Proyecto final
Intérprete

Entrega: semana de exámenes finales

Trabajar en equipos (1 o 2 personas)

Lenguaje: Python o C++

Intérprete: Instrucciones

- Declarar variables (int, bool, float, string) int num;
- Asignación de valores variables num = 10;
- Imprimir texto y valores de variables cout<<num;
- Leer desde consola de comandos cin
- Evaluar expresiones n * 10 – 5 / 2
- Operadores
  - aritméticos (+,-,*,/),
  - relacionales (>,<,>=,<=,==,!=),
  - Lógicos (and, or, not)
- Condicional (if, else)
- Ciclos (for, while)

```
Declarar  c:entero ;
Imprimir(“Dame la calificación:”);
Leer c;
Si   c >= 6:
   Imprimir(“Aprobado”);
Else:
   Imprimir(“Reprobado”);
```

Paso 1: Definir la sintaxis (ejemplo)
(Para entregar 10 de noviembre)

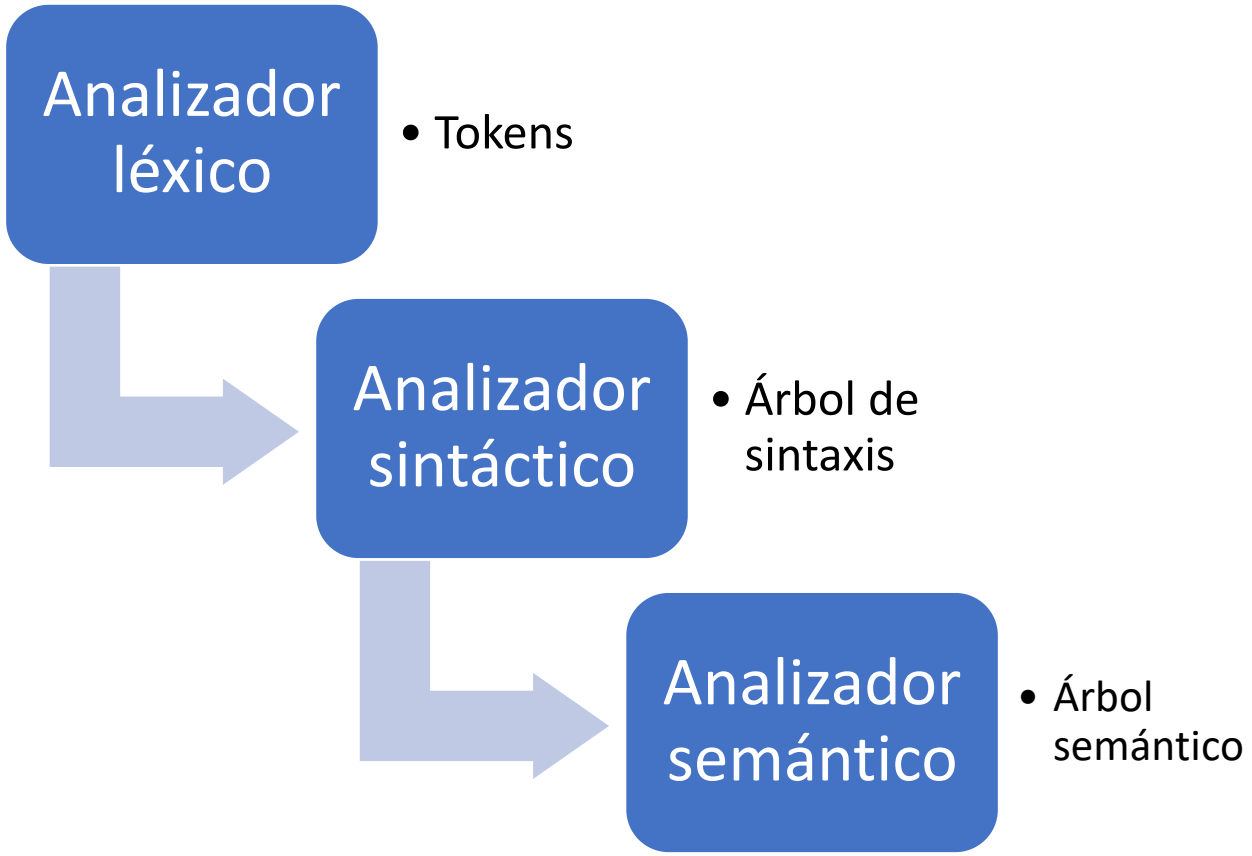
- Declarar variables (int, bool, float, string)
  Declarar nombre_variable : tipo_variable ;
  nombre_variable: empezar con letra, puede tener números y letras
  tipo_variable: cadena, entero, flotante, booleano

