



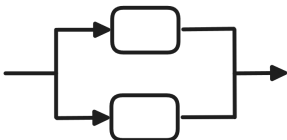
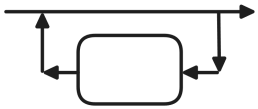

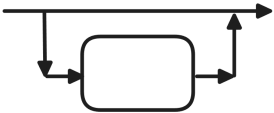
## Conceptos y Paradigmas de Lenguajes de Programación 2023

### Práctica Nro. 2

#### Sintaxis

**Objetivo:** conocer cómo se define léxicamente un lenguaje de programación y cuales son las herramientas necesarias para hacerlo

**Ejercicio 1:** Complete el siguiente cuadro:

Meta símbolos utilizados por		Símbolo utilizado en Diagramas sintacticos	Significado
BNF	EBNF		
Palabra terminal	Palabra terminal		Definición de un elemento terminal
Palabra no terminal	Palabra no terminal		Definición de un elemento no terminal
::=	::=	Diagrama con rectángulos, óvalos y flechas	Metasímbolo "Se define como"
	( )		Selección de una alternativa
<p> <p1>	{ }		Repetición
	*		Repetición de 0 o más veces
	+		Repetición de 1 o más veces
	[]		Opcional

Nota: p y p1 son producciones simbólicas

**Ejercicio 2:** ¿Cuál es la importancia de la sintaxis para un lenguaje? ¿Cuáles son sus elementos?

La sintaxis es fundamental en un lenguaje de programación porque define las **reglas** y la estructura gramatical que deben seguirse para escribir un **código válido**. Específicamente, la sintaxis dicta cómo deben organizarse y combinarse los elementos del lenguaje para crear instrucciones comprensibles para la computadora. Indica **cómo** se define un lenguaje.

Sus elementos son:

- Alfabeto o conjunto de caracteres
- Identificadores
- Operadores
- Palabra clave y palabra reservada
- Comentarios y uso de blancos

**Ejercicio 3:** ¿Explique a qué se denomina regla lexicográfica y regla sintáctica?

Las **reglas lexicográficas** especifican cómo se reconocen y clasifican los elementos básicos de la sintaxis (alfabeto, identificadores, operadores, palabras claves y reservadas, etc). Por ejemplo, se indica que "if" es una palabra clave, que "1, 2, 3, 4, 5, 6, 7, 8, 9, 0" son Int y que "&&" es un operador lógico.

Las **reglas sintácticas** se refieren a la estructura gramatical del código y cómo se combinan los elementos básicos para formar instrucciones y expresiones válidas en el lenguaje de programación. Estas reglas definen la jerarquía y la organización de las construcciones sintácticas, como declaraciones, expresiones, bucles y funciones.

**Ejercicio 4:** ¿En la definición de un lenguaje, a qué se llama palabra reservadas? ¿A qué son equivalentes en la definición de una gramática? De un ejemplo de palabra reservada en el lenguaje que más conoce. (Ada,C,Ruby,Python,...)

Las palabras reservadas son términos específicos que están reservados por el propio lenguaje para un uso particular. Estas palabras tienen un significado predefinido en la sintaxis del lenguaje y no pueden ser redefinidas por el programador para otros propósitos, como el nombre de variables o funciones.

**Ejercicio 5:** Dada la siguiente gramática escrita en BNF:

$G = (N, T, S, P)$

$N = \{ \langle \text{numero\_entero} \rangle, \langle \text{digito} \rangle \}$

$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$S = \langle \text{numero\_entero} \rangle$

$P = \{$   
     $\langle \text{numero\_entero} \rangle ::= \langle \text{digito} \rangle \langle \text{numero\_entero} \rangle \mid \langle \text{numero\_entero} \rangle \langle \text{digito} \rangle \mid$   
     $\langle \text{digito} \rangle$   
  
     $\langle \text{digito} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$   
 $\}$

**a-** Identifique las componentes de la misma

$G$  = Definición de la gramática, que está formada por 4 conjuntos

$N$  = Conjunto de símbolos no terminales

$T$  = Conjunto de símbolos terminales

$S$  = Símbolo distinguido de la gramática que pertenece a  $N$

$P$  = Conjunto de producciones

**b-** Indique por qué es ambigua y corrija la

“Una gramática es ambigua si una sentencia puede derivarse de más de una forma.”

