las reglas de **sintaxis del lenguaje**
- Las reglas de sintaxis se expresan en términos de “**palabras**”



```
if( val>10)
    var= "hola";
else
    print("chau");
//fin
```



# Esquema General del Análisis Léxico

```
if(_val>10)
var=_ "hola";
else print("
chau"); //fin
```

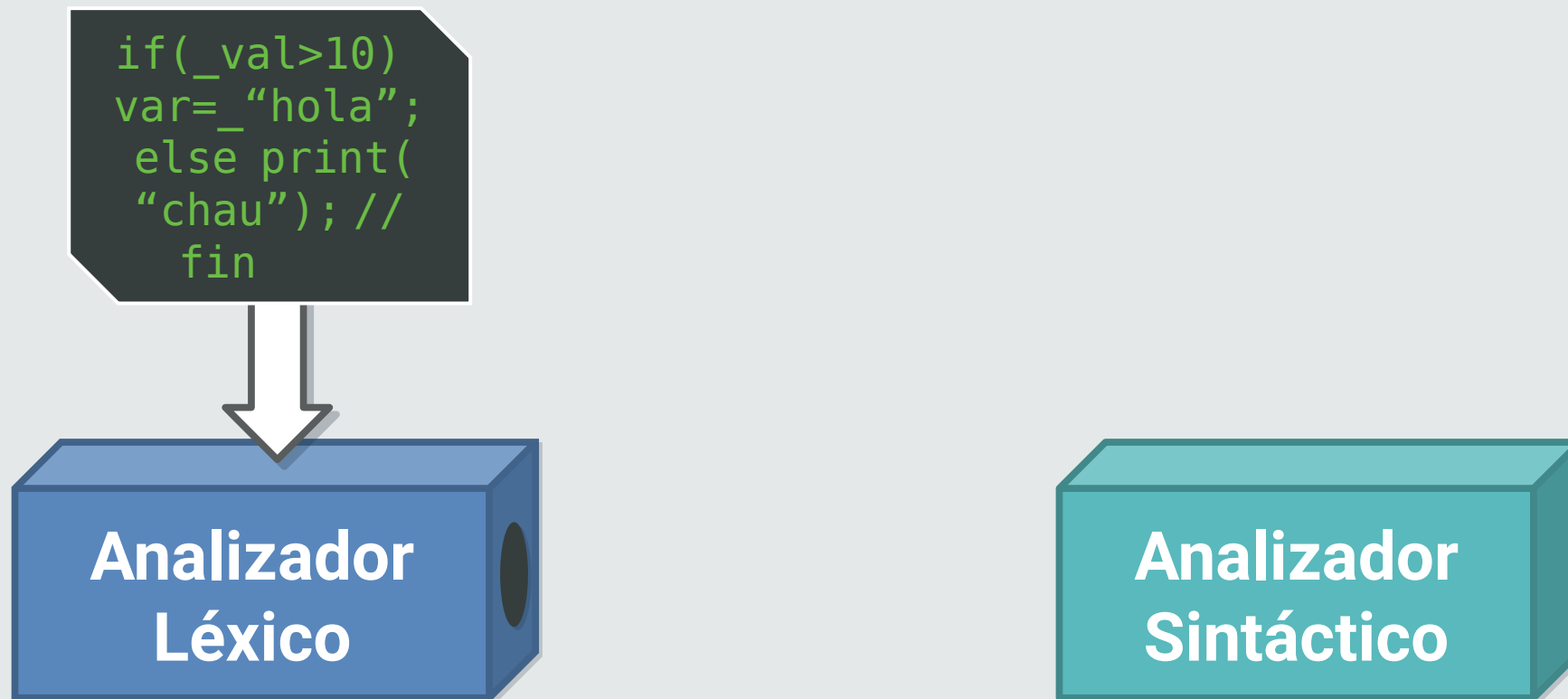
**Analizador  
Léxico**

- El **Analizador Léxico** es el encargado de leer “directamente” el **texto del programa fuente**
- Busca **construir** esas **palabras** o subcadenas **agrupando caracteres**
- **Elimina** blancos, separadores y comentarios
- Informa de **Errores Léxicos**

if ( id > num ) id = String ; else print ( String ) ;