- Asignación de valores variables
  nombre_variable <- expresión ;
  expresión: valor, resultado de una expresión

- Imprimir texto y valores de variables
  Imprimir(expresión_imprimir);

Y así sucesivamente definir la sintaxis de cada tipo de instrucción

Etapas a seguir para generar el intérprete




```
Declarar  c : entero ;
Imprimir(“Dame la calificación:”);
Leer c;
c <- “hola”;
Si   c >= 6:
   Imprimir(“Aprobado”);
Else:
   Imprimir(“Reprobado”);
```

```
Declarar  c : flotante ;
c <- 6.9;
Si   c >= 6:
   Imprimir(“Aprobado”);
Else:
   Imprimir(“Reprobado”);
```

ANALIZADOR LÉXICO

```
Declarar  c : entero ;
Imprimir(“Dame la calificación:”);
Leer c;
c <- 7.2
Si   c >= 6:
   Imprimir(“Aprobado”);
Else:
   Imprimir(“Reprobado”);
```

Token	Tipo de token
Declarar	Palabra reservada: instrucción
c	Identificador
:	Símbolo
Entero	Palabra reservada: tipo_dato
;	Símbolo
Imprimir	Palabra reservada: instrucción
(	Símbolo
“Dame la calificación:”	Valor: cadena de texto
)	Símbolo
;	Símbolo
c	Identificador
<-	Símbolo
7.2	Valor: flotante
Si	Palabra reservada: instrucción

ANALIZADOR LÉXICO

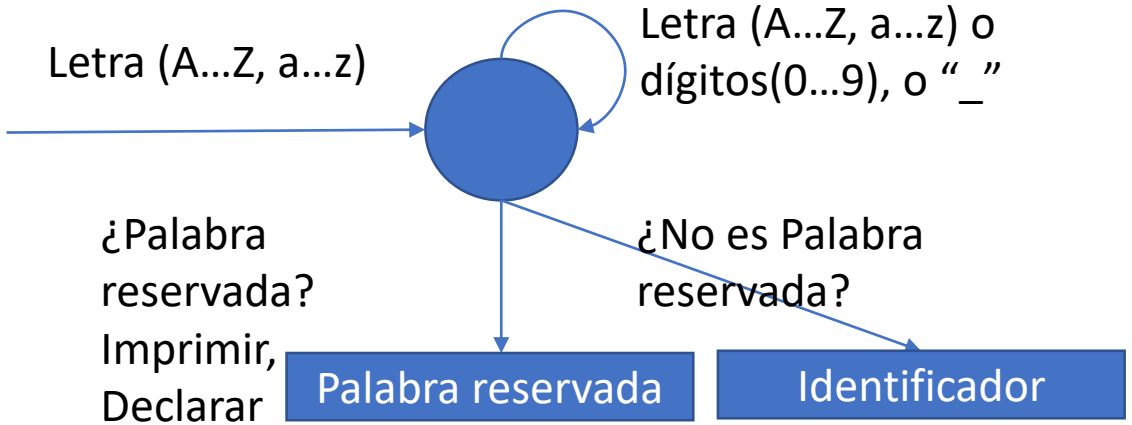
Token	# Token	Token	# Token
Palabra reservada: instrucción (Declarar, Imprimir, Para, Mientras, Si, Else)	100	Valor entero	300
Palabra reservada: tipo_dato (booleano, cadena, entero, flotante)	101	Valor flotante	301
Identificador	102	Valor cadena texto	302
Símbolo	200	Valor booleano	303
Símbolo operador aritmético (+,-,*,/)	201		
Símbolo operador relacional (<=, >=, !=, ==)	202		

Token	# de token
Declarar	100
c	102
:	200
Entero	101
;	200
Imprimir	100
(	200
“Dame la calificación:”	302
)	200
;	200
c	102
<-	200
7.2	301
Si	100