Es decir, podríamos dibujar dos árboles de derivación distintos e igualmente válidos.

En este caso, en la producción de  $\langle \text{numero\_entero} \rangle$ , se usa tanto recursividad a derecha como recursividad a izquierda, lo que no es correcto y la hace ambigua.

La forma correcta, es usar sólo una de las dos:

$\langle \text{numero\_entero} \rangle ::= \langle \text{digito} \rangle \mid \langle \text{digito} \rangle \langle \text{numero\_entero} \rangle$

**Ejercicio 6:** Defina en BNF (Gramática de contexto libre desarrollada por Backus-Naur) la gramática para la definición de una palabra cualquiera.

```
G = ( N, T, S, P)
N = {<palabra>, <mayuscula>, <minuscula> }
T = {"A", "B", "C" ... "X", "Y", "Z", "a", "b", "c", ... "x", "y", "z"}
S = <palabra>
P = {
    <palabra> ::= <mayuscula><palabra> | <minuscula><palabra> | <mayuscula> |
    <minuscula>

    <mayuscula> ::= "A" | "B" | "C" | ... | "X" | "Y" | "Z"
    <minuscula> ::= "a" | "b" | "c" | ... | "x" | "y" | "z"
}
```

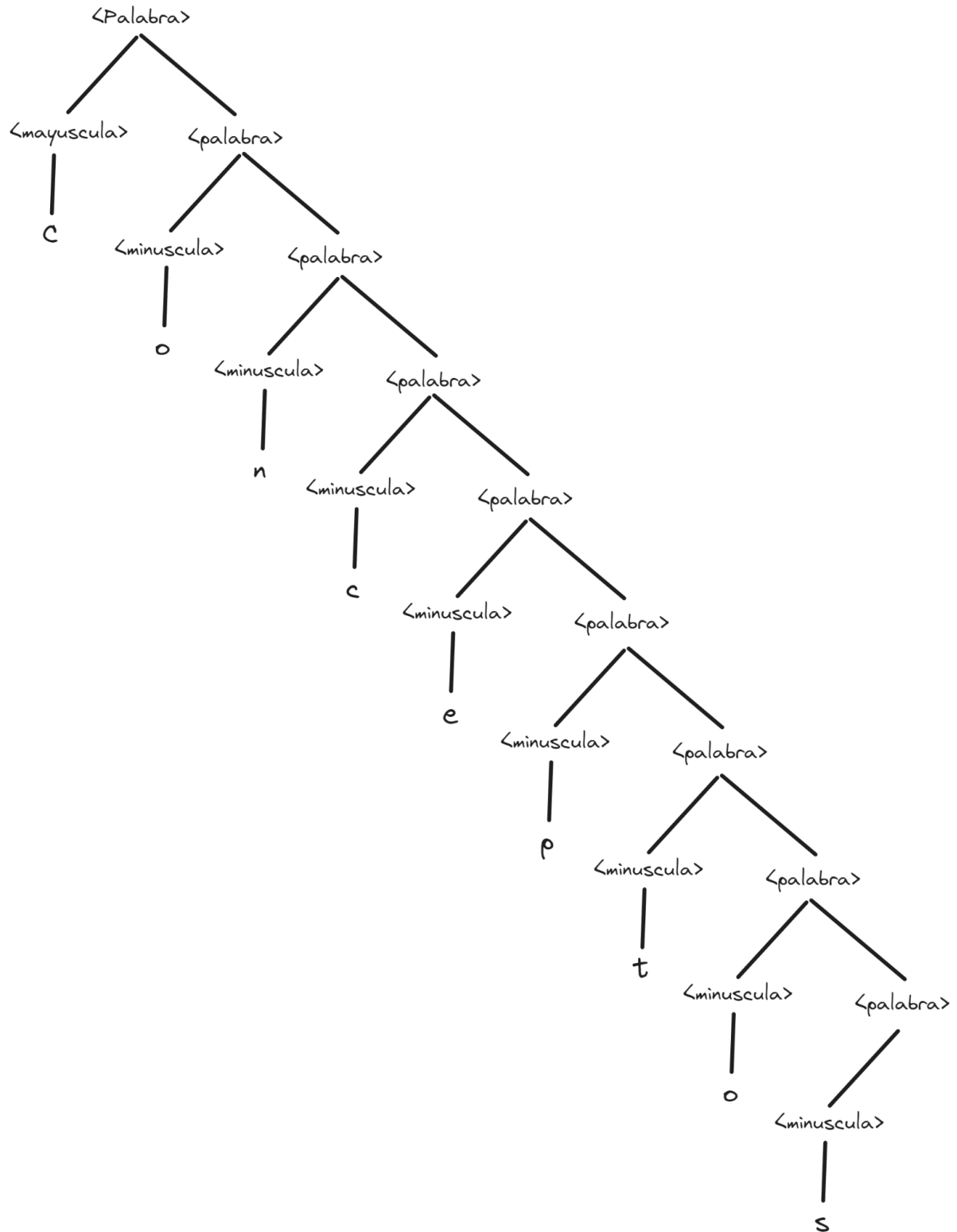
**Ejercicio 7:** Defina en EBNF la gramática para la definición de números reales.

```
G = ( N, T, S, P)
N = {<numero_real>, <digito>, <separador>, <signo> }
T = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ".", ",", +, -}
S = <numero_real>
P = {
    <numero_real> ::= [signo] {<digito>}+ [ <separador> {<digito>}+ ]

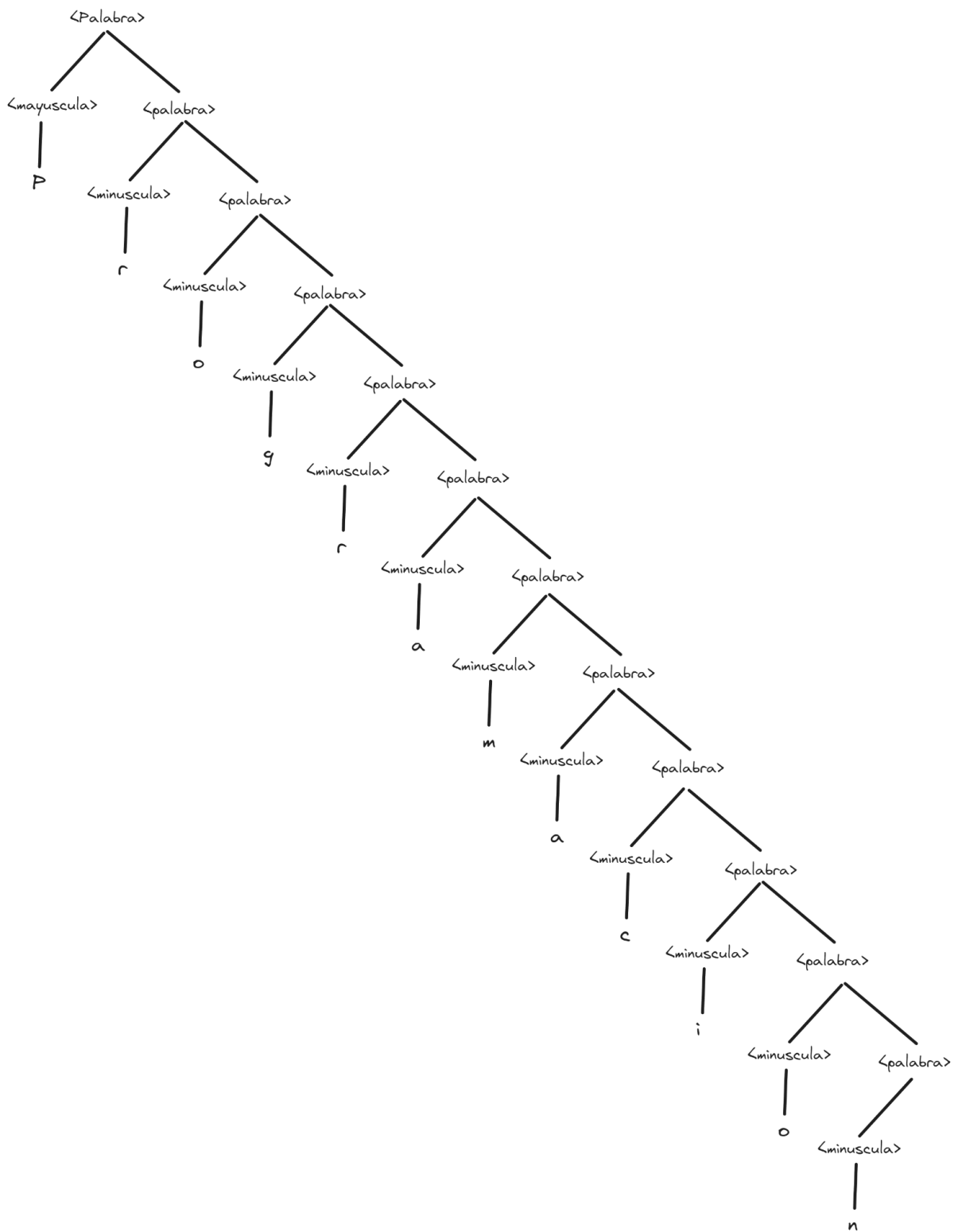
    <digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
    <separador> ::= "," | "."
    <signo> ::= + | -
}
```