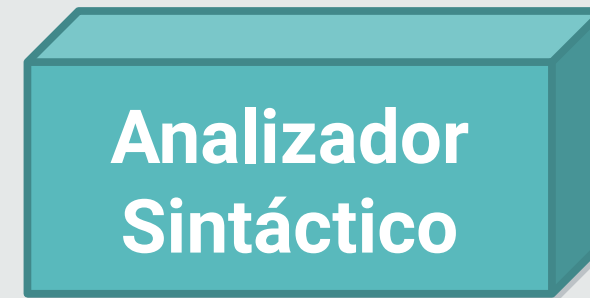
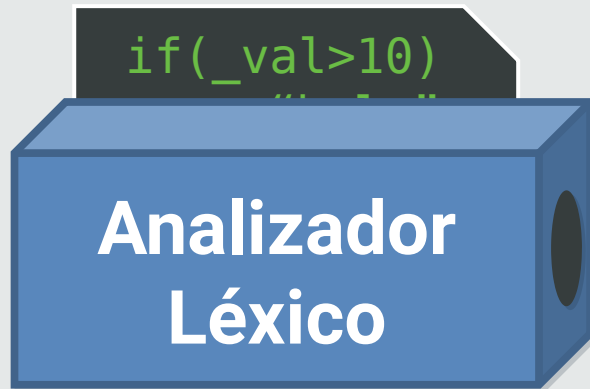
# Esquema General del Análisis Léxico

- El **Analizador Léxico** lee los caracteres directo del fuente



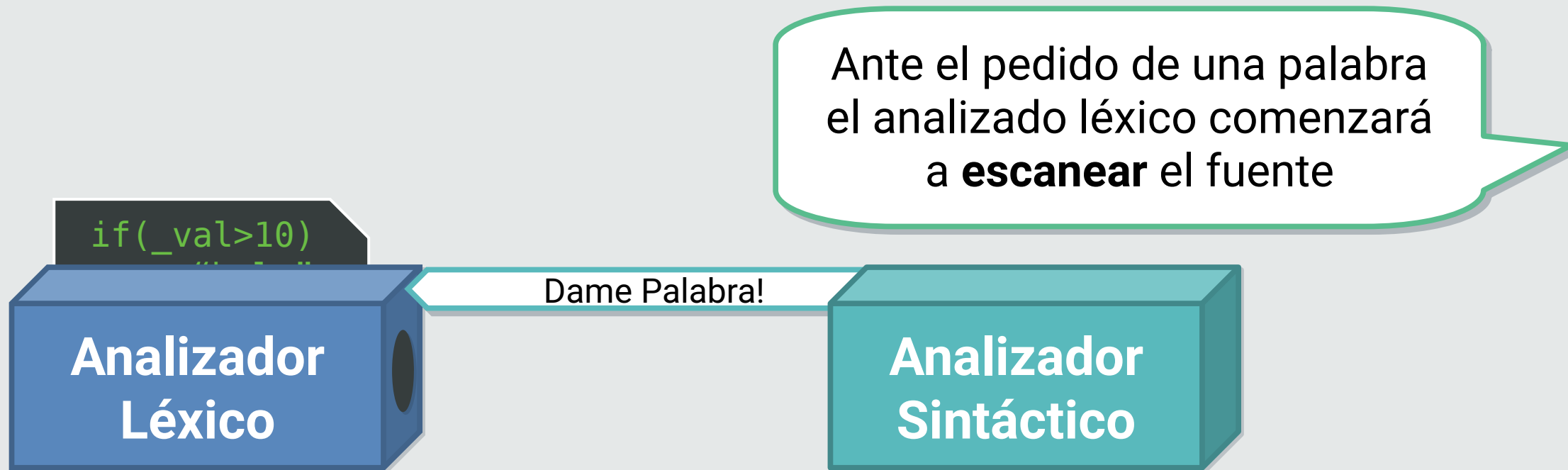
# Esquema General del Análisis Léxico

- El **Analizador Léxico** funciona por demanda!
- El analizador sintáctico irá solicitándole palabras