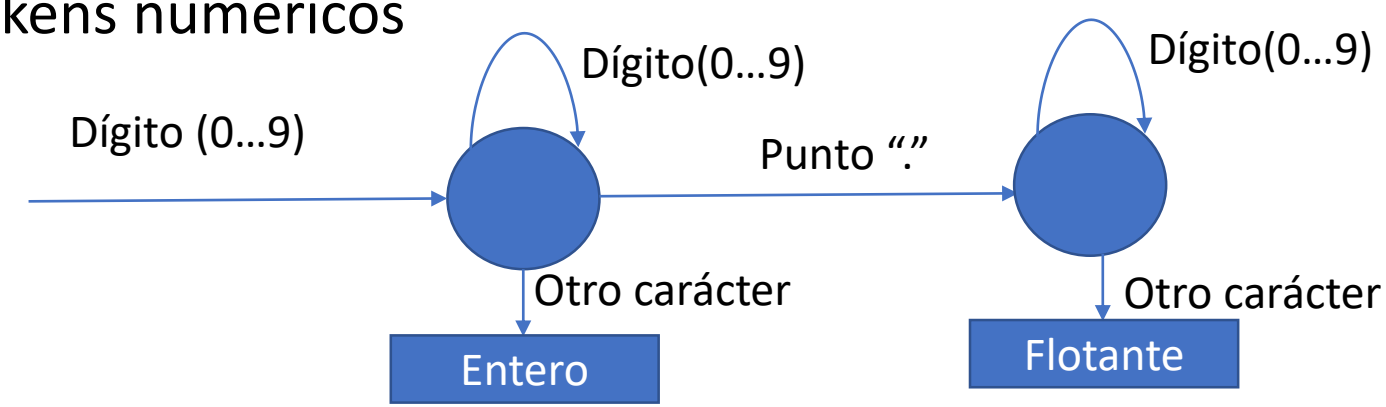
ANALIZADOR LÉXICO

```
Declarar  c : entero ;
Imprimir("Dame la calificación:");
Leer c;
c <- 77.2
Si   c >= 6392:
   Imprimir("Aprobado");
Else:
   Imprimir("Reprobado");
```