**Ejercicio 8:** Utilizando la gramática que desarrolló en los puntos 6 y 7, escriba el árbol sintáctico de:

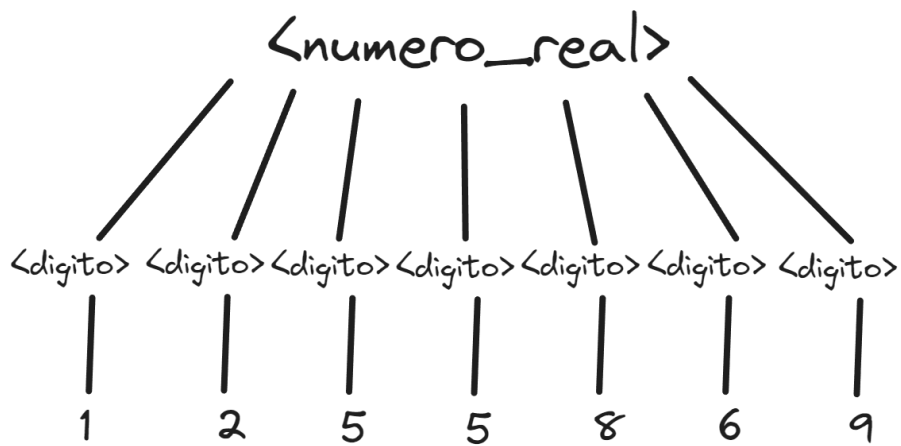
**a. Conceptos**



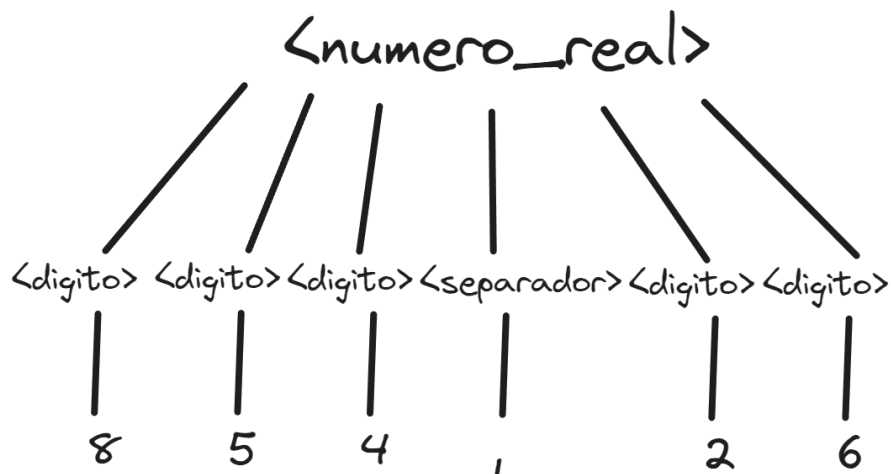