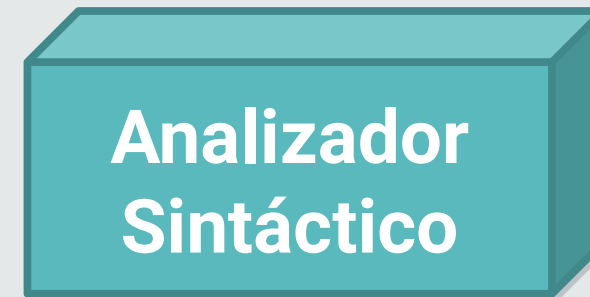
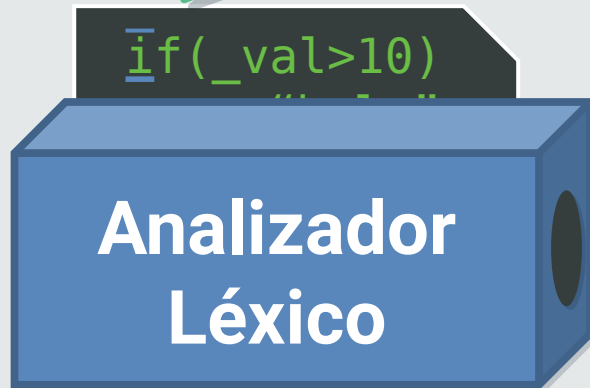
# Esquema General del Análisis Léxico

- El **Analizador Léxico** funciona por demanda!
- El analizador sintáctico irá solicitándole palabras



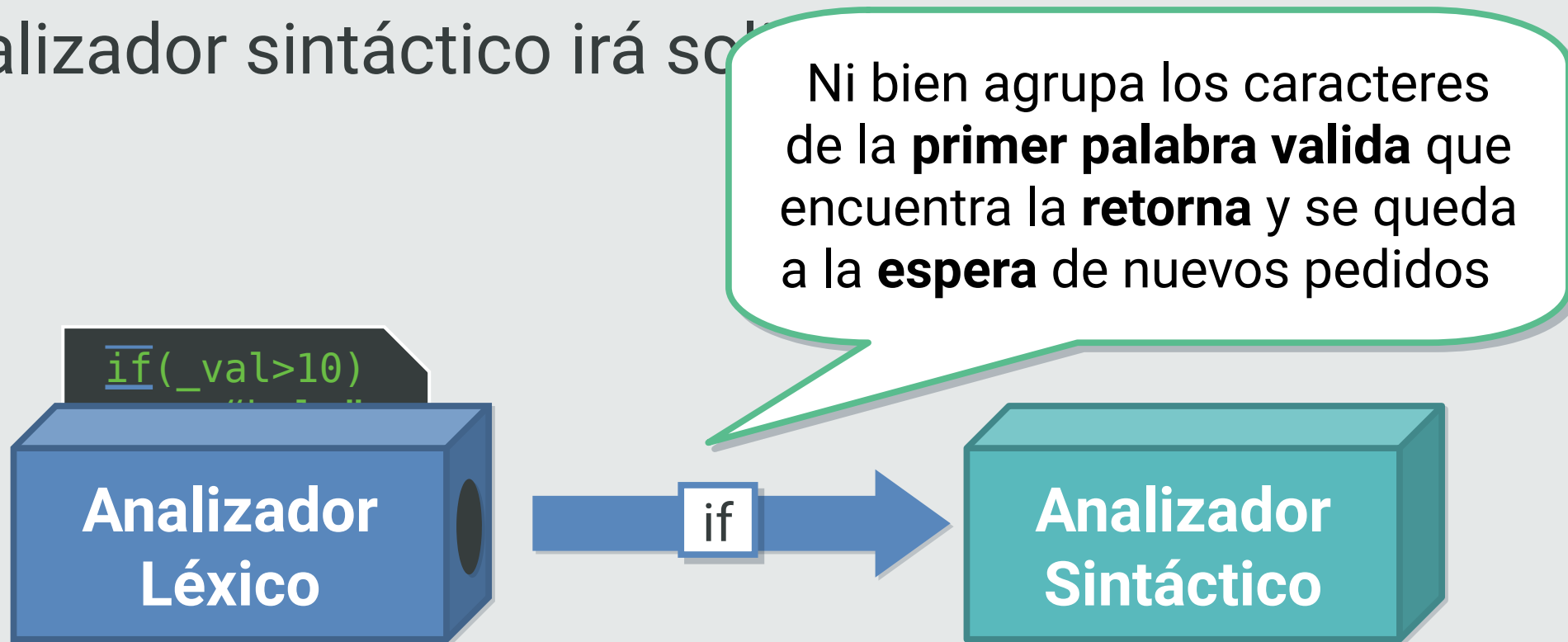
# Esquema General del Análisis Léxico

- El **Analizador Léxico** recibe el código fuente y lo procesa para extraer las palabras clave y los identificadores.
- El **analizador léxico** va recorriendo el código fuente y **agrupando caracteres** hasta que los caracteres agrupados formen una **palabra válida**.