Tokens palabra clave o identificadores



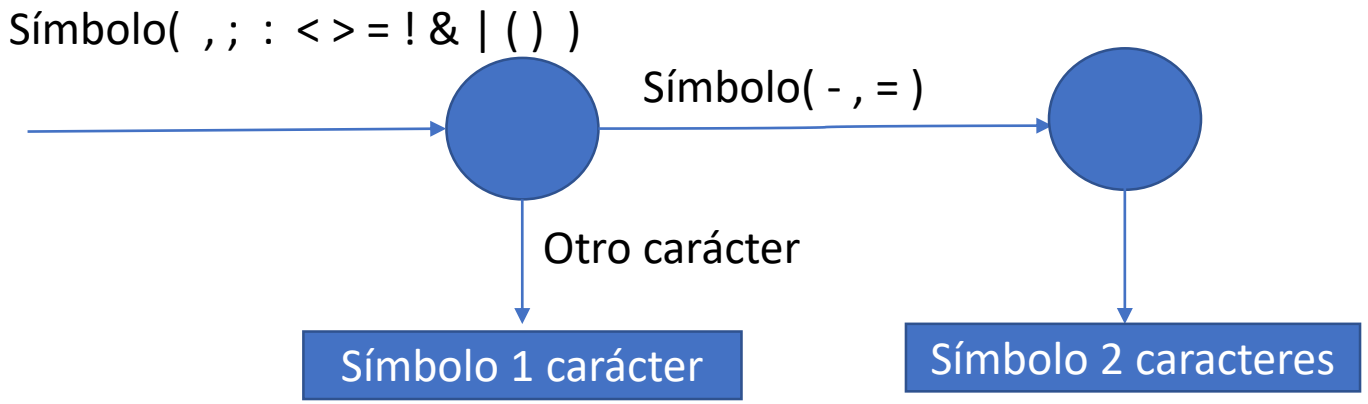
Tokens numéricos



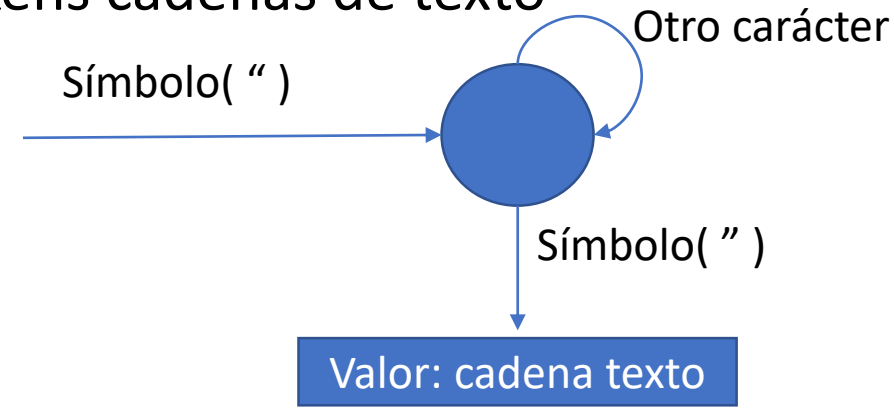
ANALIZADOR LÉXICO

```
Declarar  c : entero ;
Imprimir("Dame la calificación:");
Leer c;
c <- 77.2
Si   c >= 6392:
   Imprimir("Aprobado");
Else:
   Imprimir("Reprobado");
```