## b. Programación



c. 1255869



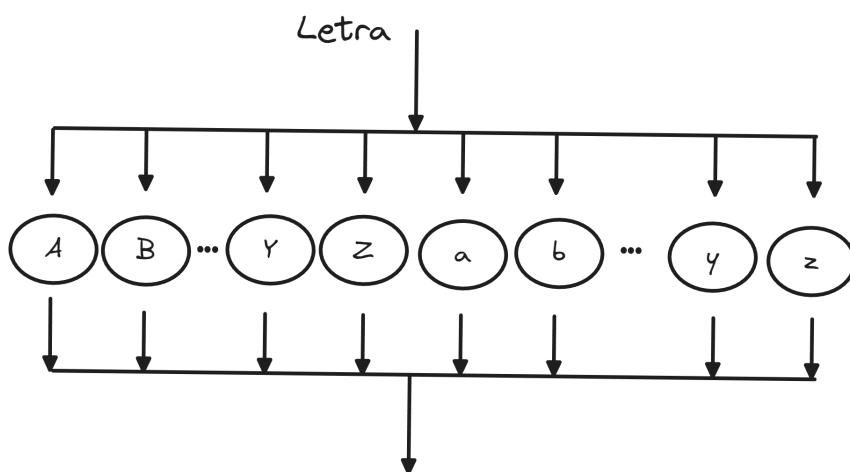
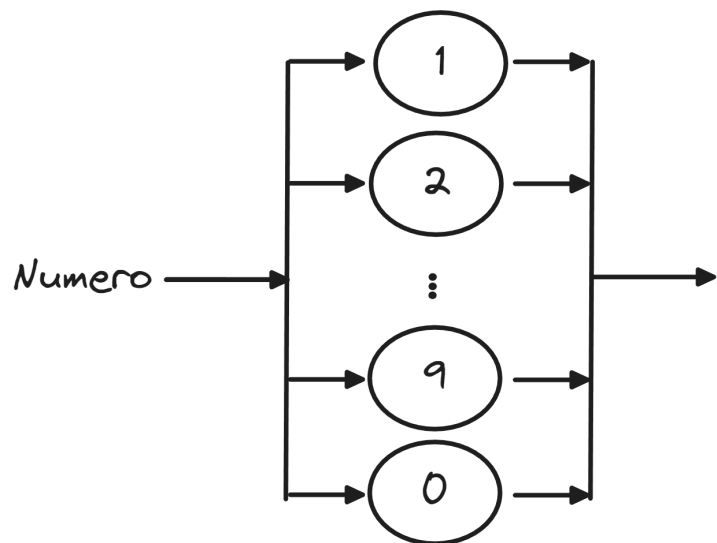
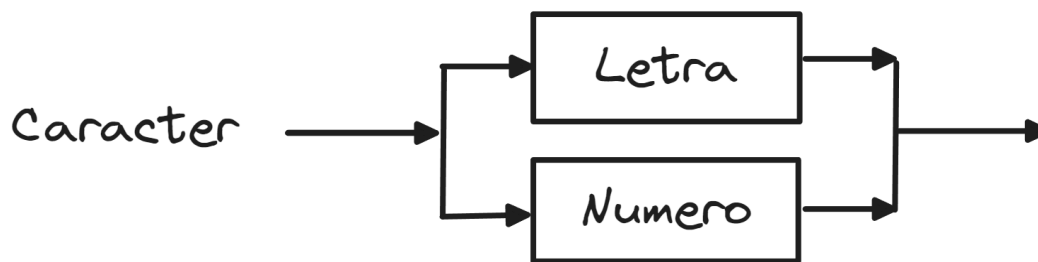
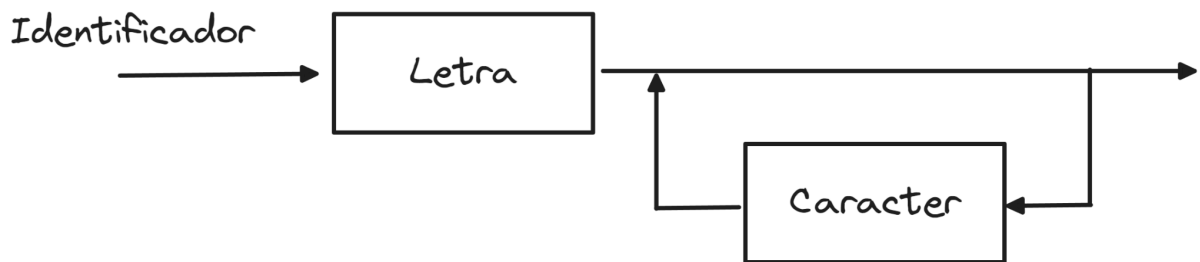
d. 854,26