# Esquema General del Análisis Léxico

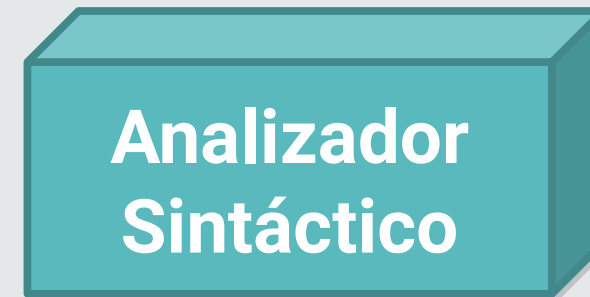
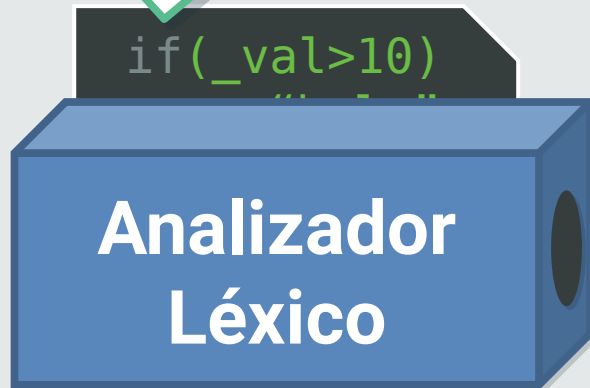
- El **Analizador Léxico** funciona por demanda!
- El analizador sintáctico irá solo





# Esquema General del Análisis Léxico

- El **Analizador Léxico** recibe el código fuente y lo transforma en tokens por demanda!
  - El **Analizador Sintáctico** recibe los tokens y los analiza formando palabras clave.
- Los **caracteres** que **usó** para formar la palabra **son consumidos** y determinan desde **dónde arrancará** a escanear ante un **nuevo pedido**



# Esquema General del Análisis Léxico

- E Los **caracteres** que **usó** para formar la palabra **son consumidos** y
- E determinan desde **dónde arrancará** a escanear ante un **nuevo pedido**