Tokens símbolos



Tokens cadenas de texto



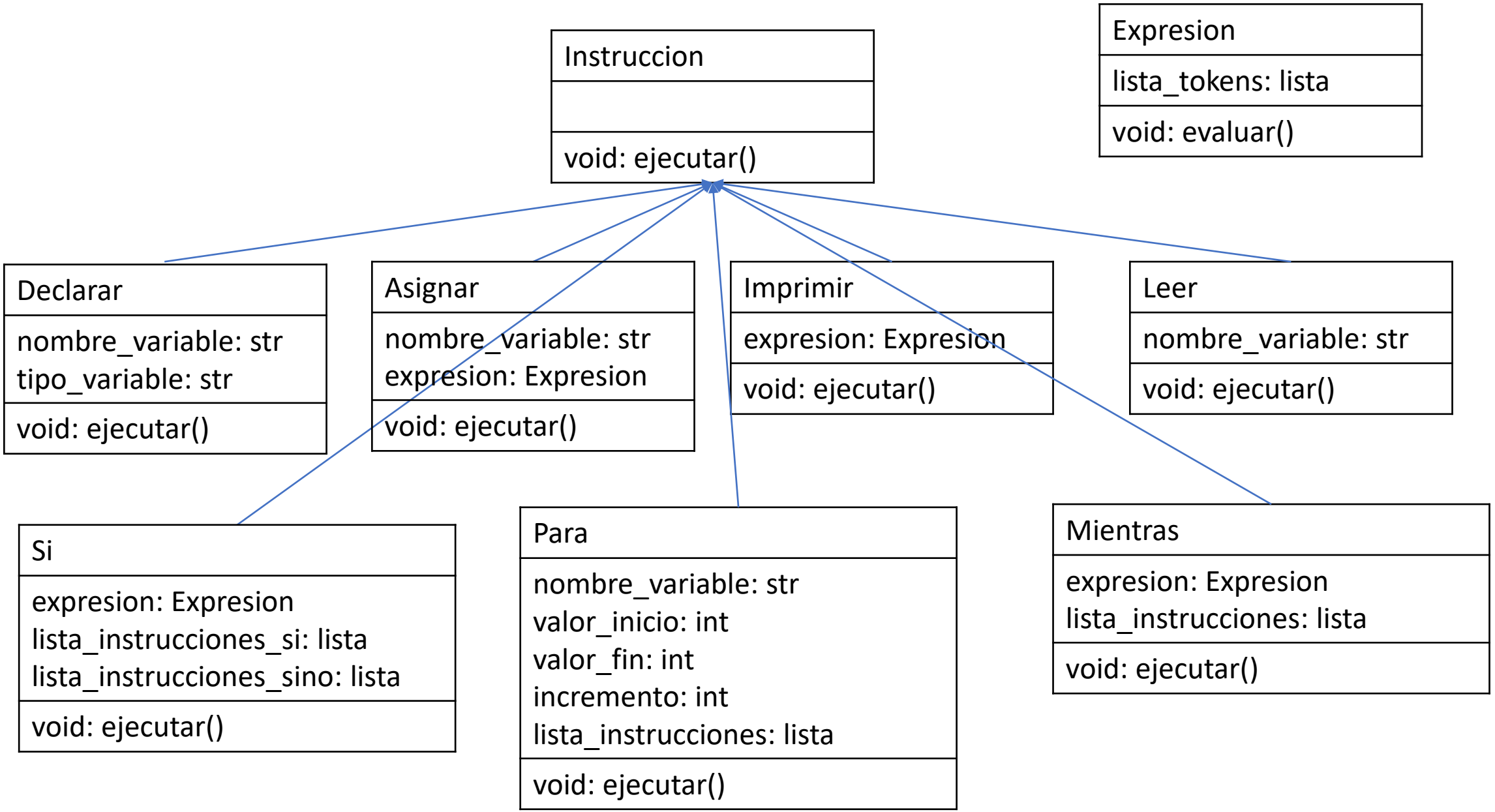
SALIDA DEL ANALIZADOR LÉXICO

Salida del analizador léxico

```
Declarar  c : entero ;
Imprimir(“Dame la calificación:”);
Leer ( c );
Si   (c >= 6){
   Imprimir(“Aprobado”);
}
```

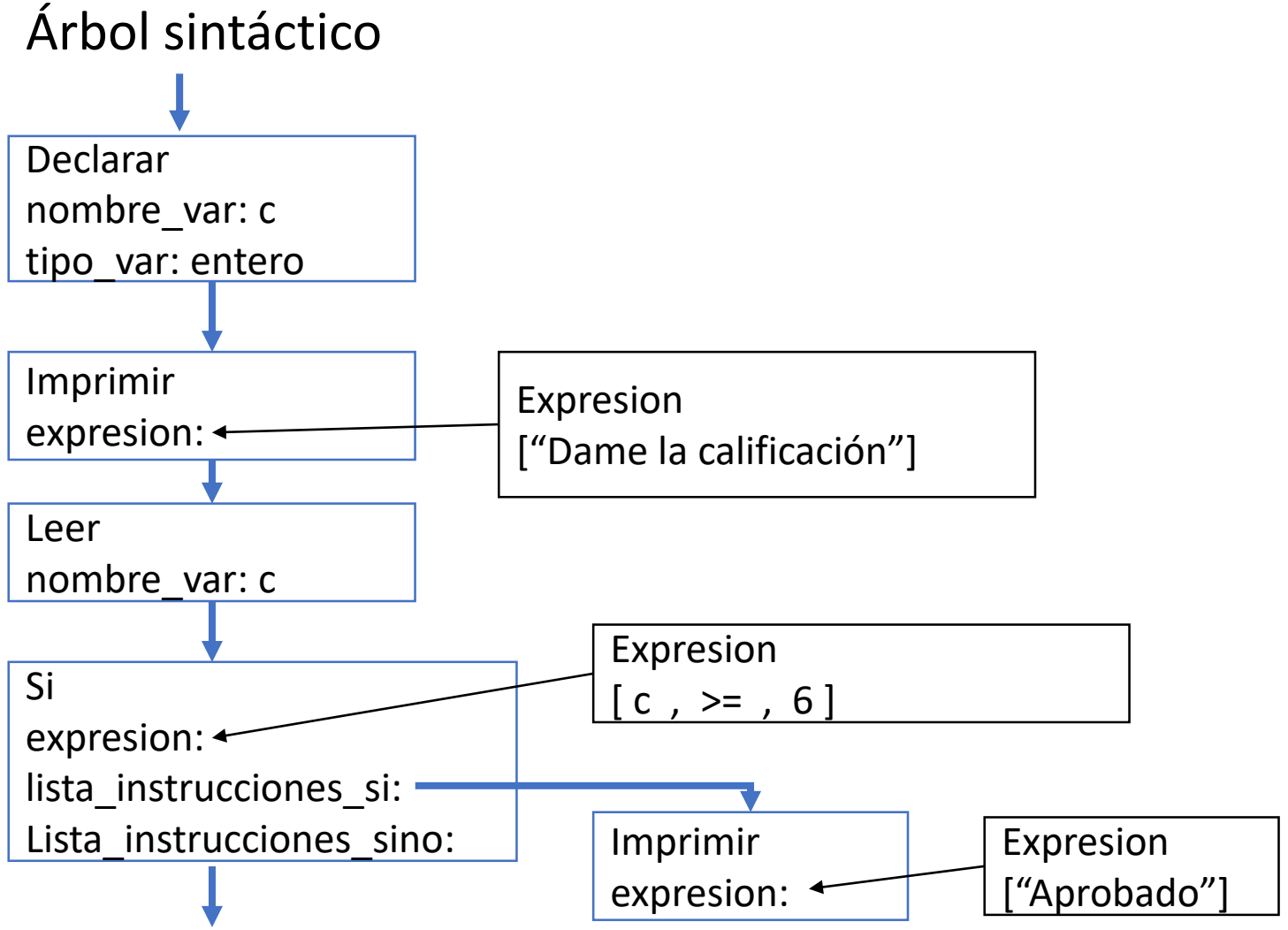
Token	#	Token	#
Declarar	100	;	303
c	102	Si	100
:	303	(	303
entero	101	c	102
;	303	>=	301
Imprimir	100	6	200
(	303	)	303
“Dame la calificación:”	203	{	303
)	303	Imprimir	100
;	303	(	303
Leer	100	“Aprobado”	203
(	303	)	303
c	102	;	303
)	303	}	303