### e. Conceptos de lenguajes

En la gramática definida en el punto 6, no está el No Terminal para el espacio, ergo no se puede representar un conjunto de palabras (String), solo palabras sueltas.

**Ejercicio 9:** Defina utilizando diagramas sintácticos la gramática para la definición de un identificador de un lenguaje de programación. Tenga presente como regla que un identificador no puede comenzar con números.





**Ejercicio 10:**

**a)** Defina con EBNF la gramática para una expresión numérica, donde intervienen variables y números. Considerar los operadores +, -, \* y / sin orden de prioridad. No considerar el uso de paréntesis.

$G = (N, T, S, P)$

$N = \{ \langle \text{expresion} \rangle, \langle \text{numero\_real} \rangle, \langle \text{digito} \rangle, \langle \text{separador} \rangle, \langle \text{signo} \rangle, \langle \text{ID} \rangle \langle \text{elem} \rangle \}$

$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ".", ",", "+, -, "a", "b", \dots "y", "z", "A", "B", \dots "Y", "Z"\}$

$S = \langle \text{expresion} \rangle$

$P = \{$

$\langle \text{expresion} \rangle ::= \langle \text{elem} \rangle [ \{ \langle \text{operador} \rangle \langle \text{elem} \rangle \}^+ ]$

$\langle \text{operador} \rangle ::= \{ +, -, *, / \}$

$\langle \text{elem} \rangle ::= \langle \text{ID} \rangle \mid \langle \text{numero\_real} \rangle$

$\langle \text{numero\_real} \rangle ::= \{ \langle \text{digito} \rangle \}^+ [ \langle \text{separador} \rangle \{ \langle \text{digito} \rangle \}^+ ]$

$\langle \text{digito} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\langle \text{separador} \rangle ::= "." \mid ","$

$\langle \text{ID} \rangle ::= \langle \text{letra} \rangle \{ \langle \text{letra} \rangle \mid \langle \text{digito} \rangle \}^*$

$\langle \text{letra} \rangle ::= "a" \mid "b" \dots "y" \mid "z" \mid "A" \mid "B" \mid \dots \mid "Y" \mid "Z"$

$\}$

**b) A la gramática definida en el ejercicio anterior agregarle prioridad de operadores.**

$G = (N, T, S, P)$

$N = \{ \langle \text{expresion} \rangle, \langle \text{numero\_real} \rangle, \langle \text{digito} \rangle, \langle \text{separador} \rangle, \langle \text{signo} \rangle, \langle \text{ID} \rangle, \langle \text{elemento} \rangle, \langle \text{termino} \rangle, \langle \text{operador\_debil} \rangle, \langle \text{operador\_fuerte} \rangle \}$