En realidad los caracteres los va consumiendo a medida que los va agrupando



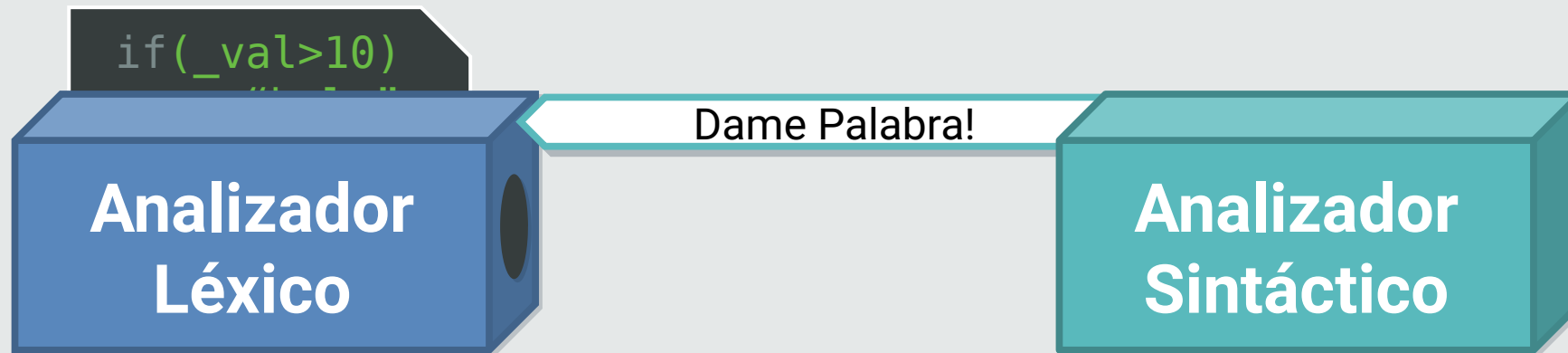
```
if(_val>10)
```

**Analizador  
Léxico**

**Analizador  
Sintáctico**

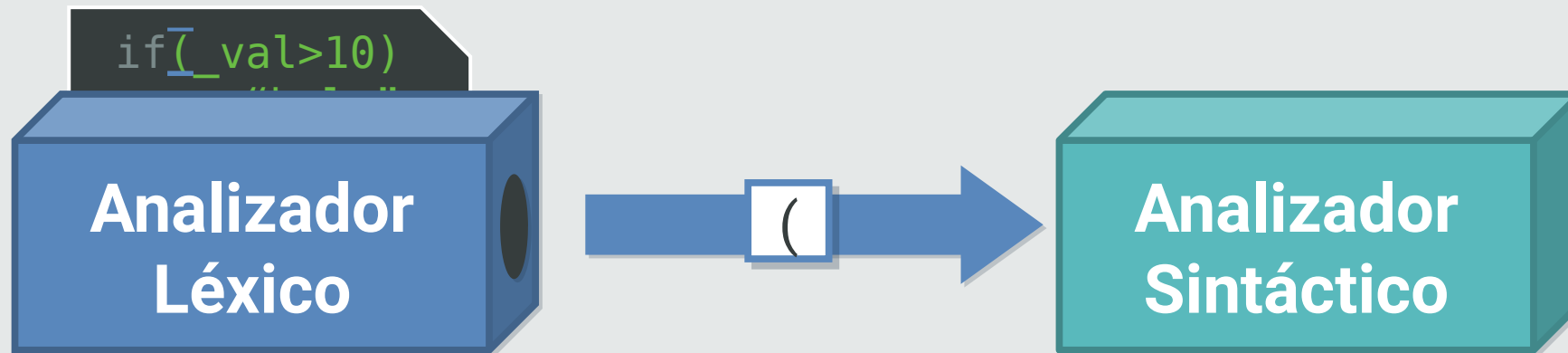
# Esquema General del Análisis Léxico

- El **Analizador Léxico** funciona por demanda!
- El analizador sintáctico irá solicitándole palabras



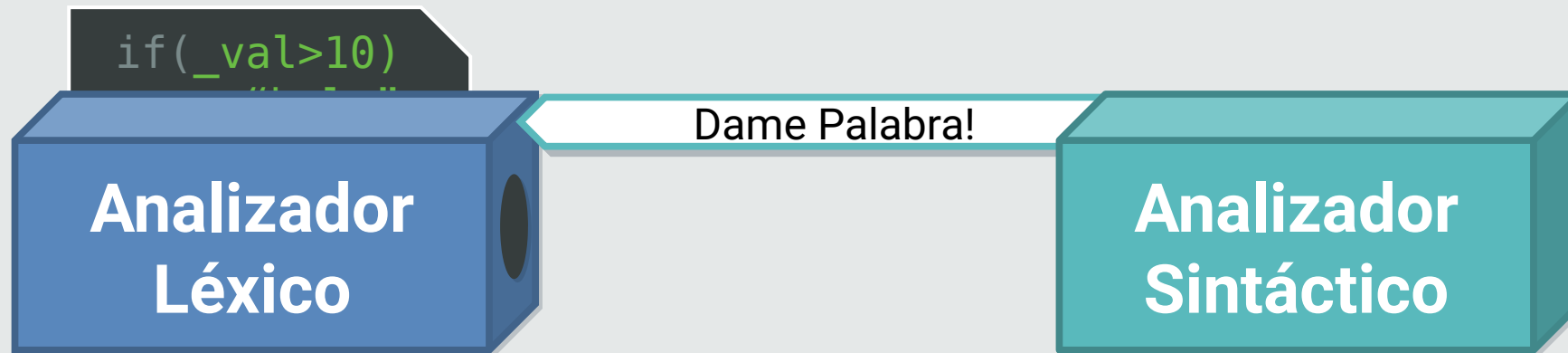
# Esquema General del Análisis Léxico

- El **Analizador Léxico** funciona por demanda!
- El analizador sintáctico irá solicitándole palabras



# Esquema General del Análisis Léxico

- El **Analizador Léxico** funciona por demanda!
- El analizador sintáctico irá solicitándole palabras