ANALIZADOR SINTÁCTICO



ANALIZADOR SINTÁCTICO

Token	#	Token	#
Declarar	100	;	303
c	102	Si	100
:	303	(	303
entero	101	c	102
;	303	>=	301
Imprimir	100	6	200
(	303	)	303
“Dame la calificación: ”	203	{	303
)	303	Imprimir	100
;	303	(	303
Leer	100	“Aprobado”	203
(	303	)	303
c	102	;	303
)	303	}	303

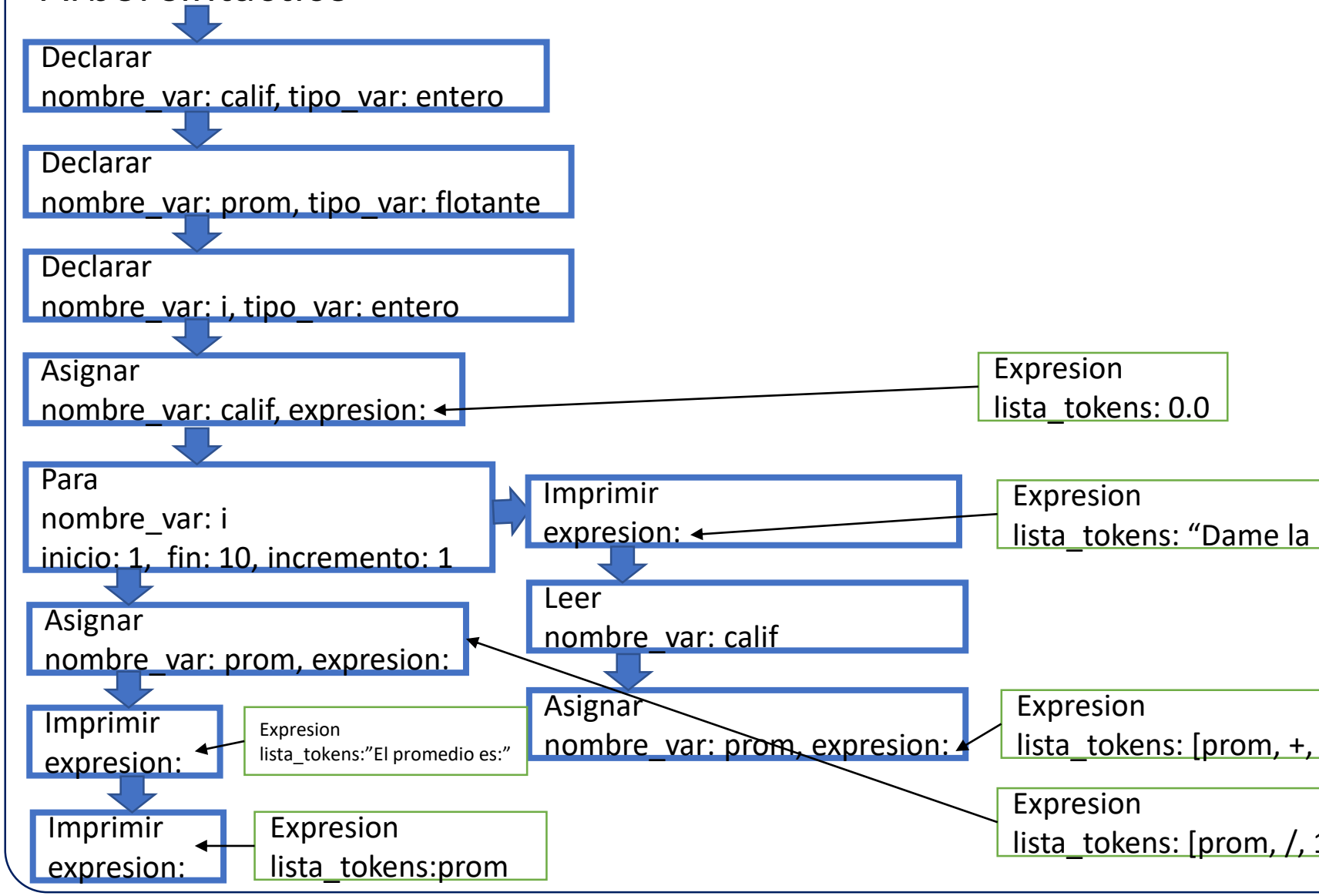


ANALIZADOR SINTÁCTICO

```
Declarar  calif : entero;  num ;
Declarar prom : flotante;