$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \text{"."}, \text{","}, \text{"+"}, \text{"-"}, \text{"a"}, \text{"b"}, \dots \text{"y"}, \text{"z"}, \text{"A"}, \text{"B"}, \dots \text{"Y"}, \text{"Z"}\}$

$S = \langle \text{expresion} \rangle$

$P = \{$

$\langle \text{expresion} \rangle ::= \text{term} \{ \langle \text{operador\_debil} \rangle \langle \text{term} \rangle \}^+$

$\langle \text{término} \rangle ::= \langle \text{elem} \rangle \{ \langle \text{operador\_fuerte} \rangle \langle \text{elem} \rangle \}^+$

$\langle \text{operador\_debil} \rangle ::= \{ + \mid - \}$

$\langle \text{operador\_fuerte} \rangle ::= \{ * \mid / \}$

$\langle \text{elem} \rangle ::= \langle \text{ID} \rangle \mid \langle \text{numero\_real} \rangle$

$\langle \text{numero\_real} \rangle ::= \{ \langle \text{digito} \rangle \}^+ [ \langle \text{separador} \rangle \{ \langle \text{digito} \rangle \}^+ ]$

$\langle \text{digito} \rangle ::= (0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9)$

$\langle \text{separador} \rangle ::= ( \text{","} \mid \text{"."} )$

$\langle \text{ID} \rangle ::= \langle \text{letra} \rangle \{ ( \langle \text{letra} \rangle \mid \langle \text{digito} \rangle ) \}^*$

$\langle \text{letra} \rangle ::= ( \text{"a"} \mid \text{"b"} \dots \text{"y"} \mid \text{"z"} \mid \text{"A"} \mid \text{"B"} \mid \dots \mid \text{"Y"} \mid \text{"Z"} )$

$\}$

**c) Describa con sus palabras los pasos y decisiones que tomó para agregarle prioridad de operadores al ejercicio anterior.**

Una expresión se define como un término, un "+" o un "-", y otro término.

Y el término se define como un elemento, un "\*" o un "/", otro elemento.

De esta manera, se separa en términos, dándole prioridad a los operadores fuertes

**Ejercicio 11:** La siguiente gramática intenta describir sintácticamente la sentencia for de ADA, indique cuál/cuáles son los errores justificando la respuesta.

$N = \{ \langle \text{sentencia\_for} \rangle, \langle \text{bloque} \rangle, \langle \text{variable} \rangle, \langle \text{letra} \rangle, \langle \text{cadena} \rangle, \langle \text{digito} \rangle, \langle \text{otro} \rangle, \langle \text{operacion} \rangle, \langle \text{llamada\_a\_funcion} \rangle, \langle \text{numero} \rangle, \langle \text{sentencia} \rangle \}$

$P = \{ \quad \langle \text{sentencia\_for} \rangle ::= \text{for (i= IN 1..10) loop } \langle \text{bloque} \rangle \text{ end loop};$

$\quad \langle \text{variable} \rangle ::= \langle \text{letra} \rangle | \langle \text{cadena} \rangle$

$\quad \langle \text{cadena} \rangle ::= \{ (\langle \text{letra} \rangle | \langle \text{digito} \rangle | \langle \text{otro} \rangle) \}^+$

$\quad \langle \text{letra} \rangle ::= ( a | .. | z | A | .. | Z )$

$\quad \langle \text{digito} \rangle ::= ( 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 )$

$\quad \langle \text{bloque} \rangle ::= \langle \text{sentencia} \rangle | \langle \text{sentencia} \rangle \langle \text{bloque} \rangle | \langle \text{bloque} \rangle \langle \text{sentencia} \rangle;$

$\quad \langle \text{sentencia} \rangle ::= \langle \text{sentencia\_asignacion} \rangle | \langle \text{llamada\_a\_funcion} \rangle |$