# Esquema General del Análisis Léxico

- El Analizador Léxico trabaja por demanda!
- El Analizador Léxico consume las palabras citándole palabras

Quando arranca un nuevo pedido y se encuentra con uno o mas **blancos** (espacio, tab, enter, etc) los **consume** y lo **descarta**

```
if(_val>10)
```

**Analizador  
Léxico**

**Analizador  
Sintáctico**

# Esquema General del Análisis Léxico

- El Analizador Léxico opera por demanda!
- El Analizador Léxico consume los blancos (espacio, tab, enter, etc) y los descarta

Cuando arranca un nuevo pedido y se encuentra con uno o mas **blancos** (espacio, tab, enter, etc) los **consume** y lo **descarta**

Es decir los ignora y no forman parte de la palabra valida

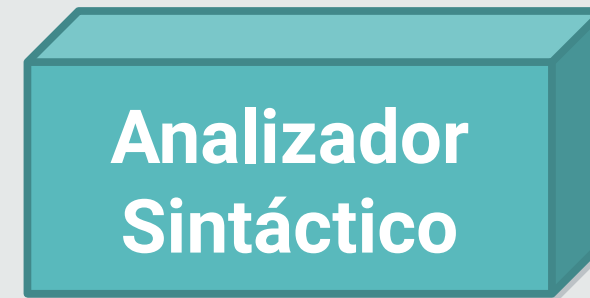
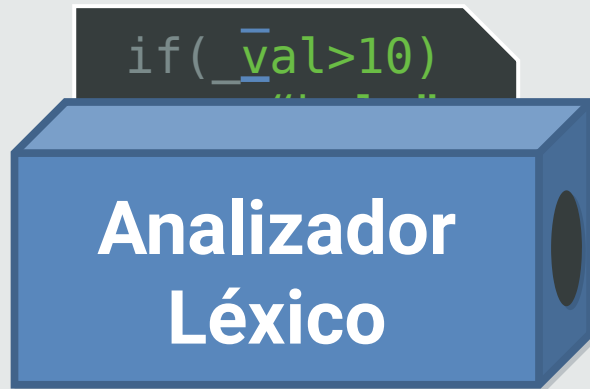
```
if(_val>10)
```

**Analizador  
Léxico**

**Analizador  
Sintáctico**

# Esquema General del Análisis Léxico

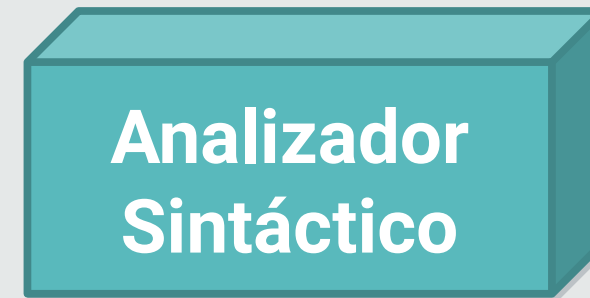
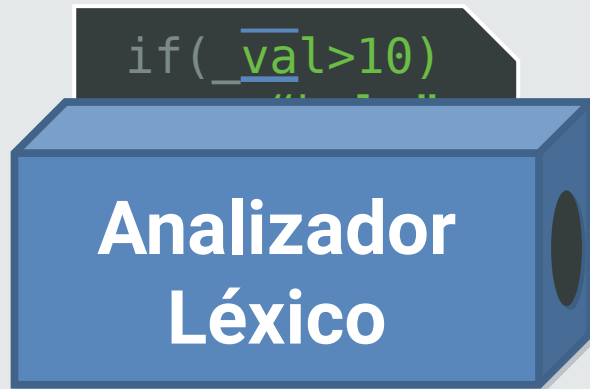
- El **Analizador Léxico** funciona por demanda!
- El analizador sintáctico irá solicitándole palabras