Declarar i : entero;
calif <- 0.0;
Para( i, 1, 10, 1){
  Imprimir(“Dame la calificación:”);
  Leer( calif );
  prom <- prom + calif;
}
prom <- prom/10;
Imprimir(“El promedio es:”);
Imprimir(prom);
```

TOKENS

Árbol sintáctico



ANALIZADOR SEMÁNTICO

Simulación de ejecución

- Tabla de símbolos

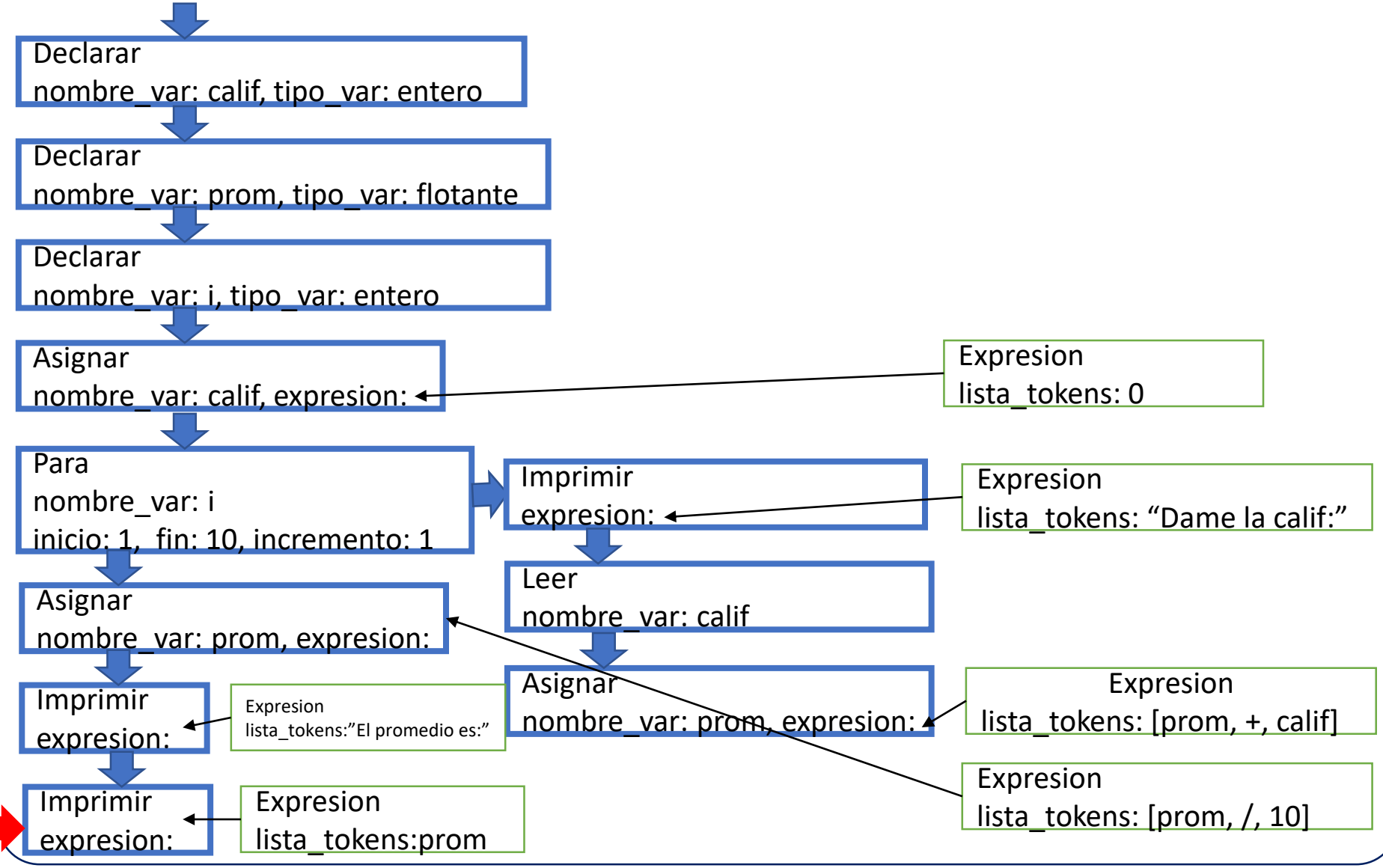
Variable	Tipo de dato	Valor
calif	entero	0
prom	flotante	0
nombre	cadena	“”