$\quad \langle \text{sentencia\_if} \rangle | \langle \text{sentencia\_for} \rangle | \langle \text{sentencia\_while} \rangle | \langle \text{sentencia\_switch} \rangle$

$\}$

**La gramática es una tupla formada por N, T, S y P**

En este caso faltan los Terminales y Símbolo distinguido

**Todos los No Terminales deben estar definidos en una producción**

No define que es  $\langle \text{otro} \rangle$ ,  $\langle \text{sentencia\_if} \rangle$ ,  $\langle \text{llamada\_a\_funcion} \rangle$ , etc...:

**No usa una sola manera de definir la sintaxis**

Usa sentencias de EBNF como  $\{ abc \}^*$  y a la vez usa recursión del BNF.

Eso no es correcto.

Además, si estuviese usando BNF, la definición del bloque es ambigua, ya que usa recursión tanto a derecha como a izquierda

**Ejercicio 12:** Realice en EBNF la gramática para la definición un tag div en html 5.  
(Puede ayudarse con el siguiente enlace  
(<https://developer.mozilla.org/es/docs/Web/HTML/Elemento/div>)

$G = (N, T, S, P)$

$N = \{<div>, <div\_apertura>, <div\_cierre>, <contenido>, <atributo\_class>, <atributo\_id>, <caracteres>\}$

$T = \{ >, <, /, 1, 2, \dots 9, 0, a, b, \dots y, z, A, B, \dots, Y, Z, \text{“ “}, =, !, \text{“ “}, :, _ \}$

$S = \{<div>\}$

$P = \{$   
 $<div> ::= \{<div\_apertura>, [<contenido>], <div\_cierre>$

$<div\_apertura> ::= \{<div, [\text{“ “}, <atributo class> \mid <atributo\_id>], >\}$

$<div\_cierre> ::= \{</div>\}$

$<contenido> ::= \{<caracter>\}^+$

$<caracter> ::= (a \mid b \mid \dots \mid y \mid z \mid A \mid B \mid \dots \mid Y \mid Z \mid 1 \mid 2 \mid \dots \mid 9 \mid 0 \mid : \mid \text{“ “} \mid ! \mid = \mid _)$   
 $\}$

**Ejercicio 13:** Defina en EBNF una gramática para la construcción de números primos. ¿Qué debería agregar a la gramática para completar el ejercicio?

La construcción de la Gramatica es igual a la de los numeros enteros.

Para verificar si se trata de un número primo o no, debería incluirse un análisis semántico, ya que las gramáticas BNF o EBNF son gramáticas libres de contexto, no se puede hacer una análisis en cuanto al sentido de lo definido.

**Ejercicio 14:** Sobre un lenguaje de su preferencia escriba en EBNF la gramática para la definición de funciones o métodos o procedimientos (considere los parámetros en caso de ser necesario)

// Funcion en Java

G = (N, T, S, P)

N = {<funcion>, <tipo>, <alcance>, <ID>, <contenido>, <alfabeto>, <caracter>, <digito>, <letra>}

T = {a, b, ... y, z, A, B, ... Y, Z, 1, 2, ... 9, 0, /, :, ,, ;, \_, -, +, \*, ?, <, >, &, |, !, {, }, [, ] }

S = {<funcion>}

P = {

<funcion> ::= { <alcance> <tipo> <ID> "(" {<tipo><ID>}\* ")" {<contenido> "}" }

<alcance> ::= (public | private | protected)

<tipo> ::= (int | double | boolean | char | String | ... | float | char | ... )

<ID> ::= <letra> {<alfabeto>}\*

<caracter> ::= (<alfabeto> | : | , | ; | \_ | + | - | \* | ? | < | > | ...)

<alfabeto> ::= (<letra> | <digito>)

<digito> ::= (0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9)

<letra> ::= "a" | "b" ... "y" | "z" | "A" | "B" | ... | "Y" | "Z"

<contenido> ::= {<caracter>}+

}