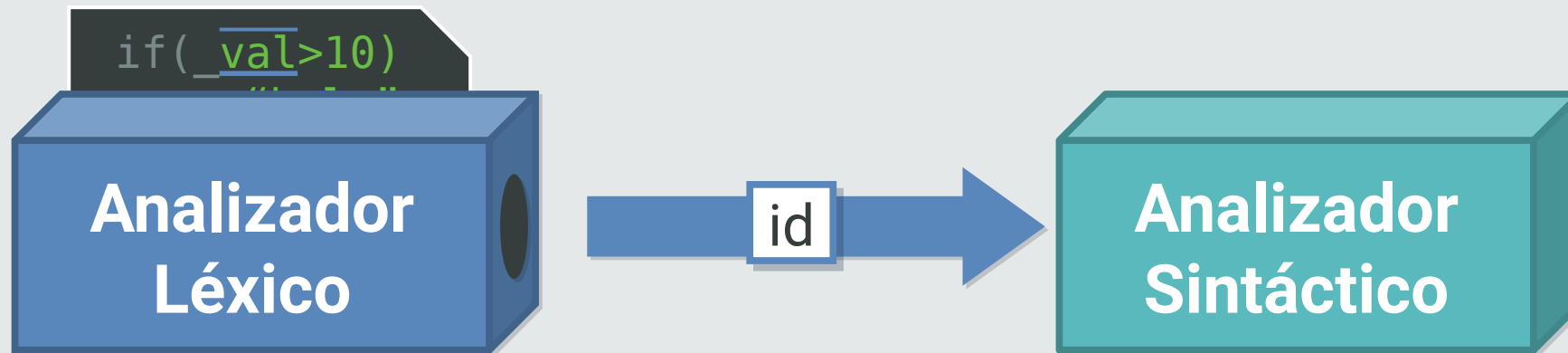
# Esquema General del Análisis Léxico

- El **Analizador Léxico** funciona por demanda!
- El analizador sintáctico irá solicitándole palabras



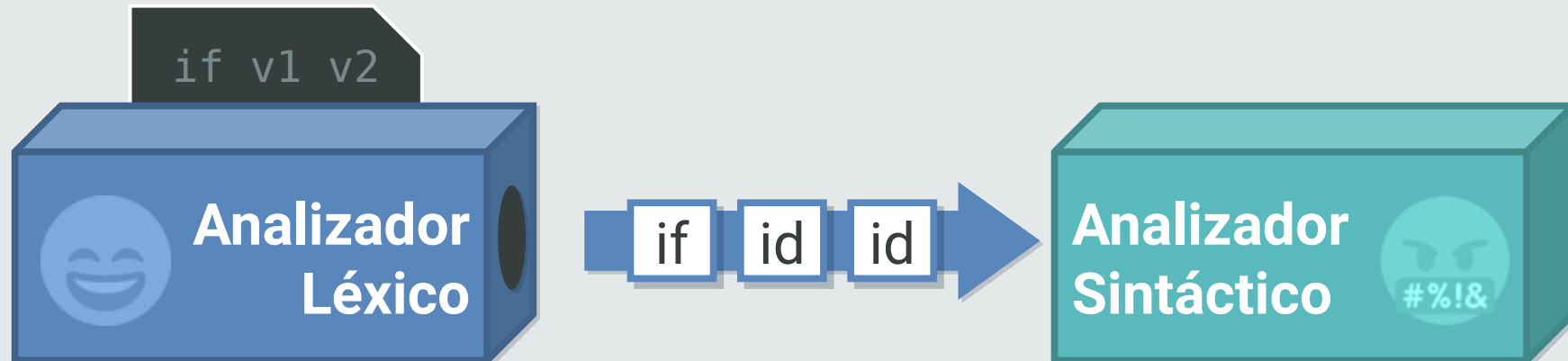
# Esquema General del Análisis Léxico

- El **Analizador Léxico** funciona por demanda!
- El analizador sintáctico irá solicitándole palabras



# Esquema General del Analizador Lexico

- Al **Analizador Léxico** no le importa el orden en que aparecen las palabras validas en el fuente!
  - Su trabajo es solamente **reconocerlas y detectar errores** cuando aparece algo que no es una palabra valida o un blanco
  - El que se encarga de ver que las **palabras validas** aparecen en **orden** correcto es el **Analizador Sintactico**



# Como seguimos?

- Las **palabras validas** son conocidas como **Tokens**
  - Los **tokens** son el eje de como vamos a desarrollar el analizador
  - Para esto vamos a ver:
    - Que características tienen los **Tokens** y como **especificarlos**
    - Como **reconocerlos** a partir de su especificación
    - Como a partir de los mecanismos para reconocerlos **implementamos** un **Analizador Léxico**

**NEXT!**