Recomendación: utilizar diccionarios

- Evaluación de expresiones
  Ejemplo: prom <- prom * 10 – 5;

ANALIZADOR SEMÁNTICO

Árbol sintáctico



Semántico

Tabla de símbolos

Variable	Tipo de dato	Valor
calif	entero	9
prom	flotante	0.9
i	entero	11

Pantalla

- Dame la calif:
- 9
- El promedio es:
- 0.9

Evaluación de expresiones

Expresión: 10 – num * calif + 2

Paso 1: reemplazar los nombres de variable por sus valores

Expresión: 10 – 10 * 9 + 2

Paso 2: Convertir la expresión a notación postfija

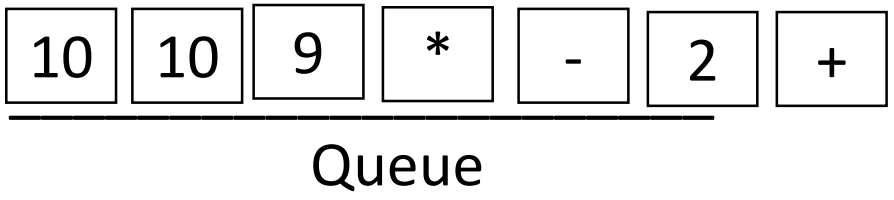
Paso 3: Evaluar la expresión postfija

Tabla de símbolos

Variable	Tipo de dato	Valor
calif	entero	9
prom	flotante	0.9
i	entero	11
num	entero	10

Evaluación de expresiones: notación postfija

Expresión: 10 – 10 * 9 + 2



- Mientras haya tokens
  - Lea un token.
  - Si el token es un valor o un número, entonces agréguelo
  - Si el token es un operador, o_1, entonces:
    - Mientras que haya un operador o_2 en el tope de
      o igual prioridad a la de o_1,
      - Retire (pop) de la pila el o_2, y póngalo
    - Ponga (push) o_1 en el tope de la pila.
  - Si el token es un paréntesis abierto, entonces póngalo en
  - Si el token es un paréntesis derecho:
    - Hasta que el token en el tope de la pila sea un
      pila y colóquelos en la cola de salida.
    - Retire (pop) el paréntesis abierto de la pila, per
    - Si la pila se termina sin encontrar un paréntesis
- Cuando no hay más tokens para leer:
  - Mientras todavía haya tokens de operadores en la pila:
    - Si el token del operador en el tope de la pila es
      correspondiente.
    - Retire (pop) al operador y póngalo en la cola de
- Fin.

Evaluación de expresiones: evaluación

Expresión: 10 – 10 * 9 + 2

10 – 90 + 2

- 80 + 2

-78

=

-78

Stack valores

Aplicación de operadores

Planeación:

- Definición sintaxis del lenguaje – Martes 10 Nov (5%)
- Analizador léxico – Martes 17 Nov (5%)
- Analizador sintáctico – Martes 24 Nov (5%)
- Analizador semántico – Martes 1 Dic